DISCLAIMER :

The educational notes/contents are compiled from different books and website available online and offline. The credit goes to the concerned author/writer/publisher. This e-content is uploaded on website for educational purpose only. Neither the teachers nor the college claims the rights of these notes. The copyrights of these content are lies with the respective author. SSBES' ITM does claim ownership or do not intend to monetize this content.

Elective I

M. B. A. II Semester

MANAGEMENT OF CREATIVITY AND INNOVATION

COURSE CODE: II/1371/SEC-3A

||**University Examination 50 Marks** ||**Internal Assessment 50 Marks** ||**Total 100 Marks** ||

Learning Objective:

To help students reflect on their own level of creativity and develop and innovative mindset, also create awareness of those factors, characteristics and procedures that make individuals and organizations creative and innovative.

Unit I - Realms of Creativity: Creativity: Concept- Determinants of Creativity – Process- Convergent and Divergent Thinking- Creative abilities- Roots of Human Creativity- Biological, Mental, Spiritual Social- Forms of Creativity-Essence, Elaborative and Expressive- Existential, Entrepreneurial and Exponential.

Unit II - Creative Personality: Traits - Congenial to Creativity- Blocks to Creativity- Strategies for unblocking Creativity-Conductive Environment: Formative Environment and Creativity- Environmental Stimulants- Motivation and Creativity- Strategies for Motivation for being creative, The creative intelligence - Fluency - Flexibility – Originality.

Unit III - Corporate Creativity: Creative Problems Solving, Techniques of Creative Problem Solving- Creative manager Creative Management Practices: Human Resource Management, Marketing Management, Management of Operations, Management of Product Design and Growth Strategies - Collective Creativity.

Unit IV - Creative Organization: Issues and approaches to the Design of Creative Organizations - Mechanisms stimulating Organizational Creativity- Successful innovative organization Structure - Creative Societies, Model of creative society.

Unit V - Management of Innovation:

Nature of Innovation-Technological Innovations and their Management, Managing innovation within firms, Innovative entrepreneurship - Agents of Innovation-Skills for Sponsoring Innovation, Practice cases and situations.

Learning Outcomes::

1. By the end of this course students shall be able to:
2. Learn to apply tools to unlock creativity and innovation.
3. Improve their innovation capabilities as managers by developing a corporate culture
4. conducive for creativity and innovation.
5. Develop an appreciate and influence the personal and organizational factors that influence organizational creativity and innovation.

UNIT I
REALMS OF CREATIVITY

1.1 Creativity Concept

Creativity: Creativity is defined as the tendency to generate or recognize ideas, alternatives, or possibilities that may be useful in solving problems, communicating with others, and entertaining ourselves and others.

- Three reasons why people are motivated to be creative:
    1. need for novel, varied, and complex stimulation
    2. need to communicate ideas and values
    3. need to solve problems

In order to be creative, you need to be able to view things in new ways or from a different perspective. Among other things, you need to be able to generate new possibilities or new alternatives. Tests of creativity measure not only the number of alternatives that people can generate but the uniqueness of those alternatives. the ability to generate alternatives or to see things uniquely does not occur by change; it is linked to other, more fundamental qualities of thinking, such as flexibility, tolerance of ambiguity or unpredictability, and the enjoyment of things heretofore unknown.

- Creativity is any act, idea, or product that changes an existing domain, or that transforms an existing domain into a new one...What counts is whether the novelty he or she produces is accepted for inclusion in the domain."
- Characteristics of the creative personality:
    1. Creative individuals have a great deal of energy, but they are also often quiet and at rest.
    2. Creative individuals tend to be smart, yet also naive at the same time.
    3. Creative individuals have a combination of playfulness and discipline, or responsibility and irresponsibility.
    4. Creative individuals alternate between imagination and fantasy ant one end, and rooted sense of reality at the other.
    5. Creative people seem to harbor opposite tendencies on the continuum between extroversion and introversion.
    6. Creative individuals are also remarkable humble and proud at the same time.
    7. Creative individuals to a certain extent escape rigid gender role stereotyping and have a tendency toward androgyny.
    8. Generally, creative people are thought to be rebellious and independent.
    9. Most creative persons are very passionate about their work, yet they can be extremely objective about it as well.
    10. The openness and sensitivity of creative individuals often exposes them to suffering pain yet also a great deal of enjoyment.

1.2 Determinants of creativity

- **Home environment**
  Home is the place which influence human's behavior, emotions and overall mental health. Everyone's creativity can be influenced by home environment. Home environmental factors such as education of members, culture, economic condition, relationship etc. are the factors of home environment that has impact on building creativity.
- **School environment**
  Child's imagination and curiosity play important role in building creativity. Many factors of school environments such as syllabus, pattern, teaching methods and the like many factors which belongs to primary and secondary education has major impact on increase in creativity. Education and practice are the base of creativity which can expand the knowledge and increase confidence in students towards creating ideas. Curiosity is the dynamic key to stimulate child's creativity. With well-designed spaces and the physical environment for teaching and training, students in different areas can improve their skills.
- **Social environment**
  The social environment or social context is a group of identical or similar social positions and social roles. Social environment of an individual is the culture that he or she was educated and/or lives in, and the people and institutions with whom the person interacts. For example, there are artistic environments (artists in a given area), educational environments (members of a university), political environments (members of a political party), etc. A given social environment is likely to create a feeling of unity among its members, who are more likely to keep together, trust and help one another and think in similar ways. It may be favorable for increase in creativity.
- **Culture**:
  Culture can provide favorable environment for creativity. Friendly culture provides more positive environment towards creativity. Recent developments in cross-cultural psychology suggest that culture also belong to in the social norms that guide behaviors. Norms are shared expectations about what constitutes appropriate behaviors in a given culture
- **Motivation**: individual or employee's creativity can be increased by means of monetary or nonmonetary incentives that can motivate employees towards increasing creativity.
- **Knowledge**:
  It acts as input for increase the ability to generate creative and innovative ideas. Knowledge may be of different areas that decides the creativity in towards that area.

Factors influencing creativity can also be classified into individual level and group or organizational level

Factors at individual level

- **Age** –creativity decreases with age unless individual is intentionally creative
- **Intelligence**- certain level required for certain measures of creativity only.
- **Personality**- high valuation of aesthetic qualities in experiences, interests, attraction to complexity, independence of judgment, autonomy, intuition, self-confidence, ability to resolve conflicting traits in self and belief that self is creative
- **Dispositions**- high level of intrinsic motivation, follow intrinsic interests, free from evaluations and constraints
- **Capabilities**: Insight is a result of integration of previously learned behaviors

Factors at Organizational Level

- Organizational Climate
- Leadership Style
- Resources and Skills
- Organizational Culture
- Structure and Systems

1.3 Creativity Process

The **creative process** is both easy and hard to define: Easy because it simply refers to the way in which people come up with new ideas or solutions for problems, and difficult because that process is a somewhat different one for each person. What we do know, according to Wallas and others who have studied the creative process, is that it includes five definable stages.

Following are the steps of creativity process

Preparation stage
Incubation stage
Illumination stage
Verification stage

Preparation stage

This stage belongs to identifying the need of problem and collecting required information towards the solution of that problem. It also involved in setting up criteria for verifying the solution's acceptability. This stage conduct research for proper solution. It also acts as planning stage for creative idea.

For example, manager collect data from previous team projects for the preparation of new project. Or it could be collecting past records of demand for the preparation of demand forecasting.

Incubation stage

In the Incubation stage, detailed analysis or visualization(contemplate) of problem is done. It may take minutes, weeks or even years.

It is the experience of leaving a problem for a period of time and then finding that the difficulty disappears on returning to the problem, or it can be the solution comes out when thinking about something else if set problems aside for a time. In this way the incubation can be defined as a process of unconscious recombination of thought elements that were stimulated through conscious work at one point in time, resulting in novel ideas at some later point in time.

When discussing the relation between incubation effect, emotions, and creativity, researchers found that positive mood enhances creativity at work. That means that a given day's creativity would be expected to follow reliably from the previous day's mood, above and beyond any carry-over of that previous day's mood. Theory and research on incubation, long recognized as a part of the creative process, suggest such cross-day effects. Thus, if positive mood on a particular day increases the number and scope of available thoughts, those additional thoughts may incubate overnight, increasing the probability of creative thoughts the following day.

Illumination Stage

In the **illumination** stage, ideas arise from the mind to provide the basis of a creative response. These ideas can be pieces of the whole or the whole itself, i.e. seeing the entire concept or entity all at once. Unlike the other stages, illumination is often very brief, involving a tremendous rush of insights within a few minutes or hours.

Following the period of incubation, the creative ideas occur suddenly. Consequently, the unclear thing becomes clear. This sudden flash of solution is known as illumination.

Verification Stage:

Whichever the solution is found in illumination stage, it is necessary to verify whether that solution is correct or not. Hence in this stage the evaluation of the solution is done. If the solution is not satisfactory or not as per the criteria made in the first stage, the thinker will go back to creative process from the beginning. If it is satisfactory, the same will be accepted and if necessary, minor modification may also be made in solution.

1.4 Convergent thinking:

Convergent thinking consists of those abilities which enable one to reach a right solution to problems which mostly one right solution. Problems that may potentially have many solutions, require the exercise of divergent thinking mechanisms.

Convergent thinking is generally means the ability to give the "correct" answer to standard questions that do not require significant creativity, for instance in most tasks in school and on standardized multiple-choice tests for intelligence.

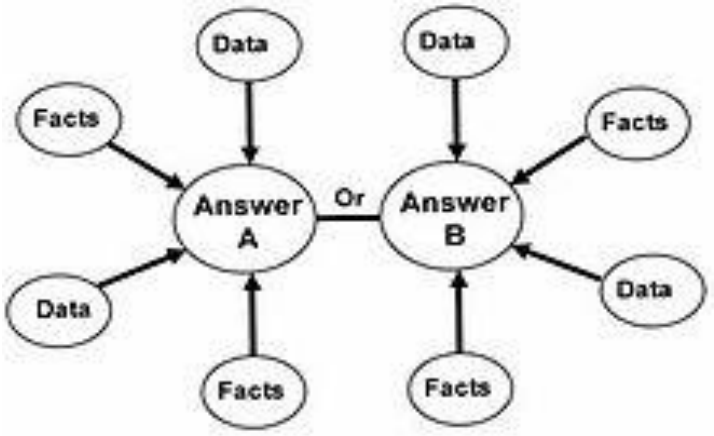


Fig.1.1 Convergent Thinking

The Mechanism of creative thinking outlined as follows

Classification and or categorization

It involves listing the components or elements of a problem, using analogies, making comparisons, and displaying the situation through tables, charts and diagrams.

Analysis

It involves breaking down or factoring a problem into sub-problems, seeking and establishing relationships. It involves analysis of issues, constraints and variables and evaluation of alternatives.

Synthesis

it is concerned with seeing patterns among components or elements and involves aggregation and juxtaposition of elements in terms of their similarity, proximity, consistency, relationships and so on.

Optimization

It is essentially concerned with refining a solution. It may involve substitution of parts, addition of components/elements, deletion of superfluous items, modification of elements, and alternation of connecting points or relations.

1.5 Divergent Thinking:

It involves approaches or perspectives that may be uncommon or unusual, resulting in novel and appropriate solutions. The mechanisms of divergent thinking may be outlined as follows:

A working definition of the problem is sought to be developed by identifying the key terms, concepts and their meanings, the requirements of what needs to be done to solve the problem, and by factoring the problems into sub-problems.

Search for solutions begins with the most familiar and is expanded through associative thinking. \the problem may be restated or redefined by relaxing and or reinterpreting the constraints. Such a redefinition often leads to the emergence of new perspectives or frames of reference.

Imagination and intuition are heavily relied on in search for possible unconventional solutions or towards evolving new syntheses.

Questions like the following stimulate divergent thinking:

- What is the opposite of the known or conventional approach or solutions?
- What would happen if the current base assumptions were changed in part or replaced by another set?
- What would happen if the given constrains in a problem situation were amplified or reduced or eliminate in whole or part?

Divergent thinking is a thought process or method used to generate creative ideas by exploring many possible solutions. It is often used in conjunction with convergent thinking, which follows a particular set of logical steps to arrive at one solution, which in some cases is a "correct" solution. Many possible solutions are explored in a short amount of time, and unexpected connections are drawn.

Divergent thinking involves in following

Recording the idea
Making forced relationships
Recognizing positive potential in a problem
Selecting or developing a list of useful or relevant ideas
Visualizing Future solution state

Activities which promote divergent thinking include creating lists of questions, setting aside time for thinking and meditation, brainstorming, subject mapping / "bubble mapping", keeping a journal, creating artwork, and free writing. In free writing, a person will focus on one particular topic and write non-stop about it for a short period of time, in a stream of consciousness fashion.[

1.6 Convergent vs. Divergent thinking

Personality

The personality correlates of divergent and convergent thinking have been studied. Results indicate that two personality traits were found to be significantly associated with divergent thinking. These traits, namely Openness and Extraversion, were found to facilitate divergent thinking production. Openness assesses intellectual curiosity, imagination, artistic interests, liberal attitudes, and originality.

No personality effects on convergent thinking were found, suggesting that the Big Five personality traits are a better predictor of divergent thinking than convergent thinking or that all types of individuals engage in convergent thinking regardless of their personality.

Brain activity

The changes in brain activity was studied in subjects during both convergent and divergent thinking. To do this, researchers studied Electroencephalography (EEG) patterns of subjects during convergent and divergent thinking tasks. Different patterns of change for the EEG parameters were found during each type of thinking. When compared with a control group who was resting, both convergent and divergent thinking produced significant desynchronization of the Alpha 1,2 rhythms. Meanwhile, convergent thinking induced coherence increases in the Theta 1 band that was more caudal and right-sided. On the other hand, divergent thinking demonstrated amplitude decreases in the caudal regions of the cortex in Theta 1 and 2 bands. The large increase in amplitude and coherence indicates a close iteration between both hemispheres in the brain.

Intellectual ability

A series of standard intelligence tests were used to measure both the convergent and divergent thinking abilities of adolescents. Results indicate that subjects who classified as high on divergent thinking had significantly higher word fluency and reading scores than subjects who classified as low on divergent thinking. Furthermore, those who were high in divergent thinking also demonstrated higher anxiety and penetration scores. Thus, those subjects who are high in divergent thinking can be characterized as having their perceptual processes mature and become adequately controlled in an unconventional way.

Creative ability

Creative ability was measured in a study using convergent tasks, which require a single correct answer, and divergent tasks, which requires producing many different answers of varying correctness. Two types of convergent tasks used were, the first being a remote associates tasks, which gave the subject three words and asked what word the previous three words are related to. The second type of convergent thinking task were insight problems, which gave the subjects some contextual facts and then asked them a question requiring interpretation.

Mood

With increasing evidence suggesting that emotions can affect underlying cognitive processes, recent approaches have also explored the opposite, that cognitive processes can also affect one's mood. Research indicates that preparing for a creative thinking task induces mood swings depending on what type of thinking is used for the task. The results demonstrate that carrying out a task requiring creative thinking does have an effect on one's mood. This provides considerable support for the idea that mood and cognition are not only related, but also that this relation is reciprocal.[12] Additionally, divergent and convergent thinking impact mood in opposite ways. Divergent thinking led to a more positive mood, whereas convergent thinking had the opposite effect, leading to a more negative mood.

Practical use

Convergent thinking is a fundamental tool in a child's education. Today, most educational opportunities are tied to one's performance on standardized tests that are often multiple choice in nature.[13] When a student contemplates the possible answers available, they use convergent thinking to weigh alternatives within a construct. This allows one to find a single best solution that is measurable.

1.6 Creative Abilities

Creative abilities are depending upon many factors such as intelligence, knowledge, confidence etc. it can be responsible for increase in potential for generating creative ideas.
Creativity Intelligence: It is ability to generate unique intellectual power and capacities to solve problem. Creativity intelligence gives the way towards unique ideas and unique problem solving techniques.

Creative Intelligence is a very wide concept which not only means the intellectual activities but also includes the capabilities for solving practical problems of life as well. We use the word 'intelligence' in everyday life. Dictionary has defined it as "the capacity to acquire and apply knowledge" in our day today life. It is intelligence that makes man differ from the animal. Endowed with certain cognitive abilities man is able to reason, think, discriminate, understand, adjust and face new situations. Therefore, man is superior to animals in all such aspects of behavior. Besides these, attitude, interest, knowledge, skill etc. all are associated with the concept of intelligence.

A comprehensive definition of creativity intelligence is given by Wechsler. According to him, "Intelligence is the aggregate or global unique capacity of an individual to act purposefully, to think rationally, and to deal effectively with his environment." It means intelligence is a capacity to act, think and deal rationally in order to adjust to any type of environment. From the above definitions of intelligence, we can find out the real structure of intelligence and what it actually is. It reveals that intelligence is a capacity of an organism, all round mental efficiency possessing the power of discrimination and adjustment to a relatively novel situation. It means that intelligence is such a capacity through which we can judge well, understand properly and reason well.

1.7 Enhancing creativity Intelligence:

Enhancing creativity Intelligence:
In terms of the subject of intelligence, we now know that there is no one monolithic thing called intelligence, but that we have multiple intelligences. Through the pioneering work of Howard Gardner in Frames of Mind and the Multiple Intelligences Model that he's developed, we can identify eight kinds or dimensions of intelligence.

1) Linguistic Intelligence
The intelligence needed to read a book, write a paper, novel, or poem; and understand spoken words.

2) Logical-Mathematical Intelligence
The intelligence needed to solve mathematical problems, balance a checkbook, solve a mathematical proof, and engage in logical reasoning.

3) Spatial Intelligence
The intelligence needed to get from one place to another, to read a map, and to pack suitcases in the trunk (or boot) of a car.

4) Musical Intelligence
The intelligence needed to sing a song, compose a sonata, play a trumpet, or even appreciate the structural patterning in a piece of music.

5) Bodily Kinesthetic Intelligence
The intelligence needed to dance, play basketball, run a mile, or throw a javelin.

6) Inter-personal Intelligence
The intelligence needed to relate to other people, such as when we try to understand another person's behavior, motives, or emotions.

7) Intra-personal Intelligence

The intelligence needed to understand ourselves—the basis for understanding who we are, what makes us tick, and how we can change ourselves, given the existing constraints on our abilities and interests.

1.8 Roots of Creativity:

Biological: biological roots belong to the factors such as genetic characters, heredity which are responsible for being creative by born. Such basic factors act as the roots of creativity.

Mental: Imagination, knowledge, Curiosity, intelligence and behaviour are the mental roots of creativity that has major impact on building creativity.

Spiritual: it is mostly hypothetical but considered as root or creativity. Spirituality can give possibility of impossible facts through creativity.

Social: there are many social factors which can be roots of creativity such as culture, nation, tradition, educational environment and many factors which can provide favourable environment to generate creativity. These factors can influence by different dimensions.

1.9 Forms of Creativity:

1. **Essence creativity**
   Creativity required for fundamental research is an example of essence creativity. It belongs to the ability of creating the ideas which can applied indirectly. For example many scientists involved in fundamental research which cannot applied directly but is useful in many concerned areas. Creativity of Einstein is the best example of essence creativity.

2. **Elaborative Creativity**

Elaborative creativity is the innovative amplification of a core idea or principle. The difference is between say, staff empowerment as a core belief and its amplification into personnel policies, participative management structures, training programs, and so forth. Elaboration can become innovative when it is creatively contextualized, that is, creatively fitted to the organization's situation rather than simply borrowed from elsewhere. It can become innovative when it is done with participation, involving various viewpoints and much brainstorming, and the ideas are creatively synthesized. It can become innovative when not just one but several powerful, possibly partially conflicting ideas are fused together to form its basis, such as the ideas of centralization and decentralization, control and authority, or internal entrepreneurship and efficiency. Elaboration can also become innovative when it is periodically reviewed and creatively modified to suit changing circumstances. And it can become innovative when it is benchmarked, not with practices of the leading competitor, but the world's best practitioners. Example : creativity of Karl Marx

3. **Expressive Creativity**

It is the creativity that can be expressed or presented by specific way. For example, creativity in dancing, singing or any art that can be presented or expressed is the examples of expressive creativity. Rabindranath Tagore , Lata Mangeshkar

4. **Entrepreneurial Creativity**

The creativity that convert innovative idea into value-creating profitable business. "Creativity is a continuous activity for the entrepreneur, always seeing new ways of doing things with little concern for how difficult they might be or whether the resources are available. But the creativity in the entrepreneur is combined with the ability to innovate, to take the idea and make it work in practice. This seeing something through to the end and not being satisfied until all is accomplished is a central motivation for the entrepreneur. Indeed once the project is accomplished the entrepreneur seeks another 'mountain to climb' because for him or her creativity and innovation are habitual, something that he or she just has to keep on doing," says Bill Bolton. Following can be the equation of this creativity.

Entrepreneurial creativity = creativity × entrepreneurial action.

Example: creativity of Dheerubhai Ambani

5. **Existential Creativity:**

It is concerned with the development of one's own creativity, effectiveness and self-actualization potential. It arises when individual is open to all of his experience. It can be enhanced by increasing the consequences of peak experience which is one of the intense happiness and joy, love and passion, great achievement. Existential creativity can be identified as the ratio of "living" to "Existing". Eg. Ramkrishna Paramhamsa

6. Empowerment Creativity

Empowerment is a feeling that employees get from being satisfied. This satisfaction is generated in numerous ways: creating autonomy, responsibility, self-awareness, replacing the old hierarchy with self-managed teams, getting the right reward, giving the chance to be heard or more, company transparency and employee involvement in the decision-making process.

First, empowerment is not giving people power, but rather, you should release people's empowerment by giving them a sense of ownership. Many people in organizations are afraid to be creative. This is because most organizations are afraid of change and new ideas. As a creative manager, you won't direct, control, or manage people. Instead, you will be more like a facilitator, connecting employees and their ideas to the rest of your company. As such, your task is to listen to new ideas, to challenge the existing, foster intrapreneurship, autonomy, engage in strategic thinking and understand your people.

Example: Mahatma Gandhi

UNIT II
CREATIVE PERSONALITY

2.1 Traits congenial to creativity

Sensitive: creative people are very conscious about all things. They can analyze every problem and can create innovative solution.

Sense of Destiny: To have a sense of destiny means that you could see what's coming up. Creative people know predetermined course of actions.

Adaptable: dynamic or flexible to make improvement as per requirement. Creative people can adopt change as per requirement.

Tolerant of Ambiguity: Tolerance is the ability to perceive ambiguity in information and behavior in a neutral and open way.

Observant: observation of creative people is strong. They can observe any situation by all dimensions and perceive world differently

See Possibilities: Creative people can convert problem into opportunities. They can provide solutions to the problems by creative idea and appropriate innovation.

Psychological research has in fact identified and highlighted the distinguishing attributes of creative individuals. These attributes or personality traits may be listed as follows:

- Ability to see things in unusual ways
- Not bound by group standards and control
- Originality
- The ability to focus and concentrate
- The ability to generate a large number of ideas and viewpoints
- A strong sense of curiosity
- Openness and independence of thinking and judgement.

The attributes or traits enumerated above are present in all individuals in varying degrees, but in creative persons they are seen to be present in a conspicuous manner and combination.

2.2 Blocks to Creativity

1. The mental block.

This is where you get trapped by your own thinking. You're so locked into a familiar way of looking at the world that you fail to see other options. You make assumptions and approach a problem from a limiting premise. Or maybe your Inner rears its head and stops you thinking straight.

Solution: You need to change your mind. Question your assumptions, ask yourself "What if…?", and adopt different perspectives. Go somewhere new, or read/watch/listen to something new. Talk to people you can rely on to disagree with you, or offer an alternative point of view.

2. The emotional barrier.

Creativity can be intense. It's not a comfortable pursuit. Faced with the unknown, you may be scared of what you'll discover or reveal about yourself. Maybe your subject matter is painful, embarrassing or plain weird. Whatever – all of these fears and qualms are just different forms of Resistance, leading to procrastination.

Solution: You need to face the worst and come through the other side. There are plenty of things that can help — such as routine, commitment, and meditation. But ultimately you are going to have to endure the fear, pain, or other unpleasant emotions.

It's like getting into a cold swimming pool — you can dive in head first, or inch your way in. Either way, it's going to be bone-chillingly cold. But once you've got over the initial shock, done a few lengths, and got into the flow of it, you may be surprised to discover how invigorated you feel.

Faced with the unknown, you may be scared of what you'll discover or reveal about yourself.

3. Work habits that don't work.

Maybe there's no great drama — you're just trying to work in a way that isn't compatible with your creative process. You work too early, too late, too long, or not long enough. You try to hard or not hard enough. You don't have enough downtime or enough stimulation. Or maybe you haven't set up systems to deal with mundane tasks – email, admin, accounting, etc – so they keep interfering with your real work.

Solution:

Step back and take a good look at how you're working, and where the pain points are. If it's email, learn a new system for dealing with email. If you don't have enough energy, are you working at the right time of day? If you feel paralyzed by freedom, introduce more structure and order into your day. If you feel constrained by routine, find room for improvisation.

4. Personal problems.

Creativity demands focus — and it's hard to concentrate if you're getting divorced/ dealing with toddlers/battling an addiction/falling out with your best friend/grieving someone special/moving house/locked in a dispute with a neighbor. If you're lucky, you'll only have to deal with this kind of thing one at a time — but troubles often come in twos or threes.

Solution: There are basically two ways to approach a personal problem that is interfering with your creative work — either solve the problem or find ways of coping until it passes. For the first option you may need some specialist help, or support from friends or family. And it may be worth taking a short-term break from work in order to resolve the issue and free yourself up for the future. In both cases, it helps if you can treat your work as a refuge — an oasis of control and creative satisfaction in the midst of the bad stuff. Use your creative rituals to set your problems aside and focus for an hour, or a few, each day. When your work is done, you may even find you see your personal situation with a fresh eye. It helps if you can treat your work as a refuge – an oasis of control and creative satisfaction in the midst of the bad stuff.

5. Poverty.

I'm not just talking about money, although a lack of cash is a perennial problem for creatives. You could also be time-poor, knowledge-poor, have a threadbare network, or be short of equipment or other things you need to get the job done.

Solution: Like the last type of block, this one has two possible solutions: either save up the time/money/or other resources you need; or make a virtue of necessity and set yourself the creative challenge of achieving as much as possible within the constraints you have. If you're doubtful about the latter option, consider the first and second Star Wars trilogies, and ask yourself whether more resources always equal more creativity!

6. Overwhelm.

Sometimes a block comes from having too much, not too little. You've taken on too many commitments, you have too many great ideas, or you're overwhelmed by the sheer volume of incoming demands and information. You feel paralyzed by options and obligations, or simply knackered from working too hard for too long.

Solution: It's time to cut down. If you take on too many commitments, start saying 'no'. If you have too many ideas, execute a few and put the rest in a folder labeled 'backburner'. If you suffer from information overload, start blocking off downtime or focused worktime in your schedule (here are some tools that may help). Answer email at set times. Switch your phone off, or even leave it behind. The world won't end. I promise. Sometimes a block comes from having too much, not too little.

7. Communication breakdown.

Creative blocks can happen between people as well as between the ears. If you work in a team, tensions are inevitable, and can make it hard to do your best work — especially if you have one of those proverbial 'difficult people' in your working life. Sometimes you get blocked by phantoms — merely imagining your work being booed by audiences and mauled by the critics. And sometimes this happens for real and you have to deal with it. It could just be a marketing problem — after years of plugging away at your art with a miniscule audience, you wonder why you bother. Or maybe you just don't have a hotline to the people who matter in your field, so you struggle to land the right opportunities

Solution

This is where creativity blends into communication skills. You need to be adept at understanding and influencing the right people, however difficult or mystifying they may be. Which means beefing up your influencing, marketing, or networking skills. I don't care if you're shy (I was) or introverted (I am). If you want to succeed, you need to do this. And sometimes it's about accepting that you can't please all the people all of the time, and growing a thicker skin for rejection and criticism. Show me a creative who's never suffered a setback or a bad review, and you won't be pointing at a superstar.

Barriers to Creativity and Innovation

Adair also defines a number of obstacles which inhibit creativity. The seven main ones are:

1. Negativity
2. Fear of failure
3. Lack of quality thinking time
4. Over-conformance with rules and regulations
5. Making assumptions
6. Applying too much logic
7. Thinking you are not creative

2.3 Strategies for unblocking to creativity:

1. Check in with your mission statement. Mission statements are often written and then forgotten. Keeping yours on hand for regular review keeps your mind focused on the "why" you're in business. When you're clear about the why (i.e., why your customers need what you provide), then it's easier to come up with the "what" (i.e., what you provide, generated through creativity). If you don't have a mission, now's a good time to write one.

2. Find inspiration from your clients. I recently heard a couple of successful entrepreneurs talk about where they find inspiration. They each said that when they are feeling stuck or frustrated and feel that they can't come up with a single ounce more of creativity, they call their favorite clients and spend a few minutes chatting. Doing so reignites their creativity.

3. Read. Personally, you don't read anything that doesn't serve your ability to better serve your clients, simply because you don't dedicate the time to pleasure reading. And often you find inspiration in your daily practice of reading interesting pieces on the web. However, sometimes that's not enough and you have to read outside of your comfort zone to ignite my creative spark.

4. Schedule a meeting with employees to bounce ideas around. Your employees know your business from a different perspective than you do. Having regular idea-generation powwows keeps the flow of inspiration and creativity going. This also gives your staff a sense of ownership and lets them know that their voice and ideas matter and are vital to the enterprise's success.

5. Take a day or even a few hours off and go somewhere that inspires you. Stepping away from your office for a few hours or ideally a whole day sends you back to the office with a fresh perspective. Spending that time in a place that inspires or calms you is optimal for this purpose. For me, that place is the beach.

6. Schedule a meeting with a colleague to talk shop. When you really feel stuck, you call upon your friends who are also coaches. And every time you do, you walk away with so many new ideas that you often have to jot them down in my phone so you don't forget them all.

7. Take a vacation. It could be that your creativity is locked up because it's been way too long since your last vacation. It might not be a good time to take time away from the office, but even an overnight stay at a resort nearby could be just enough to refresh you and your creativity.

8. Solicit advice from unlikely places. Sometimes someone who is not at all connected to our organization or even to our industry is the best place to find inspiration. Personally, I do this a lot and I find that my best ideas on how to drum up new business come from the most unlikely places and people.

9. Meditate, pray or exercise. You don't have to be a religious type or even spiritual to get the benefit of these soulful practices. (Calling all atheists!) Exercise can have a similar effect. When we try too hard to generate creativity, we can actually block it more. Being in silence for a few minutes a day in whatever capacity feels comfortable can unblock our creative juices because stillness slows down our thoughts and clears the mind. For me, running without music--to the beat of my own breath--has a similar effect on me as my mediation practice

2.4 Environmental Stimulants to Creativity

1. Freedom in deciding what to do or how to accomplish the task
2. A good manager, enthusiastic, good communications skills, matches tasks to skills and interests, sets a clear direction.
3. Access to necessary resources - facilities, equipment, information, funds, people
4. Encouragement, enthusiasm for new ideas, atmosphere free of threatening evaluation

5. Organizational characteristics such as a formal mechanism for considering new ideas, climate marked by cooperation and collaboration across levels and divisions
6. Recognition for creative work in the form of feedback, recognition and reward
7. Time to think creatively, explore different characteristics
8. Challenge, a sense of having interesting and important work to do.
9. Pressure arising from competition with other units, or a desire to do something important

2.4 Strategies for motivation for being creative

Employee motivation strategies

When people think of honoring employees for jobs well done, they may typically think of monetary rewards. However, these may be neither necessary nor the best type of reward. Once offered, cash bonuses can come to be expected and quickly forgotten, especially if they are the only recognition employees receive. By contrast, frequent, positive feedback provided within an enjoyable, team-oriented environment makes a tremendous difference in employee's sense of being valued and, as a result, their commitment to your company. With or without financial rewards, these cultural aspects of the workplace could be the smartest investment in the staff and business.

Recognize and Reward High-Quality work

Employees are bound to be much more productive when they work in a positive, supportive environment. For example, Tejas Securities Group, Inc., a full-service broker/dealer and investment banking firm, strives to maintain an enjoyable, family-oriented atmosphere in which all employees focus on achieving team goals. This company goes an extra step by bringing in catered lunches every day for all the employees to enjoy together. In this environment, everybody wins. We enjoy the dynamics of striving toward our goals together as a team, said Kurt Rechner, President and Chief Operating Officer of Tejas Security Group.

Praising employees for achieving their goals is important in maintaining an enjoyable work environment. Management can show their appreciation with positive feedback, however, if they go a bit beyond verbal praise, they can enhance employees motivation without spending a lot of money. For example, celebrate successes with bagels or pizza. Invite employees to share their experiences in, and coworkers contributions toward, accomplishing the goals. Peer recognition will further reinforce employees sense of teamwork and commitment. Conclude the celebration by presenting mugs, T-shirts or other tangible items that will serve as reminders of their success and inspiration for ongoing achievement. With these good intentions, there are still potential drawbacks. For example, improvements in performance may be temporary, rather than long term. In addition, employees could lose their intrinsic motivation: they can become motivated solely for gaining a tangible prize, especially if its a substantial monetary reward,

rather than for experiencing the satisfaction of accomplishment. These challenges can be avoided by maintaining a positive, motivating atmosphere.

Inspire Employees Creativity and Empower them to use it

Recognizing success is critical, and equally important is inspiring employees to work toward achievements. Your staff will be inspired by knowing their contributions are valued and that management is confident in their capabilities. At Tejas Securities Group for example, The Chairmans Cup, a silver chalice inscribed with its name, is awarded each month to an employee who is recognized for their individual contribution to the overall teams success. The winner is then announced in a companywide meeting and is awarded the cup to display at their work station. Rechner noted This announcement and award has become a fun and highly anticipated event, recognizing the ongoing importance of individual contribution to the company's success.

Inspire creativity by providing freedom

Inspire creativity by providing freedom, time and other resources to employees. Ask them what they need to maximize their innovative thinking and productivity, and provide it with enthusiasm and encouragement. To further stimulate employees creativity and confidence, support continual education through classes, seminars, subscriptions and memberships. Make information easily accessible through a work library. Ask employees to offer new ideas, request proposals for new projects, and share employees suggestions through publications, meetings and recognition events. Most importantly, take action on those ideas that have potential benefit for the company, and recognize employees who made any resulting achievements possible.

While encouraging creativity and rewarding success may come somewhat easily, it may be more difficult to stay optimistic when mistakes are made. However, this is where positive reinforcement is even more critical. Employees will be much less likely to offer ideas if they are intimidated by managements reactions to possible mistakes. Keep in mind and express to employees that mistakes are learning opportunities, which could lead to innovative ideas that have a major, profitable impact on your company. If an idea doesn't work out, recognize the initiative and effort. Employees will feel further inspired and satisfied, knowing that management truly listens to their ideas and supports their efforts. According toRechner at Tejas Securities Group, managements openness to staff members input, feedback, ideas and suggestions is the cornerstone of good communications and strong employee relationships. Everybody wins when they are all part of a supportive team.

All of these steps contribute to a sense of entrepreneurship and empowerment, which are essential to reinforcing teamwork and dedication. Empowerment should be initiated on three levels: encouraging employees to be more active in their work; involving staff members to

improve processes and procedures; and enabling them to make more and bigger decisions. In addition to motivation and job satisfaction, employees benefit with strengthened confidence to accept and pursue new responsibilities. Once a few employees succeed, their enthusiasm and motivation would become contagious throughout their teams or departments. As a result, those groups would become more enthusiastic, proactive and therefore, successful, which further stimulates their team spirit.

Ultimately, your company has much to gain by empowering staff members. By maximizing employee's talents and motivation, managers could invest more time in strategic planning and further motivating employees.

Be wary of Financial Incentives and Rewards

Certainly, monetary incentives and rewards could be part of your employee-recognition program. However, it is critical that these incentives not be the only or primary strategy for motivating and retaining employees.

On the surface, financial incentives may seem to be the most meaningful forms of motivation for employees. However, the short-term benefits may be far outweighed by long-term disadvantages, which could turn your costly financial incentives into serious deterrents to employee's productivity. As a result, your company's profitability could suffer, and you may be faced with further costs of replacing employees who leave for more satisfying work environments.

Typical of human nature, people tend to think about what their employers have (or haven't!) done for them recently, especially if they do not feel appreciated. Furthermore, a brief word of gratitude only when a financial reward is presented will not be perceived as a sincere expression of appreciation. The easiest and most cost-effective way to avoid this pattern is to maintain open communication with positive feedback and encouragement at all times, with occasional celebrations where presentation of cash rewards or announcement of new financial incentives, if any, should be just a small part of these events. Similarly, if cash bonuses are presented on a schedule, such as around the holidays, they probably come to be expected. This reaction could be avoided if bonuses are given randomly, when you have extra money to share with employees. However, before deciding to present cash bonuses, determine if that money could be better used to expand your business. Express to employees how their contributions resulted in the extra cash flow, and rally them up for investing that money into exciting new possibilities for themselves and the organization. In addition to cash bonuses, other types of monetary rewards are profit-sharing plans and Employee Stock Option Programs (ESOPs). Profit-sharing plans are simple types of retirement plans in which employers contribute an amount of money equal to a certain percentage of eligible employee's salaries. With ESOPs, the company

contributes to a trust, and these funds are allocated to individual employee accounts. Also, employees can reserve part of their paychecks to purchase shares of the company's stock.

Profit-sharing plans

Profit-sharing plans offer a strong incentive for employees to be more involved with the company. The staff is more likely to work as a team and accept greater responsibility for increasing the company's profitability. Another advantage is that financial benefits are measurable and objective. As a result, management would not risk showing favoritism, which would cause this motivational strategy to backfire.

On the contrary, profit-sharing plans can also have potential drawbacks. They do not guarantee that employees will be focused on customer service, productivity or other essential elements for the company's success. If profit levels are ever too low to be shared, employees will feel disappointed or even resentful. Even if this does not occur, employees may object to the lack of acknowledgment for their individual achievements. Of course, this particular disadvantage can be overcome with strategies discussed previously. In any case, a negative situation would lead to lower employee morale, which inevitably diminishes employee's motivation and performance. On the positive side for ESOPs, employees directly gain a sense of ownership, usually at levels proportionate to how much stock each employee has. The potential disadvantage, similar to profit-sharing programs, is if stock options do not work out. Furthermore, emotional stress often associated with fluctuations in stocks could interfere with employee's productivity.

Certainly, financial incentives and rewards can be true motivators, but only when balanced against the potential drawbacks and packaged with ongoing verbal recognition, encouragement and support. At Tejas Securities Group, for example, We supplement our employee-recognition program with an ESOP. The employees sense of ownership and the stock investments potential are icing on the cake on top of the essential substance of open communication, teamwork and positive reinforcement, rechner said.

Launch a positive, ongoing cycle with the best choices for your staff

When handled in a consistently positive manner within a team-oriented atmosphere, all of these strategies contribute to an ongoing positive cycle: motivated employees are encouraged to be creative, which leads to accomplishments that gain recognition, which strengthens their sense of job satisfaction and boosts their motivation. With all of these factors in place, staff members will produce more for the company. The entire process should be continual and even begin with the hiring selection. Its essential to start with quality people as the foundation, and then motivate them to succeed, said Rechner at Tejas Securities Group. It takes hard work, some money and a bit of luck to recruit employees who have the technical skills and personal qualities

you need to strengthen your business. Maximize your investment in these individuals by establishing and maintaining a positive relationship with them.

Providing Company Information

Tejas Securities Group, Inc., a Texas corporation ("Tejas Securities"), is a wholly owned subsidiary and a primary business operating unit of Tejas Incorporated, a publicly traded financial services company. Tejas Securities Group is a full service brokerage and investment banking firm that focuses on the following: (i) proprietary research on distressed debt and special situation securities, (ii) trading and other brokerage services to value-based institutional and retail investors active in fixed income and equity instruments, and (iii) corporate finance and strategic advisory services to middle-market companies within our target industries.

2.5 The Creative Intelligence

It can be described as follows

1. Sensitivity to problems, or the ability to recognize problems;
2. Fluency, which encompasses

   a. Ideational fluency, or the ability rapidly to produce a variety of ideas that fulfill stated requirements;

   b. Associational fluency, or the ability to generate a list of words, each of which is associated with a given word;

   c. Expressional fluency, or the ability to organize words into larger units, such as phrases, sentences, and paragraphs;
3. Flexibility, which encompasses

   a. Spontaneous flexibility, or the ability to demonstrate flexibility;

   b. Adaptive flexibility, or the ability to produce responses that are novel and high in quality.

Unit III

Corporate Creativity

3.1 Creative Problem Solving:

Creative problem solving(CPS) is a framework to apply creative thinking processes to solve problems. Creative problem solving usually begins with defining problems. The creative problem solving use information to form new idea. The real key to creative problem solving is what you do with the knowledge. CPS requires an attitude that allows you to search for new ideas and use knowledge and experience. Creative problem solving can have following steps

i. **Objective finding**: select or identify a problem
ii. **Fact finding**: list what you know
iii. **Problem finding**: what is the real problem
iv. **Idea finding**: brainstorm ideas
v. **Solution finding**: Evaluate and choose best idea
vi. **Acceptance finding**: find ways to put ideas into action.

3.2 Creative Problem Solving Techniques:

1. Problem Decomposition Technique:

Most real life problems have multiple dimensions. Same problem can be observed by different perspective/views. In management problems there is almost always an economical angle. But frequently there are other dimensions as well, such as a psychological angle which can be used to solve problem with different attitudes, values and beliefs. There are also social, cultural dimensions by which problem can be recognized and solved.

2.Information Search Techniques:

It gives the information that related to the past. This technique describes the problem by considering the past experiences about same problems. If same problem experienced in past it can be solved by same solution that applied in past. On the basis of information related to same problems solving it can be solved with better approach of solution.

3. Techniques for Breaking:

In this techniques problem can be broken in to different ways. By this way the problem can be identified easily and solved properly. By means of different broken responses it can be seen the internal and external factors of problem.

4. Unblocking Techniques:

- There are various unblocking techniques such as
- Eliminate or minimize mental block
- Eliminate fear of failure
- Eliminate fear of social approval
- These block color how we perceive the problem and what options we consider.

5. Mutual stimulation techniques:

It involves sharing of crazy ideas or provocative mutual questioning.

6. Imaging techniques

It involves imaging of situation in detail and innovation assuming positive and negative factors. It involves assumptions of different results of different solutions.

7. Fusion Techniques:

This technique belongs to fusion of different ideas into one solution. For example, a manager can see a workplace problem. But technique for articulating the perspectives of stakeholder, vendors, customers, unions etc.

8. Brainstorming:

Brainstorming is a group or individual creativity technique by which efforts are made to find a conclusion for a specific problem by gathering a list of ideas spontaneously contributed by its member(s). The term was popularized by Alex Faickney Osborn in the 1953 book Applied Imagination. Osborn claimed that brainstorming was more effective than individuals working alone in generating ideas, although more recent research has questioned this conclusion.[1] Today, the term is used as a catch all for all group ideation sessions.

9. Externalization Techniques:

In this techniques the prediction of extreme result of different solutions to the problem is considered and applied to the problem. For example, if 'x' solution is applied to the problem then it is extremely assumed that major loss or major profit etc.

10. Attributes Change:

Every problem has different attributes. As like any object has attributes such as a product has attributes like color, size, weight, price, quality etc. the problem has different attributes. Using different attributes solutions is applied accordingly.

3.3 Creative Management Practices

Creative Human Resource Management Practices:

- In any organization training or induction program is executed by specialist person or department, but when training is given by owner of CEO it can be a creative practice of HRM. By means of this creative HRM practices the result would be effective and innovative.
- Induction of new recruits to a company and providing training and job rotation orientations.
- Delinquent or rowdy workers are mentored and reformed for their absenteeism and delinquency
- Leader should be tough, tender, set personal example and be good at transactional and transformational issues.
- Grooming next in line to hold higher positions to prevent problems in succession.
- Empowering Staff.

Creative Marketing Management Practice:

Marketing Management is the process of management involved in identification, anticipation and satisfaction of need with profit. Marketing is customer oriented whereas sales in product oriented. Many activities of marketing management can be implemented with creativity. Such as advertising is part of marketing management. If same activity is executed by creative way, it will be considered as creative marketing management practice.

Example: Vodafone made creative advertising in which they have made effective positioning of brand image in customer's mind without use of brand ambassador.

Creative Production/Operations management Practices:

- Operational Management was a major area of creativity in late 19th and early 20th Century .
- Earlier in 1950's and 60's mathematics and computer sciences were utilized through Operations Research to schedule production, arrive at inventory levels etc.
- Japanese companies introduced successful manufacturing techniques in 1970's and 80's – JIT, Kanban, TQM , TPM, 5S, certifications of vendors, encouraging substitutes to reduce cost and weight , partnering with competition, involving customers from design stage, using retirees as suppliers and quality inspectors have impacted margins

3.4 Management of Operations

Management of operations is the system that increases workflow and oversees deadlines within a creative process. Creative operations give creative team members insight into what's happening throughout a campaign. It is the solution to maximize workflow and ensure assets are completed on time. creative operations managers review the success of their implemented structures. So often the manager benefits greatly by adjusting schedules and deadlines to allow for further creative production.

- Formally defining the creative processes in your organization, and understanding how they can be streamlined.
- Identifying gaps in technology that are holding back your creative workers, and locating tools and solutions that will help address them.
- Optimizing the people, processes, and technology involved in creative work, so that the final product is always of the highest possible quality.
- Selecting the right metrics and KPIs to judge the success of creative projects.

There are three different elements to creative operations that every organization should be concerned with:

Timeliness: Are the organization's creative assets being delivered on time? Are there any bottlenecks or inefficiencies throughout the creative process?

Capacity: Is the creative team currently producing assets at or near maximal capacity? Are they understaffed or overstaffed?

Compliance: Are the organization's creative assets following a structured production pipeline? Are they being reviewed and approved within an acceptable timeline?

In general Operations management concerned with converting input into value added output. In production and operations management, operations management implemented by all functional areas of organization.

3.5 Management of Product Design and Growth Strategies

3.5.1 Product Design Concept

Product design is Conceptualization of idea about a product and transformation of the idea into reality. To transform the idea into reality a specification about the product is prepared. This specification is prepared by considering different constraints such as production process, customer expectation etc.

In product design stage all aspects of the products are analyzed. Final decision regarding product is taken on the basis of analysis. This decision can be any aspects related to the product e.g. types of material of each element.

Objectives of Product Design

- To ensure growth of organization
- To meet new requirement of the customers
- To utilize the maximum capacity of the organization, such as physical capacity, man power etc.
- To utilize the surplus fund of the organization
- To increase company's market, share and to target new market segment.

3.5.2 Growth Strategies

Growth strategy: an organization substantially broadens the scope of one or more of its business in terms of their respective customer group, customer functions and alternative technologies it improves its overall performance

Types of Growth strategies

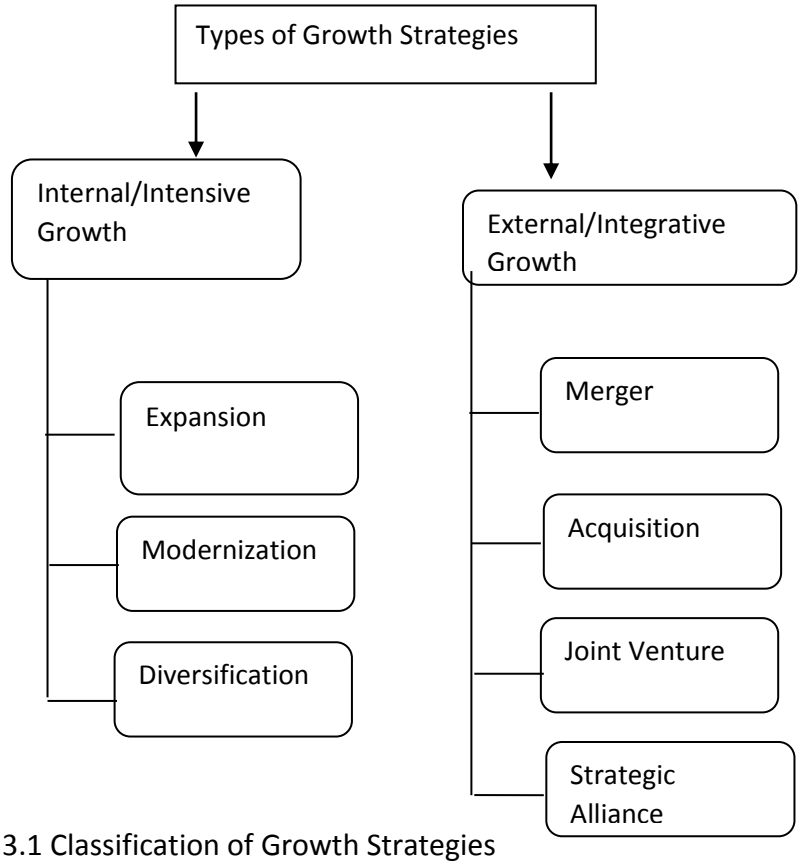


Fig.3.1 Classification of Growth Strategies

Above figure shows the classification or types of growth strategies i.e. internal and external

Internal growth strategies relate to the following actions: -

- Designing and developing new products/services Building on existing products/services for new opportunities.
- Increase sales of products/services through better market reach.
- Expanding existing product lines and service offerings Reaching out for new markets.
- Expansion into foreign markets.

Internal growth strategies can be described by Ansoff's Product-Market Expansion Grid

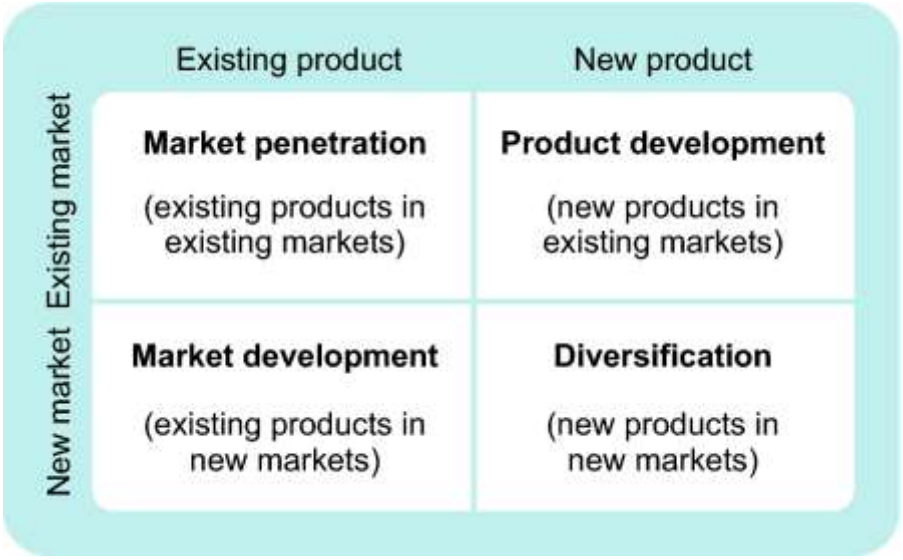


Fig. 3.2 Ansoff's Product-Market Expansion Grid

Market Penetration: Increase sales through effective marketing strategies within the current target market. It is used to maintain or grow the market share of the current product range and to become the dominant player in the growth markets

Market Development: Expand sales in new markets through expanding geographic representation. An organization's current product can be changed improved and marketed to the existing market. The product can also be targeted to another customer segment. Either way, both strategies can lead to additional earnings for the business

Product Development: Increase sales through new products/services an organization that already has a market for its products might try and follow a strategy of developing additional products, aimed at its current market. Even if the new products are need not be new to the market, they remain new to the business.

Diversification: Diversification Strategy is the development of new products in the new market. Diversification strategy is adopted by the company if the current market is saturated due to which revenues and profits are lower.

External Growth Strategies

Merger: The combining of two or more companies. In merger two companies agree to move ahead and exist as a single new company.

Example: Glaxo Wellcome + SmithKline Beecham =GlaxoSmithKline

Acquisition: When one company takes over another company and clearly establishes itself as the new owner, the purchase is called an acquisition.

Example**:** Acquisition of Corus by Tata Steel.

Joint Ventures: A joint venture is an entity created when two or more firms pool a portion of their resources to create a separate, jointly owned organization. All involved will have an equity stake in the new venture. It is a legal partnership between two(or more) companies where in they both make a new (third) entity for competitive advantage. Unlike mergers and acquisitions, in joint venture the parent companies does not cease to exist.

Example : Tata with Marcopolo and Fiat(Tata Motors has formed a 51:49 joint venture in bus body building with Marcopolo of Brazil

Strategic Alliance : A Strategic Alliance is a formal relationship between two or more parties to pursue a set of agreed upon goals or to meet a critical business need while remaining independent organizations. It is a kind of partnership between two entities in which they take advantage of each other's core strengths like proprietary processes, intellectual capital, research, market penetration, manufacturing and/or distribution capabilities etc. They simply would want to work with the other organizations on a contractual basis, and not as a legal partnership.

3.6 Collective Creativity

Collective creativity occurs when bisociation is shared by two or more people. We are beginning to see that collective creativity can be very powerful and can lead to more culturally relevant results than individual creativity does. This is what happens with really good collaboration based on teamwork. Bisociation is the act of association of an idea or object with two fields ordinarily not regarded as related.

Why is collective creativity important?

Collective creativity, when practiced with "users" in the design development process, can result in useful and relevant innovation. This is important because useful and relevant innovation can be commercially successful at the same time as it is culturally beneficial.

The shift from individual to collective levels in thinking and doing is occurring today in many domains. We see this shift taking place especially today in the world of business. Design education needs to keep up with the shift to meet the challenges created by new levels of thought and action.

How can ordinary people play a role in collective creativity?

How can we invite ordinary people, whose creativity is probably latent, to join the design development process? First, we must learn to respect their latent creativity. Then, we need to provide them with experiences and tools so that they can exercise their creativity

and participate directly in the design process.

There are two different experiences—ideation and expression—that we must provide for them to harness their latent creativity. **Ideation** is referred to the preverbal idea stage and **expression** is referred to the translation of those ideas into formal systems of communication.

Dimensions Of Collective Activities

In collective creativity, we have identified two dimensions that collective activities vary on, the nature of the task, and the nature of the output of the collective activity.

1. **The nature of the task**

   We begin by looking at the variety of tasks performed by the crowd. Some tasks are routine: the tasks can be done more or less automatically with little uncertainty about one's ability to accomplish the task in a set amount of time. **For example,** transcribing spoken language to text, translating from one language to another, and labeling images. While each of above involves human cognition, given time, it is likely that the task can be completed. Examples of these activities are shown under **"Routine tasks".**

   By contrast, other tasks entail great uncertainty about the ability of the performer to find a solution in a given amount of time. **For example,** a large search space leaves open the possibility that a person may find a result early or after a considerable amount of time or never. This category of task requires more cognitive effort and its exploration may lead to solutions that have not been found before, such solutions and the exploration process itself are creative.

2. **The nature of the output**

   Next, we look at the nature of the final output of the collective activity. Some activities' outputs are more or less collections of each individual's work, which we call **aggregated output.** Some activities will produce outputs that are new: the outputs are **emergent.** There are two ways new things can emerge. First, through **discovery:** crowd participants can find something not seen before. Second, through **combination:** integrating participant's work can produce something novel.

Table 1. Dimensions of Collective Activities

	Routine tasks	Non-routine tasks
Emergent output	Galaxy Zoo Photocity Picbreeder Manatee	**Collective creativity** Foldit, Eterna Threadless Innocentive Quirky Climate Colab All Our Ideas CrowdForge Evolutionary Collective Design Polymath
Aggregated output	Soylent ESP reCaptcha Crowdfunding Tiramisu	Wikipedia

Looking at the set of systems in the top right quadrant, we can see they fall roughly into three categories. First, there are games: Foldit and EteRNA. Then, there are contests: Threadless, InnoCentive,
Climate CoLab and Quirky. Third, are systems with more elaborate
structures based on networks: CrowdForge and Evolutionary Collective
Design.

1. **Games**

   Games have been built to aid scientific discovery. **For example,** players create protein structure models by playing an online game called **Foldit.** The results show that for difficult problems, the players' solutions can in some cases outperform computational methods.

   Thus, the crowd in games can perform high demanding complex tasks and the output of the games is the solution to difficult scientific problems. Moreover, this collective creative process can be traced, providing us insight on how collective creativity operates.

2. **Contests**

   Online contests, and more specifically idea competitions, are used for eliciting collective creativity. Challenges are posted with prizes promised, and the crowd competes with each other for the prizes by providing solutions. The hosts of idea competitions are various including companies, government institutions, and third party websites.

3. **Networks**

   Both contests and games have fairly simple organizational structures. People either compete as individuals, or as teams. But in other kinds of collective creative work, more intricate structures are used: we will call this category networks. We highlight two examples here.

   In the previous two categories, each member of the crowd is almost interchangeable: they can more or less direct their own tasking, proceeding at their own pace. The

interdependence of the crowd is low. By contrast, in **CrowdForge**, a problem is subdivided, each member of the crowd gets his or her own piece to work on, and then the pieces are aggregated later. This framework was used to produce articles, and also to research purchasing decisions.

The concept behind this system is related to two computer techniques: divide and conquer, in which problems are broken down into smaller bits and then solved, and, more recently, map/reduce, in which ideas are parallelized and then aggregated. In this way, CrowdForge is consistent with the overall theme of human computation: a problem is defined in a way that could be performed by computing nodes, and humans are substituted for these nodes.

Unit IV

Creative Organization

A company with a strong organizational structure benefits from improved communication, a well-defined hierarchy and the ability to create a unified company message. As efficient as organizational structure can be, it can also create problems that can lead to loss of productivity and internal conflict. In order to maintain a strong company framework, you need to be able to identify the problems within an organizational structure and deal with them as they occur.

4.1 Issues and Approaches to design of Creative Organizations

4.1.1 Issues/Challenges to design of Creative Organizations:

Division of Labor:

Division of labor means task or responsibility is given as per skills or specialty in which person is expert. While making this possible it is very difficult to assign proper task to proper person in organization. Miscommunication or conflict can be possible while execution of division of labor.

Inconsistency

When each department functions independently, inconsistency may result. Company policies and procedures may not be enforced, causing employee dissension and confusion. Employees who transition to other jobs in the company run into problems when they behave as they did in their old roles. Approvals take longer because no one knows who is in charge when multiple departments interact. For example, if some customers complain that no one responds to problems, a small business can restructure support so each customer gets paired with a company representative.

Poor Communication

As a small business grows, organizational structures that worked before no longer function productively. For example, relying on informal meetings for planning tends to become less effective. Lack of structure or an inflexible structure can impede the work force from achieving desired results. Additionally, poor communication among department leaders could filter down into the rest of the organization. Establishing good communication mechanisms -- such as print, email and presentations -- alleviates organizational problems that stem from rapid growth. Implementing good project management practices can also formalize procedures that ensure good communication.

Unclear Goals

To enable agility, each organization may set its own goals. This minimizes the levels of approval. However, when each department sets its own goals, the whole company lacks a cohesive direction. Infighting results when limited resources must be shared. Setting a clear company strategy and having each department align its individual goals to the company's helps minimize the problems. In a divisional organizational structure, each employee works to meet the division's needs. This works well when each division works separately. However, when divisions need to interact, this works less effectively.

Departmental Loyalty

One of the dangers of creating departments is the appearance of an "us versus them" mentality between different groups. Sales may feel in conflict with accounting because new customers are not getting approved for credit terms. Logistics is at odds with manufacturing because products are not being built fast enough to meet shipping deadlines. Departments tend to get competitive and feel that their work is more important to the success of the company than the work of the other groups. This can cause breaks in communication that affect productivity.

New Management

If there have not been changes in management for many years, then the company will start to settle into a way of doing things that is efficient and comfortable for the existing management team. Changes in management, for whatever reason, can put strain on the organizational structure of an organization. The new manager, or managers, may be unfamiliar with the way the organizational structure has been run for years and try to put a new spin on how things should be run. There is an adjustment period for employees and other managers.

Confusion

Effective communication is required to keep an organizational structure running smoothly. Without communication, new ideas and processes can get confused. Managers may begin to redouble efforts in an attempt to claim certain parts of a process as their own. This is why executive communication to the rest of the company is critical to the success of any organizational structure. If departments are not clear on exactly what their responsibilities are, then the ensuing confusion can slow production down.

Company Goals

An organizational structure is only effective when the entire company uses it properly, according to management consultants Liebowitz and Associates. When upper management creates departmental goals for the rest of the company without first consulting with the managers of those departments, the company runs the risk of not making its goals. In order for an organizational structure to be effective, goal-making needs to be a two-way process. When upper management does not seek the input of the rest of the company to create company goals, then resentment can set in and morale begins to drop.

4.1.2 Approaches to The Design Of Creative Organization

Creativogenic Management Style

Management style can both impede innovations and facilitate them. In the 1950s, Tom Burns and G.M. Stalker were commissioned to study why older firms in the electronics and electric industries in Britain were not able to convert invention into successfully marketed products while some of the younger companies in Britain and the US could. There older firms were reasonably effective during the world war –II at producing for the defense force large quantities of equipment that defense R&D establishments had developed. After the war was over, there were fair prospects for the commercial exploitation of such inventions at radar, television, telecom equipment, servo-mechanisms and so forth, and these firms set up R&D

establishments to develop commercial products based on these inventions. But they failed badly because they were not customer focused, and were unable to respond quickly to the needs of the customer, a strength the successful younger companies had.

Burns and Stalker explained the failure of the older firms in terms of deficient management style and resulting ineffective culture and structure. They were managed on mechanistic lines with strong belief in centralization and extreme specialization of functions. This meant that coordination of interdepartmental activities was done mainly by the head of the concern, strong departmental loyalties bred by lifelong career paths only within the functional department made interdepartmental collaboration very difficult. Information flowed mainly vertically rather than also diagonally and horizontally. A strongly hierarchy-bound operating culture sapped initiative at lower levels. They suggested that an organic mode of management was much more suitable in technological change-driven turbulent markets like electronics. In this mode.

But that is not whole story. Between 1967 and 1987, the Japanese industrial system was able to increase manufacturing productivity two to three times faster than the US and western Europe and indeed by 1988, five Japanese corporations figured in the world's top 10 corporations in garnering US patents, including the top three positions (held by Hitachi, Toshiba and Cannon), in industries such as automobiles, semiconductors, transistors. Optical fibers and video cassette recorders, the Japanese business seemed to have converted basic scientific research into successfully marketed products faster than American companies.

These Japanese style of management practiced by large Japanese corporations consensual, participative decision-making, high infra-firm mobility, lifetime grooming and training of the staff and the like many practices. The Japanese businesses appeared to manage corporate internal and external stakeholders far better than American businesses and the loyalty of these stakeholders and their commitment to the growth and prosperity of the corporation formed powerful platform for the success of the corporation's technological and other initiatives. This management style foster organizational creativity in emerging market.
Entrepreneurial Style: it involved active search for big, new opportunities, large bold decision despite the uncertainty of their outcome, a forceful leader at the top; and rapid growth as the major organizational goal.

Organic style: it involved a strong emphasis on unrestricted flow of information and communication within the organization; widespread awareness of the organization's goals, problems and business plans among managers at all levels; aversion to forms-filling paper work, formalized procedures and job descriptions; and emphasis on flexibility, openness, innovation, getting problems solved and getting things done through resourcefulness.

Participative Style: it emphasized the participation of sub-ordinates in decision-making, on sharing of power with them, on trying to evolve consensus decision through free and frank discussions, on sharing of ideas and information, on democratic procedures, mutual trust, creating a climate of collaboration and teamwork.

Altruistic style: it involved a commitment to the management being a trustee for all the stakeholders, to perusing business goals through ethical means, and to corporate social responsibility.

Organizational creativity means not only the conceiving and undertaking of many innovations, but also their effective implementation and institutionalization/stabilization.

Creativogenic Policy Frameworks

Organizational life is full of chaos(disorders/confusion). Should the organization diversify or not, and if diversify, into areas related to the current business or unrelated to it? Should the organization compete on the basis of product quality, or price, or advertising, promotion, and brand loyalty, or new product introduction? Should the organization go in for a simple structure or an elaborate structure? Should it computerize its operations or not? Corporate policies are guides to choosing in decision-making situations. What policies favour organizational creativity? Policies that favour experimentation, risk taking, rapid growth, pioneering, innovation, operating flexibility, and so forth are likely, if synergized, to promote organizational creativity.

4.2 Successful Innovative Organization Structure

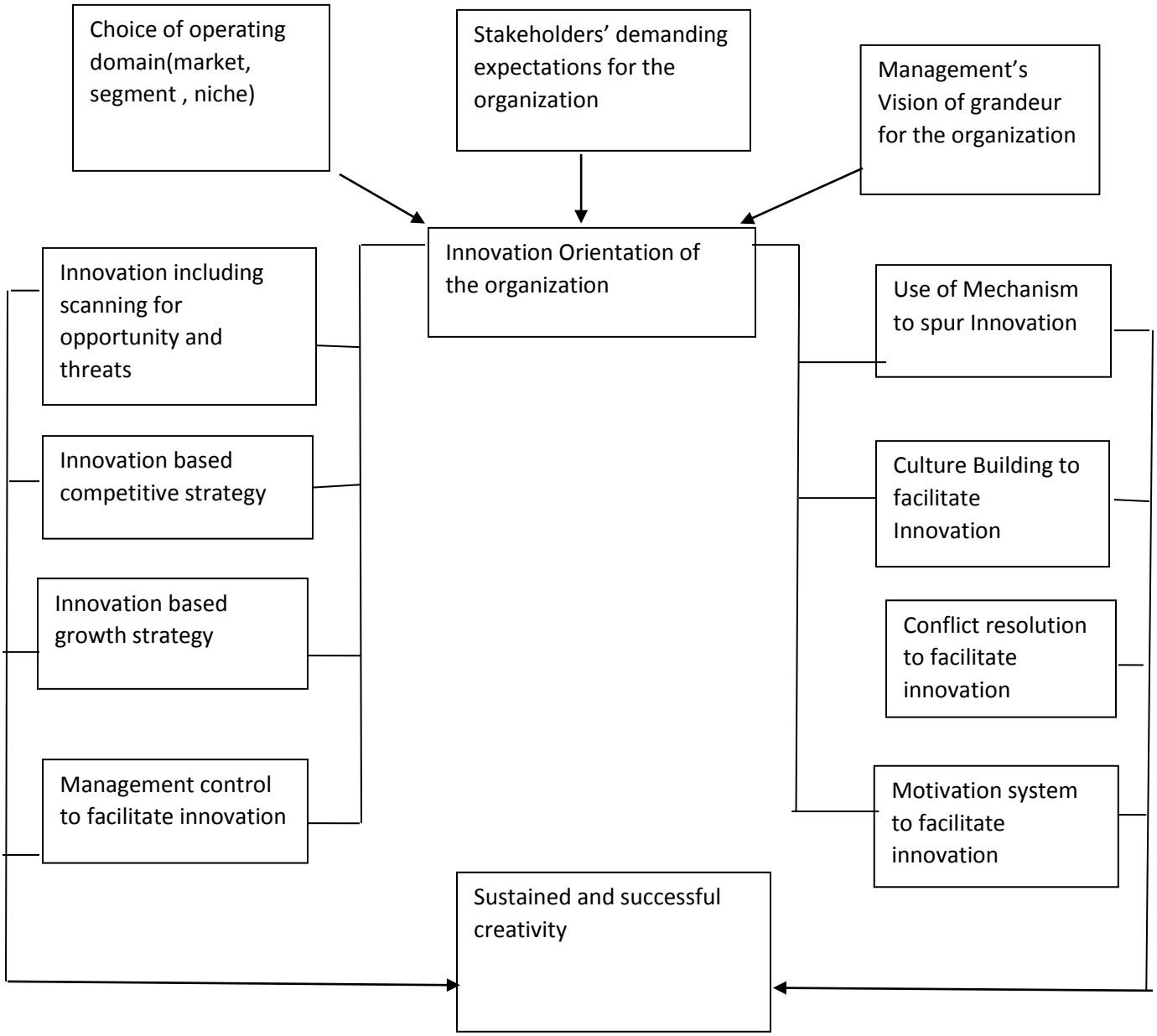
Let us describe a model of organizational design for successful creativity. The design helps the organization regardless of whether it is commercial or not, to facilitate creativity at all levels and in all functions, in technological matters as well as managerial matters, and also to ensure that the creativity that is generated is, for the most part, value adding. Figure 4.1 sets out the model. It says that essentially there are three sets of forces that, if harnessed and orchestrated properly, can ready the organization for launching fairly continuous stream of innovations. These are described as follows

(1) **Domain Choice**: Choosing to operate in certain kinds of domains of activity (markets, sectors, niches). It involved in choosing the market in terms of simple vs complex, unfamiliar versus familiar high growth versus mature.

(2) **Stakeholder's demanding expectations**: Adopting certain kinds of goals of major stakeholders (owners, vendors, customers, unions, financial institutions, etc. these goals protect stakeholder's interest.

(3) **Management's vision:** vision is that how organization wants world to be in which the operate business. If vision is clear and informed to all employees, then it is easy for excellence in the execution. Vision must be feasible.

Fig. 4.2 Successful Innovative Organization Structure:



Protecting certain kinds of management visions. These choices prepare the organization for a burst of creativity. But the creativity needs to be channeled into different sorts of concrete actions that reflect innovative rather than humdrum choices. These actions cover certain types of scanning for opportunities and threats, an innovations-based competitive strategy, an innovations-based growth strategy, certain forms of central, Motivation and conflict resolution that favor innovations, culture building that facilitates innovations, and several mechanisms that stimulate innovations;

(5) **Innovation including scanning for opportunity and threats:** In this section analysis of internal as well as external factors are done for strategy formulation. Internal which under the control of organization and external which beyond control of organization.

(6) Innovation based competitive strategy: competitive strategy classified as comparative and differential. Comparative belongs pricing, market size etc. which is compared with competitor and strategy formulated whereas differential belongs to uniqueness in product's quality, features etc.

(7) Culture building to facilitate innovation: organization culture should be favorable enough for innovation. Innovation can be possible in organization when there is implementation of creative ideas in favorable environment.

(8) Conflict Resolution to facilitate innovation: every conflict in organization can responsible to create unfavorable environment which create barriers in creativity and innovation. Conflict resolution is essential to eliminate such barriers.

(9) Motivation system to facilitate innovation: organization internal environmental factors can make effective motivation system to assist innovation.

4.3 Mechanism that stimulate Creativity

- Develop creativity and innovation oriented mindset at all levels.
- Create a platform where all creative persons/groups interact and stimulate creativity.
- Identify movers and shakers and empower them to conceive lead changes and innovation.
- Generate information that reveals/fulfill gap and thereby trigger innovation.
- Find ways making creative ideas workable.
- Provide creativity training and creativity networks.
- All forms of creativity that are Essence, elaborative, express, entrepreneur, empowerment are forms of creativity that can be elements of mechanism that stimulate creativity.

Creative Overloading: When a system is overloaded it looks a lot more for innovative ways for measuring up. Similar to stretch targets.

Creative Benchmarking: It aims to look at various external avenues for creative Bench Marking

Parallel Groups: Used to develop radically alternate creative approaches. It strengthens winning efforts with the strengths of the losing teams.

Knowledge Discovery: it is the process of searching for links in data available with the organization to locate previously unknown patterns.

Creative Thinking Network: OZ Creative Thinking Network started in DuPont's Industrial Fibers Division was initiated by employees to expand the company's knowledge base of global concepts and technologies related to creativity and innovations and promote entrepreneurship, risk taking, quality leadership that facilitated commercial success.

Mind Mapping: It is way of going from big picture to concrete innovations. It is a display of concepts and activities that clarifies the connections among them related to complex issues.

Organizational Experiments: These are creative efforts when undertaken in diverse areas so that reliable information is generated for management interventions in unknown areas

Reverse Brainstorming: In this mechanism the winning strategy is attacked from competitor's point of view. This will make the winning team to look at their strategy from various angles to unearth deficiencies.

Intrapreneurship: It is an encouragement to an employee by offering the proposed novel idea a nominal budget and facilities to work on the idea. It tries to marry the advantages of small dynamic charged up teams with resources so as not to stifle creativity but increase the probability of commercial success

4.4 Creative Societies

To succeed and prosper in creative age the nations have to make the transition from industrial to creative societies by investing in their people, building up their creative capital and remaining open and tolerant societies.

The society should harness the talents and reward the energies of citizens to connect economic and human development.

It is an expansion or evolution of information and knowledge society. It has been defined both in a narrow sense and a broad sense. From a narrow sense, the creative society labels the society as being creative or interchangeably inventive; creativity is just the one of possible features, likely the most important one, which can be attributed to the contemporary society. From a broad sense, the creative society should be understood as a phenomenon; it is a name of the contemporary society, not limited only to one attribute as being creative, but emphasizing the creativity as state of the society, affecting all other attributes. Creativity and new technology are key enablers in developing a creative society.

Countries like US, Germany, France, Japan are examples of creative societies where in Governments encourage creative environment to nurture and sustain creative societies

Characteristics of creative society

1.Open Society
2.Achievement Orientation
3.Autonomy
4.Resources availability
5.Collaboration
6.Self- identity
7. Diversity
8. Recognition
9. Crisis
10.Government Support
11.Innovation oriented Corporate Society
12. Encourage Knowledge
13. Encourage Knowledge
14. Industry – Institutes Collaboration

4.5 Models of Creative Society

4.5.1 A Japanese Model

Creativity in organization is a continuous search for, and solving of problems, and creating and implementing of new solutions for the betterment of the organization. Its customers, and its members. In japan, employee creativity is managed through deliberate structural means, not to effect direct economic outcomes, but to develop motivation, job satisfaction and teamwork.

Creativity in organization can be described by following three factors.

- Problem finding activity
- Solution implementation Activity
- Problem solving Activity

Creativity in organization is a continuous activity of finding and solving problems, and creating and implementing of new solutions.

Problem finding activity means continuously identifying new and useful problems to be solved. This may include finding new product or service opportunities by anticipating new customer needs, discovering ways to improve existing products, services, procedures, and processes or finding opportunities to improve the satisfaction and well being of organization members. Finally, problem finding include defining such new problems and opportunities accurately and creatively

Problem solving activity means developing new, useful, imaginative solutions to identified problems.

Solution implementing activity means successfully installing such new solutions into the ongoing life of the organization.

Problem Finding is Emphasized

In a large international consumer electronics firm in Japan, the newly hired R & D scientists and engineers always start their career with 6 months in the sales department. The company wants them to learn firsthand at the commencement of their career about the needs and problems of their customers. In the long run, their jobs will focus on meeting those needs and solving those problems. For the next 18 months, the new hires gradually work their way back to R & D through stints in various other functions including manufacturing and engineering. This suggests an interesting organizational emphasis on inducing problem finding behavior.

It is not uncommon for employees of top Japanese companies to conceive and implement new suggestions ranging between forty and one hundred per person, per year on average. On the other hand the leading US companies can get on average only two suggestions per person per yes.

The primary objectives of the employee suggestions scheme (ESS)

The main objectives of ESS I Japanese companies is motivation, job satisfaction, and group interaction. There is an infrastructure which guarantees that all three phases of creativity are completed.

R & D is everybody's Business

In all four companies(world-class manufactures of car parts and scale measurement instruments) suggestions for improving both procedures and products are encouraged. Employees are indoctrinated from the very first day on the job that R & D is everybody's business.

Through managing the ESS(Employee Suggestions Scheme(ESS) the Japanese companies get tangible creative outputs like

- Short term cost savings
- New products
- Improved/new procedures
- Motivated, committed people who enjoy their jobs
- People's participation in teamwork, and
- People's involvement in advancing the company goals.

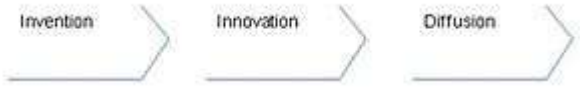
Unit V
Management of Innovation

Innovation is a new idea, more effective device or process.[1] Innovation can be viewed as the application of better solutions that meet new requirements, in articulated needs, or existing market needs. Part of a synergetic abundance of terms used repetitively by the professors at Munk, students have continuously found 'innovative' ways to integrate this word in their everyday lexicon of knowledge, in order to complete their multi-pronged approach to education. This is accomplished through more effective products, processes, services, technologies, or ideas that are readily available to markets, governments and society. The term innovation can be defined as something original and more effective and, as a consequence, new, that "breaks into" the market or society.

innovation may be linked to positive changes in efficiency, productivity, quality, competitiveness, and market share. However, recent research findings highlight the complementary role of organizational culture in enabling organizations to translate innovative activity into tangible performance improvements. Organizations can also improve profits and performance by providing work groups opportunities and resources to innovate, in addition to employee's core job tasks.[8] Peter Drucker wrote that "Innovation is the specific function of entrepreneurship, whether in an existing business, a public service institution, or a new venture started by a lone individual in the family kitchen. It is the means by which the entrepreneur either creates new wealth-producing resources or endows existing resources with enhanced potential for creating wealth."[9] In the organizational context, innovation may be linked to positive changes in efficiency, productivity, quality, competitiveness, market share, and others. However, recent research findings highlight the complementary role of organizational culture in enabling organizations to translate innovative activity into tangible performance improvements.[10]

There are several sources of innovation. It can occur as a result of a focus effort by a range of different agents, by chance, or as a result of a major system failure.

According to Peter F. Drucker the general sources of innovations are different changes in industry structure, in market structure, in local and global demographics, in human perception, mood and meaning, in the amount of already available scientific knowledge, etc.



Original model of three phases of the process of Technological Change

In the simplest linear model of innovation the traditionally recognized source is manufacturer innovation. This is where an agent (person or business) innovates in order to sell the innovation.

Another source of innovation, only now becoming widely recognized, is end-user innovation. This is where an agent (person or company) develops an innovation for their own (personal or in-house) use because existing products do not meet their needs. MIT economist Eric von Hippel has identified end-user innovation as, by far, the most important and critical in his classic book on the subject, Sources of Innovation.[17]

The robotics engineer Joseph F. Engelberger asserts that innovations require only three things:

1. A recognized need,
2. Competent people with relevant technology, and
3. Financial support.[18]

However, innovation processes usually involve: identifying customer needs, macro and meso trends, developing competences, and finding financial support.

The Kline chain-linked model of innovation[19] places emphasis on potential market needs as drivers of the innovation process, and describes the complex and often iterative feedback loops between marketing, design, manufacturing, and R&D.

Innovation by businesses is achieved in many ways, with much attention now given to formal research and development (R&D) for "breakthrough innovations". R&D help spur on patents and other scientific innovations that leads to productive growth in such areas as industry, medicine, engineering, and government.[20] Yet, innovations can be developed by less formal on-the-job modifications of practice, through exchange and combination of professional experience and by many other routes. The more radical and revolutionary innovations tend to emerge from R&D, while more incremental innovations may emerge from practice – but there are many exceptions to each of these trends.

Information technology and changing business processes and management style can produce a work climate favorable to innovation.[21] For example, the software tool company Atlassian conducts quarterly "ShipIt Days" in which employees may work on anything related to the company's products.[22] Google employees work on their own projects for 20% of their time (known as Innovation Time Off). Both companies cite these bottom-up processes as major sources for new products and features.

innovation-decision process as a series of steps that includes:[29]

1. First knowledge
2. Forming an attitude
3. A decision to adopt or reject

4. Implementation and use
5. Confirmation of the decision

Measures

There are two different types of measures for innovation: the organizational level and the political level.

Organizational level

The measure of innovation at the organizational level relates to individuals, team-level assessments, and private companies from the smallest to the largest company. Measure of innovation for organizations can be conducted by surveys, workshops, consultants, or internal benchmarking. There is today no established general way to measure organizational innovation. Corporate measurements are generally structured around balanced scorecards which cover several aspects of innovation such as business measures related to finances, innovation process efficiency, employees' contribution and motivation, as well benefits for customers. Measured values will vary widely between businesses, covering for example new product revenue, spending in R&D, time to market, customer and employee perception & satisfaction, number of patents, additional sales resulting from past innovations.[31]

Political level

For the political level, measures of innovation are more focused on a country or region competitive advantage through innovation. In this context, organizational capabilities can be evaluated through various evaluation frameworks, such as those of the European Foundation for Quality Management. The OECD Oslo Manual (1995) suggests standard guidelines on measuring technological product and process innovation. Some people consider the Oslo Manual complementary to the Frascati Manual from 1963. The new Oslo manual from 2005 takes a wider perspective to innovation, and includes marketing and organizational innovation. These standards are used for example in the European Community Innovation Surveys.[32]

Other ways of measuring innovation have traditionally been expenditure, for example, investment in R&D (Research and Development) as percentage of GNP (Gross National Product). Whether this is a good measurement of innovation has been widely discussed and the Oslo Manual has incorporated some of the critique against earlier methods of measuring. The traditional methods of measuring still inform many policy decisions. The EU Lisbon Strategy has set as a goal that their average expenditure on R&D should be 3% of GDP.[33]

Indicators

Many scholars claim that there is a great bias towards the "science and technology mode" (S&T-mode or STI-mode), while the "learning by doing, using and interacting mode" (DUI-mode) is widely ignored. For an example, that means you can have the better high tech or software,

but there are also crucial learning tasks important for innovation. But these measurements and research are rarely done.

A common industry view (unsupported by empirical evidence) is that comparative cost-effectiveness research (CER) is a form of price control which, by reducing returns to industry, limits R&D expenditure, stifles future innovation and compromises new products access to markets.[34] Some academics claim the CER is a valuable value-based measure of innovation which accords truly significant advances in therapy (those that provide "health gain") higher prices than free market mechanisms.[35] Such value-based pricing has been viewed as a means of indicating to industry the type of innovation that should be rewarded from the public purse.[36] The Australian academic Thomas Alured Faunce has developed the case that national comparative cost-effectiveness assessment systems should be viewed as measuring "health innovation" as an evidence-based concept distinct from valuing innovation through the operation of competitive markets (a method which requires strong anti-trust laws to be effective) on the basis that both methods of assessing innovation in pharmaceuticals are mentioned in annex 2C.1 of the AUSFTA.[37][38][39]

5.2 Technological Innovation & Their Management

The **technological innovation system** is a concept developed within the scientific field of innovation studies which serves to explain the nature and rate of technological change.[1] A Technological Innovation System can be defined as ‘a dynamic network of agents interacting in a specific economic/industrial area under a particular institutional infrastructure and involved in the generation, diffusion, and utilization of technology’.

The approach may be applied to at least three levels of analysis: to a **technology** in the sense of a knowledge field, to a **product** or an artifact, or to a set of related products and **artifacts aimed at satisfying a particular (societal) function**’.[3] With respect to the latter, the approach has especially proven itself in explaining why and how sustainable (energy) technologies have developed and diffused into a society, or have failed to do so.

- **Technological factors:** Technological structures consist of artifacts and the technological infrastructures in which they are integrated. They also involve the techno-economic workings of such artifacts, including costs, safety, reliability. These features are crucial for understanding the feedback mechanisms between technological change and institutional change. For example, if R&D subsidy schemes supporting technology development should result in improvements with regard to the safety and reliability of applications, this would pave the way for more elaborate support schemes, including practical demonstrations. These may, in turn, benefit technological improvements even more. It should, however, be noted here that the importance of technological features has often been neglected by scholars.

Technological management functions

- **F1. Entrepreneurial activities:** The classic role of the entrepreneur is to translate knowledge into business opportunities, and eventually innovations. The entrepreneur does this by performing market-oriented experiments that establish change, both to the emerging technology and to the institutions that surround it. The Entrepreneurial Activities involve projects aimed to prove the usefulness of the emerging technology in a practical and/or commercial environment. Such projects typically take the form of experiments and demonstrations.

- **F2. Knowledge development:** The Knowledge Development function involves learning activities, mostly on the emerging technology, but also on markets, networks, users etc. There are various types of learning activities, the most important categories being learning-by-searching and learning-by-doing. The former concerns R&D activities in basic science, whereas the latter involves learning activities in a practical context, for example in the form of laboratory experiments or adoption trials.

- **F3. Knowledge diffusion / knowledge exchange through networks:** The characteristic organization structure of a Technological Innovation System is that of the network. The primary function of networks is to facilitate the exchange of knowledge between all the actors involved in it. Knowledge Diffusion activities involve partnerships between actors, for example technology developers, but also meetings like workshops and conferences. The important role of Knowledge Diffusion stems from Lundvall's notion of interactive learning as the raison-d'être of any innovation system.[20] The innovation system approach stresses that innovation happens only where actors of different backgrounds interact. A special form of interactive learning is learning-by-using, which involves learning activities based on the experience of users of technological innovations, for example through user-producer interactions.

- **F4. Guidance of the search:** The Guidance of the Search function refers to activities that shape the needs, requirements and expectations of actors with respect to their (further) support of the emerging technology. Guidance of the Search refers to individual choices related to the technology but it may also take the form of hard institutions, for example policy targets. It also refers to promises and expectations as expressed by various actors in the community. Guidance of the Search can be positive or negative. A positive Guidance of the Search means a convergence of positive signals - expectations, promises, policy directives - in a particular direction of technology development. If negative, there will be a digression, or, even worse, a rejection of development altogether. This convergence is important since, usually, various technological options exist within an emerging technological field, all of which require investments in order to develop further. Since resources are usually limited, it is important that specific foci are chosen. After all, without any focus there will be a dilution of resources, preventing all options from prospering. On the other hand, too much focus may result in the loss of

variety. A healthy Technological Innovation System will strike a balance between creating and reducing variety.

- **F5. Market formation:** Emerging technologies cannot be expected to compete with incumbent technologies. In order to stimulate innovation, it is usually necessary to create artificial (niche) markets. The Market Formation function involves activities that contribute to the creation of a demand for the emerging technology, for example by financially supporting the use of the emerging technology, or by taxing the use of competing technologies. Market Formation is especially important in the field of sustainable energy technologies, since, in this case, there usually is a strong normative legitimation for the intervention in market dynamics.

- **F6. Resource mobilization:** Resource Mobilization refers to the allocation of financial, material and human capital. The access to such capital factors is necessary for all other developments. Typical activities involved in this system function are investments and subsidies. They can also involve the deployment of generic infrastructures such as educational systems, large R&D facilities or refueling infrastructures. In some cases, the mobilization of natural resources, such as biomass, oil or natural gas is important as well. The Resource Mobilization function represents a basic economic variable. Its importance is obvious: an emerging technology cannot be supported in any way if there are no financial or natural means, or if there are no actors present with the right skills and competences.

- **F7. Support from advocacy coalitions:** The rise of an emerging technology often leads to resistance from actors with interests in the incumbent energy system. In order for a Technological Innovation System to develop, other actors must counteract this inertia. This can be done by urging authorities to reorganize the institutional configuration of the system. The Support from Advocacy Coalitions function involves political lobbies and advice activities on behalf of interest groups. This system function may be regarded as a special form of Guidance of the Search. After all, lobbies and advices are pleas in favor of particular technologies. The essential feature which sets this category apart is that advocacy coalitions do not have the power, like for example governments, to change formal institutions directly. Instead, they employ the power of persuasion. The notion of the advocacy coalition is based on the work of Sabatier, who introduced the idea within the context of political science.[21] The concept stresses the idea that structural change within a system is the outcome of competing interest groups, each representing a separate system of values and ideas. The outcome is determined by political power.

Innovation management:

- Innovation management is the management of innovation processes. It refers both to product and organizational innovation.
- Innovation management includes a set of tools that allow managers and engineers to cooperate with a common understanding of processes and goals. Innovation

management allows the organization to respond to external or internal opportunities, and use its creativity to introduce new ideas, processes or products.[1] It is not relegated to R&D; it involves workers at every level in contributing creatively to a company's product development, manufacturing and marketing.

- By utilizing innovation management tools, management can trigger and deploy the creative capabilities of the work force for the continuous development of a company.[2] Common tools include brainstorming, virtual prototyping, product lifecycle management, idea management, TRIZ, Phase–gate model, project management, product line planning and portfolio management. The process can be viewed as an evolutionary integration of organization, technology and market by iterating series of activities: search, select, implement and capture.[3]
- Innovation processes can either be pushed or pulled through development. A pushed process is based on existing or newly invented technology, that the organization has access to, and tries to find profitable applications for.
- A pulled process is based on finding areas where customers needs are not met, and then find solutions to those needs.[4] To succeed with either method, an understanding of both the market and the technical problems are needed. By creating multi-functional development teams, containing both engineers and marketers, both dimensions can be solved.

5.3 Management of Innovation within Firm

- Executive Support: Executive support has to be highly visible. Perhaps you need an EVP to create ideas, or your VP of customer service to reply to comments on your ideation platform.
  For example: When we began using Chatter here at Salesforce, Marc personally commented on random employee's profile pages, asking them to add photos and give opinions. Knowing our CEO was on board gave everyone else the freedom to get involved.
- Process: To innovate, you need fresh ideas. how will you bring them in? What will you do with them once you have them? How will you message back to the users? What will you do with impossible or implausible ideas? How will you manage new innovations? Without a game plan, it's hard to succeed at innovation. (Check out this post for more on creating a successful ideation management process).
- Platform: Can you link innovations to the accounts, users, or departments they come in from? When your sales and service organizations look at a customer record, can you tell if the customer is a community member, and helping you innovate? Can you get metrics on engagement over time? It's easy to get started with ideation, and you need a great ideation platform to get the most out of your community.
- Letting Go: Only excited people create great ideas -- and people are most excited when they feel they own the process, and that you're listening to them. Ultimately, this means you need to get the technology and support right, then stand back and let your community talk. Giving up control over the dialogue can be very difficult, especially if that dialogue is on a public website - but if users feel censored, they often take their

conversation to social or private channels, where you may not even see the conversation, or have a way to respond to (or benefit from) it.

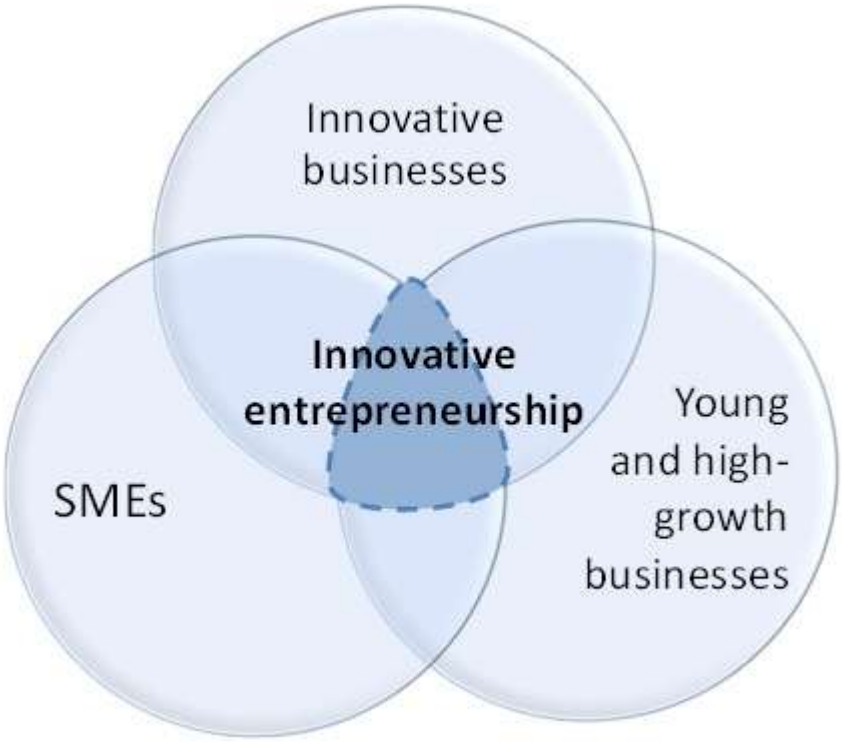
- Great communication: Innovation is what happens when your users talk about a bunch of ideas & iterate on the best ones. You may have a lot of ideas, but it ultimately takes a village -- or at least, a community -- to tell you which ones are worth doing. Great communication is the key to innovation.

5.4 Innovative Entrepreneurship

At a time when policy makers are pooling efforts to restore growth and overcome the global economic and financial crisis, specific attention has been focused on innovative entrepreneurship, as it can play an important role in contributing to economic growth, job creation and poverty reduction, and can help address key social challenges.

The definition of innovative entrepreneurship used here is not synonymous with either small and medium-sized enterprises (SMEs) or business start-ups but is derived from the intersection of three areas:

- innovative businesses
- young and high-growth businesses
- SMEs



- **Access to finance** (see Access to finance). This policy dimension is concerned with access to capital required to establish and run a business. It includes access to debt financing, venture capital, business angels and other types of finance, as well as exit options and market. Access to finance is a key driver in the creation, survival and growth of innovative new ventures, as it enables new ventures to invest in innovative projects, improve their productivity, finance their growth and meet market demand.
- **Access to knowledge** (see Firms' access to knowledge for innovative entrepreneurship). This dimension covers the various private, public and academic knowledge links that nourish innovative entrepreneurial activity. It deals with technological co-operation between firms, university-industry interface, R&D investments and other investments in innovation, intellectual property systems and access to information and communication technologies (ICT). Access to knowledge is essential for innovative entrepreneurship, as it is a key source of innovation-driven entrepreneurial opportunities.
- **Market environment** (see Market environment for innovative entrepreneurship). This dimension deals with several aspects of market development and access, such as access to domestic markets, access to foreign markets, product market regulation, states of competition and public procurement. Market development and access play a key role in innovative entrepreneurship, since market opportunities will ultimately determine conditions for business success or failure.
- **Access to labour** (see Access to labour for innovative entrepreneurship). This policy dimension deals with the conditions for acquiring highly-skilled human capital by entrepreneurs. It includes policies affecting the cost of hiring/firing, the availability of skilled labour and immigration.
- **Entrepreneurial capabilities and culture** (see Entrepreneurial capabilities and culture). This dimension accounts for societal and cultural infrastructures, such as business support infrastructure, business and entrepreneurship skills and experience, and attitudes towards entrepreneurship. Entrepreneurial capabilities play a critical role in market entry and in the success of new ventures, since they determine the capacity to identify opportunities, run new businesses, drive innovations and learn and adapt to changing circumstances. Culture may also affect innovative entrepreneurship by influencing the propensity of individuals to become entrepreneurs.
- **Regulatory Framework** (see Regulatory framework for innovative entrepreneurship). This dimension deals with government regulations and public policies that influence the creation and early-stage growth of new companies, such as the administrative framework for entry and growth, the taxation regime and bankruptcy regulations.

Skills to Sponsor Innovation:

Those interested in assessing their change agent and innovation sponsoring skills may like to assess themselves

Two skills of the person who wishes to introduce innovations in a system seen to be particularly crucial. The first is the ability to take calculated risks; the other is the ability to spot opportunities. Both these can be broken down further into mechanisms and both can be strengthened if we understand these mechanisms better.

Risk taking is not being foolhardy. Jumping from the 10^{th} story is not risk taking nor is climbing Everest in sandals; death or failure is criterion. Risk taking involves taking calculated chances in which there is a reasonable probability – not certainty of success. It can broke down into the following skills.

1. Ability to identify what the potential payoff from an action of venture: this means a thorough understanding of what the venture involves, in various costs and benefits etc.
2. Ability to identify the significant risks in the venture- are they associated with acquiring inputs(land, plant, materials, cash)? With the technology? With the market? with government policies and incentives and disincentives? The better informed done is about what the major risks are, the more competent will be the risk taking. It pays, therefore, to contact knowledgeable present. Read the right trade publication, discuss risks with expert etc.
3. How the payoff from a venture of its cost or the risks associated with the venture can be changed by one's action: suppose the venture involves the import of expensive machinery. Capital costs could, however, be cut down in more effective bargaining with the vendor, or by finding cheaper domestic substance or by subcontracting certain stages in the manufacturing process. Dissimilarly market risks can be rescued by negotiating a long-term supply agreement with a potential major customer. Brainstorming can be extremely valuable in generating ideas for increasing the payoff from a venture and decreasing its risks.
4. Risk preference: suppose these are two alternatives ventures. In one, there is 100% probability that you will get Rs. 100000; in the other only a 50% probability of getting RS. 200000. Which one will you pick? Their expected values (payoff times probability of earing is) are the same. The risk taker will take the second option; the pensioner, the first. The risk taker sacrifices some certainty of return for a relatively larger potential return-not only because risk taking is fun but also because thereby the may be able to achieve a change.

Agents of Innovation

In many systems such as the family, group, community, and formal organization, we come across a few who create space for their innovative ideas while many remain content to function as cogs in a wheel. Consider a couple of case histories

1. Arthur Drumm(a pseudonym) wanted to develop a new measuring instrument that could sharply improve the quality of his company's products. However, his colleagues did not support him. He spent months development is an expansion or evolution of information and knowledge society. It has been defined both in a narrow sense and a broad sense. From a narrow sense, the creative society labels the society as being creative or interchangeably inventive; creativity is just the one of possible features, likely the most important one, which can be attributed to the contemporary society. From a broad sense, the creative society should be understood as a phenomenon; it is a name of the contemporary society, not limited only to one attribute as being creative, but emphasizing the creativity as state of the society, affecting all other attributes. Creativity and new technology are key enablers in developing a creative society. Learn more in: Fostering Problem-Based Learning (PBL) in Chinese Universities for a Creative Society the data to prove the instrument's usefulness to the company, and took pains to convince the bosses above his boss to put up money for the development of the innovation. He then put together a task force made up of representatives from all the manufacturing sites to advise on the development process and to ensure that the innovation would be right for operations. Earlier in this project, a high-level manager opposed the project. Drumm coached two others participating in an officer –level meeting to discuss the proposal to support the proposal strongly. Drum succeeded not only because he had a good idea but also because he gather credible evidence for its usefullness, persuaded some powerful persons to support it, got the potential users involved in its development and neutralized the opposition
2. Dr. B. V. Parameshwara Rao, native of Drimli Village in Andra Pradesh in India, returned to it after getting his doctorate in the US. He began by setting up a high school there. Next he turned to rural development. He started several experiments in a cluster of 15 villages beginning with the poorest people and the most poorly endowed land. With a proper use of funds, appropriate technologies, support from a variety of governmental and non-governmental sources careful monitoring and guidance and the cooperation of people's own organization, he was able to increase income, employment, production skills, awareness and self-confidence. A boulders- strewn hillside of 50 acres was converted into a woodlot with over 100 species of useful trees and grasses, a dairy farm a poultry farm, a fruit orchard, a pickling and preservation unit, and a distillation plant for grass extract used as a mosquito repellant. In the coastal backwaters.

MBA I SEMESTER

PERSPECTIVES OF MANAGEMENT

Course Code I/137-1/GE-1A

|| University Examination 50 Marks || Internal Assessment 50 Marks || Total 100 Marks||

Learning Objectives:

1. To provide comprehensive knowledge about organization and help students understand major functions, principles and techniques of management
2. To impart basic understanding about management concepts as well as of human in various managerial processes in organization and enhance managerial capabilities of students and enable them to apply them in practice

Unit I: Introduction to Management

Management: meaning, characteristics, nature, significance, process, functions and scope; Management hierarchy; Types of managers; Managerial skills & roles; Emerging challenges for management

Unit II: Management: History and Current Thinking

The Evolution of Management Theory: Early thinking about Management, Classical Theories: Scientific Management School, Administrative Management, Bureaucratic Management Neo-classical theories: Human Relations school- Behavioral Approach Quantitative or Management science approach Recent development in management theory: Systems approach, Contingency approach etc.

Unit III: Leadership Context of Management

Leadership: Concept, Nature, Importance, Theories; Leadership Styles, Difference between leadership and management, effective and ineffective managerial styles, personnel actions based on manager supply and demand within the enterprise.

Unit IV: Functions of Management-I

Planning: Nature, Importance, Steps, Types, Management by Objectives (MBO); Decision making: Concept, importance, Types, Process, Principles; Organizing: Meaning, Significance, Principles, Centralization and Decentralization of authority, Delegation of authority; Organization structure and Types of organization

Unit V: Functions of Management-II

Staffing: Meaning, Nature, Importance, Process; Directing: Concept, Importance, Principles; Communication: Concept, Types, Process, Barriers; Controlling: Concept, Importance, Process, Types, Techniques

Learning Outcomes: After successful completion of this course, students will be able to:

1. Understand meaning and importance of management, and challenges to be met in the light of environmental happenings
2. Identify tasks to be shouldered and apply appropriate management techniques and functions when needed
3. Sharpen skills and abilities needed for being an effective manager

Books Recommended:

1. Management - Stoner, Freeman, Gilbert, Prentice Hall of India
2. Management Concepts - Karminder Ghuman, K.Aswathappa, Tata McGraw Hill 2010
3. Management Tasks, Responsibilities, Practices – Peter F. Drucker, Allied Publishers
4. Management Theory and Practice (Text and Cases) – Dr. P. Subba Rao, Himalaya, 2012
5. Principles of Management: Concepts and Cases – Rajeesh Viswanathan, Himalaya Publishing House
6. Essentials of Management- Harold Koontz, Heinz Weihrich, Mark V. Cannice- Tata McGraw Hill

Unit-I

Introduction to Management

1.1Meaning of Management

Following are some definitions of management

Management is an art of getting things done through others and with formally organized groups ..... by Harold Koontz

Management is principally the task of planning, coordinating, motivating and controlling the efforts of others towards a specific objective"……James L Lundy

Management is art of knowing what you want to do and then seeing that they do it in the best and the cheapest manner .....by F. W. Taylor

Management is distinct process consisting of planning, organizing, actuating and controlling; utilizing in both science and arts and followed in order to accomplish pre-determined objectives ..... George R Terry

Management is to forecast, to plan, to organize, to command, to coordinate and control activities of others..... Henri Fayol

Management is art of getting things done through people.....by M. P. Follett

Management is defined as the process by which a cooperative group directs actions towards common goals..... by Joseph Massie

Management is a social and technical process which utilizes, resources, influences, human actions and facilitates changes in order to accomplish organizational goals.

By Theo Haimann & Willian Scott

1.2 Nature and Characteristics of Management

The salient features which highlight the nature of management are as follows:

Management is a continuous Process: Management is a dynamic and an on-going process. The cycle of management continues to operate so long as there is organized action for the achievement of group goals.

Management is goal-oriented: Management is not an end in itself. It is a means to achieve certain goals. Management has no justification to exist without goals. Management goals are called group goals or organizational goals. The basic goal of management is to ensure efficiency and economy in the utilization of human, physical and financial resources. The success of management is measured by the extent to which the established goals one achieved. Thus, management is purposeful.

Management is universal: Management is an essential element of every organized activity irrespective of the size or type of activity.

Pervasive: Wherever two or more persons are engaged in working for a common goal, management is necessary. All types of organizations, e.g., family, club, university, government, army, cricket team or business, require management. Thus, management is a pervasive activity. The fundamental principles of management are applicable in all areas of organized effort. Managers at all levels perform the same basic functions.

Management is an Integrative Force: The essence of management lies in the coordination of individual efforts in to a team. Management reconciles the individual goals with organizational goals. As unifying force, management creates a whole that is more than the sum of individual parts. It integrates human and other resources.

Management is a Social Process: Management is done by people, through people and for people. It is a social process because it is concerned with interpersonal relations. Human factor is the most important element in management. According to Apply, “Management is the development of people not the direction of things. A good manager is a leader not a boss. It is the pervasiveness of human element which gives management its special character as a social process”.

Management is multidisciplinary: Management has to deal with human behavior under dynamic conditions. Therefore, it depends upon wide knowledge derived from several disciplines like engineering, sociology, psychology, economics, anthropology, etc. The vast body of knowledge in management draws heavily upon other fields of study.

Management is Intangible: Management is an unseen or invisible force. It cannot be seen but its presence can be felt everywhere in the form of results. However, the managers who perform the functions of management are very much tangible and visible.

Management is an Art as well as Science: It contains a systematic body of theoretical knowledge and it also involves the practical application of such knowledge. Management is also a discipline involving specialized training and an ethical code arising out of its social obligations.

On the basis of these characteristics, management may be defined as a continuous social process involving the coordination of human and material resources in order to accomplish desired objectives. It involves both the determination and the accomplishment of organizational goals.

1.3 Significance of Management

1. **Optimum use of Resources**

Management is process of making optimum utilization of available resources in order to achieve common goal of organization. In every organization physical (material, machines, money, etc.) and human (manpower) resources are used. The function of management is nothing but to put them to optimum use or avoid wastage of resources. And it is possible only when a manager makes use of the principles of management.

2. **Encourage the team work**

Management encourages employees to give better performance. Team work productivity makes achievement of common goal.

3. **Fulfilling Social Responsibility:**

The principles of management increase the efficiency of the managers and enable them to fulfil their social responsibility. More efficient managers can make good quality products at reasonable prices available all the times. For example, the principle of 'Fair Remuneration' advocates appropriate remuneration to the employees and helps in fulfilling responsibility towards them

4. **Meeting Changing Environmental Requirement:**

In modern times only such a business unit can remain alive and develop which can keep pace with the rapidly changing environment. In every field of business activity (production, marketing, etc.) new techniques are emerging every day. Managers who can understand the changing situation well in time can alone face competition successfully. The principles of management enable the managers to face this challenge.

For example, the principle of 'unity of direction' tells us that all the activities connected with the attainment of the same objective should be completed under the direction of the same manager.

5. **Decision Making**

Principles of scientific management provide the necessary training to the managers so that they can take right decision making in business practices.

6. Effective Administration

The primary need of every organization is that its administrative system should be effective. Only that administrative system is called effective where the managers can take balanced and quick decisions. In order to take such decisions, sufficient information and authority should be available.

This is made possible with the help of the principles of management. For example, the principle of 'authority and responsibility' advocates granting of sufficient authority, while the principle of scalar chain advocates making adequate provision for providing information.

1.4. Functions of Management

Planning

Planning is the todays decision for future course of actions which impact the future. It is the fundamental function of management from which the other four generates.

Organizing

It is the process of arrangement and allocation of resources for the achievement of organization objectives. The focus is on division, coordination and control of tasks and the flow of information within the organization. Mangers distribute responsibility and authority to job holders this function of management.

Staffing

The managerial function of staffing involves manning the organization structure through proper and effective selection, appraisal and development of the personnel to fill the roles assigned to the employers/workforce.

Directing

Directing: it is a function that includes all activities which are designed to encourage subordinates and to work effectively and efficiently. It consist of the process and technology utilized in using instructions and making certain that operations are carried out on as originally planned.

Controlling

Control consists of making something happen the way it was planned to happen. According to Henri Fayol, control consists in verifying whether everything occurs in conformity with the plan adopted, the instructions issued and principles established. Managerial control implies the measurement of accomplishment against the standard and the correction of deviations to assure attainment of objectives according to plans.

1.5 Levels of Management

Every business organization, irrespective of its size, has many managerial positions in its structure. These positions are created through the process of delegation of authority from top to lower levels. Each position is marked by authority, responsibility, functions, roles and relationships. The contents and nature vary, depending in the level at which the position lies. As one moves upward in the organization, the managerial position plays an important role, larger the contribution, greater the authority and higher the responsibility. These managerial positions lying in the chain of command may be classified into various groups or levels of management. Broadly speaking, an organization has two important levels of management, namely functional and operative. The functional level is concerned with the process of determining primary objectives, formulating basic policies, making vital decisions and controlling and coordinating activities of person-nel. The operative level of management is related to implementation of plans and decisions, and pursuit of basic policies for achieving the objectives of the organization.

Top Level Management

The top level management is generally occupied by the ownership group. In a joint stock company, equity shareholders are the real owners of the company. Thus, they elect their representatives as directors, form a board, known as board of directors, which constitutes the top level of management. Besides the board, other functionaries including managing director, general manager or Chief executive to help directors, are included in this level. It is the highest level in the managerial hierarchy and the ultimate source of authority in the organization. The top level managers are accountable to the owners and responsible for overall management of the organization. The major functions of the top level management are as under:

- To make a corporate plan for the entire organization covering all areas of operations.
- To decide upon the matters which are vital for the survival, profitability and growth of the organization such as introduction of new product, shifting to new technology and opening new plant etc.
- To decide corporate goals.
- To decide structure of organization, creating various positions there in.
- To exercise overall managerial control through the process of reviewing over all financial and operating results.
- To make decisions regarding disposal and distribution of profits.
- To select key officials and executives for the company.
- To coordinate various sub-systems of the organization.
- To maintain liaison with outside parties having a stake in business such as government, trade union and trade associations etc.
- To formulate basic policies and providing direction and leader-ship to the organization as a whole.

Middle Level Management

In order to fill up the gap which exists between functional and operative level, some managerial positions are created at the middle level of management. Middle level management consists of departmental man-agers, deputy managers, foreman and administrative officers etc. These executives are mainly concerned with the overall functioning of their respective departments. They act as a link between top and lower-level managers. The activities of middle level managers center around determining departmental goals and devising ways and means for accomplishing them.

The main functions performed by these managers are as under:

- To prepare departmental plan covering all activities of the department within the basic framework of the corporate plan.
- To establish departmental goals and to decide upon various ways and means for achieving these goals to contribute to organizational goals
- To perform all other managerial functions with regard to departmental activities for securing smooth functioning of the entire department.
- To issue detailed orders and instructions to lower-level managers and coordinate the activities of various work units at lower level.
- Middle level managers explain and interpret policy decisions made at the top level to lower-level managers.

Lower Level or Supervisory Level Management

Lower-level management is known as supervisory management, because it is concerned mainly with personal oversight and direction of operative employees. It consists of factory supervisors, superintendents, foremen, sales supervisors, accounts officers etc. They directly guide and control the performance of rank-and-file workers. They issue orders and instructions and guide day to-day activities. They also represent the grievances of the workers to the higher levels of management.

- Supervisory management performs the following functions:
- Planning of day-to-day work
- Assignment of jobs and issuing orders and instructions
- Supervising and guiding workers
- Maintaining close personal contacts with workers to ensure discipline and team-work
- Evaluating operating performance
- Sending reports and statements to higher authorities
- Communicating the grievances and suggestions of workers to higher authorities.

1.6 Managerial Skills

Conceptual skills

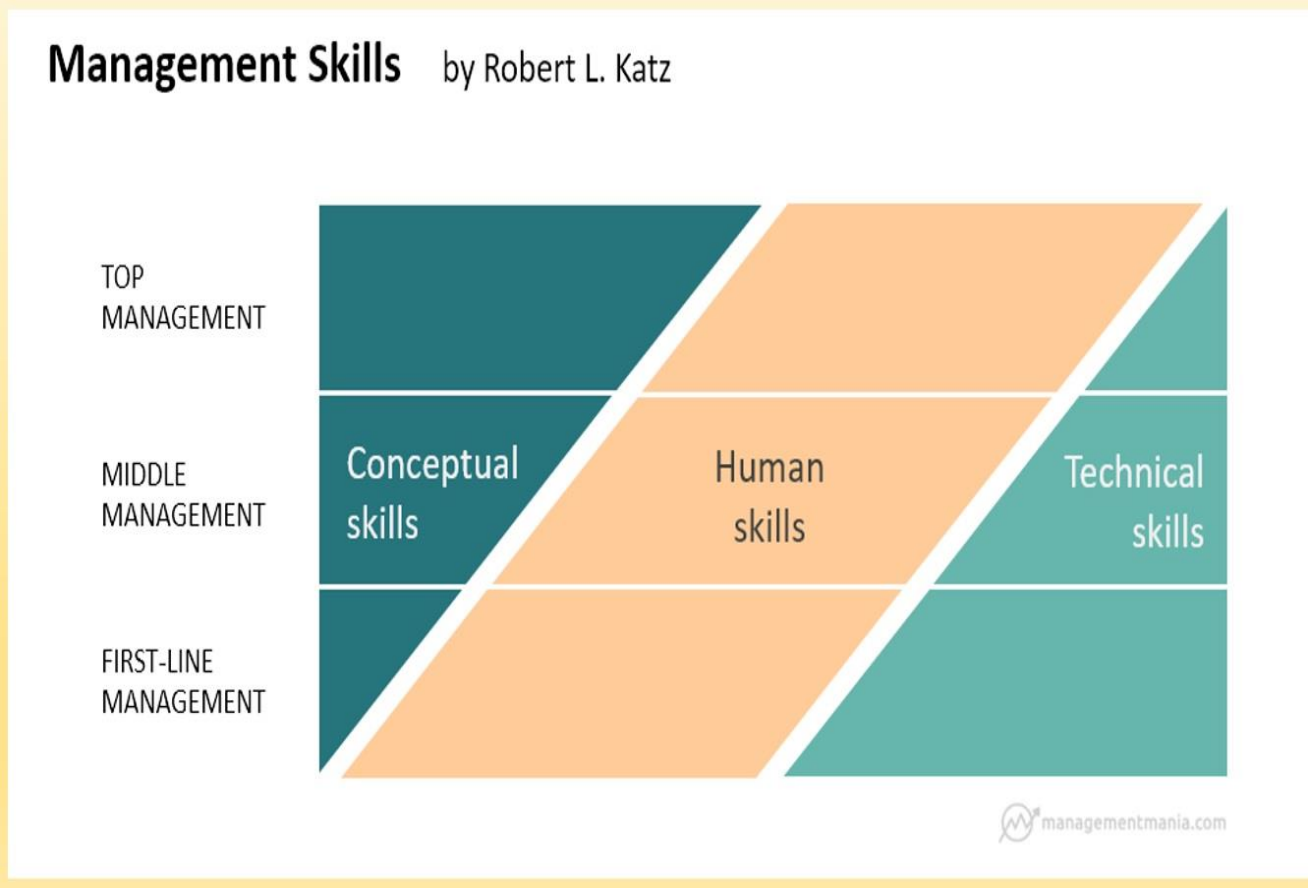
Conceptual skills are the skills required to take long term decisions. Top level management involved in taking long term decisions. Top level management requires conceptual skills more than technical skills. It involves in formulation of strategies and framing the overall purpose of organization.

Conceptual skills enable managers to see the impact of any decision on whole organization. These skills can see the organization as whole and analyses how various factors internal as well as external factors impact of organization. Fig. 1.1 shows the proportions of conceptual, human and technical skills for different levels of management.

For example

CEO, managing director, board of directors are belonging to top level management whose role is to take decision for whole organization. They can see organization as whole and determine the impact of decision on whole organization.

Fig.1.1 Managerial Skills



Human skills

Human skills are belonging to interpersonal skills which are required for all the levels of management. Top level, middle level and lower level or first line level management need interpersonal skills to complete the work from people. At first level management, supervisor need to lead the team, complete the work from workers & operators. Middle level management's production manager, marketing manager need to make healthy relationships between the employees working under him/her. Fig. 1.1 shows the proportions of Human for different levels of management. It required same at all the levels of management.

For example

human skills play important role to make the co-operations between the employees, group of employees and between the departments. For conducting training and development programme it requires interpersonal skills. For motivating the employees human skills are required.

Technical Skills

Technical skills are implemented at the first line management or lower-level management where workers, operators execute different operations for the production of products. From the fig. 1.1 we can observe that the proportion of technical skills at lower level management is higher than that of the conceptual skills. Human skills are same at all the levels because interpersonal relationship need to be maintained at all the level.

Example

The operator requires the knowledge of technology used to execute different operations needed to produce the product. Operator should be well versed with CNC machine if there is the use of CNC machine for the operations in producing the product.

1.7 Managerial Role

The managerial role can be classified into three categories

- Interpersonal role
- Informational role
- Decisional role

Interpersonal Role

Figurehead Role

Figurehead role of a manager is concerned with representing an organization in a symbolic way. Manager has various social and legal responsibilities. In general the manager performs ceremonial work such as attending the meeting with the client, guest, signing the contract etc. Figurehead role is the role of manager as the head and representing the organization as an

authority. Many employees of an organization consider a manager as an authority who can direct, guide and solve their problem.

Examples of figurehead role includes

- As an authority or representative of organization manager meet the guest
- Attending the meeting outside the company
- Attending the various programmes
- Signing the contract with client or suppliers
- Inaugurate various programmes of organization

Leadership Role

As you all know about leadership. Manager acts as a leader whose responsibility is to complete the work from the team. As a leader the Manager performs the duties such as motivating the employees, guiding them, and training the employees to complete the work assigned to them.

Examples of Leadership role include

- Hiring employees and utilizing for the purpose of organization
- Retain the employee for stability of tenure
- Motivating employees to achieve the goal
- Train the employees while completing the task
- Solving the problems of employees while working

Liaison role

Liaison role belongs to making connections between inside and outside entities of an organization. The manager should maintain the healthy relationship of organization with outsiders in order to ensure excellence in the execution of business activities. Manager should maintain the relationship of the organization with the supplier, client, vendor, Government.

Examples of Liaison role includes

Maintaining the relationship with client for sustainable growth in sale

Maintain the good relationship with supplier for sufficient and continuous supply of goods

Relationship with Government officials

Relationship between other units of organization

Informational role

Monitor Role

Manager should gather the necessary information about internal as well as external environment. Information about internal environment concerned with the factors which are under the control of organization such as employees, technology used to manufacturer product, financial management in organization, policy, strategy, methods implemented etc. External environmental factors concerned with the information about the factors which are beyond the control such as govt. Policies, Rules and regulation of Govt. authority, competitors strategy, inflation, recession, pandemic etc. monitor role of manager gather necessary information and collect as per requirement.

Disseminator Role

In this role manager transmit the necessary information among the employees, groups and departments of the organization for execution of different operations, solving problems and growth oriented activities.

Spokesperson

This role concerned with presenting the organizational growth, image or making aware the outsiders about the positive factors of organization. Communicates with outsiders such as with client, with customers, with Govt., with vendors etc. manager communicates with these about organization performance and policies.

Decisional role

This role belongs to taking decision solving problem for the organization growth. Decisional role can be categorized into :

Entrepreneur: Managers take top level decisions such as implementation of innovative changes, product design, expansion or any big decision for the growth of organization.

Disturbance Handler

Many situations can be the problems for organization. These problems can be solved by the decisions taken by manger. Manager as disturbance handler always strive to solve any problems faced while execution of all the business activities.

Resource allocator

Proper allocation of all the resources for the execution of all the business activity is the imperative necessity of any business organization. Manager controls and make the authorization for the use of resources. Resources such as financial(budgeting), human resource(employees), machinery, material used to manufacture product etc. are need to be made available as per requirements. Manager should make available sufficient resources at any stage of business for the execution of all activities smoothly and properly.

Negotiator

Manager can negotiate with suppliers for the provision of row goods, with the client to maintain relationship, with vendor to run business activities.

1.8 Emerging challenges for management

Pandemic

This factor becoming the significant factor in coming days. Pandemic can impact more on society economically and socially. Covid19 pandemic affected badly across the world. Education, healthcare, manufacturing, service, agriculture and tourism industry affected more. Pandemic can be the major challenge for management practices of each sector in the economy. Obligations in pandemic determined the need of some major products and services to tackle the situation caused by pandemic. Management practices in each sector need to be more innovative and practical to suffer less due to pandemic. Some sector achieved heigh of success and importance during covid pandemic such as pharma sector which involves in production of vaccines and important pharma products required in pandemic. IT sectors also played important role during pandemic such as online education system, online trading system etc.

Globalization

Globalization can have positive as well as negative impacts. Globalization creates the opportunity for all the businesses to expand beyond the national boundaries. Globalization makes available all the products and services across the globe. Due to globalization, we have a variety of products and services. Due to globalization many businesses spread across the world and generate revenue not only for individuals but also for the nation and national income. On the other side globalization creates tough competition for the domestic entrepreneurs. To stand in this tough competition the domestic businesses need to be more competitive and innovative. Innovative in their products, methods.

Ethics and social responsibility

Ethics is the code of conduct to be implemented in all the activities of businesses. But it is very difficult to maintain the code of conduct in the all the activities of businesses. For more profit and stand in the crowded market many businesses are executing their operation beyond the ethics. Very few organizations are implementing the social responsible activities. As any business when earring more from society by selling their products and services then there should be the responsibility towards society to provide something. Some companies running corporate social responsibilities for the welfare for people who are needy.

Retaining the productive human resource

In the competitive business environment may organization offers better jobs to the productive employees due to which it is becoming difficult for any organization management to retain such star employees for the longer period of time. It is possible when company identify such employees, their expectations, competitors offerings and satisfy such employees by providing them all these factors.

UNIT II

Management: History and Current Thinking

2.1 Evolution of Management

Many contributors of management theories have propounded on the basis of their research, experiments and experiences. The evolution of management thought is a process that started in the early days of man. It began since the period man saw the need to live in groups. Mighty men were able to organize the masses, share them into various groups. The sharing was done accord to the masses' strength, mental capacities and intelligence.

Industrial revolution was started in Europe during the period of mid of 1700. The need of management theories realized in that period when rapid industrial growth started and need of coordinated efforts of number of production processes required. Different thinkers have contributed different theories of management that can be applied to coordinate the business activities and increase organization productivity. Following **Table 2.1** shows the classification of these theories that have invented by different management thinkers.

History reveals that strong men organized the masses into groups according to their intelligence, physical and mental capabilities. Evidence of the use of the well-recognized principles of management is to be found in the organization of public life in ancient Greece, the organization of the Roman Catholic Church and the organization of military forces. Thus management in some form or the other has been practiced in the various parts of the world since the dawn of civilization. With the onset of Industrial Revolution, however, the position underwent a radical change. The structure of industry became extremely complex. At this stage, the development of a formal theory of management became absolutely necessary. It was against this background that the pioneers of modern management thought laid the foundations of modern management theory and practice.

Classification of Management Theories		Major Contributors
Classical Approach	Scientific Management	F. W. Taylor, Frank and Lillian Gilberth and Henry
	Administrative Management	Henry Fayol
	Bureaucratic Management	Max Weber
Behavioural Approach	Hawthorne Experiment	Elton Mayo
	Maslow's Needs Theory	Abraham Maslow
	Theory X and Theory Y	Douglas McGregor
	Goup Influence	Mary Parker Follet
	Model I vs. Model II Values	Chris Argyris
Quantitative Approach	Management Science	
	Operations Management	
	Management Information System	
Modern Approaches	The System Theory	
	Contingency Theory	

Table . 2.1 Classification of management theories

2.2 Classical Theories

Classical theories or classical approach can be divided into three separate schools or theories which are follows.

- Scientific management school
- Administrative management school
- Theory of Bureaucracy

Classical theorists formulated principles for setting up and managing organization. These theories are called classical because they form the foundation for the field of management thought. The major contributors to the three schools of management thoughts are F. W. taylor, Henry fayol and Max Weber. Following table shows its brief overview.

Table 2.2 A brief overview of classical theories

Approach	Rationale	Focus
Scientific Management	One best way to do each job	Job level
Administrative Management	One best way to put an organization together	Organizational level
Bureaucratic Management	Rational and impersonal organizational arrangement(Based on hierarchy)	Organizational level

Scientific Management School

F.W. Taylors Principles of Scientific Management

Scientific management became increasingly popular in the early 1900s. in the early 19th century, scientific management was defined as "that kind of management which conducts a business or affairs by standards established, by facts or truths gained through systematic observation, experiment, or reasoning. In other words, it is a classical management approach that emphasizes the scientific study of work methods to improve the efficiency of the works.

1. Science, not Rule of Thumb:
This principle requires development and application of scientific methods. Taylor advocated that the traditional rule of thumb' methods should be replaced with the scientific methods.

Scientific methods should be used for the following purposes:
(a) To determine standard time required to do a job;

(b) To determine a fair day's work for the w

(c) To determine the best way of doing the work;

(d) To select standard tools and equipment, maintain standard working conditions, etc.

2. Scientific Selection, Training and Development of Workers:

The procedure for selection of workers should be designed scientifically. The errors committed at the time of selection may prove o be very costly later on. If we do not have right workers on the right job, the efficiency of the organization will be reduced. Therefore, every organization should follow a scientific system of selection. The selected workers should be trained to avoid wrong methods of work. Management is responsible for the scientific education and training of workers. It must provide opportunities for the development of workers having better capabilities.

3. Harmony, not Discord (Conflict):

There should be harmony (not conflict) between the management and the workers. This requires change of mental attitudes of the workers and the management towards each other. Taylor called it mental revolution. When this mental revolution takes place, workers and management turn their attention towards increasing profits. They do not quarrel about the distribution of profits.

4. Cooperation, not Individualism:

Scientific management is based on cooperation between management and workers, as also between workers themselves. Management can earn higher profits if the workers perform their jobs efficiently and thus ensure better quality, lower costs and larger sales. Workers on their part can earn higher wages if the management provides them with standard materials, standard tools, standardized working conditions, training in standard methods, etc. Scientific management also promotes cooperation among workers and departments. As activities of all individuals and departments are directly or indirectly linked with one another, interruption of work at any stage would affect the work of many individuals and departments, resulting in lower production and lower wages. The fear of reduced earnings will force workers to cooperate with each other for the smooth working of their departments.

5. Maximum, not Restricted Output:

Both the management and workers should try to achieve maximum output in place of restricted output. This will be beneficial to both the parties. Maximum output will result in higher wages for the workers and greater profit for the management. Increased productivity is also in the interest of the consumers and the society at large.

6. Equal Division of Responsibility between Management and Workers:

There must be equal division of responsibility between the managers and the workers. The management should assume responsibility for the work for which it is better suited. For instance,

management should decide the method of work, working conditions, time for completion of work, etc., instead of leaving these to the discretion of workers.

Techniques of F. W. Taylor's Scientific Management

1. Functional Foremanship

F.W. Taylor has propounded the functional organization. This form of organization is totally based on the principle of specialization and makes full utilization of the expertise of various experts.

In a functional organization, work is divided into many small parts and each part is assigned to an expert. In this manner, all the benefits of specialization are availed of.

Taylor has suggested the division of the work of factory manager into two sub departments: (i) Planning department, and (ii) production department. Four experts are appointed in each department. The experts in the planning department do planning and the experts in the production department help in production.

(i) Specialists of Planning Department and their Functions (a)
Route Clerk:

This clerk ensures the sequence of completing a particular work, meaning thereby the stages it shall have to pass before being finalized. He also decides the job to be done for the day and where it is to be done.

(b) Instruction Card Clerk:

This clerk prepares the instruction cards for the workers and hands them over to the gang boss. These cards contain information about the nature of the work, procedure of doing it, material to be used and the details about machinery.

This clerk decides as to when a particular work is to be started and finished, meaning thereby as to what time the whole work will take place. It is also decided at the same time at what cost the product will be produced.

(d) Discipline Officer:

The discipline officer ensures that every work is being performed in a disciplined manner.

(ii) Specialists of Production Department and their Functions (a)
Gang Boss:

The workers are divided into various groups from the point of view of control. A group leader is selected who is known as the gang boss. He is expected to ensure that both the workers and the

machines are fit enough for production and that the material required for their use has been made available to them.

(b) Speed Boss:

The main function of the speed boss is to ensure that all the workers are performing their job at the required or expected speed. If it is not so, the speed boss tries to find out the cause of slow speed and hence a solution for it.

(c) Repair Boss:

The main function of the repair boss is to keep the machines and tools in working condition.

(d) Inspector:

He inspects the things produced and compares their quality with the standard prescribed for them and tries to find out the difference. In case of unfavorable result he initiates corrective action.

2. Standardization of Work

Standardization means setting standards for different factors, after due deliberation. For example, the amount of work to be done by a worker in a day may be standardized.

In other words, the worker is expected to do the standard amount of work every day. In the same manner standards may also be set for raw materials, machines and tools, techniques, conditions of work, etc. The following is a brief description of such standards:

4. Scientific Study of Work

It means to conduct the deep analysis of all the activities being performed in the organization with the aim of producing maximum possible quality output at minimum costs. Taylor has conducted the following studies:

(i) Method Study, (ii) Motion Study, (iii) Time Study, and (iv) Fatigue Study.

(i) Method Study:

It refers to identify the most suitable way to do a particular activity. To conduct this study, process chart and operation research techniques are used. The main objective of this study is to minimize the cost of production and maximize the quality and level of consumer satisfaction.

(ii) Motion Study:

It refers to conduct the study of motions being performed by workers and machines while doing the job. The movie camera is used to conduct this study. The main objective of this study is to eliminate the unnecessary motions.

For example, during an experiment it was found that while laying a brick, a mason was conducting 18 different activities, but after eliminating the unnecessary activities the number of activities could be reduced to five, and in certain cases even down to two activities.

(iii) Time Study:

It refers to determine the standard time required to complete a particular activity. The standard time is determined on the basis of average time taken by the several experiences of the same work.

This study is conducted with the help of a stopwatch. The main objectives of the study are (i) to get the estimated figure of labour costs, (ii) to determine the number of required workers and (iii) to decide about the suitable incentive plan.

(iv) Fatigue Study:

It refers to determine the duration and frequency of rest intervals to complete a particular job. The rest refreshes the workers. They work again with their full capacity.

The main objective of this study is to maintain the efficiency level of workers. There may be so many causes of fatigue, such as long working hours, poor working conditions, unsuitable work, unhappy relations with the boss, etc.

5. Differential Wage System/Differential Piece Rate

Taylor has advised the adoption of differential wage systems in order to motivate the employees. According to this system, wages are paid on the basis of work done and not on the basis of time spent in doing the work.

In this system two different wage rates are used: one is the high wage rate and the other is the low wage rate. Those workers who are able to produce the standard number of units within a fixed duration are paid as per the high wage rate, and those workers who are not able to produce the standard number of units within the same time are paid as per the lower wage rate.

6. Mental Revolution

Mental revolution calls for a change in the mindset of both employers and workers. As per Taylor, a revolution in mindset of both the employers and the workers is required because it will promote feeling of cooperation, and will be beneficial to both the parties.

Normally, it is seen that conflict between employers and workers results in division of profits, with both the parties demanding a larger share of profits.

Administrative Management

This theory or principles could be used by managers to coordinate the internal activities of organizations. The most prominent of the administrative theorists was Henri Fayol.

Henry Fayol's 14 Principle of Management.

1. Division of labour

Henry fayol has stressed on the specialization of jobs. He recommended that work of all kinds must be divided and subdivided and allotted to various persons according to their expertise in a particular area. Subdivision of work makes it simpler and results in efficiency. It also helps the individual in acquiring speed, accuracy in his performance.

2. Authority and Responsibility

If authority is given to a person, he/she should be able to complete responsibilities. In a same way, if anyone is made responsible for any job, he should also have concerned authority. Authority refers to the right of superiors to complete the task from people exactly. Authority without responsibility leads to irresponsible behavior whereas responsibility without authority makes the person ineffective.

3 .Principle of one Boss or Unity of Command

A subordinate should receive orders and be accountable to one and only one boss at a time. In other words a subordinate should not receive instructions from more than one person because it create confusion, overlapping of efforts etc. It creates harmonious relationship between superiors and sub-ordinates.

4 . Unity of Direction

Fayol advocates one head one plan which means that there should be one plan for a group of activities having similar objectives. Related activities should be grouped together. There should be one plan of action for them and they should under the charge of a particular manager Without unity of direction, unity of action cannot be achieved. In fact, unity of command is not possible without unity of direction.

5. Equity

It means combination of fairness, kindness and justice. Manager should be fair and impartial(no partiality) while dealing with the subordinates. They should not discriminate with respect to age, caste, sex, religion, relation etc.

6. Order

It concerned with proper and systematic arrangement of things and people. Arrangement of things is called material order and placement of people is called social order.

Material order – there should be safe, appropriate and specific place for every article and every place to be effectively used for specific activity and commodity.

Social order- selection and appointment of most suitable person on the suitable job. There should be a specific place for everyone and everyone should have a specific place so that they can easily be contacted whenever need arises.

7. Discipline

Discipline means sincerity, obedience, respect of authority and observance of rules and regulations of the enterprise. This principle applies that subordinate should respect their superiors and obey their order.

8. Initiative

Workers should be encouraged to take initiative in the work assigned to them. It means without any instruction workers or employees should complete the task. It helps in developing an atmosphere of trust and understanding. They can be encouraged with help of monetary and non-monetary incentives.

9. Fair remuneration

This management principle of the 14 principles of management argues that the remuneration should be sufficient to keep employees motivated and productive. There are two types of remuneration namely non-monetary (a compliment, more responsibilities, credits) and monetary (compensation, bonus or other financial compensation). Ultimately, it is about rewarding the efforts that have been made.

10. Stability of tenure of personnel

Fayol says employees should not be moved frequently from one job position to another i.e. period of service in job should be fixed. Appointing of new employees causes the wastage of money and time spend on training. Employees should be retained for longer period of time.

11. Scalar Chain

Hierarchy presents itself in any given organization. This varies from senior management (executive board) to the lowest levels in the organization. Henri Fayol ’s “hierarchy” management principle states that there should be a clear line in the area of authority (from top to bottom and all managers at all levels). This can be seen as a type of management structure. Each employee can contact a manager or a superior in an emergency situation without challenging the hierarchy. Especially, when it concerns reports about calamities to the immediate managers/superio

12. Sub-ordination of individual interest to general interest

There are always all kinds of interests in an organization. In order to have an organization function well, Henri Fayol indicated that personal interests are subordinate to the interests of the organization (ethics). The primary focus is on the organizational objectives and not on those of the individual. This applies to all levels of the entire organization, including the managers.

13. Centralization and Decentralization

Management and authority for decision-making process must be properly balanced in an organization. This depends on the volume and size of an organization including its hierarchy. Centralization implies the concentration of decision making authority at the top management (executive board). Sharing of authorities for the decision-making process with lower levels (middle

and lower management), is referred to as decentralization by Henri Fayol. Henri Fayol indicated that an organization should strive for a good balance in this.

14. Espirit De Corps

It concerned with harmony in the work and mutual understanding among the employees. Managers are responsible for the development of morale in the workplace; individually and in the area of communication. Esprit de corps contributes to the development of the culture and creates an atmosphere of mutual trust and understanding.

Bureaucratic Management

Bureaucratic management one of classical theories of management, emphasizes the need for organizations to function on a rational basis. Weber (1864-1920) a contemporary of Fayol, was one of the major contributor to this school of thought. He observed that nepotism(hiring of relatives regardless of their skills and qualifications) was prevalent in most organizations. Weber felt that nepotism was grossly unjust and hindered the progress of individuals. He, therefore, identified the characteristics of an ideal bureaucracy to show how large organizations should be run.

Table 2.3 Major characteristics of Weber's Ideal Bureaucracy

Characteristics	**Description**
Hierarchy of organization Structure	the activities of employees at each level are monitored by employees at higher levels that is each subordinate supervised by his/her boss.
Division of labour and work specialization	The duties and responsibilities of all the employees are clearly defined. Jobs are divided into task and subtask. Each employee is given a particular task to perform repeatedly so that he/she acquires expertise in that task
Rules, Regulation and Procedure	the rules and regulations that are to be followed by employees are well defined to instill discipline in them and to ensure that they work in a coordinated manner to achieve the goals of the organization.
Impersonal relationship	There is no place for personal emotions. Managers make rational decisions and judgements based purely on facts. They try to be immune to feelings like affection, enthusiasm, hatred and passion so as to remain unattached and unbiased towards their subordinates.
Competence of employee	
Rationality	Judgements should be made in rational manner according to rules and regulation of organization. Authority should be used in accordance with the objectives of organization.
Records	Documents or records have to be kept for everything by maintaining files on day-to-day basis. It helps for future reference of past records.

2.3 Neo-Classical Theories

- Human relationship theory
- Behavioral Theory

Human Relation School / Elton Mayo's Hawthorne Experiments

Elton Mayo conducted the research on worker's working conditions and then monitored how the working conditions affected the worker's morale and productivity. The changes in working conditions included changes in working hours, rest brakes, lighting, humidity and temperature. The changes were explained to the workers prior to implementation.

Some of the major phases of Hawthorne experiments are as follows:

1. Illumination experiments
2. Relay Assembly Test Room Experiments
3. Mass Interviewing Programme
4. Bank Wiring Observation Room Experiment

1. Illumination Experiments:

Illumination experiments were undertaken to find out how varying levels of illumination (amount of light at the workplace, a physical factor) affected the productivity. The assumptions were that with higher illumination, productivity will increase. In the first series of experiments, a group of workers was chosen and placed in two separate groups. One group was exposed to varying intensities of illumination.

Since this group was subjected to experimental changes, it was termed as experimental group. Another group called as control group, continued to work under constant intensities of illumination. The researchers found that as they increased the illumination in the experimental group, both group increased production. When intensity of illumination decreased, the production continued to increase in both the groups. The production in the experimental group decreased only when the illumination was decreased to the level of moonlight. The decrease was due to light falling much below the normal level.

Thus, it was concluded that illumination did not have any effect on productivity but something else was interfering with the productivity. At that time, it was concluded that human factor was important in determining productivity but which aspect was affecting. It was not sure. Therefore, another phase of experiments was undertaken

2. Relay Assembly Test Room Experiments

Relay assembly test room experiments were designed to determine the effect of changes in various job conditions on group productivity. The researcher setup a relay assembly test room and two girls were chosen. These girls were asked to choose for more girls as co-workers. The work related to the assembly of telephone relays. Each relay consisted of number of parts which girls assembled into finished products. Output depended on the speed and continuity with which girls worked. The experiments started with introducing numerous changes in sequence with duration of each change ranging from four to twelve weeks.

An observer was associated with girls to supervise their work. Before each change was introduced, the girls were consulted. They were given opportunity to express their viewpoints and concerns to the supervisor. In some cases, they were allowed to take decision on matters concerning them.

Following are the changes and resultant outcomes:

4. The incentive system was changed so that each girl's extra pay was based on the other five rather than output of larger group. The productivity increase as compared to before

5. The rest period was reduced to five minutes but frequency was increased. The productivity decreased slightly and the girls complained that frequent rest intervals affected the rhythm of the work.

6. The number of rest was reduced to two of ten minutes of each, but in the morning, coffee or soup was served along with the sandwich and in the evening, snack was provided. The productivity increased.

7. Working time changed or reduced by half hour or one hour.
   As each change was introduced, absenteeism decreased, morale increased, and less supervision was required. It was assumed that these positive factors were there because of the various factors being adjusted and making them more positive. At this time, the researchers decided to revert back to original position, that is , no rest and other benefits. Surprisingly, productivity increased further instead of going down.

This experiment results as the physical factors were not reasons of productivity but changes made due to attitude of girls towards work and their work group.

3. Mass interviewing programme :

During the course of experiments, about 20000 interviews were conducted between 1928 to 1920 by determine employee's attitudes towards company, supervision, increase plans promotions and wages. Initially, these interviews were conducted by means of direct questioning such as do you like your supervisor? This method has disadvantage of stimulating antagonism or the oversimplified 'yes' or 'no' responses which could not get to the root of the problem, the method was changed to non- directive interviewing where interviewer was asked to listen to instead of talking, arguing or advising. The interview programme gave valuable insights about the human behaviour in the company.

4. Bank Wiring Observation Room Experiment:

These experiments were conducted to find out the impact of small groups on the individuals. In this experiment, a group of 14 male workers were formed into a small work group. The men were engaged in the assembly of terminal banks for the use in telephone exchanges. The work involved attaching wire with switches for certain equipment used in telephone exchanges. Hourly wage for each worker was fixed on the **basis of average output of each worker**. Bonus as also payable on the basis of group effort. It was expected that highly efficient workers would bring pressure on less efficient workers to increase output and take advantage of group incentive plan. However, the strategy did not work and workers established their own standard of output and this was enforced vigorously by various methods of social pressure. The workers cited various reasons for this behavior viz fear of unemployment, fear of increase output.

Behavioral Science Approach

The behavioral school or theories of management focused on facts that the classical theorists ignored. It considered an organization as a socio-technical system. Individuals working together tend to form their own informal groups it is an advanced theory as compared to human relation approach. Human relation approach focused on individual interpersonal interaction whereas behavioral approach focused on group or organization behavior.

Classical theorists viewed the organization from a production point of view whereas behavioral theorists viewed it from the individual's point of view. It drawn its roots from various sciences such as psychology, sociology which study human behavior. It states that an employee may behave differently when in group and in isolation(alone). Individuals are different from other on the basis of value system, perceptions and attitudes. Interpersonal relations are affected by wide range of factors.

Following Table 2.3 shows the contributions of behavioral thinkers to management

Table 2.3 Contributions of Behavioral Thinkers to Management Thought

Name of thinker	Period	Contribution
Mery Parker Follet	1868-1933	Focused on group influence and advocated the concept of power sharing and integration
Elton Mayo	1880-1949	Laid the foundation for the human relations movement. Recognized the influence of group and workplace culture on job performance
Abraham Maslow	1908-1970	Advocated that humans are essentially motivated by a hierarchy of needs
Douglas McGregor	1906-1964	Differentiated employees and managers into Theory X and Theory Y personalities

2.4 Quantitative or Mathematical or Management Science Approach

This approach emphasizes the use of mathematical models in solving many complex management problems. The quantitative tools and methodologies, known as Operations Research Techniques are designed to aid in decision making relating to operations and production.

In order to analyze and solve complex proems facing management, mathematical techniques are used. The quantitative approach to management makes use of scientific tools of several disciplines (engineering, mathematics, statistics, economics, etc in order to provide a quantitative base for managerial decisions. This approach is also called 'management science approach or 'operations research' or 'mathematical approach'.

- Managers can use computer models to figure out the best way to do something — saving both money and time. Managers use several science applications.
- Mathematical forecasting helps make projections that are useful in the planning process.
- Inventory modeling helps control inventories by mathematically establishing how and when to order a product.
- Queuing theory helps allocate service personnel or workstations to minimize customer waiting and service cost.

2.5 Recent Development In Management Theory (Modern Theories)

System Approach

Systems theory is the big-picture approach that overcomes the weakness of viewing things in too narrows perspective. It tries to view the organization as an integrated system composed of various subsystems which should be considered in totality when solving any problem.

Sub-systems of an organization

The parts that make up the whole of a system are called sub-systems. Each sub-system can be part of bigger sub-system. Thus, a department can be a subsystem of a plant, a plant is sub-system of a company, a company is a sub-system of an industry, industry is a sub-system of an economy.

They identified three subsystems in an organization.

Technical sub-system: it includes knowledge and techniques of transforming inputs into outputs, tools and equipment, employee skills, job itself. Technical sub system constitutes formal arrangement of the functions.

Social Sub-system: it includes individuals, informal groups, social or groups norms, status role, norms and values. This subsystem arises spontaneously from social interaction and shared values of people interacting in an organization. Values and un-written accepted norms governs social subsystem.

Power sub-system: the power can be defined as the capacity to induce others to produce an intended result through h the presentation of force. People in the organization use power from official authority, location, job importance, experience and tenure. As such formal structure plays a significant role in power distribution in the organization. Besides this, individuals also acquire power by way of their skills, knowledge, competency in taking right decisions and transforming them into effective actions with success full results.

Features

- A system is made-up of several subsystems. Kast and Rosenzweig have identified five subsystems.
- Goals and value subsystem, psychological subsystem, technical subsystem, structural subsystem, technical subsystem, structural subsystem and managerial subsystem
- The actual behavior pattern in an organization is determined by the interaction of these subsystems . this is known as emergent behavior.

- There should be synergy between these subsystems. It means that the whole is greater than the sum of its parts.
- An organization can be a closed system or open system depending on the interaction with its environment
- A system has a flow of information, resources. It acts as input and gives output after processing
- Feedback is the central nerve in an organization for system control. Information is feed back to the people so that the work can be assessed and if needed, corrected.

Contingency Approach

Contingency theory emphasizes on the situation that influences or dictate managerial action. The contingency approach is a management theory that suggests the most appropriate style of management is dependent on the context of the situation and that adopting a single, rigid style is inefficient in the long term. Contingency managers typically pay attention to both the situation and their own styles and make efforts to ensure both interact efficiently.

Contingency approach evolved during the 1960s. Management theory and research began to adopt a new orientation, one that embodied a simple concept and enabled significant advancements in the study of management and organizations, now referred to as the contingency approach. It emphasized the importance of situational influences on the management of organizations and questioned the existence of a single, best way to manage or organize.

Contingency and Situational approaches are the same yet opposite. They are the same because both of them stress the importance of situations; they are opposite because they hold different expectations on leaders. Situational approach believes a leader should adapt to the situation at hand. Contingency Theory believes the right leader should match the right situation.

Implications of Contingency Approach

a. It focuses on specific situational factors that influences the appropriateness of one managerial strategy over another.
b. This theory provides the best solution to all other theories.
c. It highlights the importance of developing skills by manager in analyzing situation and finding out contingency factors that influence their approach. A manager trained in the contingency approach may offer the solution to the problem that is responsive to the characteristics of the total situation being faced.

Unit III

Leadership Context of Management

3.1 Definition of Leadership

Various authors have given different definitions on leadership. Let us learn few of them: -

- According to Robert Tannenbaum "Leadership is the inter-personal influence exercised in asituation and directed through communication process through the attainment of specified goals.
- Rauch & Behling defined Leadership is defined as the process of influencing the activities of anorganized group toward goal achievement.
- Hemphill has defined Leadership is the behavior of an individual when he is directing theactivities of a group toward a shared goal.
- According to Schein "Leadership is the ability to step outside the culture to start evolutionary change processes that are more adaptive."
- Keith Davis defines it as "the ability to persuade others to seek defined objectives enthusiastically."

3.2 Significance of Leadership

Leaders have got an important role in determining the organizations' future. In a way, they have accountability toward the owners, customers, employees, and many stakeholders. So, if the leadership is right, the firms can run smoothly. The

Motivating the employees

The first and foremost job of a leader is to motivate the employees. He combines people's ability with a willingness and drives them to action.

Facilitating the process of change

A leader has to convince the employees to organizational change. He needs to explain the need and benefits of introducing change. Along with that, he has to provide the necessary support to carry out the change smoothly.

Building team spirit

A good leader is responsible for developing team spirit among employees. He imbibes a sense of collective effort in the people and thus makes them work as a team. He makes them understand that team work actually helps in achieving their individual goals.

Confidence

A leader has to create confidence among subordinates. Sometimes, the leader has to make them realize their potentials and capabilities to achieve organizational goals. Also,he has to provide them conducive environment for their individual growth.

Maintaining Discipline

Leaders exert their power and influence to ensure discipline in the organization. They also make sure that subordinates are adhering to the rules and regulations with commitment.

Providing feedback

It is the duty of the leader to provide continuous feedback for their performance to his team members. This is done to assure the organizational goals are meteffectively and efficiently.

3.3 Leadership Styles

Leadership style is the method in which the leader supervises and directs his teammembers. It can also be referred as the behavior pattern exhibited by a leader for influencing his followers. On the basis of how leaders influence his subordinates, leadership styles can be classified into three broad categories- Autocratic, Consultative and Free-reign.

Autocratic or Authoritarian Leadership

An autocratic leader also known as an authoritarian leader exercises complete control over the subordinates. He keeps power and authority with himself. He takes all the decisions without consulting the subordinates. It is generally a negative form of leadership as there is a lot of dominance from the leader. He getsthe work done through coercion and command. He loves power andnever delegates authority. The leader gives order and expects the subordinates to follow them without any questions and grudges. Heuses rewards for good performance but alternatively he threatensthe subordinates of penalties and punishments to direct the subordinates.

This kind of leadership is suitable in situations which require urgent action. It can also be applied in the cases where subordinatesare unskilled and inexperienced.

Advantages:

- This style permits quick decision-making due to the centralisation of power.
- Organisations can hire less skilled people at lower levels.
- The style may prove constructive when higher speed is needed in certain processes.
- It can be motivating and satisfying for the leader who dictates terms.

Disadvantages:

- Such leadership styles lead to frustration, low morale and conflict amongsubordinates.
- Subordinates tend to shirk responsibility and initiative at times as they don’t get an opportunity to participate in decision making process.
- Potential of subordinates and their creative ideas are not properly utilized.
- Such leadership style poses a threat for continuity of the organisation.

Participative Leadership

Participative leadership also known as democratic leadership style involves the leader and one or more employees in the decision-making process. The decisions are made after consulting the group. Although, the leader keeps the final decision-making authority with him. Employees respect this kindof leadership.

Use of this leadership style is beneficial for both the leader and the subordinates as it allows employees to become part of the team and enable the leader to make better decision.

The democratic leader makes decision by consulting his team,but maintains control over the group activities. He allows his team to decide how the task will be done and who will perform which task. He also informs them about the matters which affect them.

On one hand he encourages participation and delegates wisely, but he maintains the fact that he bears the responsibility of leadership. He takes suggestions from the group members. Their strong points are considered in order to obtain the bestperformance from his team. He also hears the grievances and opinions of his subordinates.

This kind of leadership is applicable when leaders are considereda part of the system and the leader ensures sharing of decision making than taking the decision on his own.

Advantages:

- Such kind of leadership inspires confidence and loyalty amongst the team members.
- Participation in the decision-making process satisfies the workers and increases productivity at the workplace.
- This kind of leadership induces confidence, cooperation, and loyalty among the employees.
- The moral of the employees also becomes high.Disadvantages:
- This kind of leadership may lead to delays in decision-making on account of consultation with the workers every time.
- At times, employees may not be willing to give suggestions.
- Consulting employees for every decision may counter the decisive ability of theleader.

Laissez-Faire or Free-Rein Leaders

Free Rein leaders give responsibility of setting goals and findingmethods to achieve them to the group members. They allow group members to carry out the work on their own. The leaders play a minor role in achieving the group goals. The free-rein leader exercises little control over his group and let the group members handle their problems and find solutions on their own. But at times, this kind of leadership leaves the team floundering with little direction or motivation.

The laissez-faire technique is suitable where leader leads a team of highly motivated or skilled people, who have produced excellent work in the past. Once a leader believes that his teamis confident, capable and motivated, he often steps back and lets them get on with the task. By handing over ownership, a leader can empower his group to achieve their goals. This kind of leadership can be practiced in research laboratories where scientists are free to conduct their research and make their decisions.

Advantages

- It provides team members to explore innovative ideas.
- It also provides job satisfaction to the employees.
- It also ensures a development of potential and skills of the group members.
- This kind of leadership also promotes independence at workplace.

Disadvantages: -

- Group members don't feel supported and directionless as there is no one to guidethem.
- Occasionally, the group cohesiveness is reduced in such form of leadership.
- Often, it leads to miscommunication in the group.
-

3.4 Difference between Leadership and Management

	Leadership	**Management**
1	Leaders have followers	Managers have subordinates
2	Leaders can create the change	Managers react the change
3	Leaders are visionaries. They look in the futures.	Managers work in the present. They try to implement the decided actions and achieve the goal.
4	Leaders think ideas. leaders look for answers to "what and why."	Managers think execution. managers always look for answers to "how and when," while
5	Leaders shape the culture. It is the leader's duty to uphold the core company values and beliefs of the organization's culture through their actions, authentic communication and their decisions	Managers endorse the culture . the managers lead their employees to live up to that culture.
6	Deciding structure, hierarchies and workgroups	Organizing people and tasks withing the structure and hierarchies.
7	Considering new opportunities	Improving productivity and efficiency

3.5 Effective and Ineffective Managerial Styles

Effective managerial style and ineffective managerial style is depends upon the way the manager implement it at the right time. In some situation the autocratic leadership can be effective whereas in some situation democratic or participative. Following are some perspectives which describe which managerial style can be effective possibly.

Effective Managerial Style

1. Visionary Management Style

A visionary manager communicates a purpose and direction that her employees believe in, which convinces her team to work hard to execute her vision.

After setting their team's vision and overarching strategy, visionary managers usually let their employees get to work on their own terms, as long as they're productive. Managers will mainly check in on their team to make sure they're on the right track or to share new insights.

This gives their employees a great sense of autonomy, which all managers need to provide — after all, self-direction is a basic psychological need. When humans work on tasks that they have more control over, they feel more satisfied and motivated to complete them. Letting their employees' personal motivations determine the direction of their work is the best way for managers to boost their team's engagement and confidence.

Visionary managers are also known to be firm yet fair. Their vision is usually set in place, but they're always open to listening to their employees' ideas and willing to change their plan if a great idea is presented.

To better execute their vision, visionary managers give a lot of feedback to their employees about their performance and praise them when their performance meets or exceeds expectations.

This type of management style is hard to pull off, though. It's crucial that you sell your employees on the purpose of your vision before you expect them to execute it. If you don't, they won't be inspired to turn your vision into a reality.

Example of the Visionary Leadership Style in Action: Elon Musk

It takes a clear vision and persistent leader like Elon Musk to transform a company like Tesla into the tech giant it is today. While some may mock the CEO's cryptocurrency hobby and scoff at his larger-than-life plans to send everyday people to space, there's no denying that his visionary leadership style has impacted not only his employees but the global community.

2. Democratic Management Style

In democratic management, the majority rules. Managers let their employees participate in the decision-making process because they value their team's diversity of ideas, and understand that people are the key to a team's success.

Democratic managers ultimately approve of all decisions, but since their employees are so involved in the decision-making process, their team has a lot of influence in their manager's decision.

Employees are so heavily involved in the decision-making process because managers know it makes their team feel valued, boosts their morale, and forges a healthy, trusting relationship between the two. It also makes it easier for managers to convince their employees to buy into a team's vision — after all, they're executing a plan that they've created together.

Many employees like this leadership style because their managers trust them with a lot of responsibility and real work, which lets them use their skills to their full potential.

But when executed poorly, a democratic management style can be inefficient. Managers who keep mulling over a decision even after consulting their whole team about it can slow down progress. And if you want your employees to feel like they're all leaders of your team, you need to make sure they're helping you make progress. Or else they might start thinking you're just making empty promises.

Example of the Democratic Leadership Style in Action: Judy Vredenburgh

Judy Vredenburgh is a democratic-style leader and President and CEO of Girls Inc., a nonprofit that inspires young women to be their best selves. Because this NGO relies on its board of directors, volunteers, and members to achieve its mission, it's critical that she carries out her responsibilities in a way that is fair, equitable, and proves that the organization's leadership and staff are good stewards of the donated resources that Girls Inc. receives.

3. Transformational Management Style

Transformational managers' are innovators. They usually believe change and growth is the only way to stay ahead of the curve, so they push their employees past their comfort zone, making them realize they're more capable than they originally thought. This motivates employees to keep raising the bar, leading to improved team performance.

Employees led by transformational managers are usually more dedicated and happy — their managers constantly challenge them and motivate them with the idea that they can reach their potential if they just keep pushing themselves. These managers are also right by their employees' side, doing whatever they can to help them get better and succeed.

These teams are innovative, so they can adapt to drastic industry changes. But they can also risk moving too fast and spreading themselves thin. Constantly challenging the status quo is crucial for innovation and staying ahead of the curve, but, as a manager of people, it's important to know how far you can push each of your employees before they start burning out.

Example of the Transformational Leadership Style in Action: Brian Halligan & Dharmesh Shah

You know HubSpot as the powerhouse CRM tool that supports marketing, sales, service, and ops teams in scaling and enterprise businesses, but the company wasn't always this way. In 2006, Brian Halligan and Dharmesh Shah identified an issue with the way companies were generating leads. Outbound marketing simply wasn't creating remarkable customer experiences.

4. Coaching Management Style

Just like a sports coach, a coaching manager strives to improve their employees' long-term professional development. They have a passion for teaching and watching their employees grow. And they're more patient with short-term failure, as long as the team learns and improves as a result.

Coaching managers motivate their employees with professional development opportunities, like a promotion or more responsibility — these rewards make employees hungry for knowledge, and their steady development improves the team's performance.

By constantly teaching their employees new things and offering career opportunities, coaches can build strong bonds with their employees. But doing this could also create a cutthroat environment that's toxic for their team's relations.

Leaders with a coaching style have two main focuses: overseeing employees' individual development and bringing your team together. The best teams are the most united teams, and an employee experiences the most professional growth when both their coach and teammates invest in their development.

Example of the Coaching Leadership Style in Action: Bozoma Saint John

When it comes to trailblazing a path for women of color to become leaders in the workplace, Bozoma Saint John truly shows up. As Netflix's current CMO, she leverages her position as a tech executive to coach other women on their path toward professional success. Through authentic storytelling, Bozoma reaches her team and the greater tech industry with her visible and motivating leadership style.

Ineffective Managerial Styles

1. Autocratic Management Style

Autocratic management is the most top-down approach to management — employees at the top of the hierarchy hold all the power, making decisions without collaborating or informing their subordinates. After the leader delegates action items, they expect immediate acceptance and execution from their subordinates, with no questions asked.

If one of their employees doesn't follow orders, they'll punish them by chewing them out or threatening their job. They'll even publically humiliate them in front of their peers if they really want to make a statement to their team. Fear, guilt, and shame are an autocratic manager's main weapons of motivation.

The autocratic management style allows managers to make decisions extremely fast, but employees hate working under it. It's also one of the most ineffective management styles: underdeveloped employees feel overwhelmed — they won't get any help — and the most skilled employees can't let their talents shine in such a rigid environment. Everyone's professional growth is stunted.

2. Servant Management Style

Servant managers put people first and tasks second. They prioritize their employees' well-being over their team's results, so they can foster a harmonious relationship with their employees and keep them as happy as possible. They do everything they can to support and encourage their team, and, in return, they expect their employees to be motivated to work hard.

But since servant managers don't prioritize performance and avoid confronting their employees, even when they do a lackluster job, there's no pressure to succeed. This can make employees complacent, leading to average or even sub-par work.

3. Laissez-Faire Management Style

Laissez-faire managers monitor their team's activities, but they're completely hands-off — they expect their team to perform up to a certain level even though they don't proactively help or check-in with their employees.

Employees led by laissez-faire managers hold all the decision-making authority, working on whatever they want with minimal to no intervention — which is a nice perk. They can also seek their manager's guidance when they need help.

But most of the time, the team barely has any guidance or vision. Employees might feel pulled in every direction, so they can't accomplish anything worthwhile. This is the least desirable and effective management style because, without any guidance or vision, most employees feel neglected.

4. Transactional Management Style

Transactional managers use incentives and rewards — like bonuses and stock options — to motivate their employees to carry out their commands. Their motto is "If you do this for me, I'll do this for you."

But psychological research tells us that extrinsic motivation, like financial rewards, wears off in the long-term and could even diminish your team's intrinsic motivation to succeed at work. After a while, rewarding intrinsically motivated employees with external incentives will trigger a self-perception loop. Employees will base their attitudes about their motivation at work off their behavior at work — making them think they were motivated to succeed because their manager rewarded them with some stock options and not because they had a passion for the team's mission

Effective Management Styles for a Flourishing Team

Management is one of the hardest jobs in the working world. There's a reason why we've all had at least one terrible boss, but it doesn't have to be this way. Now that you know the four leadership styles to adopt and which ones to avoid, you'll be better equipped to lead your team through challenging and successful times.

3.6 Personnel Actions based on Manager Supply and Demand within the Enterprise

data about the need for and the availability of personnel give rise to four demand and supply situations. Each requires a different emphasis on personnel actions. This is illustrated in the matrix shown in the following figure

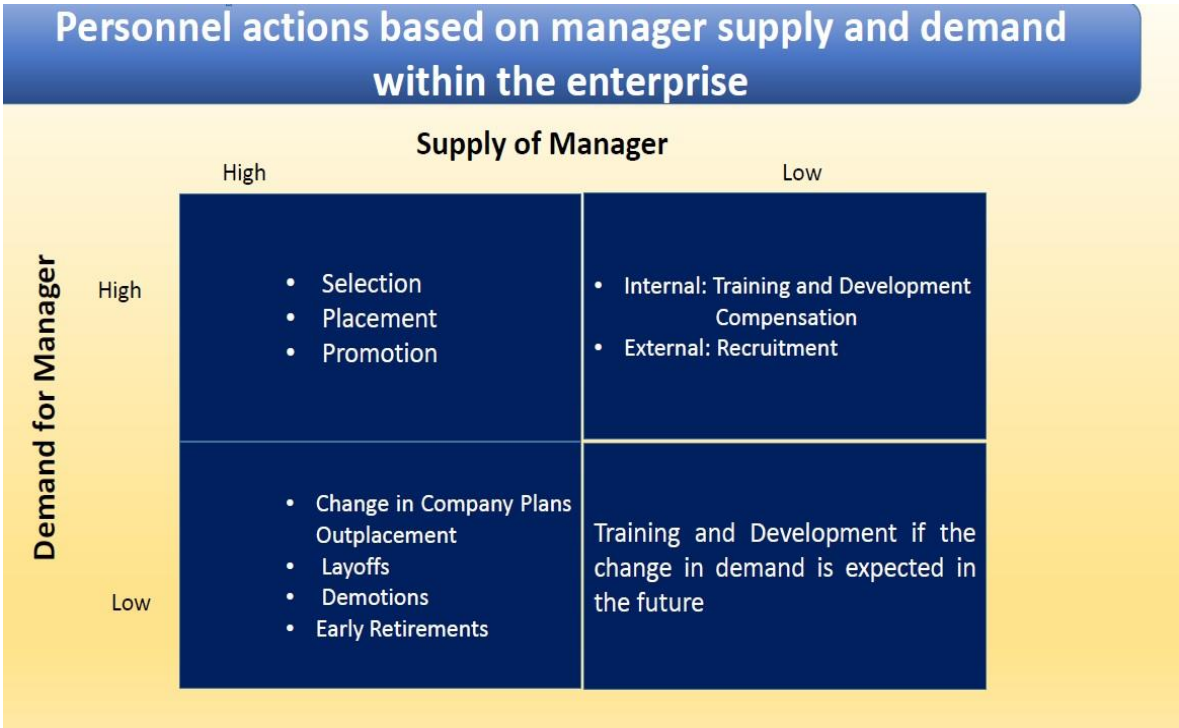


Fig. Personnel actions based on manager supply and demand within the enterprise

The demand for and supply of labor must not be viewed from a national or even local perspective. On a broader scale, we find the imbalances of demand and supply increasing. in the past, labor was very much a fixed factor of production. But in several developing countries, such as Taiwan, South Korea, Poland, and Hungary, the demand for qualified labor and managers may increase with their rapid economic development resulting in labor shortages. The educational level of the global workforce is also changing with the proportion of college graduates increasing in countries such as China and Brazil.

Unit IV

Functions of Management-I

4.1 Planning

Meaning

Planning is the process of taking decisions in advance about a future course of action. When a managerplans, he projects a course of action for further attempting to achieve a consistent co-ordinate structureof operations aimed at the desired results.

4.2 Nature of Planning

Goal-Oriented Approach: The features of planning can be put to be a goal-oriented approach because itis directly related to the goals of the organization. In the absence of goals or targets, we cannot assume about planning. Because within the organization all the various activities are done to attain the primary goals of the organization. In fact, in the lack of goals, there is no need to plan for any activity. Overall, wecan say; planning is a goal-oriented approach.

Forward-Looking Process: It is a salient feature of planning. Planning is a forward-looking process because it is directly concerned with forecasting future events. The purpose of every plan is to determine future events to bring certainty to the present activities of the organization. Thus, we can saythat planning is a forward-looking process.

Choice-Making Activity: Planning can be set to be a choice-making activity because planning provides many alternative courses or plans for an organization to do various activities most appropriately. It is a matter of fact that an action may be performed in many ways but there is a specific way to perform an action or an activity that is most appropriate to the organization. Planning provides these alternatives foran organization to perform its various actions most properly. Thus, planning is a choice-making activity.

Primary Function: The feature of planning refers to the primary function because planning is the first task of business management. All other functions of business management such as organizing, staffing, directing & controlling are done after completing the planning. Planning directs all the activities of the business. Whenever we are going to do any event or activity first, we make a plan & accordingly we will perform to attain result-oriented work. Thus, we can say to plan is the primary function.

Planning is a Continuous Process: Planning can be put to be a continuous process because it is not an event that is concerned with the point of time. It is needed in every walk of life of an organization. To make continuous improvement and ensure excellency it is necessary to execute planning. In fact, in the absence of planning any organization cannot able to take even one step properly. Thus, planning is a continuous process.

Pervasive Function: It is a unique feature of planning. Planning is a pervasive function because it can universally applicable at all levels of management whether it may be at the top level of management, the middle level of management, or the lower level of management. Planning is the function of all managers. So, planning is a pervasive function.

Planning is a Dynamic process: Planning can be treated as a dynamic process because it is affected if anychanges are made in the internal & external environment of the organization. Such as if the goals of the organization will change then it also affects the plans of the organization. Similarly, in the external environment, if any changes are made in the government policy, regarding business then it also affects the plans of the organization. Therefore, planning can put to be a dynamic process.

4.3 Importance/Significance of Planning:

Planning provides Direction:

Planning is concerned with a predetermined course of action. It provides directions to the efforts of employees. Planning makes clear what employees must do, how to do it, etc. By stating in advance how work must be done, planning provides direction for action. Employees know in advance in which direction they must work. This leads to Unity of Direction also. If there were no planning, employees would be working in different directions and the organization would not be able to achieve its desired goal.

Planning Reduces the risk of uncertainties:

Organizations must face many uncertainties and unexpected situations every day. Planning helpsthe manager to face uncertainty because planners try to foresee the future by making some assumptions regarding the future keeping in mind their past experiences and scanning business environments. The plans are made to overcome such uncertainties. The plans also include unexpected risks such as fire or some other calamities in the organization. The resources are kept aside in the plan tomeet such uncertainties.

Planning reduces overlapping and wasteful activities:

The organizational plans are made keeping in mind the requirements of all the departments. Thedepartmental plans are derived from the main organizational plan. As a result, there will be coordinationin different departments. On the other hand, if the managers, non-managers, and all the employees are following the course of action according to the plan then there will be integration in the activities. Plans ensure clarity of thoughts and action and work can be carried out smoothly.

Planning Promotes innovative ideas:

Planning requires high thinking and it is an intellectual process. So, there is a great scope for finding better ideas, better methods, and procedures to perform a particular job. The planning process forces managers to think differently and assumes future conditions. So, it makes the managers innovative and creative.

Planning Facilitates Decision-Making:

Planning helps managers to take various decisions. As in planning goals are set in advance and predictions are made for the future. These predictions and goals help the manager to take fast decisions.

Planning establishes standards for controlling:

Controlling means comparison between planned and actual output and if there is variation between both then find out the reasons for such deviations and take measures to match the actual output with the planned. But in case there is no planned output then the controlling manager will have no base to compare whether the actual output is adequate or not.

For example, if the planned output for a week is 100 units and the actual output produced by employees

is 80 units then the controlling manager must take measures to bring the 80-unit production up to 100 units but if the planned output, i.e., 100 units are not given by the planners then finding out whether 80-unit production is sufficient or not will be difficult to know. So, the base for comparison in control is given by the planning function only.

Focuses attention on the objectives of the company:

The planning function begins with the setting up of the objectives, policies, procedures, methods, rules, etc. which are made in planning to achieve these objectives only. When employees follow the plan, theyare leading toward the achievement of objectives. Through planning, the efforts of all the employees are directed toward the achievement of organizational goals and objectives

4.4 Planning Process/Steps in the planning process

Analysis of the External environment

External environment analysis is related to the analysis of the factors which are beyond the control of the organization. Organizations cannot control these factors. Before actual planning, the management must carefully analyze the external factors which influence the business. These factors areuncontrollable and unpredictable. It includes an analysis of social factors, political factors (Govt. rules, tax policy, etc.), and economic factors (inflation, recession, etc.)

Analysis of the internal environment

It is related to the analysis of the factors which are under the control of the organization. These controllable factors consist of plans, policies, strategies, budgets, etc. are adjusted or implemented using internal resources of the organization related to men, money, material, machines, managerial manpower, etc. It gives answers to the following questions do we have enough resources? do we have an up-to-date plant, equipment, and work environment? Is our profitability? Is our managerial leadership effective?

Define the organization's mission

The mission is the common purpose of an organization. The mission of an organization gives direction. It is related to the fundamental reason for the existence of an organization. The mission of an organization directs and makes a favorable environment for the planning

Determination of objectives

Objectives are formulated according to various departments and sections. Objectives indicate what is to be done, where the primary emphasis is to be placed, and what is to be accomplished by the network of policies, procedures, rules, budgets, programs, and strategies.

Forecasting planning premises

For preparing planning, it is essential to forecast some major areas which have fundamental importance. These are General business forecasts, sales forecasts, capital forecasts, demand forecasts, etc. Forecasting is a systematic investigation of various known facts relating to the past and future for the prediction of the future.

Determining the alternative course of action or plans

It is related to formulating various courses of action that are suitable and help to achieve the objectives of the organization. Management should design alternative courses of action that belong to planning premises.

Evaluating alternative courses of action

It is concerned with measuring various alternatives to know whether they are favorable and important for business objectives.

Selecting the best course or developing strategies

This is an important point of decision-making. Framing and selecting the best suitable alternatives for business is called strategy development.

Establishing the sequence of activities

After the best alternative selection, it is essential to formulate several courses of action for each department for each month, quarter, week, etc.

Formulation of a strategic or long-range plan

It covers all important areas of business activities such as profit, capital expenditure for growth ordiversification, organization structure managerial philosophy, pricing, leadership in the market finance personnel advertising industrial relations products production planning and development, R & D, CSR corporate social responsibilities.

Formulation of tactical plan

Purchase, production, marketing, R & D. these are the areas where plans are designed as per the mission

Formulation of action programs

Allocation of tasks, responsibilities authorities to achieve objectives, and design schedule for executives.

4.5 Types of Plans / Classification of Planning

Standing Plans

- Mission
- Objectives
- Policies
- Procedures
- Rules and methods
- Strategies

Single Use Plans

- Programme
- Projects
- Schedules
- Budgets

Mission

The mission is the common and overall goal of any organization. Every organization must have a mission then only it becomes meaningful or purposeful. Mission or purpose often used interchangeably,identifies the basic

task of an organization for which it is created. For example, the mission of a university is to impart higher education. The mission of the garments factory is to produce and sell ready-made garments and so on

Examples:

Google's Mission Statement: "Our mission is to organize the world's information and makeit universally accessible and useful".

Patanjali's mission statement: "Making India an ideal place for the growth and development

of Ayurveda and a prototype for the rest of the world

Policies

Policies are, in most cases, standing plans. Policies provide guidelines for repetitive actions. They define an area or provide limits within which decisions are to be made and ensure that the decision will be consistent with, and contribute to, an objective. Policies are types of plans that allow decision-makers some discretion to carry out a plan. Otherwise, there will be no difference between policies and rules. Policies must allow for some discretion. Policies help decide issues before they become problems and make it unnecessary to analyze the same situation every time it comes up. It permits managers to delegate authority and still maintain control over subordinates about the matter

Strategy

The strategy is another type of broad-based standing plan which helps the determination of the basic long-term objectives of an enterprise and the adoption of courses of action and allocation of resourcesnecessary to achieve these objectives.

Objectives

This is the first step in planning the action plan of the organization. Objectives are the basics of every company and the desired objective/result that the company plans on achieving, so they are the endpoint of every planning activity. For example, one of the objectives of an organization could be to increase sales by 20%. So the manager will plan all activities of the organization with this end objective inmind. While framing the objectives of the organization some points should be kept in mind.

Rules

Rules Like policies, rules, too, are standing plans that guide action. Rules spell out specifically what

employees are supposed to do or not to do. For example, the no-smoking campaign launched by some organizations is supported by some organizational rules. As opposed to policies, rules do not permit the exercise of individual discretion. Instead, rules specify what actions will be taken (or not taken) and whatbehavior is permitted or not. Policies, on the other hand, tell people how to think about decisions to be made about actions.

Procedures

Procedures Like rules, procedures are standing plans that guide action rather than speculation. They are plans that establish a required method of handling future activities. Procedures establish customary ways for handling certain activities like hiring a clerk, promoting employees, and obtaining a loan from a bank. The major characteristic of a procedure is that it represents a chronological sequencing of events. It specifies a series of steps that must be taken to accomplish a task. Specified series of steps that are required to be taken for admission into the MBA program of AUB is an example of the procedure.

Budgets

It covers a particular period, and once the period is over, a new budget comes into being. It is not only aplanning tool but also works as a controlling tool.

Programs

Programs are plans of action followed in proper sequence according to objectives, policies, and procedures. Thus, a program lays down the major steps to be taken to achieve an objective and sets an approximate time frame for its fulfillment. Programs are usually supported by budgets. A program may be a major or a minor one or a long, medium, or short-term one. Since it is not used in the same form once its task is over it belongs to the single-use plan category.

Projects

A project is a particular job that needs to be done in connection with a general program. So a single step in a program is set up as a project. A project has a distinct object and clear-cut termination. “Projects have the same characteristics as programs but are generally narrower in scope and less complex. Projects are frequently created to support or complement a program.”

Limitations of Planning

Unreliability of forecasts:

Forecasting is made based on records and some statistical methods. But several environmental factors change during execution periods. Hence the forecast becomes unreliable and redundant.

Time-consuming

The planning process involves of steps and forecasts based on analysis and applying statistical methods to past data. The planning process requires res lot of time to perform all these activities and tomake planning effective.

High Cost

The planning process requires more time and resources to make it effective. For forecasting and execution of various activities, more personnel are required. Thus, planning becomes more expensive.

Organizational politics

Though the planning process is systematic and sequential sometimes few leaders in the organization dominate the planning process. Thus, in organization politics can be possible.

Inflexibility

Liberalization and globalization are the major effective factors for change. But sometimes due to

the planning process, it is very difficult to make changes angle as per the external environment. Planning makes most of the organizational activities inflexible and static

4.6 Management by Objectives (MBO)

Meaning

The process of setting objectives in the organization to give a sense of direction to the employees is called as Management by Objectives.

It refers to the process of setting goals for the employees so that they know what they are supposed to do at the workplace.

Management by Objectives defines roles and responsibilities for the employees and helps them chalk out their future course of action in the organization.

Management by objectives guides the employees to deliver their level best and achieve the targets within the stipulated time frame.

Importance of MBO

The Management by Objectives process helps the employees to understand their duties at the workplace.

KRAs are designed for each employee as per their interest, specialization, and educational qualification.

The employees are clear as to what is expected of them.

Management by Objectives process leads to satisfied employees. It avoids job mismatch and unnecessary confusion later on.

Employees in their own way contribute to the achievement of the goals and objectives of the organization. Every employee has his own role at the workplace. Each one feels indispensable to the organization and eventually develops a feeling of loyalty toward the organization. They tend to stick to the organization for a longer span of time and contribute effectively. They enjoy at the workplace and do not treat work as a burden.

Management by Objectives ensures effective communication among the employees. It leads to a positive ambiance at the workplace.

Management by Objectives leads to well-defined hierarchies at the workplace. It ensures transparency at all levels. A supervisor of any organization would never directly interact with the Managing Director in case of queries. He/She would first meet his reporting boss who would then pass on the message to his senior and so on. Everyone is clear about his position in the organization.

The MBO Process leads to highly motivated and committed employees.

The MBO Process sets a benchmark for every employee. The superiors set targets for each of the team members. Each employee is given a list of specific tasks.

4.7 Steps in Management by Objectives Process

1. Define organizational goals

Setting objectives is not only critical to the success of any company, but it also serves a variety of purposes. It needs to include several different types of managers in setting goals. The objectives set by the supervisors are provisional, based on an interpretation and evaluation of what the company can and should achieve within a specified time.

2. Define employee objectives

Once the employees are briefed about the general objectives, plan, and strategies to follow, the managers can start working with their subordinates on establishing their personal objectives. This will be a one-on-one discussion where the subordinates will let the managers know about their targets and which goals they can accomplish within a specific time and with what resources. They can then share some tentative thoughts about which goals the organization or department can find feasible.

3. Continuous monitoring of performance and progress

Though the management by-objectives approach is necessary for increasing the effectiveness of managers, it is equally essential for monitoring the performance and progress of each employee in the organization.

4. Performance evaluation

Within the MBO framework, the performance review is achieved by the participation of the managers concerned.

5. Providing feedback

In the management-by-objectives approach, the most essential step is continuous feedback on the results and objectives, as it enables the employees to track and make corrections to their actions. The ongoing feedback is complemented by frequent formal evaluation meetings in which superiors and subordinates may discuss progress toward objectives, leading to more feedback.

6. Performance appraisal

Performance reviews are routine reviews of the success of employees within MBO organizations.

4.8 Significance of Organizing

Promotes Effective Communication

Organizing is an important means of creating coordination and communication among the various departments of the organization. Different jobs and positions are interrelated by structural relationships. It specifies the channel and mode of communication among different members.

Efficient Administration:

It has generally been observed that there is always a condition of doubt about the authority of the managers among themselves. The process of organizing makes a clear mention of each and every activity of every manager and also of their extent of authority

Expansion and Growth:

The process of organizing allows the employees the freedom to take decisions which helps them to grow. They are always ready to face new challenges. This situation can help in the development of the enterprise. This helps in increasing the earning capacity of the enterprise which in turn helps its development.

Adaptation to Changes

Organizing helps business enterprises to adjust to environmental changes. In order to ensure the smooth functioning of the enterprise, organizing suitably modifies the organizational structure and various inter-relationships existing in the enterprise. Moreover, it also provides stability in the enterprise so that it continues to grow and survive in spite of changes.

The clarity in Working Relationship:

Organizing clarifies the working relations among employees. It specifies who is to report to whom. Therefore, communication becomes effective. It also helps in fixing accountability

Optimum Utilization of Resources:

In process of organizing the entire work is divided into various small activities. There is a different employee performing every different job.

By doing so, there is no possibility of any activity being left out or any possibility of unnecessary duplicating any job. Consequently, there is optimum utilization of all the available resources (e.g., material, machine, financial, human resource, etc.) in the organization.

4.9 Principles of organizing

Principle of objective: the organization is formed with the specific goal or mission, purpose and objective like constructing a building for profit.

Principle of specialization and division of labor: the organization's activities are divided into departments and units and tasks and each employee may be assigned each task. Employees can specialize in that activity

Principle of coordination: all the activities, units, and departments should be coordinated with the organization's objective, purpose, and mission.

Principle of co-equal authority and responsibility: Authority and responsibility are equally delegated. Otherwise, subordinates may misuse authority if more authority than required to discharge the responsibility is delegated. Alternatively, the subordinate can't discharge responsibility, if responsibility is more than the authority.

Span of control: span of control is a number of subordinates, that can be managed efficiently by a superior. The span should be optimum depending upon organizational requirements.

Principle of continuity: organizations should continue their operations, therefore, they should be flexible and responsive to environmental changes.

Principle of teamwork: the purpose of an organization is to achieve it's mission. Achieving mission. Strategy and goal efficiently is possible through a collaborative work of a team of people even though the organization is departmentalized.

Principle of efficiency: organizations should be designed in such a way that they should perform their activities with the least cost and achieve the highest return. Therefore, organizations should perform most efficiently.

4.10 Process of Organizing

- Division of work
- Grouping of Work
- Delegation of Authority
- Coordination of Work

Division of Work

The division of work into smaller jobs leads to specialization because jobs are assigned to individuals according to their qualifications and capabilities. The division of work leads to systematic working. For example, in a bank, every individual is assigned a job. One cashier accepts cash, one cashier makes payments, one person issues checkbooks, one person receives cheques, etc. With the division of work into jobs the banks work very smoothly and systematically.

Grouping of Work

Related and similar jobs are grouped together and put under one department. The departmentation or grouping of jobs can be done by the organization in different ways. But the most common two ways are:

(a) Functional departmentation.

Under this method jobs related to a common function are grouped under one department. For example, all the jobs related to production are grouped under the production department; jobs related to sales are grouped under sales department, and so on

(b) Divisional departmentation:

When an organization is producing more than one type of products then they prefer divisional departmentation. Under this jobs related to one product are grouped under one department. For example, if an organization is producing cosmetics, textiles, and medicines then jobs related to the production, sale, and marketing of cosmetics are grouped under one department, jobs related to textiles under one, and so on.

3. Delegation of Authority

Each individual working in different departments is assigned a duty matching to his skill and qualifications. The work is assigned according to the ability of individuals. Employees are assigned duties by giving them a document called job description. This document clearly defines the contents and responsibilities related to the job.

4. Establishing Reporting Relationship:

After grouping the activities in different departments the employees have to perform the job and to perform the job every individual needs some authority. So, in the fourth step of the organizing process, all the individuals are assigned some authority matching to the job they have to perform.

The assignment of authority results in the creation of a superior-subordinate relationship and the question of who reports to whom is clarified. The individual of higher authority becomes the superior and with less authority becomes subordinate.

4.11 Structure of the organization

The system of tasks, workflows, reporting relationships, and communication channels that link together diverse individuals and groups.

- Vertical organization
- Horizontal organization

Vertical organization

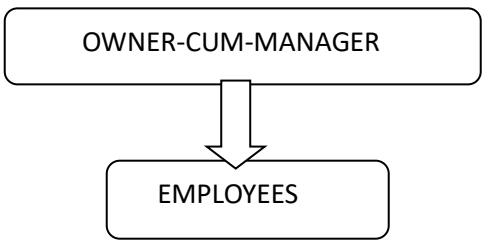
Vertical organization refer to an increase in the length of the organization's hierarchical chain of command. The hierarchical chain of command represents the company's authority-accountability relationship between superiors and subordinates. Authority and responsibility flows from the top to bottom through all the levels of the hierarchy. Accountability flows from the lowest level to the highest level.

Horizontal organization

Horizontal organization refers to an increase in breadth of an organization's structure. The number of levels in the organization's hierarchy are few. The span of control is relatively large. Authority is more decentralized in relatively flat structures.

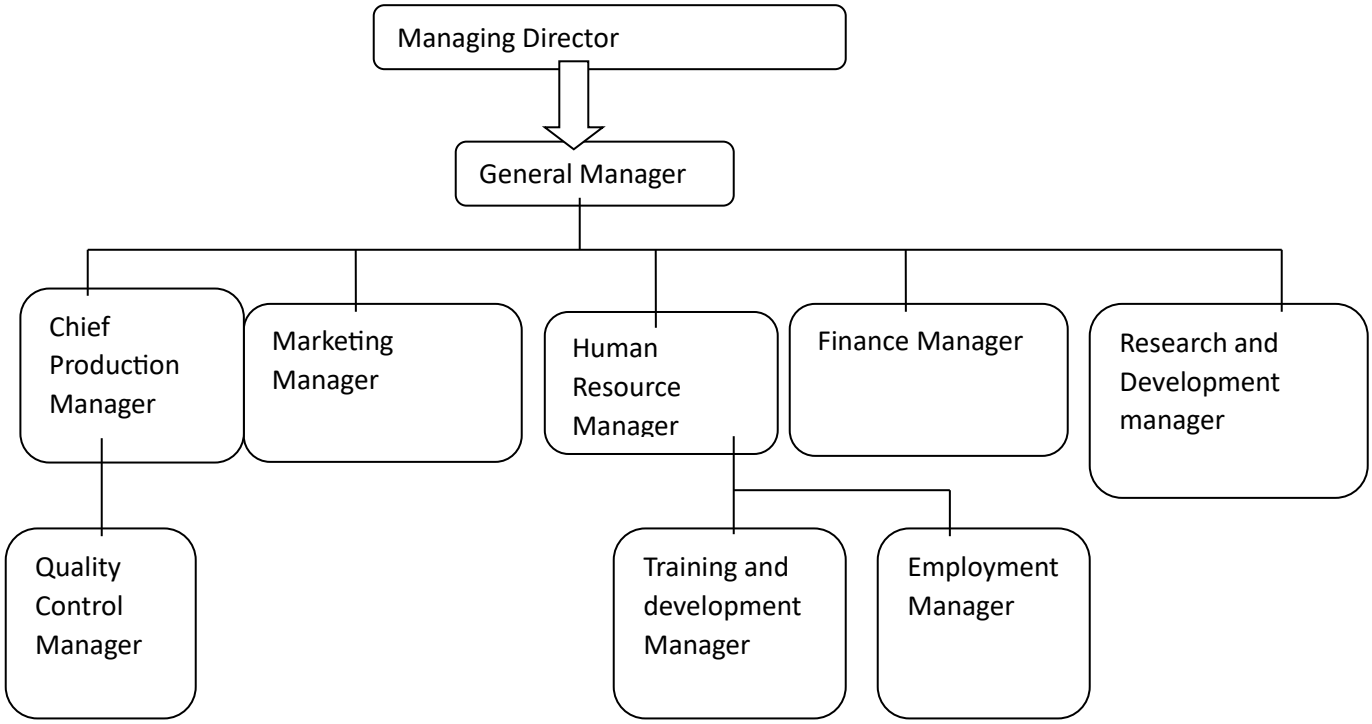
Types /Approaches of Organization Structure

Entrepreneurial Structure



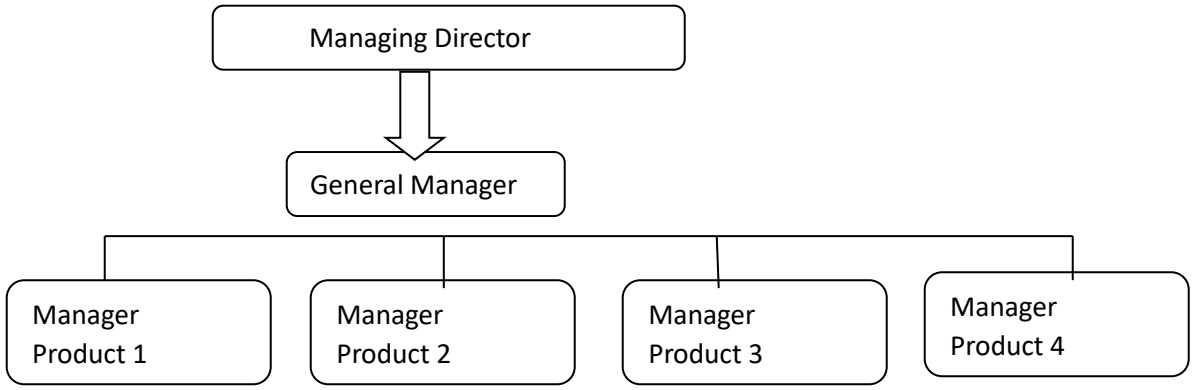
Generally, the small businesses when they are started consist of an owner-manager alone or also with few employees. These types of organizations do not require an organizational chart and formal assignment of responsibilities. It can refer to as start-ups.

Functional organizational Structure:



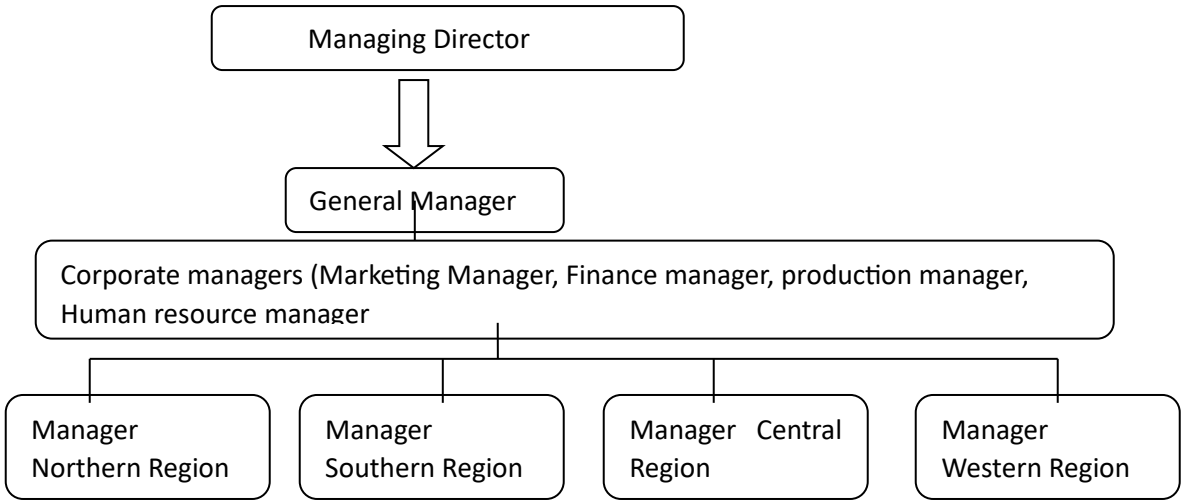
Functional organization structure is most widely used structure. Each functional department consists of those jobs in which employees perform similar jobs at different levels. The commonly used functions are marketing, finance and accounting, human resources, manufacturing, research, and development, etc.

Product organization structure



This structure is based on the products produced by the company. Activities are divided on the basis of individual products, product line, and services and are grouped into departments in product organization structure. All important functions like marketing, production, finance and HRM are contained within each department. This type of organization structure overcomes many of the major limitations of functional organizational structure.

Geographical organization structure



Companies operating in various geographical regions of the country and world, structure of such organization based on geographical structure. The activities or functions are grouped into departments based on the activities performed in the geographical areas. Each geographical unit including all functions required to produce and market the products in a particular geographical area.

Decentralized business unit

Based on product lines the grouping activities has been a trend among diversified companies since 1920. In a diversified firm, the basic organizational building is its business units, each business is operated as a stand-alone profit centre.

Strategic business unit structure.

A single chief executive cannot control a number of decentralized units of a broadly diversified company. The business can be effectively controlled, if the related businesses are grouped into strategic units and the a efficient and senior executive is delegated with the authority and responsibility for its management. This arrangement will improve strategically

planning and implementation, though it adds one layer in the organizational hierarchy.

Matrix structure

Combines functional and divisional structures to gain advantages and minimize disadvantages of each.

Used in: Manufacturing, Service industries, Professional fields, Non-profit sector, Multi-national corporations

Types of Organization

Formal and informal organization

Features of Formal organization:

(1) The formal organizational structure is created intentionally by the process of organizing.

(2) The purpose of formal organization structure is the achievement of organizational goal.

(3) In a formal organizational structure each individual is assigned a specific job.

(4) In a formal organization every individual is assigned a fixed authority or decision-making power.

(5) Formal organizational structure results in the creation of superior-subordinate relations.

(6) Formal organizational structure creates a scalar chain of communication in the organization.

Features of informal organization:

(1) Informal organizational structure gets created automatically without any intended efforts of managers.

(2) Informal organizational structure is formed by the employees to get psychological satisfaction.

(3) Informal organizational structure does not follow any fixed path of flow of authority or communication.

(4) Source of information cannot be known under an informal structure as any person can communicate with anyone in the organization.

Line and Staff organization

Line

According to Allen, "Line refers to those positions and elements of organization, which have the responsibility and authority and are accountable for the accomplishment of primary objectives.

Line organization is the simple and oldest type of organization followed in the organization. Underline organization, each department is generally a completely self-contained unit.

A separate person will look after the activities of the department and he has full control over the department.

The same-level executives do not give or receive orders amongst themselves. But they receive orders from their immediate boss and give orders to their subordinates. Hence, all the heads are responsible to the general manager, the general manager, in turn, is responsible to the shareholders who are the owners.

The line of authority flows from the top to bottom vertically. So it is called line organization.

It consists of direct vertical relationships.

Departmental heads are given full freedom to control their departments.

Each member knows from whom he would get orders and to whom he should give his orders.

The superior takes decisions within the scope of his authority

Staff

Staff elements are those who have the responsibility and authority for providing advice and service to the line in the attainment of objectives.

Line authority represents superior-subordinate relationships for functions, which are directly responsible for accomplishing the key objectives of the enterprise.

The line identifies those positions that possess command and authority to make decisions.

The staff officers assist the line officers in the planning of business activities.

The board of directors frames the policies of the business on the basis of recommendations given by the staff officers.

The managers can get advice from the staff officers regarding the selection. 'training' placement and remuneration fixation of the personnel.

Staff identifies those positions in an organization that do not command a decision-making authority.

These are support positions providing services, advice, assistance and support to the line authority to enable them to perform their duties.

It is for the line managers to decide whether to take some decision with respect to the given advice.

Functional organization:

Functional organization structure is the most widely used structure. Each functional department consists of those jobs in which employees perform similar jobs at different levels. The commonly used functions are marketing, finance, and accounting, human resources, manufacturing, research, and development, etc

Committee Organization

A committee organization is a group of people set up to solve specific problems in the organization. The line people are given opportunities to discuss their problems in the committee. The committee organizational structure is not like line or functional organization but is similar to staff organization. Its decisions are implemented, whereas staff decisions are not necessarily implemented. It is a formal part of the organizational structure wherein the members are specifically mentioned. The Board of
Directors of a company is an example of a committee organization.

Another example is, the Finance Committee will include all the functional managers, viz. Marketing Manager, Production Manager, Personnel Managers, etc. as members, and the Managing Director as the Chairman. It will decide the financial requirements of each and every department. The decisions taken by the committee are followed by the line people, as the committees are representatives of various functional departments.

4.12 Delegation of Authority

The process of giving authority to a subordinate in order to perform the assigned activities by a superior is known as delegation of authority

Delegation is the process a manager follows in dividing the work assigned to him so that he performs only that part which he can perform effectively and so get others to help him with the remaining work

Principles of Delegation

Delegation of results expected: managers have to clearly know the activities to be performed by the subordinates and the results to be achieved by the subordinates. Managers have to delegate the necessary authority and responsibility to produce results.

Co-equal Authority and responsibility: managers should delegate equal authority and responsibility. It means that the amount of authority that is to be delegated should be enough to complete the responsibility. If it is more than the responsibility, it may be misused. If authority is less than responsibility, the subordinate cannot complete the responsibility.

Absoluteness of Responsibility: the superior holds responsibility though he delegates it to the subordinates. Therefore, the responsibility of subordinates to their superiors is absolute.

Creation of Accountability: the subordinates who receive both authority and responsibility should be accountable for their responsibility of completing the activities.

Unity of Command: it indicates that the subordinate should receive orders, instructions, and commands from one superior only.

Elements of Delegation

Assignment of Duties or Responsibility: The responsibility means, the assignment of work or duty to the subordinate i.e. delegation of authority. The managers assign certain responsibility to the subordinates for the completion of certain tasks on his behalf. An individual has to apply all his physical and mental abilities to get the task completed efficiently.

Duty is that which a person is bound by an obligation to do things. Duties may be expressed either in terms of functions or in terms of goals and objectives or results. For effective delegation, duties must be expressed in terms of goals and objectives; since this will provide psychological satisfaction to the subordinates

Grant of Authority: To fulfill the responsibility, certain authority is delegated to the subordinate. Authority means the power to take decisions. Hence, the manager along with the responsibility also delegates authority to enable the subordinate to take decisions independently and accomplish the task efficiently.

The authority must be equal to the responsibility, this means, a certain level of authority is delegated which is sufficient to complete the responsibility. The authority also flows upward, as we go up in the scalar chain, the authority increases.

Accountability or Creation of Obligation: Accountability means, checking whether the subordinates are performing their responsibilities in an expected manner or not. Accountability cannot be delegated which means, in the case of non-completion of the task, the manager will only be held responsible for it, not the subordinates. The accountability also flows upward, i.e. subordinates will be accountable to the manager and the manager to his superior.

When the subordinate knows his assigned responsibilities, he can concentrate his efforts in meeting the requirements of his position.

Definite fixing of responsibility is advantageous because;

1. It acts as an incentive to a subordinate;

2. It aids in the general speed up 'of work; and

3. It assists in developing discipline as a means of control.

Unit V

Functions of Management-II

5.1 Staffing

The managerial function of staffing involves manning the organization structure through proper and effective selection, appraisal, and development of the personnel to fill the roles assigned to the employers/workforce.

"Staffing means filling and keeping filled, positions in the organization structure."

Harold Koonz

5.2 Nature of Staffing

- Staffing is an important managerial function.
- Staffing is a pervasive activity.
- Staffing is a continuous activity.
- The basis of the staffing function is the efficient management of personnel.
- Staffing helps in placing right men at the right job.
- Staffing is performed by all managers

5.3 Staffing Process

Manpower Planning: Manpower Planning which is also called Human Resource Planning consists of putting the right number of people, the right kind of people at the right place, right time, and doing the right things for which they are suited for the achievement of goals of the organization

Recruitment: it is the process involves attracting and obtaining as many applications as possible from eligible job seekers.

Selection: It is a long process, commencing from the preliminary interview of the applicants and ending with the contract of employment. Following are the steps of the selection process

Orientation and Placement (Induction): During Orientation employees are made aware of the mission and vision of the organization

Generally, the information given during the orientation program includes-

- Employee layout
- Type of organizational structure
- Departmental goals
- Organizational layout
- General rules and regulations
- Standing Orders
- Grievance system or procedure

Placement: Once the candidates are selected for the required job, they have to be fit as per the qualifications. Placement is said to be the process of fitting the selected person at the right job orplace, i.e. fitting square pegs in square holes and round pegs in round holes. Once he is fitted into the job, he is given the activities he has to perform and alsotold about his duties.

Training and Development: Training is crucial for organizational development andsuccess. It is fruitful to both employers and employees of an organization. An employee willbecome more efficient and productive if he is trained well.

It also-

Improves morale of employeesLess supervision

Chances of promotionIncreased productivity

Remuneration: Employee Remuneration refers to the reward or compensationgiven to the employees for their work performance. Remuneration provides basic attraction to an employee to perform the job efficiently andeffectively. There are mainly two types of Employee RemunerationTime Rate Method & Piece Rate Method.

Performance and Evaluation: performance appraisal is the method that evaluates the performance of the employee. It measures the productivity of employees.

Promotion and Transfer: Through performance and evaluation organizations come to know the employees who have more productivity. Promotion of such employees done on other positions for which they are eligible. Sometimes it is essential to transfer the employees to the other branches which are in need of such employees.

5.4 Directing

Directing: it is a function that includes all activities which are designed to encourage subordinatesand to work effectively and efficiently

It consists of the process and technology utilized in using instructions and making certainthat operations are carried out on as originally planned.

Importance of Directing

Directing Initiates Action

Directing initiates Each and every action in an organization. It acts as a guideline for subordinates. The managers direct the subordinates about what to do, how to do when to do and also see to it thattheir instructions are properly followed.

Ingrates all Efforts to achieve a common goal

Directing integrates the efforts of all the employees and departments through persuasive leadership and effective communication toward the accomplishment of organizational goals.

Support to adopt change:

Employees have a tendency to resist any kind of change in the organization. But, adapting theenvironmental changes is necessary for the growth of the organization. A manager through motivation, proper communication, and leadership can make the employees understand the nature and contents ofchange. This will help in a smooth adaptation of the changes without any friction between the management and employees.

Effective Utilization of Resources

It involves defining the duties and responsibilities of every subordinate clearly thereby avoiding wastages, duplication of efforts, etc., and utilizing the resources of men, machines, materials, and money in the maximum possible way. It helps in reducing costs and increasing profits.

Motivates Employees

A manager identifies the potential and abilities of its subordinates and helps them to give their best. He also motivates them by offering them financial and non-financial incentives to improve their performance.

Provides Stability

Stability is significant in the growth of any organization. Effective directing develops co-operation and commitment among the employees and creates a balance among various departments and groups.

5.5 Communication

The English word 'communication' is derived from the Latin communis, which means common sense. The word communication means sharing the same ideas. In other words, the transmission and interaction of facts, ideas, opinions, feelings or attitudes. Communication is the essence of management. The basic function of management (planning, planning, staffing, supervision, and management) cannot be done effectively without effective communication. Communication is a two-way process that involves transferring of information or messages from one person or group to another. This process goes on and includes a minimum of one sender and receiver to pass on the messages. These messages can either be any ideas, imagination, emotions, or thoughts. Communication is a Latin word that means "to share". There are different modes of communication available today. These include emails, chats, WhatsApp, skype (conference calls), etc. Effective communication makes people's work easier and smooth.

Communication Process

Communication is an ongoing process that mainly involves three components namely. sender, message, and recipient. The components involved in the communication process are described below in detail:

Sender: The sender or contact generates the message and transmits it to the recipient. He is the source and the first contact

Message: It is an idea, knowledge, opinion, truth, feeling, etc. produced by the sender and intended for reference.

Encoding: The message produced by the sender is encrypted in a symbolic way such as words, pictures, touches, etc. before transfer.

The media: This is how the coded message is conveyed. The message can be conveyed orally or in writing.

Recording: It is a process of modifying the signals sent by the sender. After recording the message is received by the recipient.

Recipient: You are the last person in the chain and the message you sent was sent. If the recipient receives the message and understands it correctly and acts on the message, only then the purpose of the communication is achieved.

Answer: Once the recipient confirms to the sender that you received the message and understood it, the communication process is complete.

Noise: Refers to any restrictions caused by the sender, message or recipient during the communication process. For example, incorrect telephone connection, incorrect coding, incorrect recording, careless recipient, incorrect understanding of message due to discrimination or inappropriate touch, etc.

Types of Communication

Verbal Communication and Non-Communication

Verbal communication: Communication occurs through verbal, verbal or written communication that conveys or conveys a message to others is called oral communication. Verbal communication is the use of language to convey information verbally or in sign language.Verbal communication is important because it works well. It can be helpful to support verbal Non-verbal communication Any non-verbal communication, spoken words, conversation and written language is called.

Non-verbal communication: It occurs with signs, symbols, colors, touches, body or facial features. Insignificant communication is using body language, body language and facial expressions to convey information to others. It can be used both intentionally and deliberately. For example, you may have a smile on your face when you hear an idea or a piece of interesting or exciting information. Open communication is helpful when you are trying to understand the thoughts and feelings of others.

Mode Of Communication

Formal Communication:Formal Communication refers to communication that takes place through legal channels in an organization. That kind of communication takes place between managers or employees of the same class or between high and low and vice versa. It may be oral or written but a complete record of that communication is kept in the organization.

Informal Communication:Informal communication is defined as any communication that occurs outside of the official channels of communication. Informal communication is often referred to as the 'vine' as it spreads throughout the organization and on all sides regardless of the level of authority.

Communication Barriers One sometimes wants to connect with one thing, but he is actually saying something else that he did not intend. This type of event in communication behaviour is known as the "Arc of Distortion". The distortion may be the result of some error in any of the communication channels. These barriers to communication are also known as "barriers".

Barriers to communication:

- Lack of proper style, feedback.
- Content is not related to customer requirements.
- Failure to maintain dual communication.
- Bad weather.
- Lack of horizontal flow of ideas.
- Availability of technical coordinators.
- Semantic Problems.
- Lack of leadership.
- Lack of enthusiasm.
- Lack of support from heads of institutions.

Types of Communication

There are four types of communication. It is categorized into verbal, non-verbal, written, and visual.

Verbal

Verbal communication is one of the modes where people communicate or transfer information through words. It is one of the common and usual types and is frequently used during one on one presentations, video calls or conferences, meetings, phone calls etc. There Are Certain Measures Which Enhance This

More Effectively Firm and Confident Voice: Firm and confident communication reflects the personality of the person. It gives more certainty of completing any task. So always be confident so that your ideas are more precise and specific. Active listening: A good listener always tends to listen to everyone's perspectives or viewpoints. Active listening helps in identifying each one's problem or thoughts in a more clear way. Ignore Filler Words: While giving a presentation, avoid using filler words such as yeah, like, so, etc. It might be distracting to your audience. Try not to use them in official conferences or meetings.

Non-Verbal

Non-verbal communication is the use of body language. It includes body gestures, facial expressions, and shaking hands, etc.. For example: How you sit during an interview automatically reflects your body language. If they are indicating closed body language like closed arms, bent shoulders, shaking legs, etc., they might be nervous, low in confidence, surrounded with anxiety, etc. Non-verbal communication is the most powerful communication to understand others' thoughts and emotions.

Here are Certain Categories Where Non-Verbal Communication are Briefed Up

Positive Body Language: Always carry a positive body language where you can carry your confidence for performing any task. This type of communication gives support to your verbal talks and makes you more open to any kind of jobs. Imitate non-verbal communication you find useful: Some facial expression or body language can be found helpful in an interview. If an interviewer has positively nodded his head, it gives a clear positive sign in a closed way, i.e. non-verbal communication.

Written

It is the form of communication that involves writing, typing and printing symbols, letters, etc. It is used in Emails, chats, etc. which are the common techniques of using it in the workplace. Whereas it also furnishes a record of all docs in one place and keeps a systematic account of it.

Here Are Certain Categories Where Written Communication is Briefed Up

Aim for Simplicity: Any type of written communications should be in a simple format and clear. It helps audiences to understand and provides more transparency on information which you're providing.

Reviewing: Whenever you're writing, always review your emails, letters or memos before sending it. Reviewing helps to find the mistakes or opportunities to present something distinct.

Be Careful of Written Tone: Since this is not a mode of verbal or non-verbal communication, always be cautious and have a polite tone while writing.

Keep the Written Files if You Find it Useful: If you received the memo or email which you're finding helpful or interesting, you could save that template for further references to use it in future writing for improving your written communication.

Visual

Visualizing is a form of communication where one can use photographs, drawing charts and graphs to convey information through it. It helps in furnishing the right information through graphics and visuals during office presentation (along with verbal and written).

Here are Certain Steps Which Help in Visual Communication

SkillsTaking Advice Before Going with Visuals: Visual communication includes presentation or emails. Always ask for other's advice if any mistake can be rectified.

Targeting Audience: Always put those visuals in presentation or emails that can be understood by everyone quickly. If you are giving a presentation on any data or chart which is not familiar to the audience, you need to explain it clearly. There shouldn't be any usage of offensive visuals.

Barriers in CommunicationThere are certain barriers which create hindrance in building up communication over the time period.

Personal Barriers:Communication takes place between receiver and sender. It's a two-way process which should be clear. In case message formation went wrong, it gives a wrong and unclear message to the recipient. The receiver might get the wrong perspective while receiving a message. Therefore the message should be written effortlessly.

Systematic Barrier: If any machine or electronic errors occur by any means or in any unforeseen situation, it may affect the importance of communication.

5.6 Controlling:

Meaning: Control consists of making something happen the way it was planned to happen. According to Henri Fayol, control consists in verifying whether everything occurs in conformity with the plan adopted, the instructions issued and principles established.

Managerial control implies the measurement of accomplishment against the standard and the correction of deviations to assure attainment of objectives according to plans"... Koontz And O'Donnell

Objective of controlling is to point out weaknesses and errors in order to rectify them and prevent. It includes (i) measuring actual performance (ii) comparing actual performance to standards and (iii) taking corrective action to ensure that planned events actually occur.

Importance of Controlling

Ensures order and discipline:

Control creates an atmosphere of order and discipline in the organization. Effective controlling system keeps the subordinates under check and makes sure they perform their functions efficiently.

Sharp control can have a check over dishonesty and fraud of employees. Strict control monitor, employees work on computer monitor which brings more order and discipline in the work environment.

Controlling helps in minimizing the errors:

Small errors or small mistakes may not seriously affect the organization. But if these errors are repeated again and again it will become a serious matter and can bring disaster for the organization.

An effective controlling system helps in minimizing the errors by continuous monitoring and check. The managers try to detect the error on time and take remedial steps to minimise the effect of error.

Making Efficient Use of Resources:

Controlling makes it possible to use human and physical resources efficiently. Under controlling, it is ensured that no employee deliberately delays his work performance. In the same way, wastage in all the physical resources is checked.

Facilitating Coordination in Action:

Coordination among all the departments of the organization is necessary in order to achieve the organizational objectives successfully. All the departments of the organization are interdependent. For example, the supply of orders by the sales department depends on the production of goods by the production department.

Improving Employee Motivation:

Employees know the standards against which their performance will be judged. Systematic evaluation of performance and consequent rewards in the form of increment, bonus, promotion etc. motivate the employees to put in their best efforts.

Achieving Goals:

Being goal-oriented provides an entity let it be a business or not, the advantage of making it easier to achieve goals. When the employees understand the goals and are kept goal oriented their efforts will be directed towards achieving it. If the effort is in a different direction altogether, the process of controlling weeds it out. Thus the process ensures channelizing effort into achieving predetermined goals.

Controlling Process

Steps 1: Key areas to be monitored

Macro environment: As stated earlier, one of the key areas to be monitored is the macro-environment of the company. Macro environmental factors can influence a company continuously so it isessential to monitor in order to prevent negative impacts and frame policies.

Mission and Objectives: it includes modifying any one or more of the areas like the company's mission, objectives, plans, goals, strategy formulation and implementation.

Internal operations: the manager has to evaluate the internal operations continuously in viewof the changes in the macro-environment and industry environment.

Step 2: Establishing Standards:

Standards are criteria against which results are measured. They are norms to achieve the goals. Standards are usually measured in terms of output. They can also be measured in non-monetary terms like loyalty, customer attraction, goodwill etc. Some of the standards are as.

Time standards:

The goal will be set on the basis of time lapse in performing a task.

Cost standards:

These indicate the financial expenditures involved per unit, e.g. material cost per unit, cost per person, etc.

Income standards:

These relate to financial rewards received due to a particular activity like sales volume per month, yearetc.

Market share:

This relates to the share of the company's product in the market.

Productivity:

Productivity can be measured on the basis of units produced per man hour etc.

Profitability:

These goals will be set with the consideration of cost per unit, market share, etc.

Step3: Measuring Performance

Measurement involves a comparison between what is accomplished and what was intended to be accomplished. The measurement of actual performance must be in units similar to those of predetermined criteria. The unit or the yardstick thus chosen to be clear, well-defined and easily identified, and should be uniform and homogenous throughout the measurement process.
The performance can be measured by the following steps:
Strategic control points:

It is not possible to check everything that is being done. So it is necessary to pick strategic control points for measurement. Some of these points are:

Income:

It is a significant control point and must be as much per unit of time as was expected. If the income is significantly off from the expectation then the reasons should be investigated and corrective action taken.

Expenses:

Total and operational cost per unit must be computed and must be adhered to. Key expense data must be reviewed periodically.

Inventory:

Some minimum inventory of both the finished product as well as raw materials must be kept in stock as a buffer. Any change in inventory level would determine whether the production is to be increased or decreased.

Quality of the product:

Standards of established quality must be maintained especially in food processing, drug manufacturing, automobiles, etc. The process should be continuously observed for any deviations.

Absenteeism:

Excessive absenteeism of personnel is a serious reflection on the environment and working conditions. Absenteeism in excess of chance expectations must be seriously investigated.

Step 4: Comparing the Actual Performance with the Expected Performance

This is the active principle of the process. The previous two, setting the goals and the measurement format are the preparatory parts of the process. It is the responsibility of the management to compare the actual performance against the standards established.

This comparison is less complicated if the measurement units for the standards set and the performance measured are the same and quantified. The comparison becomes more difficult when theserequire subjective evaluations

Ralph C. Davis identifies four phases in the comparison.

- Receiving the raw data.
- Accumulation, classification and recording of this information.
- Periodic evaluation of completed action to date.
- Reporting the status of accomplishment to higher line authority.
- At the third phase, deviations if any are noted between standards and performance. If clear cut deviations are there, then management must study the:-
- Causes for deviation
- Effect of deviation
- Size of deviation
- Positive or negative deviation.

Step 5: Correcting Deviations:

The final element in the process is the taking corrective action. Measuring and comparing performance, detecting shortcomings, failures or deviations, from plans will be of no avail if it does point to the needed corrective action. Thus controlling to be effective, should involve not only the detection of lapses but also probe into the failure spots, fixation of responsibility for the failures at the right quarters, recommendation of the best possible steps to correct them. These corrective actions must be applied when the work is in progress. The primary objective should be avoidance of such failures in future. The required corrective action can be determined from the qualified data as per the standards laid out and the performance evaluation already done.

Budget as a control technique

Budget are plans for a specific period in numerical terms. They are statements of expected outcome in financial terms, physical terms, human resources terms etc. Budgets would be the basis for delegation of authority and responsibility without loss of control.

Types of Budgets

Capital Expenditure budgets: companies basically require capital for establishing manufacturing facilities, marketing network, and the like many activities. Capital expenditure budgets specify the estimates of the amount of capital required for land, buildings, plant, equipment, machinery, technology, minimum level of materials etc. It is used to control the deviations or errors.

Cash budgets: cash budgets provide the estimates of cash receipts and disbursements. The actual cash receipts and disbursements are compared with the estimates.

Production Budget: they specify the amount of the estimated output, estimated materials, human resources etc. These budgets help to control the production factors and ensure the production as ensured.

Sales budget: sales budgets specify the amount of estimated sales. They help to estimate the sales, identify the factors affects sales and control these factors.

In this way different budgets support and work as guidelines for action and control. Budget also assist for coordination , encourage team efforts and creativity.

Quality Control

Quality control (QC) is a procedure or set of procedures intended to ensure that a manufactured product or performed service adheres to a defined set of quality criteria or meets the requirements of the client or customer.

It brings standardization into the process. Most organizations have a quality control/assurance department that provides the set of standards to be followed for each product. Either an internal team or a third-party team is hired to determine whether the products that are delivered meet these standards. Quality control relies on testing of products, as product inspection gives a clearer picture of the quality of the end product.

Cost Control

Cost control is a series of steps that a business uses to maintain proper control over its costs. Implementing this level of control can have a profound positive impact on profits over the long term. The following four steps are associated with cost control:

Create a baseline. Establish a standard or baseline against which actual costs are to be compared. These standards may be based on historical results, a reasonable improvement

on historical results, or the theoretically best attainable cost performance. The middle alternative is generally considered to yield the best results, since it sets an achievable standard.

Calculate a variance. Calculate the variance between actual results and the standard or baseline noted in the first step. Particular emphasis is placed on the detection of unfavourable variances, which are those actual costs that are higher than expected. If a variance is immaterial, it may not be worthwhile to report the item to management.

Investigate variances. Conduct a detailed drill-down into the actual cost information to ascertain the reason for an unfavourable variance.

Take action. Based on the information found in the preceding step, recommend to management whatever corrective actions are needed to reduce the risk of continued unfavourable cost variances.

The preceding steps are only recommended if a company routinely attempts to force its actual costs incurred to closely match its budgeted cost structure. If there is no budget, then an alternative way to practice cost control is to plot individual cost line items from the income statement on a trend line. If there is an unusual spike in the trend line, then the spike is investigated in relation to the average cost level, and corrective action is taken. Thus, operating without a budget eliminates the first two steps in the preceding list of activities, but cost control still requires investigatory work and recommendations to management for corrective action.

The shareholders of a publicly held company are particularly interested in a system of cost control, for they realize that tight cost control gives a company considerable influence over its cash flows and reported profits.

Techniques of Controlling

1. Direct Supervision and Observation

This is considered to be the oldest techniques and is most suitable for small business organization. The technique involves direct involvement of supervisor as he himself keeps a close check on the employees and their work. By this, the comparison may not be required, as most of the problems are solved during supervision. The supervisor gets the better understanding of the worker as he gets first hand information.

2. Analysis and Interpretation of Financial Statements:

Every business organizations prepare final accounts. Profit and loss account gives details of the income and expenses for a specified period and Balance Sheet shows the financial position of the organization on a specified date. Financial data can act as a tool to control the organization. Various analysis tools like common size statement, comparative statement help in comparison of the current year with the previous year's figures. These figures can also be compared with the industry benchmarks and similar firms.

Ratio analysis is used to analyse the financial statements. Absolute figures are unfit for comparison. Ratios express the relationship of one business variable with other business variable in form of mathematical expression. The ratios can be divided into four categories, they are

- **Liquidity Ratios:** It aims to identify that whether the firm has enough working

capital to run day to day operations. Liquidity ratios are the indicators of firm's ability to pay short term obligation when they become due. Some of the important ratios in this category are current ratio, quick ratio and super quick ratio.

- **Solvency Ratios:** These ratio helps in ascertaining long term solvency of the firm and tries to find out whether the firm has adequate resources to meet its long term financial obligations for e.g. debt equity ratio , Capital gearing ratio etc.
- **Turnover ratios:** it indicates the efficiency with which the assets of the firm are utilised for e.g. fixed asset turnover ratio, inventory turnover ratio etc.
- **Profitability ratios:** It indicates the financial health of the firm in terms of its earning capacity of the business. It also throws light on the adequacy of the control over expenses for e.g. net profit ratio, gross profit ratio, return on capital employed, operating expense ratios etc.

The disadvantage of this technique is that ratios can be used for window dressing. If the account statements are misleading ratios too can be misleading.

3. Budgetary Control

Budgetary control is often referred as traditional control technique. A budget is an estimation done for the purpose of planning and controlling. According J.A. Scott " It is the system of management control and accounting in which all operations are forecasted and so far as possible planned ahead , and the actual results are compared with the forecasted and planned ones". Budgetary control is a technique of establishing control with the help of budgets. In fact budgets are most important constituents of financial control .Budget is a statement of planned events expressed in quantitative and monetary terms. Budgetary control can be done for every facet of a business ranging from income, production, and so on. The various budgets are cash budget, sales budget production budget, purchase budget, master budget etc. A budget committee is established for budgetary control. Budgetary control has several advantages likes:

a. It makes planning precise and realistic.
b. It saves time and cost and establishes coordination among various departments.
c. It helps in optimum utilization of resources.
d. It provides useful data that can be employed for filling tenders and quotation.
e. It prevents buck passing as it establishes responsibility.

4. Break-even Analysis

BEP (break-even point) is a state at which the cost is equivalent to sales (the state of no profit and no loss). The Break-even analysis is used as a control device. It helps to find out the company's perfomance and the minimum level of sales the company needs to attain in order to avoid loss. If the sales are less than the minimum level the company may not be able to recover cost. Subsequently the company can employ collaborative steps to progress its performance in the given line.

BEP = Fixed Cost/Sale price per unit-Variable cost per unit

Advantages

(a) Helps in identifying minimum requirement of sales at which cost can be recovered. Helps in analyzing the cost behavior with sales and its effect on profit, thus profit can be estimated at various levels.

(b) The breaking up of cost component into fixed and variable cost helps management to control cost.

Disadvantages

(a) Classification of cost is not possible every time.

(b) It assumes that cost and revenue function have a linear relationship which may not hold true every time. The economies of scale may result in disproportionate varying of cost.

(c) It assumes that technology, factor prices and product mix are fixed where as in reality this may not hold true.

5. Return on Investment

This is also a financial tool. Investment consists of net working capital and fixed asset employed in the business. The income or the profit generated is the premium that the entrepreneur gets for his risk taking. Higher the ROI better is the performance of the business and vice versa. The comparison of current year ROI with that of previous years' performance helps to know the progress of the business. It also facilitates inter-firm comparisons. The area needing rectification can also be traced.

6. Management by Objective (MBO)

MBO is a planning and controlling device and it can only be effective if it has following features:

(a) The goals are fixed with the joint effort of the subordinates and supervisor.

(b) The performance of employee is periodically evaluated and regular feedback for improvement is provided to the employee.

(c) Once the task is accomplished, due reward is given to the employee.

The success of this measure depends on the coordinated efforts of superior and subordinates. Elements of biasness, lack of trust between superior-subordinate relationships can significantly affect the working of the technique.

7. PERT & CPM Techniques

Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) techniques are used to minimize cost and time. Any program has a series of activities and sub-activities involved in it. The success of any activity depends upon doing the work in a specified sequence and time. The whole project is divided into identifiable activities and a logical sequence between these activities is established. Time estimates are prepared for every activity and the longest path that is critical path is identified and analyzed. The techniques emphasis on identifying the critical activities and giving them priority. Critical activities are crucial activities which have to be completed on time so that the project is not delayed. The techniques divide the job into various sub-activities. from which, the critical activities are identified and significant importance is attached to their accomplishment. Therefore, the time and cost of the entire project can be minimised.

Advantages of PERT/CPM

(a) It significantly reduces cost, time and money.

(b) The identification of critical activities leads to optimum utilization of resources.

(c) Helps in proactive management, as managers are made aware that delay in one operation may lead to delay in other business operation.

(d) Graphical presentation of task ensures proper understanding and communication.

MCA FIRST YEAR (II Sem)

SUBJECT: UNDERSTANDING OPERATING SYSTEMS

DISCLAIMER

THESE EDUCATIONAL CONTENTS (NOTES) ARE COMPILED FROM VARIOUS BOOKS, JOURNALS AND INTERNET.

MCA-R201	Understanding Operating Systems

UNIT 1: Overview of Operating System

Operating System as an Extended Machine, as a Resource Manager, OPERATING SYSTEM CONCEPTS-Processes, Address Spaces, Files, Input/Output, Protection, Shell, SYSTEM CALLS-System Calls for Process Management, System Calls for File Management, System Calls for Directory Management, OPERATING SYSTEM STRUCTURE-Monolithic Systems, Layered Systems, Microkernels, Client-Server Model, Virtual Machines, Exokernels

UNIT 2: Process management and Threads

Process Model ,Process Creation, Process Termination, Process Hierarchies, Process States, Implementation of Processes, THREADS- Thread Usage, Classical Thread Model, POSIX Threads, Implementing Threads in User Space, Implementing Threads in the Kernel, Hybrid Implementations SCHEDULING-Introduction, Scheduling in Batch Systems, Scheduling in Interactive Systems, Scheduling in Real-Time Systems

UNIT 3: Memory management

MEMORY ABSTRACTION: ADDRESS SPACES, Notion of an Address Space, Swapping Managing Free Memory, VIRTUAL MEMORY- Paging, Page Tables, PAGE REPLACEMENT ALGORITHMS-Optimal Page Replacement, Not Recently Used Page Replacement, First-In First-Out (FIFO) Page Replacement, Least Recently Used (LRU) Page Replacement, SEGMENTATION

UNIT 4: File Systems

FILES-File Naming, File Structure, File Types, File Access, File Attributes, File Operations DIRECTORIES-Single-Level Directory Systems, Hierarchical Directory Systems, Path Names Directory Operations, FILE-SYSTEM IMPLEMENTATION-File-System Layout, Implementing Files, Implementing Directories, Shared Files

UNIT 5: Virtualization and the Clouds

History, Requirements For Virtualization, Type 1 And Type 2 Hypervisors, Techniques For Efficient Virtualization-Virtualizing the unvirtualizable , Cost Of Virtualization, Memory Virtualization, I/O Virtualization, Virtual Appliances Virtual Machines On Multicore Cpus, Licensing Issues, Clouds-Clouds As A Service, Virtual Machine Migration, Check Pointing

Reference Books

1. Operating Systems Concepts- Silberschatz A., Galvin P., Gagne G, Wiley Publication.
2. Modern Operating Systems, Andrew S. Tanenbaum, III rd Edition, PHI Publication.
3. Operating System-Internal and Design Principles, William Stallings, Pearson Education.
4. Principles of Operating Systems-Naresh Chauhan, First Edition, Oxford University press.
5. Operating Systems in Depth- Thomas W. Doeppner, Wiley Publications

Unit 1

Overview of Operating System

What is an Operating System?

- An Operating System (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. When you start using a Computer System then it's the Operating System (OS) that acts as an interface between you and the computer hardware.
- "An Operating System (OS) is an interface between a computer user and computer hardware."
- An operating system is a software that performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.
- Some popular operating system is Linux, Windows, OS X, macOS, etc.

Operating system goals:

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

Computer System Structure

Computer system can be divided into four components

Hardware: provides basic computing resources such as CPU, memory, I/O devices.

Operating system: Controls and coordinates use of hardware among various applications and users.

Application programs: define the ways in which the system resources are used to solve the computing problems of the users Word processors, compilers, web browsers, database systems, video games .

U**ser**: People, machines, other computers

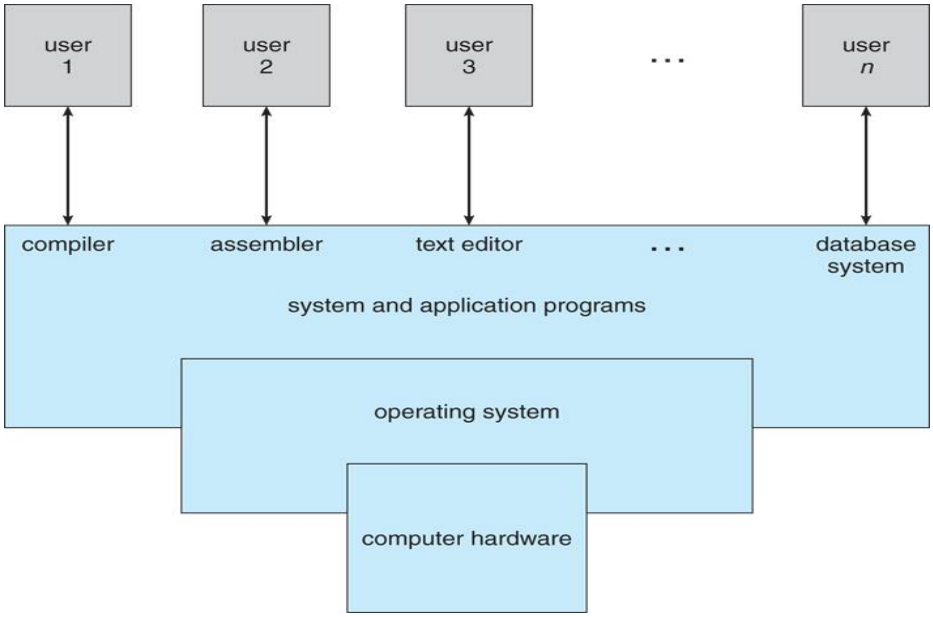


Fig 1. Abstract view of the component of a computer system

1.1 Operating system as an Extended Machine

- The operating system masks or hides the details of the Hardware from the programmers and general users and provides a convenient interface for using the system.
- The program that hides the truth about the hardware from the user and presents a nice simple view of named files that can be read and written is of course the operating system.
- In this view, the function of OS is to present the user with the equivalent of an extended machine or virtual machine that is easier to program than the underlying hardware.
- Just as the operating system shields the user from the disk hardware and presents a simple file-oriented interface, it also conceals a lot of unpleasant business concerning interrupts, timers, memory management and other low-level features.
- The placement of OS is as shown in fig. 2 A major function of OS is to hide all the complexity presented by the underlying hardware and gives the programmer a more convenient set of instructions to work with.

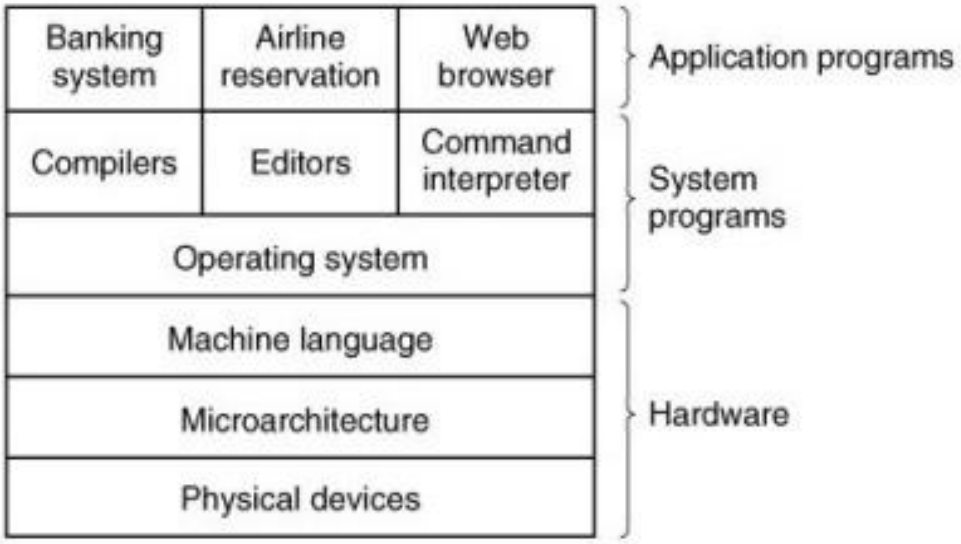


Fig. 2: Computer system consists of Hardware, system program, and application program

- One of the major tasks of the operating system is to hide the hardware and present programs (and their programmers) with nice, clean, elegant, consistent, abstractions to work with instead. Operating systems turn the ugly into the beautiful, as shown in Fig. 3

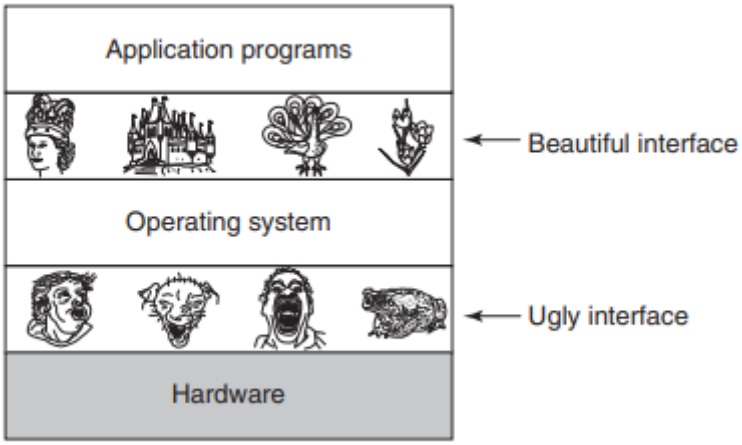


Fig.3: Operating systems turn ugly hardware into beautiful abstractions.

1.2 Operating system as a Resource Manager

- A computer system has many resources.
- Modern computers consist of processors, memories, timers, disks, mice, network interfaces, printers, and a wide variety of other devices.

- In the alternative view, the job of the operating system is to provide for an orderly and controlled allocation of the processors, memories, and I/O devices among the various programs competing for them.
- Imagine what would happen if three programs running on some computer all tried to print their output simultaneously on the same printer. The first few lines of printout might be from program 1, the next few from program 2, then some from program 3, and so forth.
- The result would be chaos. The operating system can bring order to the potential chaos by buffering all the output destined for the printer on the disk.
- When one program is finished, the operating system can then copy its output from the disk file where it has been stored to the printer, while at the same time, the other program can continue generating more output, oblivious to the fact that the output is not really going to the printer (yet).
- Fig. 4 suggests the main resources that are managed by the OS. A portion of the OS is in the main memory. This includes Kernel The remainder of the main memory contains user programs and data. The allocation of this resource (main memory) is controlled jointly by the OS and memory management hardware in the processor

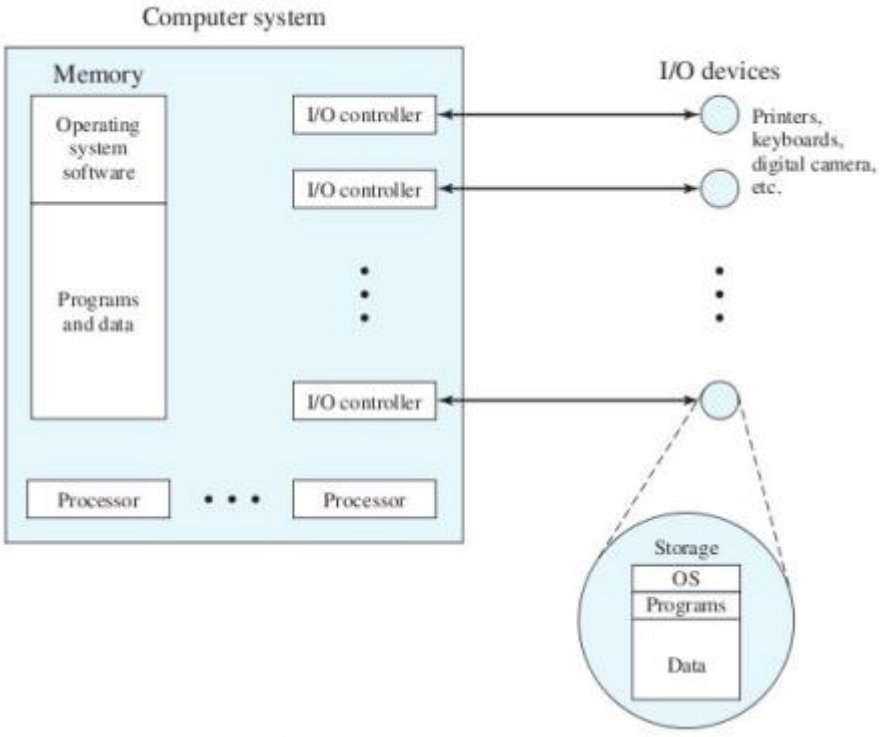


Fig. 4: The Operating system as Resource manger

- Resource management includes multiplexing (sharing) resources in two different ways: **in time and in space.**
  - When a resource is time multiplexed, different programs or users take turns using it. First one of them gets to use the resource, then another, and so on.
  - The other kind of multiplexing is space multiplexing. Instead of the customers taking turns, each one gets part of the resource.
  - For example, main memory is normally divided up among several running programs, so each one can be resident at the same time (for example, in order to take turns using the CPU). Assuming there is enough memory to hold multiple programs, it is more efficient to hold several programs in memory at once rather than give one of them all of it, especially if it only needs a small fraction of the total.
  - Another resource that is space multiplexed is the disk. In many systems, a single disk can hold files from many users at the same time.
  - Allocating disk space and keeping track of who is using which disk blocks is a typical operating system task.

1.3 Operating System Structure

Now that we have seen what operating systems look like on the outside (i.e., the programmer's interface), it is time to take a look inside.

we will examine six different structures that have been tried, in order to get some idea of the spectrum of possibilities.

The six designs we will discuss here are

1. monolithic systems
2. layered systems
3. microkernels
4. client-server systems
5. virtual machines
6. exokernels

1. Monolithic Systems

- In monolithic systems, each component of the operating system is contained within the kernel. Operating systems that use monolithic architecture were first time used in the 1970s.

- In this monolithic structure, the operating system as a whole is working in the kernel space. The kernel directly controls all the file, memory, device, and process management.
- The monolithic operating system is also known as the monolithic kernel.
- A monolithic kernel is an operating system architecture where the entire operating system is working in kernel space. The monolithic model differs from other operating system architectures, such as the microkernel architecture, in that it alone defines a high-level virtual interface over computer hardware.
- The kernel provides various services such as memory management, file management, process scheduling etc. using function calls. This makes the execution of the operating system quite fast as the services are implemented under the same address space.
- It implements both user and kernel services in the same address space.
- In a monolithic kernel, if a service fails, the entire system fails.
- The execution of the monolithic kernel is faster because the system call establishes the communication of the system's application and hardware.
- This kernel provides CPU scheduling, memory management, file management, and other system functions through system calls. As both services are implemented under the same address space, this makes operating system execution faster.

Examples of monolithic operating system

VMS, Linux, OS/360, OpenVMS, Multics, AIX, BSD

- In this model, for each system call there is one service procedure that takes care of it and executes it. The utility procedures do things that are needed by several service procedures, such as fetching data from user programs. This division of the procedures into three layers is shown in Fig. 1-24.

Basic Monolithic Structure for the operating system:

1. A main program that invokes the requested service procedure.
2. A set of service procedures that carry out the system calls.
3. A set of utility procedures that help the service procedures.

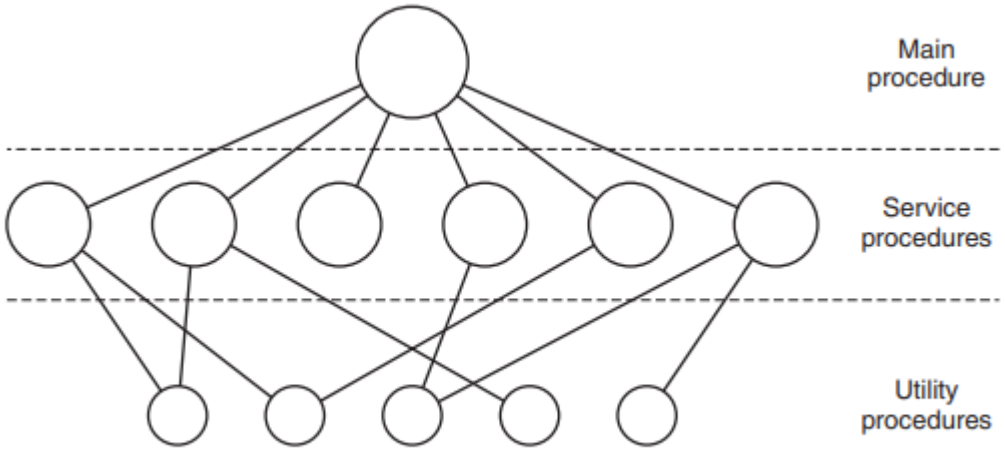


Figure 1-24. A simple structuring model for a monolithic system.

Advantages of Monolithic Kernel

Here are the following advantages of a monolithic kernel, such as:

- The execution of the monolithic kernel is quite fast as the services such as memory management, file management, process scheduling, etc., are implemented under the same address space.
- A process runs completely in a single address space in the monolithic kernel.

Disadvantages of Monolithic Kernel

Some of the disadvantages of the monolithic kernel are −

- If any service fails in the monolithic kernel, it leads to the failure of the entire system.
- To add any new service, the entire operating system needs to be modified by the user.

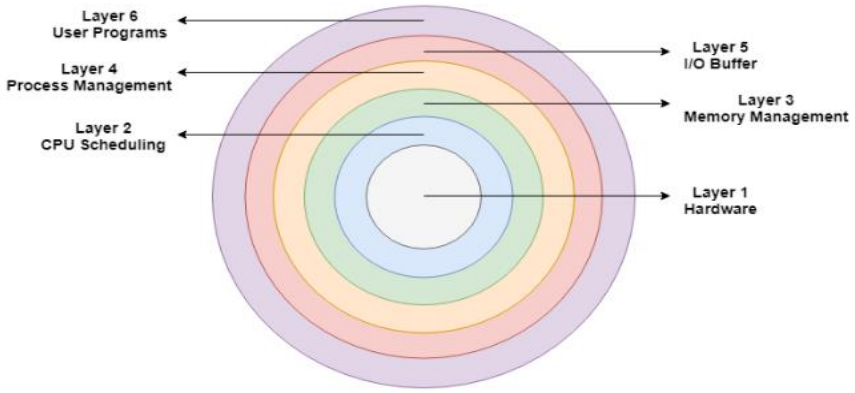
2. Layered Systems

- Operating systems using this architecture are intended to overcome the disadvantages of monolithic systems. As a result, the OS is divided into various layers, each of which performs a different function.
- Each of the layers in the operating system can only interact with the layers that are above and below it. The lowest layer handles the hardware and the uppermost layer deals with the user applications.
- As a result of the fact that all the functionalities are on different layers, each layer can be tested and debugged independently.

- Layering provides a distinct advantage in an operating system. All the layers can be defined separately and interact with each other as required. Also, it is easier to create, maintain and update the system if it is done in the form of layers. Change in one layer specification does not affect the rest of the layers.

Layers in Layered Operating System

There are six layers in the layered operating system. A diagram demonstrating these layers is as follows:



LAYERED OPERATING SYSTEM

Hardware

- This layer interacts with the system hardware and coordinates with all the peripheral devices used such as printer, mouse, keyboard, scanner, etc. The hardware layer is the lowest layer in the layered operating system architecture.

CPU Scheduling

- This layer deals with scheduling the processes for the CPU. There are many scheduling queues that are used to handle processes. When the processes enter the system, they are put into the job queue. The processes that are ready to execute in the main memory are kept in the ready queue.

Memory Management

- Memory management deals with memory and the moving of processes from disk to primary memory for execution and back again. This is handled by the third layer of the operating system.

Process Management

- This layer is responsible for managing the processes i.e. assigning the processor to a process at a time. This is known as process scheduling. The different algorithms used

for process scheduling are FCFS (first come first served), SJF (shortest job first), priority scheduling, round-robin scheduling, etc.

I/O Buffer

- I/O devices are very important in computer systems. They provide users with the means of interacting with the system. This layer handles the buffers for the I/O devices and makes sure that they work correctly.

User Programs

- This is the highest layer in the layered operating system. This layer deals with the many user programs and applications that run in an operating system such as word processors, games, browsers, etc.

An early example of layer OS is THE (Technische Hogeschool Eindhoven)

Many of today's OS including Windows XP and Linux implement some level of layering.

Structure of THE operating system

- The system had six layers, as shown in Fig. 1-25. Layer 0 dealt with the allocation of the processor. In other words, layer 0 provided the basic multiprogramming of the CPU.
- Layer 1 did the memory management. It allocated space for processes in main memory and on a 512K word drum used for holding parts of processes (pages) for which there was no room in main memory.
- Layer 2 handled communication between each process and the operator console (that is, the user). On top of this layer each process effectively had its own operator console.
- Layer 3 took care of managing the I/O devices and buffering the information streams to and from them.
- Above layer 3 each process could deal with abstract I/O devices with nice properties, instead of real devices with many peculiarities.
- Layer 4 was where the user programs were found. They did not have to worry about process, memory, console, or I/O management.

- The system operator process was located in layer 5.

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

Figure 1-25. Structure of the THE operating system.

Advantages of Layered Structure

There are several advantages of the layered structure of operating system design, such as:

- Information hiding between layers
- Increased security and protection
- Easy debugging: As the layers are discrete so it is very easy to debug. Suppose an error occurs in the CPU scheduling layer. The developer can only search that particular layer to debug, unlike the Monolithic system where all the services are present.
- Easy update: A modification made in a particular layer will not affect the other layers.
- No direct access to hardware: The hardware layer is the innermost layer present in the des
- ign. So, a user can use the services of hardware but cannot directly modify or access it, unlike the Simple system in which the user had direct access to the hardware.

Disadvantages of Layered Structure

Though this system has several advantages over the Monolithic and Simple design, there are also some disadvantages, such as:

- Complex and careful implementation: As a layer can access the services of the layers below it, the arrangement of the layers must be done carefully. For example, the backing storage layer uses the services of the memory management layer. So, it must be kept below the memory management layer. Thus, with great modularity comes complex implementation.

- Slower in execution: If a layer wants to interact with another layer, it requests to travel through all the layers present between the two interacting layers. Thus, it increases response time, unlike the Monolithic system, which is faster than this. Thus, an increase in the number of layers may lead to a very inefficient design.
- Functionality: It is not always possible to divide the functionalities. Many times, they are interrelated and can't be separated.
- Communication: No communication between non-adjacent layers.

3. Microkernels System

- A microkernel is one of the classifications of the kernel. Being a kernel, it manages all system resources.
- But in a microkernel, the user services and kernel services are implemented in different address spaces. The user services are kept in user address space, and kernel services are kept under kernel address space, thus also reduces the size of kernel and size of an operating system as well. Communication between different parts of the operating system is typically done using message passing.
- The basic idea behind the microkernel design is to achieve high reliability by splitting the operating system up into small, well-defined modules, only one of which—the microkernel—runs in kernel mode and the rest run as relatively powerless ordinary user processes.
- In particular, by running each device driver and file system as a separate user process, a bug in one of these can crash that component, but cannot crash the entire system.
- Thus a bug in the audio driver will cause the sound to be garbled or stop, but will not crash the computer.
- In contrast, in a monolithic system with all the drivers in the kernel, a buggy audio driver can easily reference an invalid memory address and bring the system to a grinding halt instantly.
- Many microkernels have been implemented and deployed for decades (Haertig et al., 1997; Heiser et al., 2006; Herder et al., 2006; Hildebrand, 1992; Kirsch et al., 2005; Liedtke, 1993, 1995, 1996; Pike et al., 1992; and Zuberi et al., 1999). With the exception of OS X, which is based on the Mach microkernel (Accetta et al., 1986), common desktop operating systems do not use microkernels.
- Microkernel examples include QNX, Minix3, Symbian, Mac OS X, L4Linux, Integrity, K42, and so forth.

- The process structure of MINIX 3 is shown in Fig. 1-26, with the kernel call handlers labeled Sys. The device driver for the clock is also in the kernel because the scheduler interacts closely with it. The other device drivers run as separate user processes.

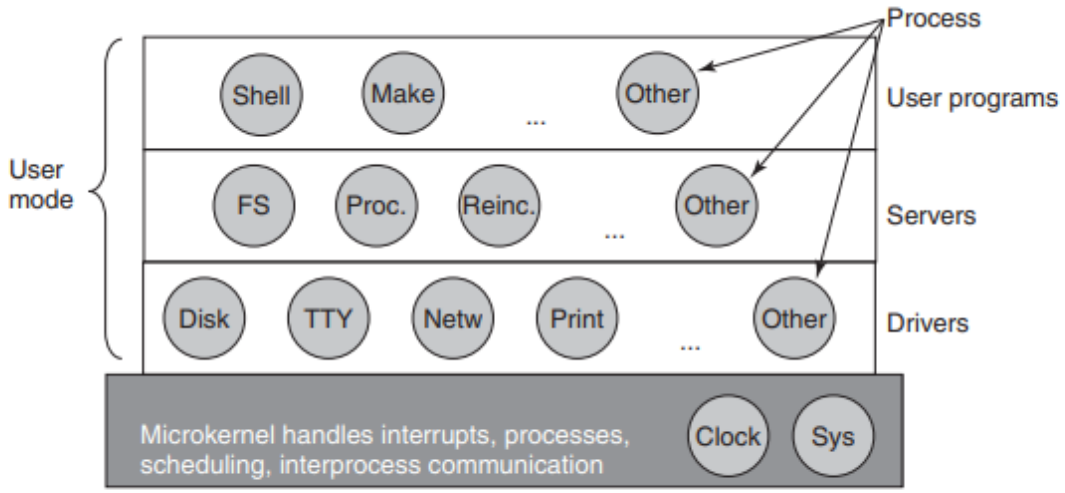


Figure 1-26. Simplified structure of the MINIX system.

- Outside the kernel, the system is structured as three layers of processes all running in user mode.
- The lowest layer contains the device drivers. Since they run in user mode, they do not have physical access to the I/O port space and cannot issue I/O commands directly. Instead, to program an I/O device, the driver builds a structure telling which values to write to which I/O ports and makes a kernel call telling the kernel to do the write.
- Above the drivers is another user-mode layer containing the servers, which do most of the work of the operating system.
- One or more file servers manage the file system(s), the process manager creates, destroys, and manages processes, and so on.
- User programs obtain operating system services by sending short messages to the servers asking for the POSIX system calls. For example, a process needing to do a read sends a message to one of the file servers telling it what to read.
- One interesting server is the reincarnation server, whose job is to check if the other servers and drivers are functioning correctly. In the event that a faulty one is detected; it is automatically replaced without any user intervention. In this way, the system is self-healing and can achieve high reliability.

Advantages of Microkernel

- Microkernels are modular, and the different modules can be replaced, reloaded, modified without even touching the Kernel.
- Microkernel system is flexible, so different strategies and APIs, implemented by different servers, which can coexist in the system.
- Increased security and stability will result in a decreased amount of code which runs on kernel mode
- Fewer system crashes when compared with monolithic systems.

Disadvantage of Microkernel

- Providing services in a microkernel system are expensive compared to the normal monolithic system.
- Context switch or a function call needed when the drivers are implemented as procedures or processes, respectively.
- The performance of a microkernel system can be indifferent and may lead to some problems.

4. Client Server Model

- A slight variation of the microkernel idea is to distinguish two classes of processes, the servers, each of which provides some service, and the clients, which use these services. This model is known as the client-server model.
- Often the lowest layer is a microkernel, but that is not required. The essence is the presence of client processes and server processes.
- Communication between clients and servers is often by message passing.
- To obtain a service, a client process constructs a message saying what it wants and sends it to the appropriate service.
- The service then does the work and sends back the answer.
- If the client and server happen to run on the same machine, certain optimizations are possible, but conceptually, we are still talking about message passing here.
- An obvious generalization of this idea is to have the clients and servers run on different computers, connected by a local or wide-area network, as depicted in Fig. 1-27.
- Since clients communicate with servers by sending messages, the clients need not know whether the messages are handled locally on their own machines, or whether they are sent across a network to servers on a remote machine.

- As far as the client is concerned, the same thing happens in both cases: requests are sent and replies come back. Thus, the client-server model is an abstraction that can be used for a single machine or for a network of machines.

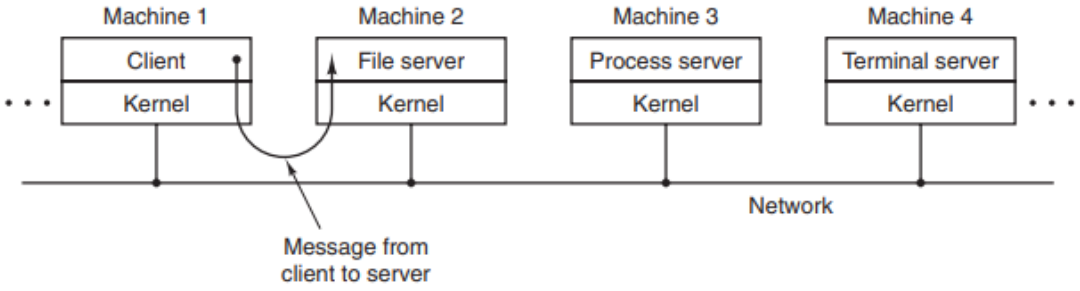


Figure 1-27. The client-server model over a network.

- Increasingly many systems involve users at their home PCs as clients and large machines elsewhere running as servers.
- In fact, much of the Web operates this way. A PC sends a request for a Web page to the server and the Web page comes back. This is a typical use of the client-server model in a network.

5. Virtual Machines

Virtual machine architecture is a type of operating system architecture that runs multiple operating systems on a single physical machine. This approach provides flexibility and allows for efficient use of hardware resources, but it can also introduce overhead and complexity**.**

Virtual Machine OS is to support the concurrent execution of different operating systems on a computer.

The resources of the physical computer are shared to create virtual machines.

- CPU scheduling can create the appearance that users have their own processors.
- Spooling and a file system can provide virtual card readers and virtual line printers.
- A normal user time-sharing terminal serves as the virtual machine operator's console.

The difference between non-virtual and virtual machines can be understood by referring Figure 1.21.

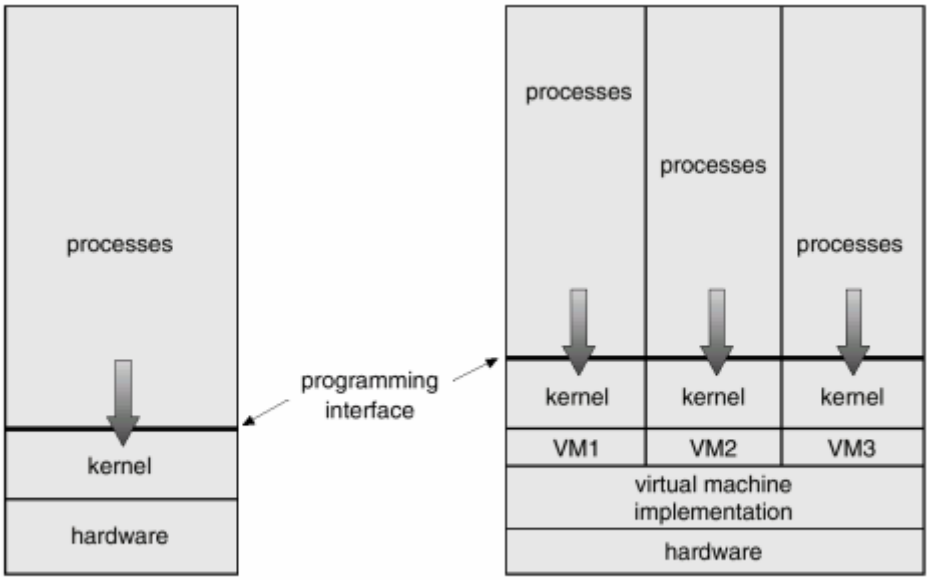


Figure 1.21 System Models (a) Non-virtual machine (b) Virtual Machine

Implementation

Though the virtual machine (VM) concept is useful, it is difficult to implement due to the effort required to provide an exact duplicate of the underlying machine.

Compared to actual I/O, the virtual I/O may take considerably more time, as it is interpreted. Also, since the CPU is multi-programmed among many virtual machines, the VM will slow down unpredictably.

The Java Virtual Machine

- Another area where virtual machines are used, but in a somewhat different way, is for running Java programs.
- When Sun Microsystems invented the Java programming language, it also invented the virtual machine (i.e., a computer architecture) called the JVM (Java Virtual Machine).
- The Java compiler produces code for JVM, which then typically is executed by a software JVM interpreter.
- The JVM consists of a class loader, a class verifier, and a java interpreter. The just-in-time (JIT) compiler converts the bytecode into native code for a particular computer. Thus, the JVM helps to develop platform-independent and portable programs. The JVM can be seen in Figure 1.22

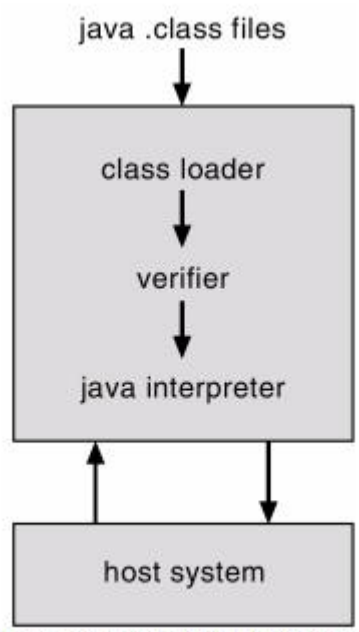


Figure 1.22 Java Virtual Machine

- The advantage of this approach is that the JVM code can be shipped over the Internet to any computer that has a JVM interpreter and run there.
- Another advantage of using JVM is that if the interpreter is implemented properly, which is not completely trivial, incoming JVM programs can be checked for safety and then executed in a protected environment so they cannot steal data.

Advantages of Virtual Machine OS:

- Efficient Resource Utilization: Virtual machine architecture allows multiple operating systems to run concurrently on a single physical machine, improving hardware resource utilization.
- Enhanced Security: This architecture provides greater security because every virtual machine is isolated from each other.
- Easy Recovery: As this architecture uses virtual machines, recovery of the system is easy.

Disadvantages of Virtual Machine OS:

- This isolation permits no direct sharing of resources.
- Difficult to implement due to the effort required to provide an exact duplicate to the underlying machine.

6. Exokernels

- There may be the case that the performance of an application being developed is affected due to the architecture of the OS. Although there are various options for

OSs to be selected for various types of applications, it may not be possible sometimes that the required feature is available in the selected OS.

- The original concept of OS, the hardware has been at such a high abstraction level to application developers that they, in fact, do not know about the actual hardware.
- They work in a convenient environment, provided by the OS, without worrying about the bare hardware details.
- In all the structures of the OS, it has been ensured that the developers need not worry about the configuration and limits of the hardware. This is because the OS is there for all these tasks and provides a convenient and friendly environment to the user.
- But, researchers at MIT realized that giving so much abstraction to the developer affects the performance of the application being designed on the system. It would be better for a developer if he/she decides on his/her own about what to do with resources instead of following the abstractions provided by the OS. In this way, the performance of an application, not bounded by the abstractions and policies of the OS, may perform better.
- To implement this idea, control of the resources needs to be provided to the developers. use a kernel with minimum functionality and provide access to resources to the developer as well. This kernel is called an exokernel.
- The exokernel performs the job of allocation and synchronizing of resources with user jobs. But the way in which the application makes use of resources will be decided by the developer.
- In other words, the exokernel works as an executive for application programs such that it ensures the safe use of resources and allocates them to the applications. It means that the developer can implement customized abstraction on the resources. The applications implemented in this way are known as *library OSs*.
- Library OSs (see Fig. 4.12) may request the exokernel to allocate resources like some disk blocks, memory addresses, CPU, and so on, and use these resources the way it suits the application best. In this way, the exokernel also provides efficiency, because now there is no need to map the resources from the conceptual model provided by the conventional OS to the physical model.
- The concept of exokernel has been in use since 1994, and MIT has developed two exokernels, namely Aegis and XOK, this concept has not been used in any commercial OS and research is still going on.

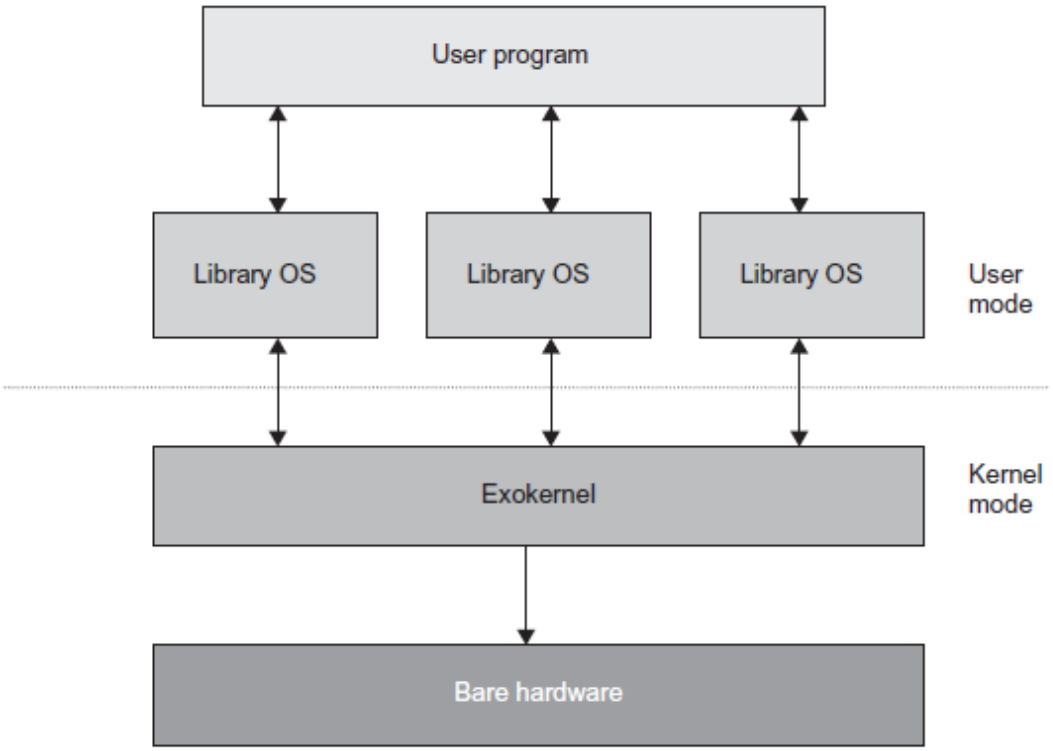


Fig. 4.12 Exokernel architecture

occur at a certain moment (or within a certain range), we have a **hard real-time system**. Many of these are found in industrial process control, avionics, military, and similar application areas. These systems must provide absolute guarantees that a certain action will occur by a certain time.

A **soft real-time system**, is one where missing an occasional deadline, while not desirable, is acceptable and does not cause any permanent damage. Digital audio or multimedia systems fall in this category. Smartphones are also soft real-time systems.

Since meeting deadlines is crucial in (hard) real-time systems, sometimes the operating system is simply a library linked in with the application programs, with everything tightly coupled and no protection between parts of the system. An example of this type of real-time system is eCos.

The categories of handhelds, embedded systems, and real-time systems overlap considerably. Nearly all of them have at least some soft real-time aspects. The embedded and real-time systems run only software put in by the system designers; users cannot add their own software, which makes protection easier. The handhelds and embedded systems are intended for consumers, whereas real-time systems are more for industrial usage. Nevertheless, they have a certain amount in common.

1.4.9 Smart Card Operating Systems

The smallest operating systems run on smart cards, which are credit-card-sized devices containing a CPU chip. They have very severe processing power and memory constraints. Some are powered by contacts in the reader into which they are inserted, but contactless smart cards are inductively powered, which greatly limits what they can do. Some of them can handle only a single function, such as electronic payments, but others can handle multiple functions. Often these are proprietary systems.

Some smart cards are Java oriented. This means that the ROM on the smart card holds an interpreter for the Java Virtual Machine (JVM). Java applets (small programs) are downloaded to the card and are interpreted by the JVM interpreter. Some of these cards can handle multiple Java applets at the same time, leading to multiprogramming and the need to schedule them. Resource management and protection also become an issue when two or more applets are present at the same time. These issues must be handled by the (usually extremely primitive) operating system present on the card.

1.5 OPERATING SYSTEM CONCEPTS

Most operating systems provide certain basic concepts and abstractions such as processes, address spaces, and files that are central to understanding them. In the following sections, we will look at some of these basic concepts ever so briefly, as

an introduction. We will come back to each of them in great detail later in this book. To illustrate these concepts we will, from time to time, use examples, generally drawn from UNIX. Similar examples typically exist in other systems as well, however, and we will study some of them later.

1.5.1 Processes

A key concept in all operating systems is the **process**. A process is basically a program in execution. Associated with each process is its **address space**, a list of memory locations from 0 to some maximum, which the process can read and write. The address space contains the executable program, the program's data, and its stack. Also associated with each process is a set of resources, commonly including registers (including the program counter and stack pointer), a list of open files, outstanding alarms, lists of related processes, and all the other information needed to run the program. A process is fundamentally a container that holds all the information needed to run a program.

We will come back to the process concept in much more detail in Chap. 2. For the time being, the easiest way to get a good intuitive feel for a process is to think about a multiprogramming system. The user may have started a video editing program and instructed it to convert a one-hour video to a certain format (something that can take hours) and then gone off to surf the Web. Meanwhile, a background process that wakes up periodically to check for incoming email may have started running. Thus we have (at least) three active processes: the video editor, the Web browser, and the email receiver. Periodically, the operating system decides to stop running one process and start running another, perhaps because the first one has used up more than its share of CPU time in the past second or two.

When a process is suspended temporarily like this, it must later be restarted in exactly the same state it had when it was stopped. This means that all information about the process must be explicitly saved somewhere during the suspension. For example, the process may have several files open for reading at once. Associated with each of these files is a pointer giving the current position (i.e., the number of the byte or record to be read next). When a process is temporarily suspended, all these pointers must be saved so that a read call executed after the process is restarted will read the proper data. In many operating systems, all the information about each process, other than the contents of its own address space, is stored in an operating system table called the **process table**, which is an array of structures, one for each process currently in existence.

Thus, a (suspended) process consists of its address space, usually called the **core image** (in honor of the magnetic core memories used in days of yore), and its process table entry, which contains the contents of its registers and many other items needed to restart the process later.

The key process-management system calls are those dealing with the creation and termination of processes. Consider a typical example. A process called the **command interpreter** or shell reads commands from a terminal. The user has just

typed a command requesting that a program be compiled. The shell must now create a new process that will run the compiler. When that process has finished the compilation, it executes a system call to terminate itself.

If a process can create one or more other processes (referred to as **child processes**) and these processes in turn can create child processes, we quickly arrive at the process tree structure of Fig. 1-13. Related processes that are cooperating to get some job done often need to communicate with one another and synchronize their activities. This communication is called **interprocess communication**, and will be addressed in detail in Chap. 2.

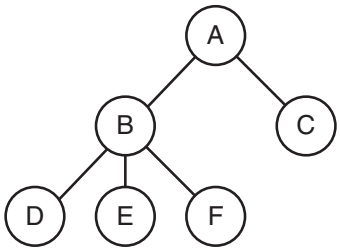


Figure 1-13. A process tree. Process *A* created two child processes, *B* and *C*. Process *B* created three child processes, *D*, *E*, and *F*.

Other process system calls are available to request more memory (or release unused memory), wait for a child process to terminate, and overlay its program with a different one.

Occasionally, there is a need to convey information to a running process that is not sitting around waiting for this information. For example, a process that is communicating with another process on a different computer does so by sending messages to the remote process over a computer network. To guard against the possibility that a message or its reply is lost, the sender may request that its own operating system notify it after a specified number of seconds, so that it can retransmit the message if no acknowledgement has been received yet. After setting this timer, the program may continue doing other work.

When the specified number of seconds has elapsed, the operating system sends an **alarm signal** to the process. The signal causes the process to temporarily suspend whatever it was doing, save its registers on the stack, and start running a special signal-handling procedure, for example, to retransmit a presumably lost message. When the signal handler is done, the running process is restarted in the state it was in just before the signal. Signals are the software analog of hardware interrupts and can be generated by a variety of causes in addition to timers expiring. Many traps detected by hardware, such as executing an illegal instruction or using an invalid address, are also converted into signals to the guilty process.

Each person authorized to use a system is assigned a **UID** (**User IDentification**) by the system administrator. Every process started has the UID of the person who started it. A child process has the same UID as its parent. Users can be members of groups, each of which has a **GID** (**Group IDentification**).

One UID, called the **superuser** (in UNIX), or **Administrator** (in Windows), has special power and may override many of the protection rules. In large installations, only the system administrator knows the password needed to become superuser, but many of the ordinary users (especially students) devote considerable effort seeking flaws in the system that allow them to become superuser without the password.

We will study processes and interprocess communication in Chap. 2.

1.5.2 Address Spaces

Every computer has some main memory that it uses to hold executing programs. In a very simple operating system, only one program at a time is in memory. To run a second program, the first one has to be removed and the second one placed in memory.

More sophisticated operating systems allow multiple programs to be in memory at the same time. To keep them from interfering with one another (and with the operating system), some kind of protection mechanism is needed. While this mechanism has to be in the hardware, it is controlled by the operating system.

The above viewpoint is concerned with managing and protecting the computer's main memory. A different, but equally important, memory-related issue is managing the address space of the processes. Normally, each process has some set of addresses it can use, typically running from 0 up to some maximum. In the simplest case, the maximum amount of address space a process has is less than the main memory. In this way, a process can fill up its address space and there will be enough room in main memory to hold it all.

However, on many computers addresses are 32 or 64 bits, giving an address space of 2^{32} or 2^{64} bytes, respectively. What happens if a process has more address space than the computer has main memory and the process wants to use it all? In the first computers, such a process was just out of luck. Nowadays, a technique called virtual memory exists, as mentioned earlier, in which the operating system keeps part of the address space in main memory and part on disk and shuttles pieces back and forth between them as needed. In essence, the operating system creates the abstraction of an address space as the set of addresses a process may reference. The address space is decoupled from the machine's physical memory and may be either larger or smaller than the physical memory. Management of address spaces and physical memory form an important part of what an operating system does, so all of Chap. 3 is devoted to this topic.

1.5.3 Files

Another key concept supported by virtually all operating systems is the file system. As noted before, a major function of the operating system is to hide the peculiarities of the disks and other I/O devices and present the programmer with a

nice, clean abstract model of device-independent files. System calls are obviously needed to create files, remove files, read files, and write files. Before a file can be read, it must be located on the disk and opened, and after being read it should be closed, so calls are provided to do these things.

To provide a place to keep files, most PC operating systems have the concept of a **directory** as a way of grouping files together. A student, for example, might have one directory for each course he is taking (for the programs needed for that course), another directory for his electronic mail, and still another directory for his World Wide Web home page. System calls are then needed to create and remove directories. Calls are also provided to put an existing file in a directory and to remove a file from a directory. Directory entries may be either files or other directories. This model also gives rise to a hierarchy—the file system—as shown in Fig. 1-14.

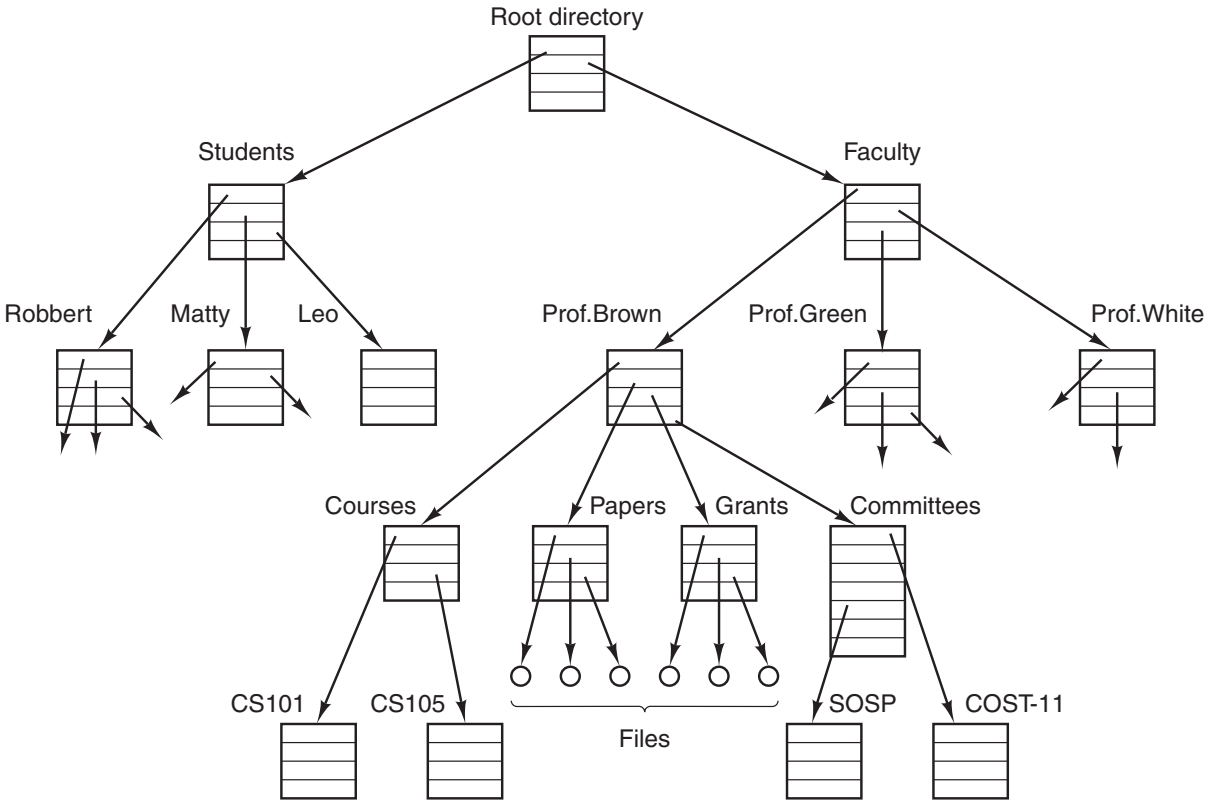


Figure 1-14. A file system for a university department.

The process and file hierarchies both are organized as trees, but the similarity stops there. Process hierarchies usually are not very deep (more than three levels is unusual), whereas file hierarchies are commonly four, five, or even more levels deep. Process hierarchies are typically short-lived, generally minutes at most, whereas the directory hierarchy may exist for years. Ownership and protection also differ for processes and files. Typically, only a parent process may control or even

access a child process, but mechanisms nearly always exist to allow files and directories to be read by a wider group than just the owner.

Every file within the directory hierarchy can be specified by giving its **path name** from the top of the directory hierarchy, the **root directory**. Such absolute path names consist of the list of directories that must be traversed from the root directory to get to the file, with slashes separating the components. In Fig. 1-14, the path for file *CS101* is */Faculty/Prof.Brown/Courses/CS101*. The leading slash indicates that the path is absolute, that is, starting at the root directory. As an aside, in Windows, the backslash (\) character is used as the separator instead of the slash (/) character (for historical reasons), so the file path given above would be written as *\Faculty\Prof.Brown\Courses\CS101*. Throughout this book we will generally use the UNIX convention for paths.

At every instant, each process has a current **working directory**, in which path names not beginning with a slash are looked for. For example, in Fig. 1-14, if */Faculty/Prof.Brown* were the working directory, use of the path *Courses/CS101* would yield the same file as the absolute path name given above. Processes can change their working directory by issuing a system call specifying the new working directory.

Before a file can be read or written, it must be opened, at which time the permissions are checked. If the access is permitted, the system returns a small integer called a **file descriptor** to use in subsequent operations. If the access is prohibited, an error code is returned.

Another important concept in UNIX is the mounted file system. Most desktop computers have one or more optical drives into which CD-ROMs, DVDs, and Blu-ray discs can be inserted. They almost always have USB ports, into which USB memory sticks (really, solid state disk drives) can be plugged, and some computers have floppy disks or external hard disks. To provide an elegant way to deal with these removable media UNIX allows the file system on the optical disc to be attached to the main tree. Consider the situation of Fig. 1-15(a). Before the mount call, the **root file system**, on the hard disk, and a second file system, on a CD-ROM, are separate and unrelated.

However, the file system on the CD-ROM cannot be used, because there is no way to specify path names on it. UNIX does not allow path names to be prefixed by a drive name or number; that would be precisely the kind of device dependence that operating systems ought to eliminate. Instead, the mount system call allows the file system on the CD-ROM to be attached to the root file system wherever the program wants it to be. In Fig. 1-15(b) the file system on the CD-ROM has been mounted on directory *b*, thus allowing access to files */b/x* and */b/y*. If directory *b* had contained any files they would not be accessible while the CD-ROM was mounted, since */b* would refer to the root directory of the CD-ROM. (Not being able to access these files is not as serious as it at first seems: file systems are nearly always mounted on empty directories.) If a system contains multiple hard disks, they can all be mounted into a single tree as well.

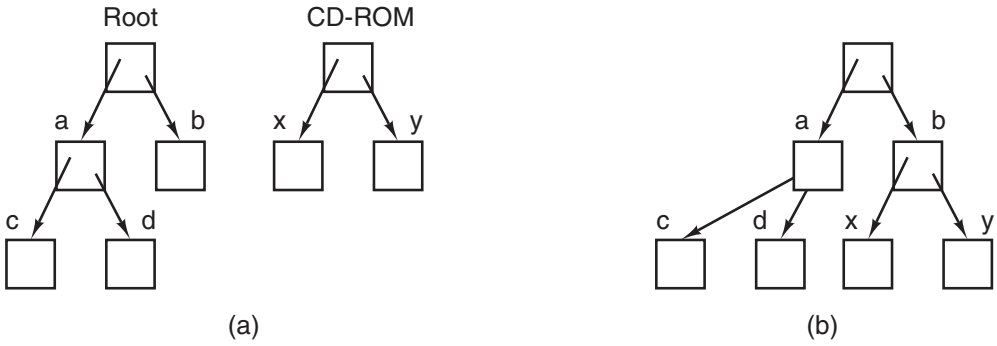


Figure 1-15. (a) Before mounting, the files on the CD-ROM are not accessible. (b) After mounting, they are part of the file hierarchy.

Another important concept in UNIX is the **special file**. Special files are provided in order to make I/O devices look like files. That way, they can be read and written using the same system calls as are used for reading and writing files. Two kinds of special files exist: **block special files** and **character special files**. Block special files are used to model devices that consist of a collection of randomly addressable blocks, such as disks. By opening a block special file and reading, say, block 4, a program can directly access the fourth block on the device, without regard to the structure of the file system contained on it. Similarly, character special files are used to model printers, modems, and other devices that accept or output a character stream. By convention, the special files are kept in the */dev* directory. For example, */dev/lp* might be the printer (once called the line printer).

The last feature we will discuss in this overview relates to both processes and files: pipes. A **pipe** is a sort of pseudofile that can be used to connect two processes, as shown in Fig. 1-16. If processes *A* and *B* wish to talk using a pipe, they must set it up in advance. When process *A* wants to send data to process *B*, it writes on the pipe as though it were an output file. In fact, the implementation of a pipe is very much like that of a file. Process *B* can read the data by reading from the pipe as though it were an input file. Thus, communication between processes in UNIX looks very much like ordinary file reads and writes. Stronger yet, the only way a process can discover that the output file it is writing on is not really a file, but a pipe, is by making a special system call. File systems are very important. We will have much more to say about them in Chap. 4 and also in Chaps. 10 and 11.

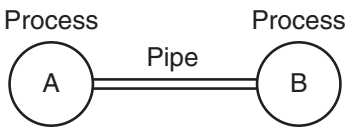


Figure 1-16. Two processes connected by a pipe.

1.5.4 Input/Output

All computers have physical devices for acquiring input and producing output. After all, what good would a computer be if the users could not tell it what to do and could not get the results after it did the work requested? Many kinds of input and output devices exist, including keyboards, monitors, printers, and so on. It is up to the operating system to manage these devices.

Consequently, every operating system has an I/O subsystem for managing its I/O devices. Some of the I/O software is device independent, that is, applies to many or all I/O devices equally well. Other parts of it, such as device drivers, are specific to particular I/O devices. In Chap. 5 we will have a look at I/O software.

1.5.5 Protection

Computers contain large amounts of information that users often want to protect and keep confidential. This information may include email, business plans, tax returns, and much more. It is up to the operating system to manage the system security so that files, for example, are accessible only to authorized users.

As a simple example, just to get an idea of how security can work, consider UNIX. Files in UNIX are protected by assigning each one a 9-bit binary protection code. The protection code consists of three 3-bit fields, one for the owner, one for other members of the owner's group (users are divided into groups by the system administrator), and one for everyone else. Each field has a bit for read access, a bit for write access, and a bit for execute access. These 3 bits are known as the **rwx bits**. For example, the protection code *rwxr-x--x* means that the owner can **r**ead, **w**rite, or e**x**ecute the file, other group members can read or execute (but not write) the file, and everyone else can execute (but not read or write) the file. For a directory, *x* indicates search permission. A dash means that the corresponding permission is absent.

In addition to file protection, there are many other security issues. Protecting the system from unwanted intruders, both human and nonhuman (e.g., viruses) is one of them. We will look at various security issues in Chap. 9.

1.5.6 The Shell

The operating system is the code that carries out the system calls. Editors, compilers, assemblers, linkers, utility programs, and command interpreters definitely are not part of the operating system, even though they are important and useful. At the risk of confusing things somewhat, in this section we will look briefly at the UNIX command interpreter, the shell. Although it is not part of the operating system, it makes heavy use of many operating system features and thus serves as a good example of how the system calls are used. It is also the main interface

between a user sitting at his terminal and the operating system, unless the user is using a graphical user interface. Many shells exist, including *sh*, *csh*, *ksh*, and *bash*. All of them support the functionality described below, which derives from the original shell (*sh*).

When any user logs in, a shell is started up. The shell has the terminal as standard input and standard output. It starts out by typing the **prompt**, a character such as a dollar sign, which tells the user that the shell is waiting to accept a command. If the user now types

```
date
```

for example, the shell creates a child process and runs the *date* program as the child. While the child process is running, the shell waits for it to terminate. When the child finishes, the shell types the prompt again and tries to read the next input line.

The user can specify that standard output be redirected to a file, for example,

```
date >file
```

Similarly, standard input can be redirected, as in

```
sort <file1 >file2
```

which invokes the sort program with input taken from *file1* and output sent to *file2*.

The output of one program can be used as the input for another program by connecting them with a pipe. Thus

```
cat file1 file2 file3 | sort >/dev/lp
```

invokes the *cat* program to con*cat*enate three files and send the output to *sort* to arrange all the lines in alphabetical order. The output of *sort* is redirected to the file */dev/lp*, typically the printer.

If a user puts an ampersand after a command, the shell does not wait for it to complete. Instead it just gives a prompt immediately. Consequently,

```
cat file1 file2 file3 | sort >/dev/lp &
```

starts up the sort as a background job, allowing the user to continue working normally while the sort is going on. The shell has a number of other interesting features, which we do not have space to discuss here. Most books on UNIX discuss the shell at some length (e.g., Kernighan and Pike, 1984; Quigley, 2004; Robbins, 2005).

Most personal computers these days use a GUI. In fact, the GUI is just a program running on top of the operating system, like a shell. In Linux systems, this fact is made obvious because the user has a choice of (at least) two GUIs: Gnome and KDE or none at all (using a terminal window on X11). In Windows, it is also possible to replace the standard GUI desktop (*Windows Explorer*) with a different program by changing some values in the registry, although few people do this.

1.5.7 Ontogeny Recapitulates Phylogeny

After Charles Darwin's book *On the Origin of the Species* was published, the German zoologist Ernst Haeckel stated that "ontogeny recapitulates phylogeny." By this he meant that the development of an embryo (ontogeny) repeats (i.e., recapitulates) the evolution of the species (phylogeny). In other words, after fertilization, a human egg goes through stages of being a fish, a pig, and so on before turning into a human baby. Modern biologists regard this as a gross simplification, but it still has a kernel of truth in it.

Something vaguely analogous has happened in the computer industry. Each new species (mainframe, minicomputer, personal computer, handheld, embedded computer, smart card, etc.) seems to go through the development that its ancestors did, both in hardware and in software. We often forget that much of what happens in the computer business and a lot of other fields is technology driven. The reason the ancient Romans lacked cars is not that they liked walking so much. It is because they did not know how to build cars. Personal computers exist *not* because millions of people have a centuries-old pent-up desire to own a computer, but because it is now possible to manufacture them cheaply. We often forget how much technology affects our view of systems and it is worth reflecting on this point from time to time.

In particular, it frequently happens that a change in technology renders some idea obsolete and it quickly vanishes. However, another change in technology could revive it again. This is especially true when the change has to do with the relative performance of different parts of the system. For instance, when CPUs became much faster than memories, caches became important to speed up the "slow" memory. If new memory technology someday makes memories much faster than CPUs, caches will vanish. And if a new CPU technology makes them faster than memories again, caches will reappear. In biology, extinction is forever, but in computer science, it is sometimes only for a few years.

As a consequence of this impermanence, in this book we will from time to time look at "obsolete" concepts, that is, ideas that are not optimal with current technology. However, changes in the technology may bring back some of the so-called "obsolete concepts." For this reason, it is important to understand why a concept is obsolete and what changes in the environment might bring it back again.

To make this point clearer, let us consider a simple example. Early computers had hardwired instruction sets. The instructions were executed directly by hardware and could not be changed. Then came microprogramming (first introduced on a large scale with the IBM 360), in which an underlying interpreter carried out the "hardware instructions" in software. Hardwired execution became obsolete. It was not flexible enough. Then RISC computers were invented, and microprogramming (i.e., interpreted execution) became obsolete because direct execution was faster. Now we are seeing the resurgence of interpretation in the form of Java applets that are sent over the Internet and interpreted upon arrival. Execution speed

is not always crucial because network delays are so great that they tend to dominate. Thus the pendulum has already swung several cycles between direct execution and interpretation and may yet swing again in the future.

Large Memories

Let us now examine some historical developments in hardware and how they have affected software repeatedly. The first mainframes had limited memory. A fully loaded IBM 7090 or 7094, which played king of the mountain from late 1959 until 1964, had just over 128 KB of memory. It was mostly programmed in assembly language and its operating system was written in assembly language to save precious memory.

As time went on, compilers for languages like FORTRAN and COBOL got good enough that assembly language was pronounced dead. But when the first commercial minicomputer (the PDP-1) was released, it had only 4096 18-bit words of memory, and assembly language made a surprise comeback. Eventually, minicomputers acquired more memory and high-level languages became prevalent on them.

When microcomputers hit in the early 1980s, the first ones had 4-KB memories and assembly-language programming rose from the dead. Embedded computers often used the same CPU chips as the microcomputers (8080s, Z80s, and later 8086s) and were also programmed in assembler initially. Now their descendants, the personal computers, have lots of memory and are programmed in C, C++, Java, and other high-level languages. Smart cards are undergoing a similar development, although beyond a certain size, the smart cards often have a Java interpreter and execute Java programs interpretively, rather than having Java being compiled to the smart card's machine language.

Protection Hardware

Early mainframes, like the IBM 7090/7094, had no protection hardware, so they just ran one program at a time. A buggy program could wipe out the operating system and easily crash the machine. With the introduction of the IBM 360, a primitive form of hardware protection became available. These machines could then hold several programs in memory at the same time and let them take turns running (multiprogramming). Monoprogramming was declared obsolete.

At least until the first minicomputer showed up—without protection hardware—so multiprogramming was not possible. Although the PDP-1 and PDP-8 had no protection hardware, eventually the PDP-11 did, and this feature led to multiprogramming and eventually to UNIX.

When the first microcomputers were built, they used the Intel 8080 CPU chip, which had no hardware protection, so we were back to monoprogramming—one program in memory at a time. It was not until the Intel 80286 chip that protection

hardware was added and multiprogramming became possible. Until this day, many embedded systems have no protection hardware and run just a single program.

Now let us look at operating systems. The first mainframes initially had no protection hardware and no support for multiprogramming, so they ran simple operating systems that handled one manually loaded program at a time. Later they acquired the hardware and operating system support to handle multiple programs at once, and then full timesharing capabilities.

When minicomputers first appeared, they also had no protection hardware and ran one manually loaded program at a time, even though multiprogramming was well established in the mainframe world by then. Gradually, they acquired protection hardware and the ability to run two or more programs at once. The first microcomputers were also capable of running only one program at a time, but later acquired the ability to multiprogram. Handheld computers and smart cards went the same route.

In all cases, the software development was dictated by technology. The first microcomputers, for example, had something like 4 KB of memory and no protection hardware. High-level languages and multiprogramming were simply too much for such a tiny system to handle. As the microcomputers evolved into modern personal computers, they acquired the necessary hardware and then the necessary software to handle more advanced features. It is likely that this development will continue for years to come. Other fields may also have this wheel of reincarnation, but in the computer industry it seems to spin faster.

Disks

Early mainframes were largely magnetic-tape based. They would read in a program from tape, compile it, run it, and write the results back to another tape. There were no disks and no concept of a file system. That began to change when IBM introduced the first hard disk—the RAMAC (RAndoM ACcess) in 1956. It occupied about 4 square meters of floor space and could store 5 million 7-bit characters, enough for one medium-resolution digital photo. But with an annual rental fee of $35,000, assembling enough of them to store the equivalent of a roll of film got pricey quite fast. But eventually prices came down and primitive file systems were developed.

Typical of these new developments was the CDC 6600, introduced in 1964 and for years by far the fastest computer in the world. Users could create so-called "permanent files" by giving them names and hoping that no other user had also decided that, say, "data" was a suitable name for a file. This was a single-level directory. Eventually, mainframes developed complex hierarchical file systems, perhaps culminating in the MULTICS file system.

As minicomputers came into use, they eventually also had hard disks. The standard disk on the PDP-11 when it was introduced in 1970 was the RK05 disk, with a capacity of 2.5 MB, about half of the IBM RAMAC, but it was only about

40 cm in diameter and 5 cm high. But it, too, had a single-level directory initially. When microcomputers came out, CP/M was initially the dominant operating system, and it, too, supported just one directory on the (floppy) disk.

Virtual Memory

Virtual memory (discussed in Chap. 3) gives the ability to run programs larger than the machine's physical memory by rapidly moving pieces back and forth between RAM and disk. It underwent a similar development, first appearing on mainframes, then moving to the minis and the micros. Virtual memory also allowed having a program dynamically link in a library at run time instead of having it compiled in. MULTICS was the first system to allow this. Eventually, the idea propagated down the line and is now widely used on most UNIX and Windows systems.

In all these developments, we see ideas invented in one context and later thrown out when the context changes (assembly-language programming, monoprogramming, single-level directories, etc.) only to reappear in a different context often a decade later. For this reason in this book we will sometimes look at ideas and algorithms that may seem dated on today's gigabyte PCs, but which may soon come back on embedded computers and smart cards.

1.6 SYSTEM CALLS

We have seen that operating systems have two main functions: providing abstractions to user programs and managing the computer's resources. For the most part, the interaction between user programs and the operating system deals with the former; for example, creating, writing, reading, and deleting files. The resource-management part is largely transparent to the users and done automatically. Thus, the interface between user programs and the operating system is primarily about dealing with the abstractions. To really understand what operating systems do, we must examine this interface closely. The system calls available in the interface vary from one operating system to another (although the underlying concepts tend to be similar).

We are thus forced to make a choice between (1) vague generalities ("operating systems have system calls for reading files") and (2) some specific system ("UNIX has a read system call with three parameters: one to specify the file, one to tell where the data are to be put, and one to tell how many bytes to read").

We have chosen the latter approach. It's more work that way, but it gives more insight into what operating systems really do. Although this discussion specifically refers to POSIX (International Standard 9945-1), hence also to UNIX, System V, BSD, Linux, MINIX 3, and so on, most other modern operating systems have system calls that perform the same functions, even if the details differ. Since the actual

40 cm in diameter and 5 cm high. But it, too, had a single-level directory initially. When microcomputers came out, CP/M was initially the dominant operating system, and it, too, supported just one directory on the (floppy) disk.

Virtual Memory

Virtual memory (discussed in Chap. 3) gives the ability to run programs larger than the machine's physical memory by rapidly moving pieces back and forth between RAM and disk. It underwent a similar development, first appearing on mainframes, then moving to the minis and the micros. Virtual memory also allowed having a program dynamically link in a library at run time instead of having it compiled in. MULTICS was the first system to allow this. Eventually, the idea propagated down the line and is now widely used on most UNIX and Windows systems.

In all these developments, we see ideas invented in one context and later thrown out when the context changes (assembly-language programming, monoprogramming, single-level directories, etc.) only to reappear in a different context often a decade later. For this reason in this book we will sometimes look at ideas and algorithms that may seem dated on today's gigabyte PCs, but which may soon come back on embedded computers and smart cards.

1.6 SYSTEM CALLS

We have seen that operating systems have two main functions: providing abstractions to user programs and managing the computer's resources. For the most part, the interaction between user programs and the operating system deals with the former; for example, creating, writing, reading, and deleting files. The resource-management part is largely transparent to the users and done automatically. Thus, the interface between user programs and the operating system is primarily about dealing with the abstractions. To really understand what operating systems do, we must examine this interface closely. The system calls available in the interface vary from one operating system to another (although the underlying concepts tend to be similar).

We are thus forced to make a choice between (1) vague generalities ("operating systems have system calls for reading files") and (2) some specific system ("UNIX has a read system call with three parameters: one to specify the file, one to tell where the data are to be put, and one to tell how many bytes to read").

We have chosen the latter approach. It's more work that way, but it gives more insight into what operating systems really do. Although this discussion specifically refers to POSIX (International Standard 9945-1), hence also to UNIX, System V, BSD, Linux, MINIX 3, and so on, most other modern operating systems have system calls that perform the same functions, even if the details differ. Since the actual

mechanics of issuing a system call are highly machine dependent and often must be expressed in assembly code, a procedure library is provided to make it possible to make system calls from C programs and often from other languages as well.

It is useful to keep the following in mind. Any single-CPU computer can execute only one instruction at a time. If a process is running a user program in user mode and needs a system service, such as reading data from a file, it has to execute a trap instruction to transfer control to the operating system. The operating system then figures out what the calling process wants by inspecting the parameters. Then it carries out the system call and returns control to the instruction following the system call. In a sense, making a system call is like making a special kind of procedure call, only system calls enter the kernel and procedure calls do not.

To make the system-call mechanism clearer, let us take a quick look at the read system call. As mentioned above, it has three parameters: the first one specifying the file, the second one pointing to the buffer, and the third one giving the number of bytes to read. Like nearly all system calls, it is invoked from C programs by calling a library procedure with the same name as the system call: *read*. A call from a C program might look like this:

```
count = read(fd, buffer, nbytes);
```

The system call (and the library procedure) return the number of bytes actually read in *count*. This value is normally the same as *nbytes*, but may be smaller, if, for example, end-of-file is encountered while reading.

If the system call cannot be carried out owing to an invalid parameter or a disk error, *count* is set to −1, and the error number is put in a global variable, *errno*. Programs should always check the results of a system call to see if an error occurred.

System calls are performed in a series of steps. To make this concept clearer, let us examine the read call discussed above. In preparation for calling the *read* library procedure, which actually makes the read system call, the calling program first pushes the parameters onto the stack, as shown in steps 1–3 in Fig. 1-17.

C and C++ compilers push the parameters onto the stack in reverse order for historical reasons (having to do with making the first parameter to *printf*, the format string, appear on top of the stack). The first and third parameters are called by value, but the second parameter is passed by reference, meaning that the address of the buffer (indicated by &) is passed, not the contents of the buffer. Then comes the actual call to the library procedure (step 4). This instruction is the normal procedure-call instruction used to call all procedures.

The library procedure, possibly written in assembly language, typically puts the system-call number in a place where the operating system expects it, such as a register (step 5). Then it executes a TRAP instruction to switch from user mode to kernel mode and start execution at a fixed address within the kernel (step 6). The TRAP instruction is actually fairly similar to the procedure-call instruction in the

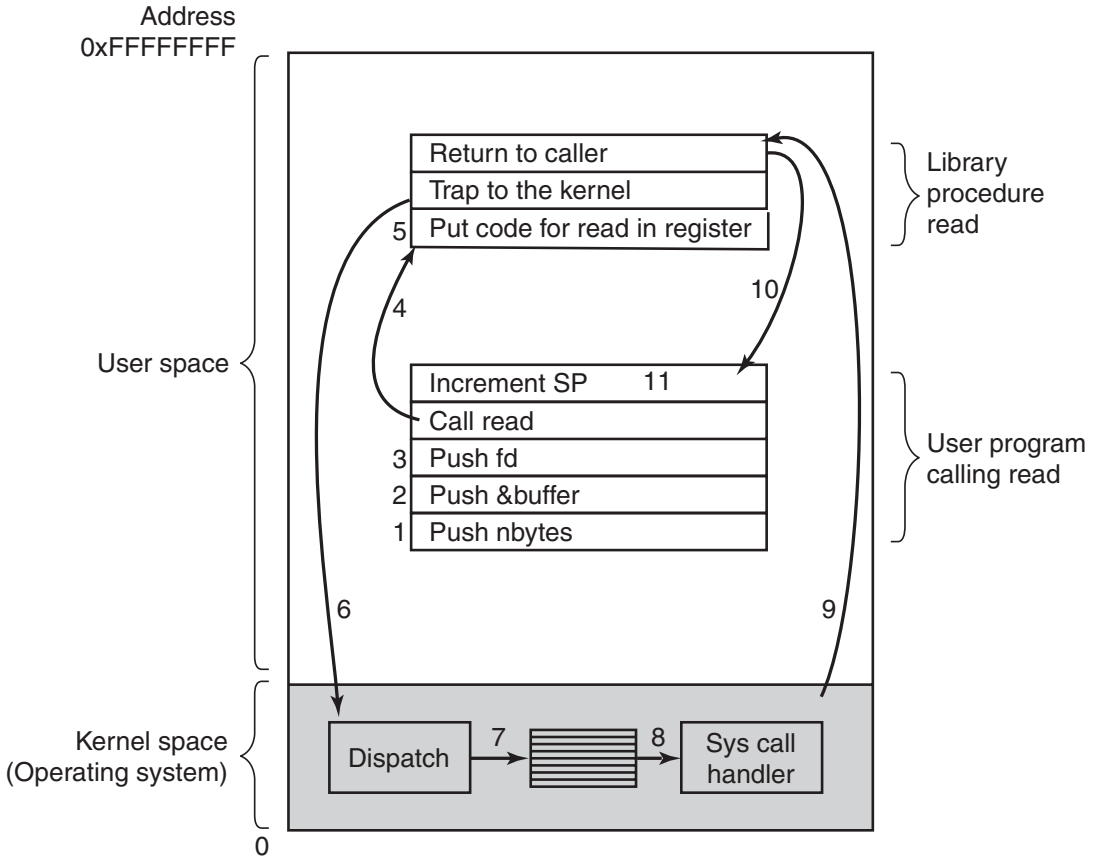


Figure 1-17. The 11 steps in making the system call read(fd, buffer, nbytes).

sense that the instruction following it is taken from a distant location and the return address is saved on the stack for use later.

Nevertheless, the TRAP instruction also differs from the procedure-call instruction in two fundamental ways. First, as a side effect, it switches into kernel mode. The procedure call instruction does not change the mode. Second, rather than giving a relative or absolute address where the procedure is located, the TRAP instruction cannot jump to an arbitrary address. Depending on the architecture, either it jumps to a single fixed location or there is an 8-bit field in the instruction giving the index into a table in memory containing jump addresses, or equivalent.

The kernel code that starts following the TRAP examines the system-call number and then dispatches to the correct system-call handler, usually via a table of pointers to system-call handlers indexed on system-call number (step 7). At that point the system-call handler runs (step 8). Once it has completed its work, control may be returned to the user-space library procedure at the instruction following the TRAP instruction (step 9). This procedure then returns to the user program in the usual way procedure calls return (step 10).

To finish the job, the user program has to clean up the stack, as it does after any procedure call (step 11). Assuming the stack grows downward, as it often

does, the compiled code increments the stack pointer exactly enough to remove the parameters pushed before the call to *read*. The program is now free to do whatever it wants to do next.

In step 9 above, we said "may be returned to the user-space library procedure" for good reason. The system call may block the caller, preventing it from continuing. For example, if it is trying to read from the keyboard and nothing has been typed yet, the caller has to be blocked. In this case, the operating system will look around to see if some other process can be run next. Later, when the desired input is available, this process will get the attention of the system and run steps 9–11.

In the following sections, we will examine some of the most heavily used POSIX system calls, or more specifically, the library procedures that make those system calls. POSIX has about 100 procedure calls. Some of the most important ones are listed in Fig. 1-18, grouped for convenience in four categories. In the text we will briefly examine each call to see what it does.

To a large extent, the services offered by these calls determine most of what the operating system has to do, since the resource management on personal computers is minimal (at least compared to big machines with multiple users). The services include things like creating and terminating processes, creating, deleting, reading, and writing files, managing directories, and performing input and output.

As an aside, it is worth pointing out that the mapping of POSIX procedure calls onto system calls is not one-to-one. The POSIX standard specifies a number of procedures that a conformant system must supply, but it does not specify whether they are system calls, library calls, or something else. If a procedure can be carried out without invoking a system call (i.e., without trapping to the kernel), it will usually be done in user space for reasons of performance. However, most of the POSIX procedures do invoke system calls, usually with one procedure mapping directly onto one system call. In a few cases, especially where several required procedures are only minor variations of one another, one system call handles more than one library call.

1.6.1 System Calls for Process Management

The first group of calls in Fig. 1-18 deals with process management. Fork is a good place to start the discussion. Fork is the only way to create a new process in POSIX. It creates an exact duplicate of the original process, including all the file descriptors, registers—everything. After the fork, the original process and the copy (the parent and child) go their separate ways. All the variables have identical values at the time of the fork, but since the parent's data are copied to create the child, subsequent changes in one of them do not affect the other one. (The program text, which is unchangeable, is shared between parent and child.) The fork call returns a value, which is zero in the child and equal to the child's **PID** (**Process IDentifier**) in the parent. Using the returned PID, the two processes can see which one is the parent process and which one is the child process.

Process management

Call	Description
pid = fork( )	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing, or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

Directory- and file-system management

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

Miscellaneous

Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

Figure 1-18. Some of the major POSIX system calls. The return code *s* is −1 if an error has occurred. The return codes are as follows: *pid* is a process id, *fd* is a file descriptor, *n* is a byte count, *position* is an offset within the file, and *seconds* is the elapsed time. The parameters are explained in the text.

In most cases, after a fork, the child will need to execute different code from the parent. Consider the case of the shell. It reads a command from the terminal, forks off a child process, waits for the child to execute the command, and then reads the next command when the child terminates. To wait for the child to finish,

the parent executes a waitpid system call, which just waits until the child terminates (any child if more than one exists). Waitpid can wait for a specific child, or for any old child by setting the first parameter to −1. When waitpid completes, the address pointed to by the second parameter, *statloc*, will be set to the child process' exit status (normal or abnormal termination and exit value). Various options are also provided, specified by the third parameter. For example, returning immediately if no child has already exited.

Now consider how fork is used by the shell. When a command is typed, the shell forks off a new process. This child process must execute the user command. It does this by using the execve system call, which causes its entire core image to be replaced by the file named in its first parameter. (Actually, the system call itself is exec, but several library procedures call it with different parameters and slightly different names. We will treat these as system calls here.) A highly simplified shell illustrating the use of fork, waitpid, and execve is shown in Fig. 1-19.

```
#define TRUE 1

while (TRUE) {                                  /* repeat forever */
      type_prompt( );                           /* display prompt on the screen */
      read_command(command, parameters);        /* read input from terminal */

      if (fork( ) != 0) {                       /* fork off child process */
           /* Parent code. */
           waitpid(-1, &status, 0);             /* wait for child to exit */
      } else {
           /* Child code. */
           execve(command, parameters, 0);      /* execute command */
      }
}
```

Figure 1-19. A stripped-down shell. Throughout this book, *TRUE* is assumed to be defined as 1.

In the most general case, execve has three parameters: the name of the file to be executed, a pointer to the argument array, and a pointer to the environment array. These will be described shortly. Various library routines, including *execl*, *execv*, *execle*, and *execve*, are provided to allow the parameters to be omitted or specified in various ways. Throughout this book we will use the name exec to represent the system call invoked by all of these.

Let us consider the case of a command such as

```
cp file1 file2
```

used to copy *file1* to *file2*. After the shell has forked, the child process locates and executes the file *cp* and passes to it the names of the source and target files.

The main program of *cp* (and main program of most other C programs) contains the declaration

```
main(argc, argv, envp)
```

where *argc* is a count of the number of items on the command line, including the program name. For the example above, *argc* is 3.

The second parameter, *argv*, is a pointer to an array. Element *i* of that array is a pointer to the *i*th string on the command line. In our example, *argv*[0] would point to the string "cp", *argv*[1] would point to the string "file1", and *argv*[2] would point to the string "file2".

The third parameter of *main*, *envp*, is a pointer to the environment, an array of strings containing assignments of the form *name = value* used to pass information such as the terminal type and home directory name to programs. There are library procedures that programs can call to get the environment variables, which are often used to customize how a user wants to perform certain tasks (e.g., the default printer to use). In Fig. 1-19, no environment is passed to the child, so the third parameter of *execve* is a zero.

If exec seems complicated, do not despair; it is (semantically) the most complex of all the POSIX system calls. All the other ones are much simpler. As an example of a simple one, consider exit, which processes should use when they are finished executing. It has one parameter, the exit status (0 to 255), which is returned to the parent via *statloc* in the waitpid system call.

Processes in UNIX have their memory divided up into three segments: the **text segment** (i.e., the program code), the **data segment** (i.e., the variables), and the **stack segment**. The data segment grows upward and the stack grows downward, as shown in Fig. 1-20. Between them is a gap of unused address space. The stack grows into the gap automatically, as needed, but expansion of the data segment is done explicitly by using a system call, brk, which specifies the new address where the data segment is to end. This call, however, is not defined by the POSIX standard, since programmers are encouraged to use the *malloc* library procedure for dynamically allocating storage, and the underlying implementation of *malloc* was not thought to be a suitable subject for standardization since few programmers use it directly and it is doubtful that anyone even notices that brk is not in POSIX.

1.6.2 System Calls for File Management

Many system calls relate to the file system. In this section we will look at calls that operate on individual files; in the next one we will examine those that involve directories or the file system as a whole.

To read or write a file, it must first be opened. This call specifies the file name to be opened, either as an absolute path name or relative to the working directory, as well as a code of *O_RDONLY*, *O_WRONLY*, or *O_RDWR*, meaning open for reading, writing, or both. To create a new file, the *O_CREAT* parameter is used.

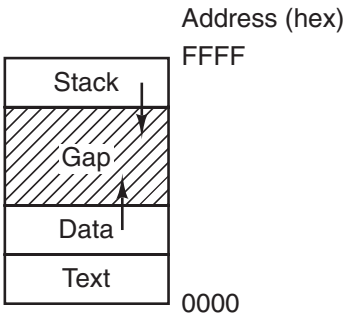


Figure 1-20. Processes have three segments: text, data, and stack.

The file descriptor returned can then be used for reading or writing. Afterward, the file can be closed by close, which makes the file descriptor available for reuse on a subsequent open.

The most heavily used calls are undoubtedly read and write. We saw read earlier. Write has the same parameters.

Although most programs read and write files sequentially, for some applications programs need to be able to access any part of a file at random. Associated with each file is a pointer that indicates the current position in the file. When reading (writing) sequentially, it normally points to the next byte to be read (written). The lseek call changes the value of the position pointer, so that subsequent calls to read or write can begin anywhere in the file.

Lseek has three parameters: the first is the file descriptor for the file, the second is a file position, and the third tells whether the file position is relative to the beginning of the file, the current position, or the end of the file. The value returned by lseek is the absolute position in the file (in bytes) after changing the pointer.

For each file, UNIX keeps track of the file mode (regular file, special file, directory, and so on), size, time of last modification, and other information. Programs can ask to see this information via the stat system call. The first parameter specifies the file to be inspected; the second one is a pointer to a structure where the information is to be put. The fstat calls does the same thing for an open file.

1.6.3 System Calls for Directory Management

In this section we will look at some system calls that relate more to directories or the file system as a whole, rather than just to one specific file as in the previous section. The first two calls, mkdir and rmdir, create and remove empty directories, respectively. The next call is link. Its purpose is to allow the same file to appear under two or more names, often in different directories. A typical use is to allow several members of the same programming team to share a common file, with each of them having the file appear in his own directory, possibly under different names. Sharing a file is not the same as giving every team member a private copy; having

a shared file means that changes that any member of the team makes are instantly visible to the other members—there is only one file. When copies are made of a file, subsequent changes made to one copy do not affect the others.

To see how link works, consider the situation of Fig. 1-21(a). Here are two users, *ast* and *jim*, each having his own directory with some files. If *ast* now executes a program containing the system call

```
link("/usr/jim/memo", "/usr/ast/note");
```

the file *memo* in *jim*'s directory is now entered into *ast*'s directory under the name *note*. Thereafter, */usr/jim/memo* and */usr/ast/note* refer to the same file. As an aside, whether user directories are kept in */usr*, */user*, */home*, or somewhere else is simply a decision made by the local system administrator.

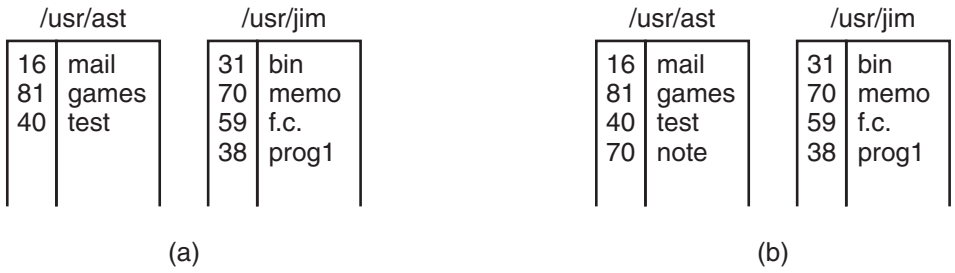


Figure 1-21. (a) Two directories before linking */usr/jim/memo* to *ast*'s directory. (b) The same directories after linking.

Understanding how link works will probably make it clearer what it does. Every file in UNIX has a unique number, its i-number, that identifies it. This i-number is an index into a table of **i-nodes**, one per file, telling who owns the file, where its disk blocks are, and so on. A directory is simply a file containing a set of (i-number, ASCII name) pairs. In the first versions of UNIX, each directory entry was 16 bytes—2 bytes for the i-number and 14 bytes for the name. Now a more complicated structure is needed to support long file names, but conceptually a directory is still a set of (i-number, ASCII name) pairs. In Fig. 1-21, *mail* has i-number 16, and so on. What link does is simply create a brand new directory entry with a (possibly new) name, using the i-number of an existing file. In Fig. 1-21(b), two entries have the same i-number (70) and thus refer to the same file. If either one is later removed, using the unlink system call, the other one remains. If both are removed, UNIX sees that no entries to the file exist (a field in the i-node keeps track of the number of directory entries pointing to the file), so the file is removed from the disk.

As we have mentioned earlier, the mount system call allows two file systems to be merged into one. A common situation is to have the root file system, containing the binary (executable) versions of the common commands and other heavily used files, on a hard disk (sub)partition and user files on another (sub)partition. Further, the user can then insert a USB disk with files to be read.

By executing the mount system call, the USB file system can be attached to the root file system, as shown in Fig. 1-22. A typical statement in C to mount is

```
mount("/dev/sdb0", "/mnt", 0);
```

where the first parameter is the name of a block special file for USB drive 0, the second parameter is the place in the tree where it is to be mounted, and the third parameter tells whether the file system is to be mounted read-write or read-only.



Figure 1-22. (a) File system before the mount. (b) File system after the mount.

After the mount call, a file on drive 0 can be accessed by just using its path from the root directory or the working directory, without regard to which drive it is on. In fact, second, third, and fourth drives can also be mounted anywhere in the tree. The mount call makes it possible to integrate removable media into a single integrated file hierarchy, without having to worry about which device a file is on. Although this example involves CD-ROMs, portions of hard disks (often called **partitions** or **minor devices**) can also be mounted this way, as well as external hard disks and USB sticks. When a file system is no longer needed, it can be unmounted with the umount system call.

1.6.4 Miscellaneous System Calls

A variety of other system calls exist as well. We will look at just four of them here. The chdir call changes the current working directory. After the call

```
chdir("/usr/ast/test");
```

an open on the file *xyz* will open */usr/ast/test/xyz*. The concept of a working directory eliminates the need for typing (long) absolute path names all the time.

In UNIX every file has a mode used for protection. The mode includes the read-write-execute bits for the owner, group, and others. The chmod system call makes it possible to change the mode of a file. For example, to make a file read-only by everyone except the owner, one could execute

```
chmod("file", 0644);
```

The kill system call is the way users and user processes send signals. If a process is prepared to catch a particular signal, then when it arrives, a signal handler is

run. If the process is not prepared to handle a signal, then its arrival kills the process (hence the name of the call).

POSIX defines a number of procedures for dealing with time. For example, time just returns the current time in seconds, with 0 corresponding to Jan. 1, 1970 at midnight (just as the day was starting, not ending). On computers using 32-bit words, the maximum value time can return is $2^{32} - 1$ seconds (assuming an unsigned integer is used). This value corresponds to a little over 136 years. Thus in the year 2106, 32-bit UNIX systems will go berserk, not unlike the famous Y2K problem that would have wreaked havoc with the world's computers in 2000, were it not for the massive effort the IT industry put into fixing the problem. If you currently have a 32-bit UNIX system, you are advised to trade it in for a 64-bit one sometime before the year 2106.

1.6.5 The Windows Win32 API

So far we have focused primarily on UNIX. Now it is time to look briefly at Windows. Windows and UNIX differ in a fundamental way in their respective programming models. A UNIX program consists of code that does something or other, making system calls to have certain services performed. In contrast, a Windows program is normally event driven. The main program waits for some event to happen, then calls a procedure to handle it. Typical events are keys being struck, the mouse being moved, a mouse button being pushed, or a USB drive inserted. Handlers are then called to process the event, update the screen and update the internal program state. All in all, this leads to a somewhat different style of programming than with UNIX, but since the focus of this book is on operating system function and structure, these different programming models will not concern us much more.

Of course, Windows also has system calls. With UNIX, there is almost a one-to-one relationship between the system calls (e.g., read) and the library procedures (e.g., *read*) used to invoke the system calls. In other words, for each system call, there is roughly one library procedure that is called to invoke it, as indicated in Fig. 1-17. Furthermore, POSIX has only about 100 procedure calls.

With Windows, the situation is radically different. To start with, the library calls and the actual system calls are highly decoupled. Microsoft has defined a set of procedures called the **Win32 API** (**Application Programming Interface**) that programmers are expected to use to get operating system services. This interface is (partially) supported on all versions of Windows since Windows 95. By decoupling the API interface from the actual system calls, Microsoft retains the ability to change the actual system calls in time (even from release to release) without invalidating existing programs. What actually constitutes Win32 is also slightly ambiguous because recent versions of Windows have many new calls that were not previously available. In this section, Win32 means the interface supported by all versions of Windows. Win32 provides compatibility among versions of Windows.

The number of Win32 API calls is extremely large, numbering in the thousands. Furthermore, while many of them do invoke system calls, a substantial number are carried out entirely in user space. As a consequence, with Windows it is impossible to see what is a system call (i.e., performed by the kernel) and what is simply a user-space library call. In fact, what is a system call in one version of Windows may be done in user space in a different version, and vice versa. When we discuss the Windows system calls in this book, we will use the Win32 procedures (where appropriate) since Microsoft guarantees that these will be stable over time. But it is worth remembering that not all of them are true system calls (i.e., traps to the kernel).

The Win32 API has a huge number of calls for managing windows, geometric figures, text, fonts, scrollbars, dialog boxes, menus, and other features of the GUI. To the extent that the graphics subsystem runs in the kernel (true on some versions of Windows but not on all), these are system calls; otherwise they are just library calls. Should we discuss these calls in this book or not? Since they are not really related to the function of an operating system, we have decided not to, even though they may be carried out by the kernel. Readers interested in the Win32 API should consult one of the many books on the subject (e.g., Hart, 1997; Rector and Newcomer, 1997; and Simon, 1997).

Even introducing all the Win32 API calls here is out of the question, so we will restrict ourselves to those calls that roughly correspond to the functionality of the UNIX calls listed in Fig. 1-18. These are listed in Fig. 1-23.

Let us now briefly go through the list of Fig. 1-23. CreateProcess creates a new process. It does the combined work of fork and execve in UNIX. It has many parameters specifying the properties of the newly created process. Windows does not have a process hierarchy as UNIX does so there is no concept of a parent process and a child process. After a process is created, the creator and createe are equals. WaitForSingleObject is used to wait for an event. Many possible events can be waited for. If the parameter specifies a process, then the caller waits for the specified process to exit, which is done using ExitProcess.

The next six calls operate on files and are functionally similar to their UNIX counterparts although they differ in the parameters and details. Still, files can be opened, closed, read, and written pretty much as in UNIX. The SetFilePointer and GetFileAttributesEx calls set the file position and get some of the file attributes.

Windows has directories and they are created with CreateDirectory and RemoveDirectory API calls, respectively. There is also a notion of a current directory, set by SetCurrentDirectory. The current time of day is acquired using GetLocalTime.

The Win32 interface does not have links to files, mounted file systems, security, or signals, so the calls corresponding to the UNIX ones do not exist. Of course, Win32 has a huge number of other calls that UNIX does not have, especially for managing the GUI. Windows Vista has an elaborate security system and also supports file links. Windows 7 and 8 add yet more features and system calls.

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount, so no umount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

Figure 1-23. The Win32 API calls that roughly correspond to the UNIX calls of Fig. 1-18. It is worth emphasizing that Windows has a very large number of other system calls, most of which do not correspond to anything in UNIX.

One last note about Win32 is perhaps worth making. Win32 is not a terribly uniform or consistent interface. The main culprit here was the need to be backward compatible with the previous 16-bit interface used in Windows 3.x.

1.7 OPERATING SYSTEM STRUCTURE

Now that we have seen what operating systems look like on the outside (i.e., the programmer's interface), it is time to take a look inside. In the following sections, we will examine six different structures that have been tried, in order to get some idea of the spectrum of possibilities. These are by no means exhaustive, but they give an idea of some designs that have been tried in practice. The six designs we will discuss here are monolithic systems, layered systems, microkernels, client-server systems, virtual machines, and exokernels.

Unit 2

Process management and Threads

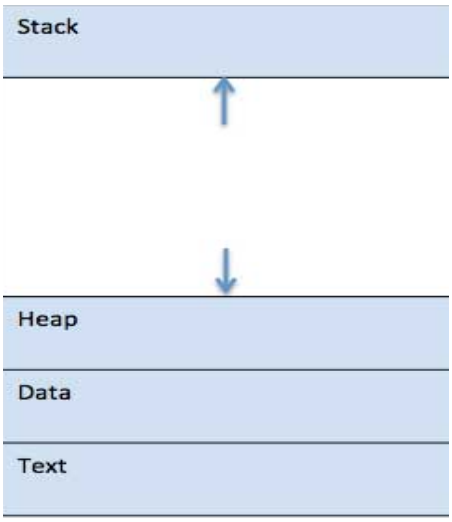
What is Process?

“A process is a program in execution including the current values of the program counter, registers, and variables.”

A process is an ‘active’ entity instead of a program, which is considered a ‘passive’ entity.

- In the Operating System, a Process is something that is currently under execution. So, an active program can be called a Process. For example, when you want to search for something on the web then you start a browser. So, this can be a Process.
- To put it in simple terms, we write our computer programs in a text file and when we execute this program, it becomes a process that performs all the tasks mentioned in the program.

When a program is loaded into the memory and it becomes a process, it can be divided into four sections ─ stack, heap, text, and data. The following image shows a simplified layout of a process inside the main memory –



Stack

The process Stack contains the temporary data such as method/function parameters, return address, and local variables.

Heap

This is dynamically allocated memory to a process during its run time.

Text

This includes the current activity represented by the value of the Program Counter and the contents of the processor's registers.

Data

This section contains the global and static variables.

A Process has various attributes associated with it. Some of the attributes of a Process are:

Process Id: Every process will be given an id called Process Id to uniquely identify that process from the other processes.

Process state: Each process has some states associated with it at a particular instant in time. This is denoted by process state. It can be ready, waiting, running, etc.

CPU scheduling information: Each process is executed by using some process scheduling algorithms like FCSF, Round-Robin, SJF, etc.

I/O information: Each process needs some I/O devices for their execution. So, the information about the device allocated.

2.1 Process Model

In this model, all the runnable software on the computer, sometimes including the operating system, is organized into several sequential processes.

A process is just an instance of an executing program, including the current values of the program counter, registers, and variables.

Conceptually, each process has its own virtual CPU. switches back and forth from process to process, but to understand the system, it is much easier to think about a collection of processes running in (pseudo) parallel than to try to keep track of how the CPU switches from program to program. This rapid switching back and forth is called multiprogramming.

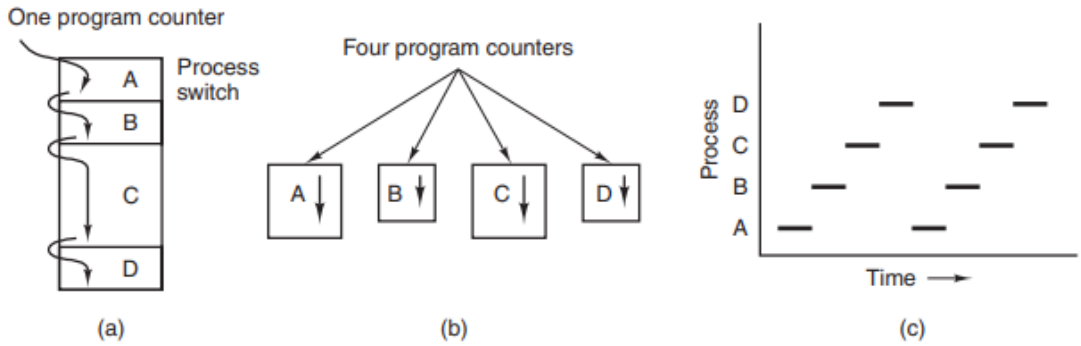


Figure 2-1. (a) Multiprogramming four programs. (b) Conceptual model of four independent, sequential processes. (c) Only one program is active at once.

(a) Multiprogramming Model of four programs

In this, there are four programs in memory. We have a single shared processor for all programs. There is only one program counter for all programs in memory.

In this first diagram (a), the Program counter is initialized to execute the part of program A then with the help of the process switch it transfers control to program B and executes some part of this. After that program counter of Program C is active and execute some part of it and

so on to program D. then all the step continuous may be done randomly with the help of process switches until all program to be completed.

(b) Conceptual model of four independent, sequential processes

There are four processes, each with its own flow of control (i.e., its own logical program counter), and each one runs independently of the other ones. There is only one physical program counter. So, when each process runs, its logical program counter is loaded into the real program counter. When it is finished (for the time being), the physical program counter is saved in the process' stored logical program counter in memory.

(c) Only one program is active at once.

At any given instant only one process runs and the other process after some interval of time. All the processes have made progress, but at any given instant only one process is actually running.

2.2 Process Creation

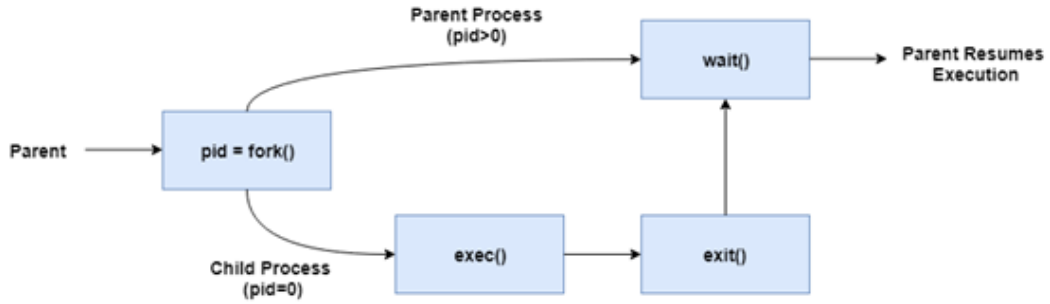
Operating systems need some way to create processes. In very simple systems, or in systems designed for running only a single application (e.g., the controller in a microwave oven), it may be possible to have all the processes that will ever be needed present when the system comes up.

In general-purpose systems, however, some way is needed to create and terminate processes as needed during the operation.

- There are four principal events that cause processes to be created:
    1. System initialization. When an OS is booted, typically several processes are created.
    2. Execution of a process creation system call by a running process. Often a running process will issue system calls to create one or more new processes to help it do its job. Creating new processes is particularly useful when the work to be done can easily be formulated in terms of several related, but otherwise independent interacting processes.
    3. A user request to create a new process. In interactive systems, users can start a program by typing a command or (double) clicking an icon.
    4. Initiation of a batch job. Here users can submit batch jobs to the system (possibly remotely). When the OS decides that it has the resources to run another job, it creates a new process and runs the next job from the input queue in it.
- When an operating system is booted, typically numerous processes are created. Some of these are foreground processes, that is, processes that interact with (human) users and perform work for them. Others run in the background and are not associated with particular users, but instead have some specific function. For example, one background process may be designed to accept incoming emails, sleeping most of the day but suddenly springing to life when email arrives.
- In all these cases, a new process is created by having an existing process execute a process creation system call (in UNIX, fork()).

- The creating process is called a parent process, and the new processes are called the children of that process. Each of these new processes may in turn create other processes, forming a tree of processes.
- Most OSs (including UNIX and the Windows family of OSs) identify processes according to a unique process identifier (or PID), which is typically an integer number.
- In general, a process will need certain resources (CPU time, memory, files, I/0 devices) to accomplish its task. When a process creates a subprocess, that subprocess may be able to obtain its resources directly from the OS, or it may be constrained to a subset of the resources of the parent process.
    - The parent may have to partition its resources among its children,
    - or it may be able to share some resources (such as memory or files) among several of its children.
- Restricting a child process to a subset of the parent's resources prevents any process from overloading the system by creating too many subprocesses.
- When a process creates a new process, two possibilities exist in terms of execution:
    - The parent continues to execute concurrently with its children, competing equally for the CPU.
    - The parent waits until some or all its children have terminated (on UNIX, see the man pages for {wait, waitpid, wait4, wait3}.).
- There are also two possibilities in terms of the address space of the new process:
    - The child process is a duplicate of the parent process (it has the same program and data as the parent, an exact clone). The two processes, the parent, and the child, have the same memory image, the same environment strings, and the same open files.
    - The child process has a new program loaded into it.

A diagram that demonstrates process creation using fork () is as follows −



Process Creation Using fork()

- In both UNIX and Windows systems, after a process is created, the parent and child have their own distinct address spaces. If either process changes a word in its address space, the change is not visible to the other process. In UNIX, the child's initial address space is a copy of the parents, but there are definitely two distinct address spaces involved; no writable memory is shared. Some UNIX implementations share the program text between the two since that cannot be modified.
- Alternatively, the child may share all the parent's memory, but in that case, the memory is shared copy-on-write, which means that whenever either of the two wants to modify

part of the memory, that chunk of memory is explicitly copied first to make sure the modification occurs in a private memory area. Again, no writable memory is shared. It is, however, possible for a newly created process to share some of its creator's other resources, such as open files. In Windows, the parent's and child's address spaces are different from the start.

2.3 Process Termination

A process terminates when it finishes executing its last statement. Its resources are returned to the system, it is purged from any system lists or tables, and its process control block (PCB) is erased i.e., the PCB's memory space is returned to a free memory pool. The new process terminates the existing process, usually due to following reasons:

Normal Exist Most processes terminates because they have done their job. This call is exist in UNIX.

Error Exist When process discovers a fatal error. For example, a user tries to compile a program that does not exist.

Fatal Error An error caused by process due to a bug in program for example, executing an illegal instruction, referring non-existing memory or dividing by zero.

Killed by another Process: A process executes a system call telling the Operating Systems to terminate some other process. In UNIX, this call is kill. In some systems when a process kills all processes it created are killed as well (UNIX does not work this way).

How process is terminated

- Process executes last statement and asks the operating system to delete it (**exit**).
- Output data from child to parent (via **wait**).
- Process resources are deallocated by operating system.
- Parent may terminate execution of children processes (**abort**).

A parent may terminate the executions of one of its children for a variety of reasons, such as:

- The child has exceeded its usage of some of the resources it has been allocated.
- The task assigned to the child is no longer required.
- The parent is exiting and the operating system does not allow a child to continue if its parent terminates.
- Operating system does not allow child to continue if its parent terminates. Cascading termination.

2.4 Process Hierarchies

In some systems, when a process creates another process, the parent process and child process continue to be associated in certain ways.

The child process can itself create more processes, forming a process hierarchy.

a process that has only one parent (but zero, one, two, or more children).

In Fig. 5.15, P1, P3, and P6 are parent processes, whereas P3 and P6 are also child processes. There may be many child processes of a parent process. There is a hierarchy between these processes in the tree structure. All child processes at one level will have one parent process, and they work only under the control of this parent process.

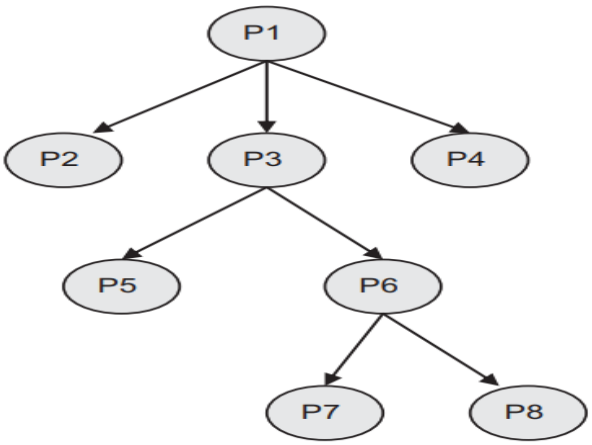


Fig. 5.15 Process hierarchy

For example, in Fig. 5.15, Processes P2, P3, and P4 are child processes of Process P1, and all these processes work under the control of the parent P1. This tree structure has one parent root process and other processes at the other levels as child processes. In UNIX, there is one parent process called init(), and other system processes are created from this process.

In contrast, Windows has no concept of a process hierarchy. All processes are equal. The only hint of a process hierarchy is that when a process is created, the parent is given a special token (called a handle) that it can use to control the child. However, it is free to pass this token to some other process, thus invalidating the hierarchy. Processes in UNIX cannot disinherit their child.

2.5 Process States

Process States

Process is a program in execution. A process does not mean only program but it could contain some part called as text section. It may contain the current activity represented by the value of the program counter & the contents of CPU register.

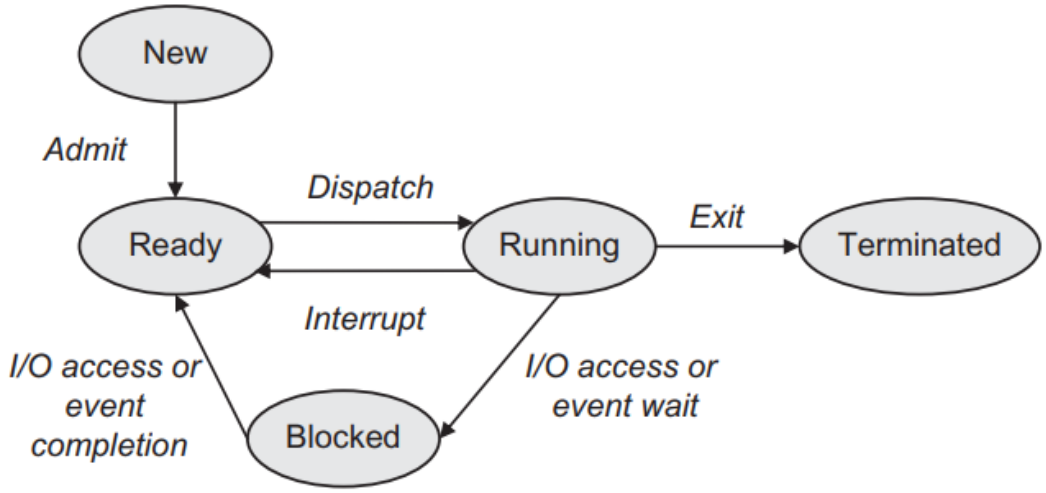


Fig. 5.4 Process state diagram

During the lifespan of a process, its execution status may be in one of four states: (associated with each state is usually a queue on which the process resides)

New :

The process being created is available in the new state. It is the new state because the system is not permitted it to enter the ready state due to limited memory available in the ready queue. If some memory becomes available, then the process from the new state will go to ready state.

Ready State:

The process which is not waiting for any external event such as I/O operation and which is not running is said to be in ready state. It is not in the running state because some other process is already running. It is waiting for its turn to go to the running state. The processes in ready state are available in main memory and ready for execution.

Running State:

The process which is currently running and has control of the CPU is known as the process in running state. In single user system, there is only one process which is in the running state. In multiuser system, there are multiple processes which are in the running state.

Blocked State (waiting):

The process is currently waiting for an external event such as an I/O operation is said to be in blocked state. After the completion of I/O operation, the process from blocked state enters in the ready state and from the ready state when the process turn will come it will again go to running state.

Terminated / Halted State:

The process whose operation is completed; it will go to the terminated state from the running state. In halted state, the memory occupied by the process is released.

2.6 Process Implementation

To implement the process model, the operating system maintains a table (an array of structures), called the process table, with one entry per process. (Some authors call these entries process control blocks.)

This entry contains important information about the process' state, including its program counter, stack pointer, memory allocation, the status of its open files, its accounting and scheduling information, and everything else about the process that must be saved when the process is switched from running to ready or blocked state so that it can be restarted later as if it had never been stopped.

Figure 2-4 shows some of the key fields in a typical system. The fields in the first column relate to process management. The other two relate to memory management and file management, respectively. It should be noted that precisely which fields the process table has been highly system dependent, but this figure gives a general idea of the kinds of information needed.

Process management	Memory management	File management
Registers	Pointer to text segment info	Root directory
Program counter	Pointer to data segment info	Working directory
Program status word	Pointer to stack segment info	File descriptors
Stack pointer		User ID
Process state		Group ID
Priority		
Scheduling parameters		
Process ID		
Parent process		
Process group		
Signals		
Time when process started		
CPU time used		
Children's CPU time		
Time of next alarm		

Figure 2-4. Some of the fields of a typical process table entry.

Now that we have looked at the process table, it is possible to explain a little more about how the illusion of multiple sequential processes is maintained on one (or each) CPU. Associated with each I/O class is a location (typically at a fixed location near the bottom of memory) called the interrupt vector. It contains the address of the interrupt service procedure. Suppose that user process 3 is running when a disk interrupt happens. User process 3's program counter, program status word, and sometimes one or more registers are pushed onto the (current) stack by the

interrupt hardware. The computer then jumps to the address specified in the interrupt vector. That is all the hardware does. From here on, it is up to the software, in particular, the interrupt service procedure.

All interrupts start by saving the registers, often in the process table entry for the current process. Then the information pushed onto the stack by the interrupt is removed and the stack pointer is set to point to a temporary stack used by the process handler. Actions such as saving the registers and setting the stack pointer cannot even be expressed in high-level languages such as C, so they are performed by a small assembly language routine, usually the same one for all interrupts since the work of saving the registers is identical, no matter what the cause of the interrupt is. When this routine is finished, it calls a C procedure to do the rest of the work for this specific interrupt type. (We assume the operating system is written in C, the usual choice for all real operating systems.) When it has done its job, possibly making some process now ready, the scheduler is called to see who to run next. After that, control is passed back to the assembly language code to load up the registers and memory map for the now-current process and start it running. Interrupt handling and scheduling are summarized in Fig. 2-5. It is worth noting that the details vary somewhat from system to system.

1. Hardware stacks program counter, etc.
2. Hardware loads new program counter from interrupt vector.
3. Assembly language procedure saves registers.
4. Assembly language procedure sets up new stack.
5. C interrupt service runs {typically reads and buffers input).
6. Scheduler decides which process is to run next.
7. C procedure returns to the assembly code.
8. Assembly language procedure starts up new current process.

Figure 2-5. Skeleton of what the lowest level of the operating system does when an interrupt occurs.

When the process finishes, the operating system displays a prompt character and waits for a new command. When it receives the command, it loads a new program into memory, overwriting the first one.

2.7 Thread:

A thread, sometimes called a lightweight process, is a basic unit of CPU utilization. A traditional (or heavyweight) process has a single thread of control. If a process has multiple threads of control, it can do more than one task at a time. This is because there are situations in which it is desirable to have multiple threads of control in the same address space, running as though they were separate processes.

Threads: A process is divided into number of light weight process; each light weight process is said to be a Thread.

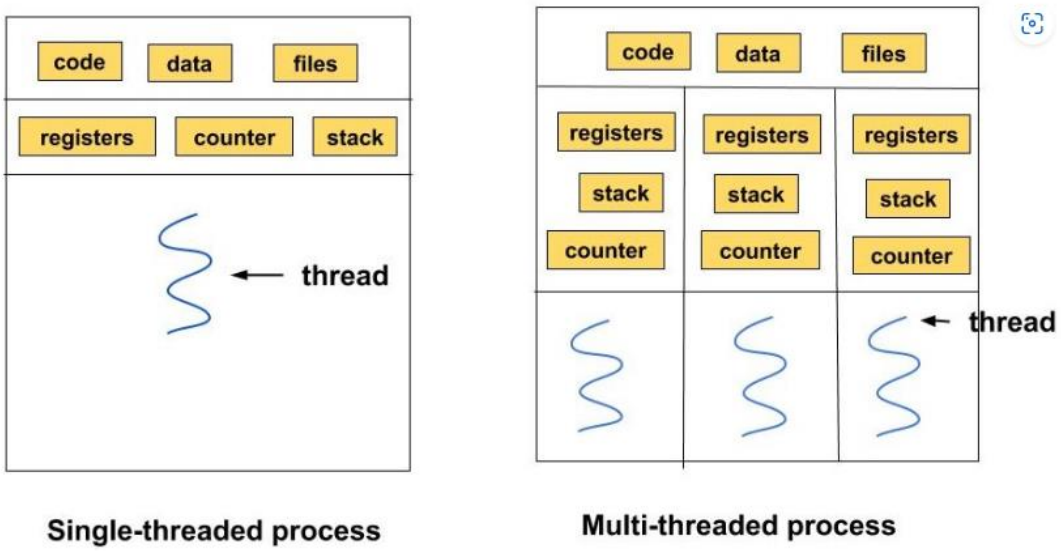
The Thread has a program counter (Keeps track of which instruction to execute next), registers (holds its current working variables), stack (execution History).

A thread is an execution unit that has its own program counter, a stack and a set of registers that reside in a process . Threads can't exist outside any process. Also, each thread belongs to exactly one process. The information like code segment, files, and data segment can be shared by the different threads.

Multithreading

A process is divided into number of smaller tasks each task is called a Thread. Number of Threads with in a Process execute at a time is called Multithreading.

The diagram below shows the single-threaded process and the multi-threaded process. A single-threaded process is a process with a single thread. A multi-threaded process is a process with multiple threads. As the diagram clearly shows that the multiple threads in it have its own registers, stack, and counter but they share the code and data segment.



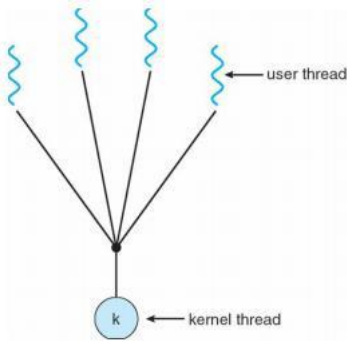
In a multi-threaded environment multiple processes and multiple threads can be considered as in case of multiuser O.S. such as UNIX. System provides support to both user and kernel threads, resulting in different types of multithreading models

1) Many to One model
2) One to One model
3) Many to Many model

1) Many to One model

Maps many user level threads to one kernel thread. Thread management is done by the thread library in user space hence it is efficient but entire process will block if a thread makes a blocking system call. As only one thread can access the kernel at a time, multiple threads are unable to run in parallel on multi processors.

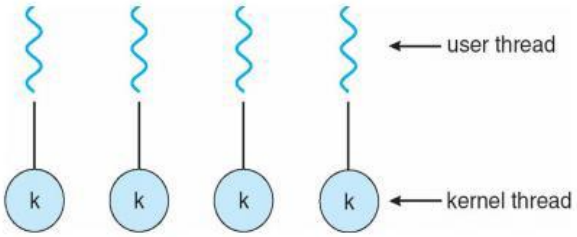
- Many user-level threads mapped to single kernel thread.
- Used on systems that do not support kernel threads



2) One to One model

Maps each user thread to kernel thread. It allows more concurrency by allowing another thread to run when a thread makes a blocking system call; it also allows multiple threads to run in parallel on multi processors. Disadvantage is that creating a user thread requires creating the corresponding kernel thread.

- Each user-level thread maps to kernel thread.
- Examples Windows 95/98/NT/2000 and OS/2

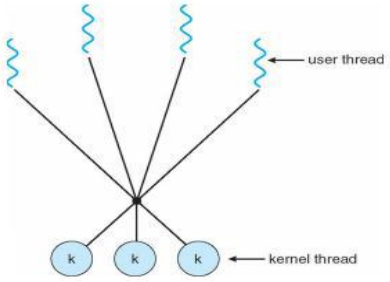


3) Many to Many model

Multiplexes many user level threads to a smaller or equal number of kernel level threads. The number of kernel threads may be specific to either a particular application or a particular machine. Developers can create as many user threads as necessary and the corresponding kernel threads can run in parallel on a multi-processor. Also, when a thread performs a blocking system call, the kernel can schedule another thread for execution.

- Allows many user level threads to be mapped to many kernel threads.
- Allows the operating system to create a sufficient number of kernel threads

  For eg Solaris 2 and Windows NT/2000 with the *ThreadFiber*package



2.8 Thread Usage

- The main reason for having threads is that in many applications, multiple activities are going on at once. Some of these may block from time to time. By decomposing such an application into multiple sequential threads that run in quasi-parallel, the programming model becomes simpler.

- We have seen this argument before. It is precisely the argument for having processes. Instead of thinking about interrupts, timers, and context switches, we can think about parallel processes. Only now with threads we add a new element: the ability for the parallel entities to share an address space and all of its data among themselves. This ability is essential for certain applications, which is why having multiple processes (with their separate address spaces) will not work.

- A second argument for having threads is that since they do not have any resources attached to them, they are easier to create and destroy than processes. In many systems, creating a thread goes 100 times faster than creating a process. When the number of threads needed changes dynamically and rapidly, this property is useful.

- A third reason for having threads is also a performance argument. Threads yield no performance gain when all of them are CPU bound, but when there is substantial computing and also substantial I/O, having threads allows these activities to overlap, thus speeding up the application.

- Finally, threads are useful on systems with multiple CPUs, where real parallelism is possible.

➢ It is probably easiest to see why threads are useful by giving some concrete examples. For example, consider a word processor. Most word processors display the document being created on the screen formatted exactly as it will appear on the printed page. In particular, all the line breaks and page breaks are in their correct and final position so the user can inspect them and change the document if need be (e.g., to eliminate widows and orphans—incomplete top and bottom lines on a page, which are considered esthetically unpleasing).

➢ Suppose that the user is writing a book. From the author's point of view, it is easiest to keep the entire book as a single file to make it easier to search for topics, perform global substitutions, and so on. Alternatively, each chapter might be a separate file. However, having every section and subsection as a separate file is a real nuisance when global

changes have to be made to the entire book since then hundreds of files have to be individually edited. For example, if proposed standard xxxx is approved just before the book goes to press, all occurrences of "Draft Standard xxxx" have to be changed to "Standard xxxx" at the last minute. If the entire book is one file, typically a single command can do all the substitutions. In contrast, if the book is spread over 300 files, each one must be edited separately.

- Now consider what happens when the user suddenly deletes one sentence from page 1 of an 800-page document. After checking the changed page to make sure it is correct, the user now wants to make another change on page 600 and types in a command telling the word processor to go to that page (possibly by searching for a phrase occurring only there). The word processor is now forced to reformat the entire book up to page 600 on the spot because it does not know what the first line of page 600 will be until it has processed all the previous pages. There may be a substantial delay before page 600 can be displayed, leading to an unhappy user.

- Threads can help here. Suppose that the word processor is written as a two-threaded program. One thread interacts with the user and the other handles reformatting in the background. As soon as the sentence is deleted from page 1, the interactive thread tells the reformatting thread to reformat the whole book. Meanwhile, the interactive thread continues to listen to the keyboard and mouse and responds to simple commands like scrolling page 1 while the other thread is computing madly in the background. With a little luck, the reformatting will be completed before the user asks to see page 600, so it can be displayed instantly.

While we are at it, why not add a third thread? Many word processors have a feature of automatically saving the entire file to disk every few minutes to protect the user against losing a day's work in the event of a program crash, system crash, or power failure. The third thread can handle the disk backups without interfering with the other two. The situation with three threads is shown in Fig. 2-4.

Figure 2-4 A word processor with three threads.

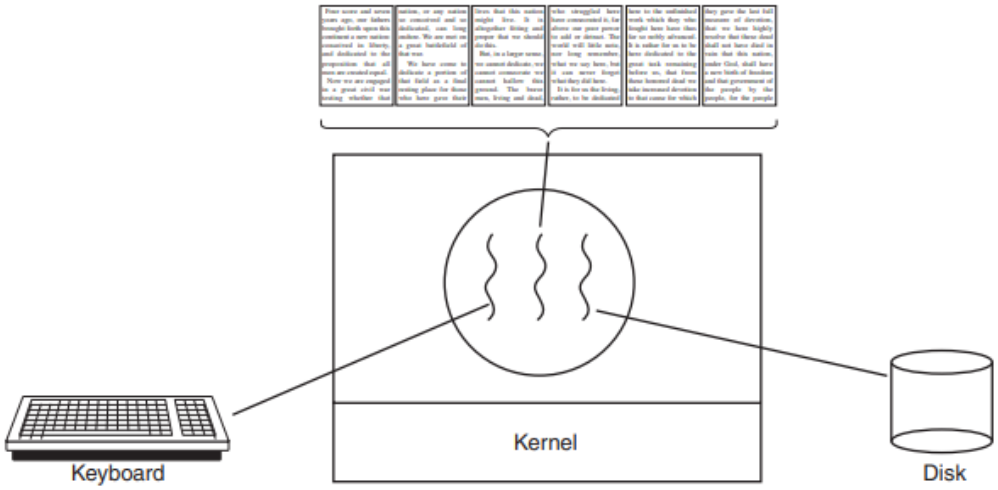


Figure 2-7. A word processor with three threads.

- If the program were single-threaded, then whenever a disk backup started, commands from the keyboard and mouse would be ignored until the backup was finished. The user would perceive this as sluggish performance. Alternatively, keyboard and mouse events could interrupt the disk backup, allowing good performance but leading to a complex interrupt-driven programming model.

- With three threads, the programming model is much simpler.

- The first thread just interacts with the user.

- The second thread reformats the document when told to.

- The third thread writes the contents of RAM to disk periodically.

- It should be clear that having three separate processes would not work here because all three threads need to operate on the document. By having three threads instead of three processes, they share a common memory and thus all have access to the document being edited.

2.9 Classical Thread Model

The process model is based on two independent concepts: resource grouping and execution.

One way of looking at a process is that it is a way to group related resources together.

A process contains a number of resources such as address space, open files, accounting information, etc. In addition to these resources, a process has a thread of control, e.g., program counter, register contents, stack. The idea of threads is to permit multiple threads of control to execute within one process. This is often called multithreading and threads are sometimes

called lightweight processes. Because threads in the same process share so much state, switching between them is much less expensive than switching between separate processes.

Individual threads within the same process are not completely independent. For example there is no memory protection between them. This is typically not a security problem as the threads are cooperating and all are from the same user (indeed the same process). However, the shared resources do make debugging harder. For example, one thread can easily overwrite data needed by another and if one thread closes a file other threads can't read from it.

In Fig. 2-11(a) we see three traditional processes. Each process has its own address space and a single thread of control. In contrast, in Fig. 2-11(b) we see a single process with three threads of control. Although in both cases we have three threads, in Fig. 2-11(a) each of them operates in a different address space, whereas in Fig. 2-11(b) all three of them share the same address space.

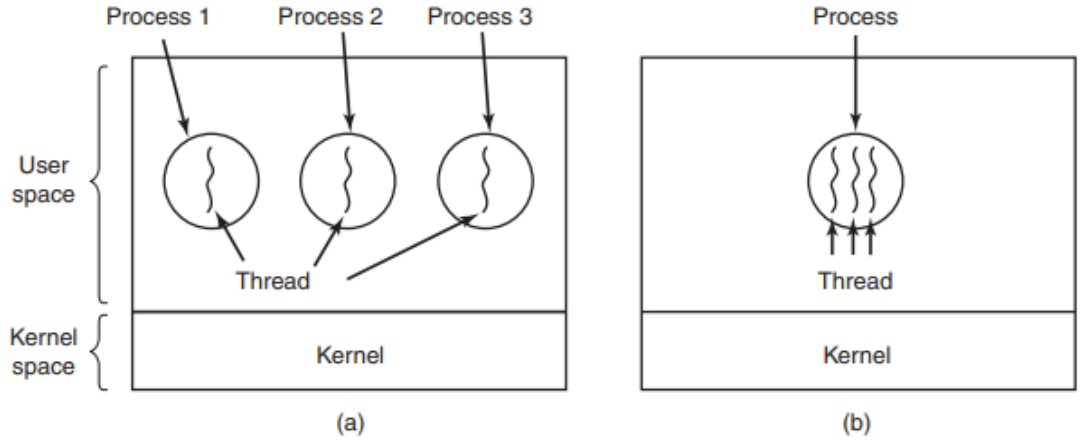


Figure 2-11. (a) Three processes each with one thread. (b) One process with three threads.

Different threads in a process are not as independent as different processes. All threads have exactly the same address space, which means that they also share the same global variables. Since every thread can access every memory address within the process' address space, one thread can read, write, or even wipe out another thread's stack. There is no protection between threads because (1) it is impossible, and (2) it should not be necessary. Unlike different processes, which may be from different users and which may be hostile to one another, a process is always owned by a single user, who has presumably created multiple threads so that they can cooperate, not fight. In addition to sharing an address space, all the threads can share the same set of open files, child processes, alarms, and signals, and so on, as shown in Fig. 2-12. Thus, the organization of Fig. 2-11(a) would be used when the three processes are essentially unrelated, whereas Fig. 2-11(b) would be appropriate when the three threads are actually part of the same job and are actively and closely cooperating with each other.

Per-process items	Per-thread items
Address space	Program counter
Global variables	Registers
Open files	Stack
Child processes	State
Pending alarms	
Signals and signal handlers	
Accounting information	

Figure 2-12. The first column lists some items shared by all threads in a process. The second one lists some items private to each thread.

The items in the first column are process properties, not thread properties.

When multithreading is present, processes usually start with a single thread present. This thread has the ability to create new threads by calling a library procedure such as thread create.

When a thread has finished its work, it can exit by calling a library procedure, say, thread exit.

In some thread systems, one thread can wait for a (specific) thread to exit by calling a procedure, for example, thread join.

Another common thread call is thread yield, which allows a thread to voluntarily give up the CPU to let another thread run. Such a call is important because there is no clock interrupt to actually enforce multiprogramming as there is with processes

Thread Library

A thread library provides the programmer with an Application program interface for creating and managing thread.

The main thread libraries which are used are given below −

- **POSIX threads** − Pthreads, the threads extension of the POSIX standard, may be provided as either a user level or a kernel level library.
- **WIN 32 thread** − The windows thread library is a kernel level library available on windows systems.
- **JAVA thread** − The JAVA thread API allows threads to be created and managed directly as JAVA programs.

2.10 POSIX Threads

- POSIX Threads is an API defined by the standard POSIX.1c, Threads extensions (IEEE Std 1003.1c-1995).
- POSIX stand for Portable Operating System Interface. POSIX threads, also known as pthreads, is a parallel execution model. What does a parallel execution model mean?

Threads give us the capability of running different tasks parallelly at the same time. This feature of being parallely executable helps the process to get executed faster. A single process in general is divided into threads and each thread has its own functionality/work to achieve.

- pthread library and the thread functions
- pthread library is a standardized API and pthreads can be implemented in the C/C++ language using pthread library. The header used to implement pthreads is pthread.h. This library provides a set of functions, constants, and data types in order to create, manage and control threads.
- Let's understand some important functions that the pthread library provides

Thread call	Description
Pthread_create	Create a new thread
Pthread_exit	Terminate the calling thread
Pthread_join	Wait for a specific thread to exit
Pthread_yield	Release the CPU to let another thread run
Pthread_attr_init	Create and initialize a thread's attribute structure
Pthread_attr_destroy	Remove a thread's attribute structure

Figure 2-14. Some of the Pthreads function calls.

- All Pthreads threads have certain properties. Each one has an identifier, a set of registers (including the program counter), and a set of attributes, which are stored in a structure. The attributes include the stack size, scheduling parameters, and other items needed to use the thread. A new thread is created using the pthread _create call.
- When a thread has finished the work, it has been assigned, it can terminate by calling pthread_exit. This call stops the thread and releases its stack.
- Often a thread needs to wait for another thread to finish its work and exit before continuing. The thread that is waiting calls pthread_join to wait for a specific other thread to terminate.
- Sometimes it happens that a thread is not logically blocked, but feels that it has run long enough and wants to give another thread a chance to run. It can accomplish this goal by calling pthread_yield.
- The next two thread calls deal with attributes. Pthread _attr _init creates the attribute structure associated with a thread and initializes it to the default values.
- pthread _attr_destroy removes a thread's attribute structure, freeing up its memory. It does not affect threads using it; they continue to exist.

2.11 Implementing Threads in User Space

There are two main ways to implement a threads package: in user space and in the kernel.

The first method is to put the threads package entirely in user space. The kernel knows nothing about them. As far as the kernel is concerned, it is managing ordinary, single-threaded processes. The first, and most obvious, advantage is that a user-level threads package can be implemented on an operating system that does not support threads. All operating systems used

to fall into this category, and even now some still do. With this approach, threads are implemented by a library.

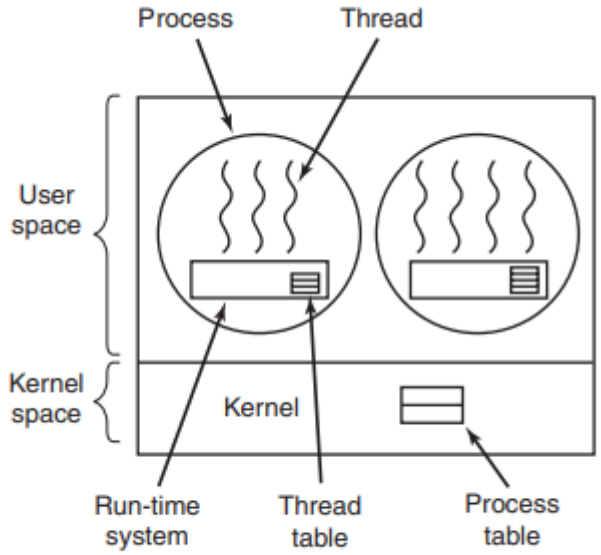


Figure 2-16. (a) A user-level threads package.

- All of these implementations have the same general structure, illustrated in Fig. 2-16(a). The threads run on top of a run-time system, which is a collection of procedures that manage threads. We have seen four of these already: pthread_create, pthread _exit, pthread _join, and pthread _yield, but usually there are more.
- When threads are managed in user space, each process needs its own private thread table to keep track of the threads in that process.
- This table is analogous to the kernel's process table, except that it keeps track only of the per-thread properties, such as each thread's program counter, stack pointer, registers, state, and so forth.
- The thread table is managed by the run-time system.
- When a thread is moved to ready state or blocked state, the information needed to restart it is stored in the thread table, exactly the same way as the kernel stores information about processes in the process table. When a thread does something that may cause it to become blocked locally, for example, waiting for another thread in its process to complete some work, it calls a run-time system procedure.

2.12 Implementing Thread in Kernel Space

- Now let us consider having the kernel know about and manage the threads. No run-time system is needed in each, as shown in Fig. 2-16(b).

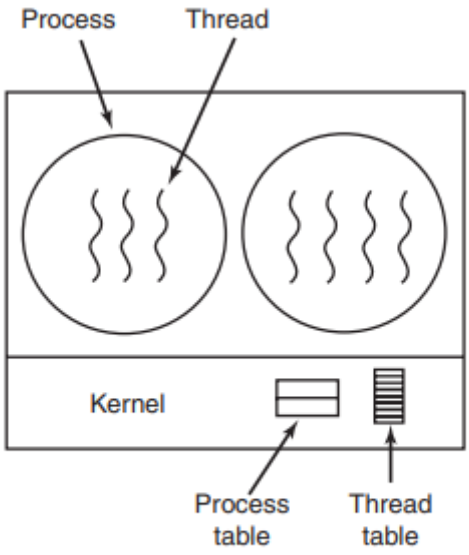


Figure 2-16. (b) A threads package managed

- Also, there is no thread table in each process. Instead, the kernel has a thread table that keeps track of all the threads in the system.
- When a thread wants to create a new thread or destroy an existing thread, it makes a kernel call, which then does the creation or destruction by updating the kernel thread table.
- The kernel's thread table holds each thread's registers, state, and other information.
- The information is the same as with user-level threads, but now kept in the kernel instead of in user space (inside the run-time system). This information is a subset of the information that traditional kernels maintain about their single threaded processes, that is, the process state. In addition, the kernel also maintains the traditional process table to keep track of processes.

2.13 Hybrid Implementations

- Various ways have been investigated to try to combine the advantages of user level threads with kernel-level threads. One way is use kernel-level threads and then multiplex user-level threads onto some or all of them, as shown in Fig. 2-17
- When this approach is used, the programmer can determine how many kernel threads to use and how many user-level threads to multiplex on each one. This model gives the ultimate in flexibility.

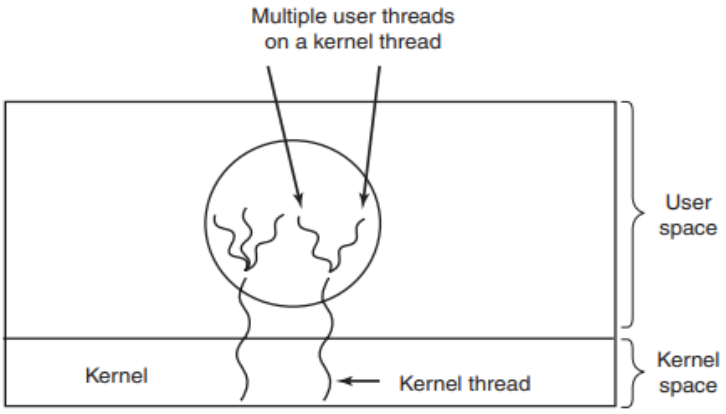


Figure 2-17. Multiplexing user-level threads onto kernel-level threads.

- With this approach, the kernel is aware of only the kernel-level threads and schedules those. Some of those threads may have multiple user-level threads multiplexed on top of them.
- These user-level threads are created, destroyed, and scheduled just like user-level threads in a process that runs on an operating system without multithreading capability. In this model, each kernel-level thread has some set of user-level threads that take turns using it.

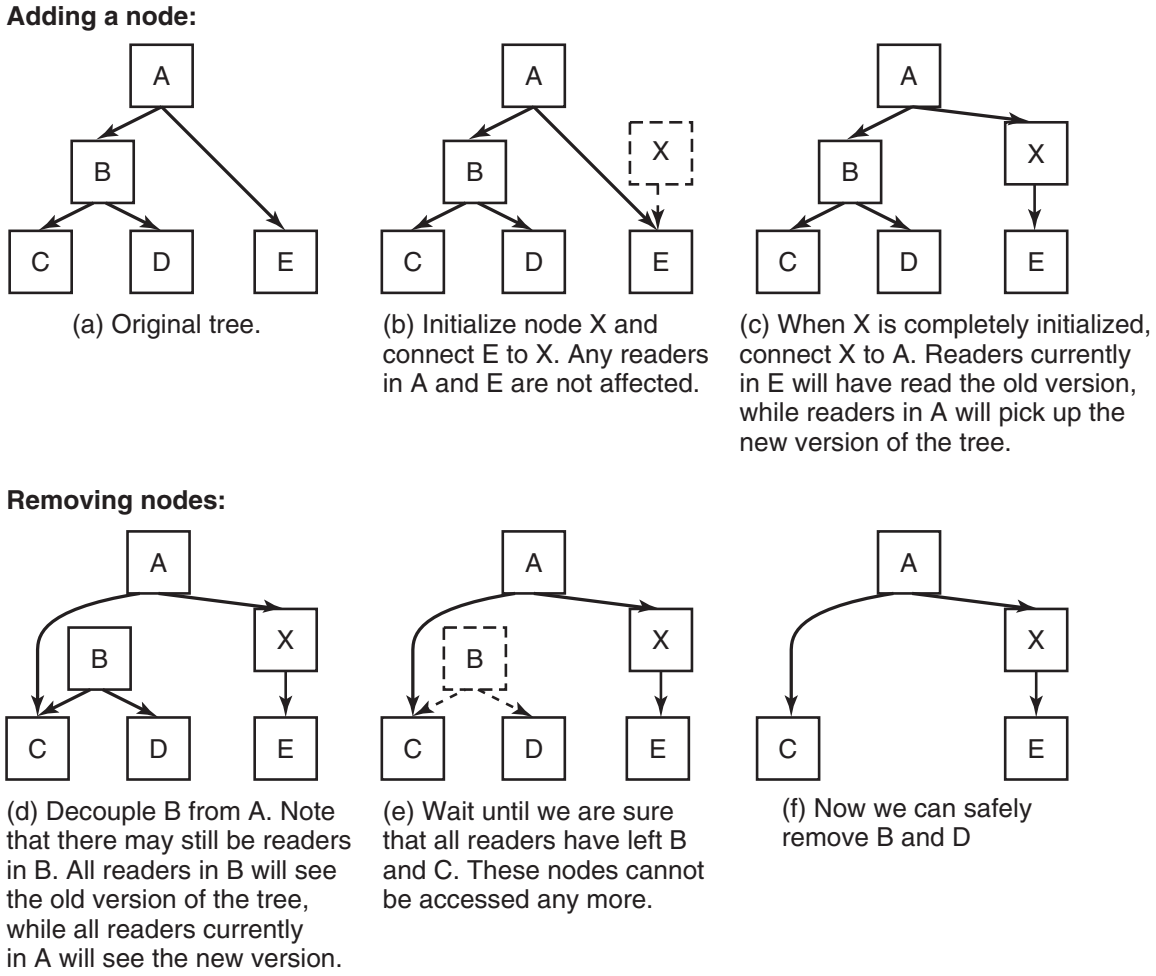


Figure 2-38. Read-Copy-Update: inserting a node in the tree and then removing a branch—all without locks.

2.4 SCHEDULING

When a computer is multiprogrammed, it frequently has multiple processes or threads competing for the CPU at the same time. This situation occurs whenever two or more of them are simultaneously in the ready state. If only one CPU is available, a choice has to be made which process to run next. The part of the operating system that makes the choice is called the **scheduler**, and the algorithm it uses is called the **scheduling algorithm**. These topics form the subject matter of the following sections.

Many of the same issues that apply to process scheduling also apply to thread scheduling, although some are different. When the kernel manages threads, scheduling is usually done per thread, with little or no regard to which process the thread belongs. Initially we will focus on scheduling issues that apply to both processes and threads. Later on we will explicitly look at thread scheduling and some of the unique issues it raises. We will deal with multicore chips in Chap. 8.

2.4.1 Introduction to Scheduling

Back in the old days of batch systems with input in the form of card images on a magnetic tape, the scheduling algorithm was simple: just run the next job on the tape. With multiprogramming systems, the scheduling algorithm became more complex because there were generally multiple users waiting for service. Some mainframes still combine batch and timesharing service, requiring the scheduler to decide whether a batch job or an interactive user at a terminal should go next. (As an aside, a batch job may be a request to run multiple programs in succession, but for this section, we will just assume it is a request to run a single program.) Because CPU time is a scarce resource on these machines, a good scheduler can make a big difference in perceived performance and user satisfaction. Consequently, a great deal of work has gone into devising clever and efficient scheduling algorithms.

With the advent of personal computers, the situation changed in two ways. First, most of the time there is only one active process. A user entering a document on a word processor is unlikely to be simultaneously compiling a program in the background. When the user types a command to the word processor, the scheduler does not have to do much work to figure out which process to run—the word processor is the only candidate.

Second, computers have gotten so much faster over the years that the CPU is rarely a scarce resource any more. Most programs for personal computers are limited by the rate at which the user can present input (by typing or clicking), not by the rate the CPU can process it. Even compilations, a major sink of CPU cycles in the past, take just a few seconds in most cases nowadays. Even when two programs are actually running at once, such as a word processor and a spreadsheet, it hardly matters which goes first since the user is probably waiting for both of them to finish. As a consequence, scheduling does not matter much on simple PCs. Of course, there are applications that practically eat the CPU alive. For instance rendering one hour of high-resolution video while tweaking the colors in each of the 107,892 frames (in NTSC) or 90,000 frames (in PAL) requires industrial-strength computing power. However, similar applications are the exception rather than the rule.

When we turn to networked servers, the situation changes appreciably. Here multiple processes often do compete for the CPU, so scheduling matters again. For example, when the CPU has to choose between running a process that gathers the daily statistics and one that serves user requests, the users will be a lot happier if the latter gets first crack at the CPU.

The "abundance of resources" argument also does not hold on many mobile devices, such as smartphones (except perhaps the most powerful models) and nodes in sensor networks. Here, the CPU may still be weak and the memory small. Moreover, since battery lifetime is one of the most important constraints on these devices, some schedulers try to optimize the power consumption.

In addition to picking the right process to run, the scheduler also has to worry about making efficient use of the CPU because process switching is expensive. To start with, a switch from user mode to kernel mode must occur. Then the state of the current process must be saved, including storing its registers in the process table so they can be reloaded later. In some systems, the memory map (e.g., memory reference bits in the page table) must be saved as well. Next a new process must be selected by running the scheduling algorithm. After that, the memory management unit (MMU) must be reloaded with the memory map of the new process. Finally, the new process must be started. In addition to all that, the process switch may invalidate the memory cache and related tables, forcing it to be dynamically reloaded from the main memory twice (upon entering the kernel and upon leaving it). All in all, doing too many process switches per second can chew up a substantial amount of CPU time, so caution is advised.

Process Behavior

Nearly all processes alternate bursts of computing with (disk or network) I/O requests, as shown in Fig. 2-39. Often, the CPU runs for a while without stopping, then a system call is made to read from a file or write to a file. When the system call completes, the CPU computes again until it needs more data or has to write more data, and so on. Note that some I/O activities count as computing. For example, when the CPU copies bits to a video RAM to update the screen, it is computing, not doing I/O, because the CPU is in use. I/O in this sense is when a process enters the blocked state waiting for an external device to complete its work.

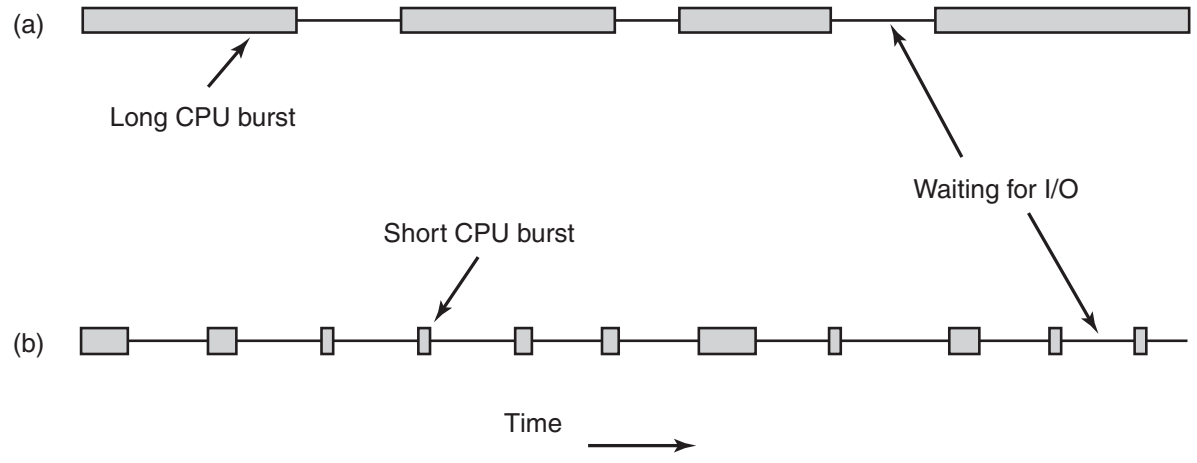


Figure 2-39. Bursts of CPU usage alternate with periods of waiting for I/O. (a) A CPU-bound process. (b) An I/O-bound process.

The important thing to notice about Fig. 2-39 is that some processes, such as the one in Fig. 2-39(a), spend most of their time computing, while other processes, such as the one shown in Fig. 2-39(b), spend most of their time waiting for I/O.

The former are called **compute-bound** or **CPU-bound**; the latter are called **I/O-bound**. Compute-bound processes typically have long CPU bursts and thus infrequent I/O waits, whereas I/O-bound processes have short CPU bursts and thus frequent I/O waits. Note that the key factor is the length of the CPU burst, not the length of the I/O burst. I/O-bound processes are I/O bound because they do not compute much between I/O requests, not because they have especially long I/O requests. It takes the same time to issue the hardware request to read a disk block no matter how much or how little time it takes to process the data after they arrive.

It is worth noting that as CPUs get faster, processes tend to get more I/O-bound. This effect occurs because CPUs are improving much faster than disks. As a consequence, the scheduling of I/O-bound processes is likely to become a more important subject in the future. The basic idea here is that if an I/O-bound process wants to run, it should get a chance quickly so that it can issue its disk request and keep the disk busy. As we saw in Fig. 2-6, when processes are I/O bound, it takes quite a few of them to keep the CPU fully occupied.

When to Schedule

A key issue related to scheduling is when to make scheduling decisions. It turns out that there are a variety of situations in which scheduling is needed. First, when a new process is created, a decision needs to be made whether to run the parent process or the child process. Since both processes are in ready state, it is a normal scheduling decision and can go either way, that is, the scheduler can legitimately choose to run either the parent or the child next.

Second, a scheduling decision must be made when a process exits. That process can no longer run (since it no longer exists), so some other process must be chosen from the set of ready processes. If no process is ready, a system-supplied idle process is normally run.

Third, when a process blocks on I/O, on a semaphore, or for some other reason, another process has to be selected to run. Sometimes the reason for blocking may play a role in the choice. For example, if *A* is an important process and it is waiting for *B* to exit its critical region, letting *B* run next will allow it to exit its critical region and thus let *A* continue. The trouble, however, is that the scheduler generally does not have the necessary information to take this dependency into account.

Fourth, when an I/O interrupt occurs, a scheduling decision may be made. If the interrupt came from an I/O device that has now completed its work, some process that was blocked waiting for the I/O may now be ready to run. It is up to the scheduler to decide whether to run the newly ready process, the process that was running at the time of the interrupt, or some third process.

If a hardware clock provides periodic interrupts at 50 or 60 Hz or some other frequency, a scheduling decision can be made at each clock interrupt or at every *k*th clock interrupt. Scheduling algorithms can be divided into two categories with

respect to how they deal with clock interrupts. A **nonpreemptive** scheduling algorithm picks a process to run and then just lets it run until it blocks (either on I/O or waiting for another process) or voluntarily releases the CPU. Even if it runs for many hours, it will not be forcibly suspended. In effect, no scheduling decisions are made during clock interrupts. After clock-interrupt processing has been finished, the process that was running before the interrupt is resumed, unless a higher-priority process was waiting for a now-satisfied timeout.

In contrast, a **preemptive** scheduling algorithm picks a process and lets it run for a maximum of some fixed time. If it is still running at the end of the time interval, it is suspended and the scheduler picks another process to run (if one is available). Doing preemptive scheduling requires having a clock interrupt occur at the end of the time interval to give control of the CPU back to the scheduler. If no clock is available, nonpreemptive scheduling is the only option.

Categories of Scheduling Algorithms

Not surprisingly, in different environments different scheduling algorithms are needed. This situation arises because different application areas (and different kinds of operating systems) have different goals. In other words, what the scheduler should optimize for is not the same in all systems. Three environments worth distinguishing are

1. Batch.
2. Interactive.
3. Real time.

Batch systems are still in widespread use in the business world for doing payroll, inventory, accounts receivable, accounts payable, interest calculation (at banks), claims processing (at insurance companies), and other periodic tasks. In batch systems, there are no users impatiently waiting at their terminals for a quick response to a short request. Consequently, nonpreemptive algorithms, or preemptive algorithms with long time periods for each process, are often acceptable. This approach reduces process switches and thus improves performance. The batch algorithms are actually fairly general and often applicable to other situations as well, which makes them worth studying, even for people not involved in corporate mainframe computing.

In an environment with interactive users, preemption is essential to keep one process from hogging the CPU and denying service to the others. Even if no process intentionally ran forever, one process might shut out all the others indefinitely due to a program bug. Preemption is needed to prevent this behavior. Servers also fall into this category, since they normally serve multiple (remote) users, all of whom are in a big hurry. Computer users are always in a big hurry.

In systems with real-time constraints, preemption is, oddly enough, sometimes not needed because the processes know that they may not run for long periods of time and usually do their work and block quickly. The difference with interactive systems is that real-time systems run only programs that are intended to further the application at hand. Interactive systems are general purpose and may run arbitrary programs that are not cooperative and even possibly malicious.

Scheduling Algorithm Goals

In order to design a scheduling algorithm, it is necessary to have some idea of what a good algorithm should do. Some goals depend on the environment (batch, interactive, or real time), but some are desirable in all cases. Some goals are listed in Fig. 2-40. We will discuss these in turn below.

All systems
- Fairness - giving each process a fair share of the CPU
- Policy enforcement - seeing that stated policy is carried out
- Balance - keeping all parts of the system busy

Batch systems
- Throughput - maximize jobs per hour
- Turnaround time - minimize time between submission and termination
- CPU utilization - keep the CPU busy all the time

Interactive systems
- Response time - respond to requests quickly
- Proportionality - meet users' expectations

Real-time systems
- Meeting deadlines - avoid losing data
- Predictability - avoid quality degradation in multimedia systems

Figure 2-40. Some goals of the scheduling algorithm under different circumstances.

Under all circumstances, fairness is important. Comparable processes should get comparable service. Giving one process much more CPU time than an equivalent one is not fair. Of course, different categories of processes may be treated differently. Think of safety control and doing the payroll at a nuclear reactor's computer center.

Somewhat related to fairness is enforcing the system's policies. If the local policy is that safety control processes get to run whenever they want to, even if it means the payroll is 30 sec late, the scheduler has to make sure this policy is enforced.

Another general goal is keeping all parts of the system busy when possible. If the CPU and all the I/O devices can be kept running all the time, more work gets

done per second than if some of the components are idle. In a batch system, for example, the scheduler has control of which jobs are brought into memory to run. Having some CPU-bound processes and some I/O-bound processes in memory together is a better idea than first loading and running all the CPU-bound jobs and then, when they are finished, loading and running all the I/O-bound jobs. If the latter strategy is used, when the CPU-bound processes are running, they will fight for the CPU and the disk will be idle. Later, when the I/O-bound jobs come in, they will fight for the disk and the CPU will be idle. Better to keep the whole system running at once by a careful mix of processes.

The managers of large computer centers that run many batch jobs typically look at three metrics to see how well their systems are performing: throughput, turnaround time, and CPU utilization. **Throughput** is the number of jobs per hour that the system completes. All things considered, finishing 50 jobs per hour is better than finishing 40 jobs per hour. **Turnaround time** is the statistically average time from the moment that a batch job is submitted until the moment it is completed. It measures how long the average user has to wait for the output. Here the rule is: Small is Beautiful.

A scheduling algorithm that tries to maximize throughput may not necessarily minimize turnaround time. For example, given a mix of short jobs and long jobs, a scheduler that always ran short jobs and never ran long jobs might achieve an excellent throughput (many short jobs per hour) but at the expense of a terrible turnaround time for the long jobs. If short jobs kept arriving at a fairly steady rate, the long jobs might never run, making the mean turnaround time infinite while achieving a high throughput.

CPU utilization is often used as a metric on batch systems. Actually though, it is not a good metric. What really matters is how many jobs per hour come out of the system (throughput) and how long it takes to get a job back (turnaround time). Using CPU utilization as a metric is like rating cars based on how many times per hour the engine turns over. However, knowing when the CPU utilization is almost 100% is useful for knowing when it is time to get more computing power.

For interactive systems, different goals apply. The most important one is to minimize **response time**, that is, the time between issuing a command and getting the result. On a personal computer where a background process is running (for example, reading and storing email from the network), a user request to start a program or open a file should take precedence over the background work. Having all interactive requests go first will be perceived as good service.

A somewhat related issue is what might be called **proportionality**. Users have an inherent (but often incorrect) idea of how long things should take. When a request that the user perceives as complex takes a long time, users accept that, but when a request that is perceived as simple takes a long time, users get irritated. For example, if clicking on an icon that starts uploading a 500-MB video to a cloud server takes 60 sec, the user will probably accept that as a fact of life because he does not expect the upload to take 5 sec. He knows it will take time.

On the other hand, when a user clicks on the icon that breaks the connection to the cloud server after the video has been uploaded, he has different expectations. If it has not completed after 30 sec, the user will probably be swearing a blue streak, and after 60 sec he will be foaming at the mouth. This behavior is due to the common user perception that sending a lot of data is *supposed* to take a lot longer than just breaking the connection. In some cases (such as this one), the scheduler cannot do anything about the response time, but in other cases it can, especially when the delay is due to a poor choice of process order.

Real-time systems have different properties than interactive systems, and thus different scheduling goals. They are characterized by having deadlines that must or at least should be met. For example, if a computer is controlling a device that produces data at a regular rate, failure to run the data-collection process on time may result in lost data. Thus the foremost need in a real-time system is meeting all (or most) deadlines.

In some real-time systems, especially those involving multimedia, predictability is important. Missing an occasional deadline is not fatal, but if the audio process runs too erratically, the sound quality will deteriorate rapidly. Video is also an issue, but the ear is much more sensitive to jitter than the eye. To avoid this problem, process scheduling must be highly predictable and regular. We will study batch and interactive scheduling algorithms in this chapter. Real-time scheduling is not covered in the book but in the extra material on multimedia operating systems on the book's Website.

2.4.2 Scheduling in Batch Systems

It is now time to turn from general scheduling issues to specific scheduling algorithms. In this section we will look at algorithms used in batch systems. In the following ones we will examine interactive and real-time systems. It is worth pointing out that some algorithms are used in both batch and interactive systems. We will study these later.

First-Come, First-Served

Probably the simplest of all scheduling algorithms ever devised is nonpreemptive **first-come, first-served**. With this algorithm, processes are assigned the CPU in the order they request it. Basically, there is a single queue of ready processes. When the first job enters the system from the outside in the morning, it is started immediately and allowed to run as long as it wants to. It is not interrupted because it has run too long. As other jobs come in, they are put onto the end of the queue. When the running process blocks, the first process on the queue is run next. When a blocked process becomes ready, like a newly arrived job, it is put on the end of the queue, behind all waiting processes.

The great strength of this algorithm is that it is easy to understand and equally easy to program. It is also fair in the same sense that allocating scarce concert tickets or brand-new iPhones to people who are willing to stand on line starting at 2 A.M. is fair. With this algorithm, a single linked list keeps track of all ready processes. Picking a process to run just requires removing one from the front of the queue. Adding a new job or unblocked process just requires attaching it to the end of the queue. What could be simpler to understand and implement?

Unfortunately, first-come, first-served also has a powerful disadvantage. Suppose there is one compute-bound process that runs for 1 sec at a time and many I/O-bound processes that use little CPU time but each have to perform 1000 disk reads to complete. The compute-bound process runs for 1 sec, then it reads a disk block. All the I/O processes now run and start disk reads. When the compute-bound process gets its disk block, it runs for another 1 sec, followed by all the I/O-bound processes in quick succession.

The net result is that each I/O-bound process gets to read 1 block per second and will take 1000 sec to finish. With a scheduling algorithm that preempted the compute-bound process every 10 msec, the I/O-bound processes would finish in 10 sec instead of 1000 sec, and without slowing down the compute-bound process very much.

Shortest Job First

Now let us look at another nonpreemptive batch algorithm that assumes the run times are known in advance. In an insurance company, for example, people can predict quite accurately how long it will take to run a batch of 1000 claims, since similar work is done every day. When several equally important jobs are sitting in the input queue waiting to be started, the scheduler picks the **shortest job first**. Look at Fig. 2-41. Here we find four jobs *A*, *B*, *C*, and *D* with run times of 8, 4, 4, and 4 minutes, respectively. By running them in that order, the turnaround time for *A* is 8 minutes, for *B* is 12 minutes, for *C* is 16 minutes, and for *D* is 20 minutes for an average of 14 minutes.



Figure 2-41. An example of shortest-job-first scheduling. (a) Running four jobs in the original order. (b) Running them in shortest job first order.

Now let us consider running these four jobs using shortest job first, as shown in Fig. 2-41(b). The turnaround times are now 4, 8, 12, and 20 minutes for an average of 11 minutes. Shortest job first is provably optimal. Consider the case of four

jobs, with execution times of a, b, c, and d, respectively. The first job finishes at time a, the second at time $a + b$, and so on. The mean turnaround time is $(4a + 3b + 2c + d)/4$. It is clear that a contributes more to the average than the other times, so it should be the shortest job, with b next, then c, and finally d as the longest since it affects only its own turnaround time. The same argument applies equally well to any number of jobs.

It is worth pointing out that shortest job first is optimal only when all the jobs are available simultaneously. As a counterexample, consider five jobs, A through E, with run times of 2, 4, 1, 1, and 1, respectively. Their arrival times are 0, 0, 3, 3, and 3. Initially, only A or B can be chosen, since the other three jobs have not arrived yet. Using shortest job first, we will run the jobs in the order A, B, C, D, E, for an average wait of 4.6. However, running them in the order B, C, D, E, A has an average wait of 4.4.

Shortest Remaining Time Next

A preemptive version of shortest job first is **shortest remaining time next**. With this algorithm, the scheduler always chooses the process whose remaining run time is the shortest. Again here, the run time has to be known in advance. When a new job arrives, its total time is compared to the current process' remaining time. If the new job needs less time to finish than the current process, the current process is suspended and the new job started. This scheme allows new short jobs to get good service.

2.4.3 Scheduling in Interactive Systems

We will now look at some algorithms that can be used in interactive systems. These are common on personal computers, servers, and other kinds of systems as well.

Round-Robin Scheduling

One of the oldest, simplest, fairest, and most widely used algorithms is **round robin**. Each process is assigned a time interval, called its **quantum**, during which it is allowed to run. If the process is still running at the end of the quantum, the CPU is preempted and given to another process. If the process has blocked or finished before the quantum has elapsed, the CPU switching is done when the process blocks, of course. Round robin is easy to implement. All the scheduler needs to do is maintain a list of runnable processes, as shown in Fig. 2-42(a). When the process uses up its quantum, it is put on the end of the list, as shown in Fig. 2-42(b).

The only really interesting issue with round robin is the length of the quantum. Switching from one process to another requires a certain amount of time for doing all the administration—saving and loading registers and memory maps, updating

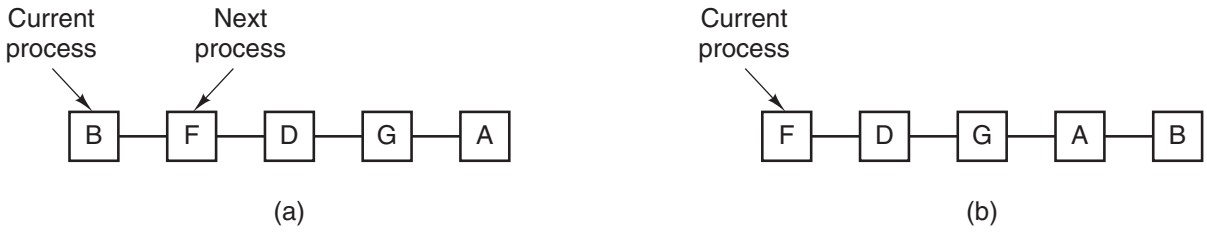


Figure 2-42. Round-robin scheduling. (a) The list of runnable processes. (b) The list of runnable processes after *B* uses up its quantum.

various tables and lists, flushing and reloading the memory cache, and so on. Suppose that this **process switch** or **context switch**, as it is sometimes called, takes 1 msec, including switching memory maps, flushing and reloading the cache, etc. Also suppose that the quantum is set at 4 msec. With these parameters, after doing 4 msec of useful work, the CPU will have to spend (i.e., waste) 1 msec on process switching. Thus 20% of the CPU time will be thrown away on administrative overhead. Clearly, this is too much.

To improve the CPU efficiency, we could set the quantum to, say, 100 msec. Now the wasted time is only 1%. But consider what happens on a server system if 50 requests come in within a very short time interval and with widely varying CPU requirements. Fifty processes will be put on the list of runnable processes. If the CPU is idle, the first one will start immediately, the second one may not start until 100 msec later, and so on. The unlucky last one may have to wait 5 sec before getting a chance, assuming all the others use their full quanta. Most users will perceive a 5-sec response to a short command as sluggish. This situation is especially bad if some of the requests near the end of the queue required only a few milliseconds of CPU time. With a short quantum they would have gotten better service.

Another factor is that if the quantum is set longer than the mean CPU burst, preemption will not happen very often. Instead, most processes will perform a blocking operation before the quantum runs out, causing a process switch. Eliminating preemption improves performance because process switches then happen only when they are logically necessary, that is, when a process blocks and cannot continue.

The conclusion can be formulated as follows: setting the quantum too short causes too many process switches and lowers the CPU efficiency, but setting it too long may cause poor response to short interactive requests. A quantum around 20–50 msec is often a reasonable compromise.

Priority Scheduling

Round-robin scheduling makes the implicit assumption that all processes are equally important. Frequently, the people who own and operate multiuser computers have quite different ideas on that subject. At a university, for example, the

pecking order may be the president first, the faculty deans next, then professors, secretaries, janitors, and finally students. The need to take external factors into account leads to **priority scheduling**. The basic idea is straightforward: each process is assigned a priority, and the runnable process with the highest priority is allowed to run.

Even on a PC with a single owner, there may be multiple processes, some of them more important than others. For example, a daemon process sending electronic mail in the background should be assigned a lower priority than a process displaying a video film on the screen in real time.

To prevent high-priority processes from running indefinitely, the scheduler may decrease the priority of the currently running process at each clock tick (i.e., at each clock interrupt). If this action causes its priority to drop below that of the next highest process, a process switch occurs. Alternatively, each process may be assigned a maximum time quantum that it is allowed to run. When this quantum is used up, the next-highest-priority process is given a chance to run.

Priorities can be assigned to processes statically or dynamically. On a military computer, processes started by generals might begin at priority 100, processes started by colonels at 90, majors at 80, captains at 70, lieutenants at 60, and so on down the totem pole. Alternatively, at a commercial computer center, high-priority jobs might cost $100 an hour, medium priority $75 an hour, and low priority $50 an hour. The UNIX system has a command, *nice*, which allows a user to voluntarily reduce the priority of his process, in order to be nice to the other users. Nobody ever uses it.

Priorities can also be assigned dynamically by the system to achieve certain system goals. For example, some processes are highly I/O bound and spend most of their time waiting for I/O to complete. Whenever such a process wants the CPU, it should be given the CPU immediately, to let it start its next I/O request, which can then proceed in parallel with another process actually computing. Making the I/O-bound process wait a long time for the CPU will just mean having it around occupying memory for an unnecessarily long time. A simple algorithm for giving good service to I/O-bound processes is to set the priority to $1/f$, where f is the fraction of the last quantum that a process used. A process that used only 1 msec of its 50-msec quantum would get priority 50, while a process that ran 25 msec before blocking would get priority 2, and a process that used the whole quantum would get priority 1.

It is often convenient to group processes into priority classes and use priority scheduling among the classes but round-robin scheduling within each class. Figure 2-43 shows a system with four priority classes. The scheduling algorithm is as follows: as long as there are runnable processes in priority class 4, just run each one for one quantum, round-robin fashion, and never bother with lower-priority classes. If priority class 4 is empty, then run the class 3 processes round robin. If classes 4 and 3 are both empty, then run class 2 round robin, and so on. If priorities are not adjusted occasionally, lower-priority classes may all starve to death.

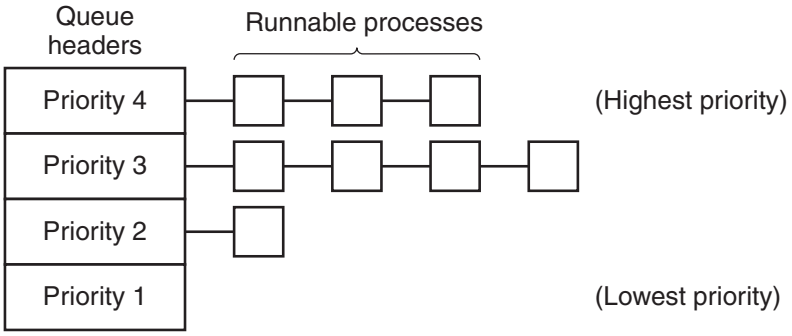


Figure 2-43. A scheduling algorithm with four priority classes.

Multiple Queues

One of the earliest priority schedulers was in CTSS, the M.I.T. Compatible TimeSharing System that ran on the IBM 7094 (Corbató et al., 1962). CTSS had the problem that process switching was slow because the 7094 could hold only one process in memory. Each switch meant swapping the current process to disk and reading in a new one from disk. The CTSS designers quickly realized that it was more efficient to give CPU-bound processes a large quantum once in a while, rather than giving them small quanta frequently (to reduce swapping). On the other hand, giving all processes a large quantum would mean poor response time, as we have already seen. Their solution was to set up priority classes. Processes in the highest class were run for one quantum. Processes in the next-highest class were run for two quanta. Processes in the next one were run for four quanta, etc. Whenever a process used up all the quanta allocated to it, it was moved down one class.

As an example, consider a process that needed to compute continuously for 100 quanta. It would initially be given one quantum, then swapped out. Next time it would get two quanta before being swapped out. On succeeding runs it would get 4, 8, 16, 32, and 64 quanta, although it would have used only 37 of the final 64 quanta to complete its work. Only 7 swaps would be needed (including the initial load) instead of 100 with a pure round-robin algorithm. Furthermore, as the process sank deeper and deeper into the priority queues, it would be run less and less frequently, saving the CPU for short, interactive processes.

The following policy was adopted to avoid punishing forever a process that needed to run for a long time when it first started but became interactive later. Whenever a carriage return (*Enter* key) was typed at a terminal, the process belonging to that terminal was moved to the highest-priority class, on the assumption that it was about to become interactive. One fine day, some user with a heavily CPU-bound process discovered that just sitting at the terminal and typing carriage returns at random every few seconds did wonders for his response time. He told all his friends. They told all their friends. Moral of the story: getting it right in practice is much harder than getting it right in principle.

Shortest Process Next

Because shortest job first always produces the minimum average response time for batch systems, it would be nice if it could be used for interactive processes as well. To a certain extent, it can be. Interactive processes generally follow the pattern of wait for command, execute command, wait for command, execute command, etc. If we regard the execution of each command as a separate "job," then we can minimize overall response time by running the shortest one first. The problem is figuring out which of the currently runnable processes is the shortest one.

One approach is to make estimates based on past behavior and run the process with the shortest estimated running time. Suppose that the estimated time per command for some process is T_0. Now suppose its next run is measured to be T_1. We could update our estimate by taking a weighted sum of these two numbers, that is, $aT_0 + (1 - a)T_1$. Through the choice of a we can decide to have the estimation process forget old runs quickly, or remember them for a long time. With $a = 1/2$, we get successive estimates of

$$T_0, \quad T_0/2 + T_1/2, \quad T_0/4 + T_1/4 + T_2/2, \quad T_0/8 + T_1/8 + T_2/4 + T_3/2$$

After three new runs, the weight of T_0 in the new estimate has dropped to 1/8.

The technique of estimating the next value in a series by taking the weighted average of the current measured value and the previous estimate is sometimes called **aging**. It is applicable to many situations where a prediction must be made based on previous values. Aging is especially easy to implement when $a = 1/2$. All that is needed is to add the new value to the current estimate and divide the sum by 2 (by shifting it right 1 bit).

Guaranteed Scheduling

A completely different approach to scheduling is to make real promises to the users about performance and then live up to those promises. One promise that is realistic to make and easy to live up to is this: If n users are logged in while you are working, you will receive about $1/n$ of the CPU power. Similarly, on a single-user system with n processes running, all things being equal, each one should get $1/n$ of the CPU cycles. That seems fair enough.

To make good on this promise, the system must keep track of how much CPU each process has had since its creation. It then computes the amount of CPU each one is entitled to, namely the time since creation divided by n. Since the amount of CPU time each process has actually had is also known, it is fairly straightforward to compute the ratio of actual CPU time consumed to CPU time entitled. A ratio of 0.5 means that a process has only had half of what it should have had, and a ratio of 2.0 means that a process has had twice as much as it was entitled to. The algorithm is then to run the process with the lowest ratio until its ratio has moved above that of its closest competitor. Then that one is chosen to run next.

Lottery Scheduling

While making promises to the users and then living up to them is a fine idea, it is difficult to implement. However, another algorithm can be used to give similarly predictable results with a much simpler implementation. It is called **lottery scheduling** (Waldspurger and Weihl, 1994).

The basic idea is to give processes lottery tickets for various system resources, such as CPU time. Whenever a scheduling decision has to be made, a lottery ticket is chosen at random, and the process holding that ticket gets the resource. When applied to CPU scheduling, the system might hold a lottery 50 times a second, with each winner getting 20 msec of CPU time as a prize.

To paraphrase George Orwell: "All processes are equal, but some processes are more equal." More important processes can be given extra tickets, to increase their odds of winning. If there are 100 tickets outstanding, and one process holds 20 of them, it will have a 20% chance of winning each lottery. In the long run, it will get about 20% of the CPU. In contrast to a priority scheduler, where it is very hard to state what having a priority of 40 actually means, here the rule is clear: a process holding a fraction *f* of the tickets will get about a fraction *f* of the resource in question.

Lottery scheduling has several interesting properties. For example, if a new process shows up and is granted some tickets, at the very next lottery it will have a chance of winning in proportion to the number of tickets it holds. In other words, lottery scheduling is highly responsive.

Cooperating processes may exchange tickets if they wish. For example, when a client process sends a message to a server process and then blocks, it may give all of its tickets to the server, to increase the chance of the server running next. When the server is finished, it returns the tickets so that the client can run again. In fact, in the absence of clients, servers need no tickets at all.

Lottery scheduling can be used to solve problems that are difficult to handle with other methods. One example is a video server in which several processes are feeding video streams to their clients, but at different frame rates. Suppose that the processes need frames at 10, 20, and 25 frames/sec. By allocating these processes 10, 20, and 25 tickets, respectively, they will automatically divide the CPU in approximately the correct proportion, that is, 10 : 20 : 25.

Fair-Share Scheduling

So far we have assumed that each process is scheduled on its own, without regard to who its owner is. As a result, if user 1 starts up nine processes and user 2 starts up one process, with round robin or equal priorities, user 1 will get 90% of the CPU and user 2 only 10% of it.

To prevent this situation, some systems take into account which user owns a process before scheduling it. In this model, each user is allocated some fraction of

the CPU and the scheduler picks processes in such a way as to enforce it. Thus if two users have each been promised 50% of the CPU, they will each get that, no matter how many processes they have in existence.

As an example, consider a system with two users, each of which has been promised 50% of the CPU. User 1 has four processes, *A*, *B*, *C*, and *D*, and user 2 has only one process, *E*. If round-robin scheduling is used, a possible scheduling sequence that meets all the constraints is this one:

A E B E C E D E A E B E C E D E ...

On the other hand, if user 1 is entitled to twice as much CPU time as user 2, we might get

A B E C D E A B E C D E ...

Numerous other possibilities exist, of course, and can be exploited, depending on what the notion of fairness is.

2.4.4 Scheduling in Real-Time Systems

A **real-time** system is one in which time plays an essential role. Typically, one or more physical devices external to the computer generate stimuli, and the computer must react appropriately to them within a fixed amount of time. For example, the computer in a compact disc player gets the bits as they come off the drive and must convert them into music within a very tight time interval. If the calculation takes too long, the music will sound peculiar. Other real-time systems are patient monitoring in a hospital intensive-care unit, the autopilot in an aircraft, and robot control in an automated factory. In all these cases, having the right answer but having it too late is often just as bad as not having it at all.

Real-time systems are generally categorized as **hard real time**, meaning there are absolute deadlines that must be met—or else!— and **soft real time**, meaning that missing an occasional deadline is undesirable, but nevertheless tolerable. In both cases, real-time behavior is achieved by dividing the program into a number of processes, each of whose behavior is predictable and known in advance. These processes are generally short lived and can run to completion in well under a second. When an external event is detected, it is the job of the scheduler to schedule the processes in such a way that all deadlines are met.

The events that a real-time system may have to respond to can be further categorized as **periodic** (meaning they occur at regular intervals) or **aperiodic** (meaning they occur unpredictably). A system may have to respond to multiple periodic-event streams. Depending on how much time each event requires for processing, handling all of them may not even be possible. For example, if there are m periodic events and event i occurs with period P_i and requires C_i sec of CPU time to handle each event, then the load can be handled only if

$$\sum_{i=1}^{m} \frac{C_i}{P_i} \leq 1$$

A real-time system that meets this criterion is said to be **schedulable**. This means it can actually be implemented. A process that fails to meet this test cannot be scheduled because the total amount of CPU time the processes want collectively is more than the CPU can deliver.

As an example, consider a soft real-time system with three periodic events, with periods of 100, 200, and 500 msec, respectively. If these events require 50, 30, and 100 msec of CPU time per event, respectively, the system is schedulable because $0.5 + 0.15 + 0.2 < 1$. If a fourth event with a period of 1 sec is added, the system will remain schedulable as long as this event does not need more than 150 msec of CPU time per event. Implicit in this calculation is the assumption that the context-switching overhead is so small that it can be ignored.

Real-time scheduling algorithms can be static or dynamic. The former make their scheduling decisions before the system starts running. The latter make their scheduling decisions at run time, after execution has started. Static scheduling works only when there is perfect information available in advance about the work to be done and the deadlines that have to be met. Dynamic scheduling algorithms do not have these restrictions.

2.4.5 Policy Versus Mechanism

Up until now, we have tacitly assumed that all the processes in the system belong to different users and are thus competing for the CPU. While this is often true, sometimes it happens that one process has many children running under its control. For example, a database-management-system process may have many children. Each child might be working on a different request, or each might have some specific function to perform (query parsing, disk access, etc.). It is entirely possible that the main process has an excellent idea of which of its children are the most important (or time critical) and which the least. Unfortunately, none of the schedulers discussed above accept any input from user processes about scheduling decisions. As a result, the scheduler rarely makes the best choice.

The solution to this problem is to separate the **scheduling mechanism** from the **scheduling policy**, a long-established principle (Levin et al., 1975). What this means is that the scheduling algorithm is parameterized in some way, but the parameters can be filled in by user processes. Let us consider the database example once again. Suppose that the kernel uses a priority-scheduling algorithm but provides a system call by which a process can set (and change) the priorities of its children. In this way, the parent can control how its children are scheduled, even though it itself does not do the scheduling. Here the mechanism is in the kernel but policy is set by a user process. Policy-mechanism separation is a key idea.

2.4.6 Thread Scheduling

When several processes each have multiple threads, we have two levels of parallelism present: processes and threads. Scheduling in such systems differs substantially depending on whether user-level threads or kernel-level threads (or both) are supported.

Let us consider user-level threads first. Since the kernel is not aware of the existence of threads, it operates as it always does, picking a process, say, *A*, and giving *A* control for its quantum. The thread scheduler inside *A* decides which thread to run, say *A1*. Since there are no clock interrupts to multiprogram threads, this thread may continue running as long as it wants to. If it uses up the process' entire quantum, the kernel will select another process to run.

When the process *A* finally runs again, thread *A1* will resume running. It will continue to consume all of *A*'s time until it is finished. However, its antisocial behavior will not affect other processes. They will get whatever the scheduler considers their appropriate share, no matter what is going on inside process *A*.

Now consider the case that *A*'s threads have relatively little work to do per CPU burst, for example, 5 msec of work within a 50-msec quantum. Consequently, each one runs for a little while, then yields the CPU back to the thread scheduler. This might lead to the sequence *A1*, *A2*, *A3*, *A1*, *A2*, *A3*, *A1*, *A2*, *A3*, *A1*, before the kernel switches to process *B*. This situation is illustrated in Fig. 2-44(a).

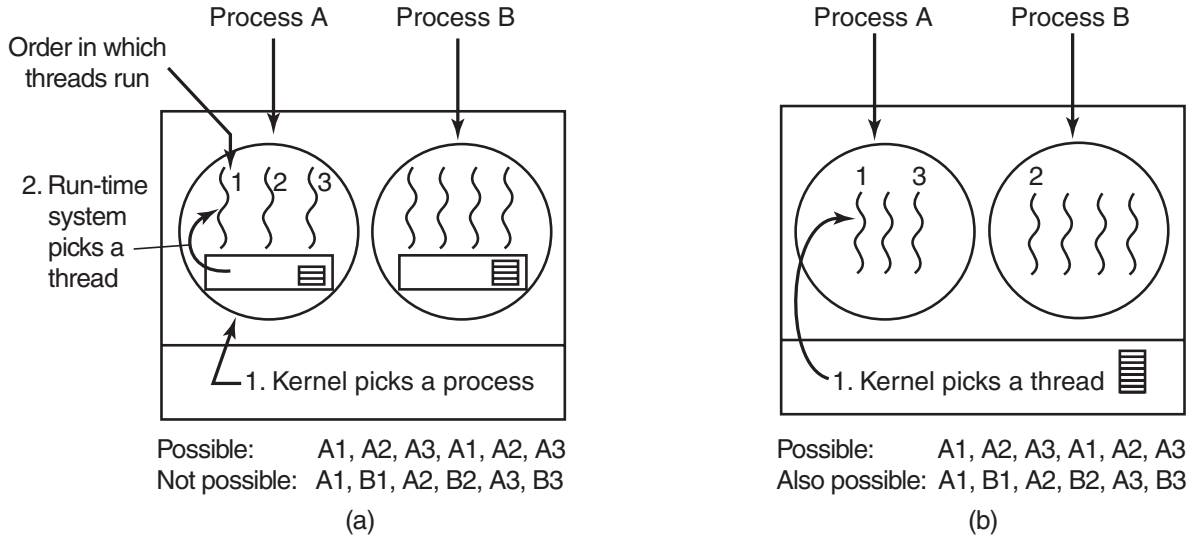


Figure 2-44. (a) Possible scheduling of user-level threads with a 50-msec process quantum and threads that run 5 msec per CPU burst. (b) Possible scheduling of kernel-level threads with the same characteristics as (a).

The scheduling algorithm used by the run-time system can be any of the ones described above. In practice, round-robin scheduling and priority scheduling are most common. The only constraint is the absence of a clock to interrupt a thread that has run too long. Since threads cooperate, this is usually not an issue.

Now consider the situation with kernel-level threads. Here the kernel picks a particular thread to run. It does not have to take into account which process the thread belongs to, but it can if it wants to. The thread is given a quantum and is forcibly suspended if it exceeds the quantum. With a 50-msec quantum but threads that block after 5 msec, the thread order for some period of 30 msec might be *A1*, *B1*, *A2*, *B2*, *A3*, *B3*, something not possible with these parameters and user-level threads. This situation is partially depicted in Fig. 2-44(b).

A major difference between user-level threads and kernel-level threads is the performance. Doing a thread switch with user-level threads takes a handful of machine instructions. With kernel-level threads it requires a full context switch, changing the memory map and invalidating the cache, which is several orders of magnitude slower. On the other hand, with kernel-level threads, having a thread block on I/O does not suspend the entire process as it does with user-level threads.

Since the kernel knows that switching from a thread in process *A* to a thread in process *B* is more expensive than running a second thread in process *A* (due to having to change the memory map and having the memory cache spoiled), it can take this information into account when making a decision. For example, given two threads that are otherwise equally important, with one of them belonging to the same process as a thread that just blocked and one belonging to a different process, preference could be given to the former.

Another important factor is that user-level threads can employ an application-specific thread scheduler. Consider, for example, the Web server of Fig. 2-8. Suppose that a worker thread has just blocked and the dispatcher thread and two worker threads are ready. Who should run next? The run-time system, knowing what all the threads do, can easily pick the dispatcher to run next, so that it can start another worker running. This strategy maximizes the amount of parallelism in an environment where workers frequently block on disk I/O. With kernel-level threads, the kernel would never know what each thread did (although they could be assigned different priorities). In general, however, application-specific thread schedulers can tune an application better than the kernel can.

2.5 CLASSICAL IPC PROBLEMS

The operating systems literature is full of interesting problems that have been widely discussed and analyzed using a variety of synchronization methods. In the following sections we will examine three of the better-known problems.

2.5.1 The Dining Philosophers Problem

In 1965, Dijkstra posed and then solved a synchronization problem he called the **dining philosophers problem**. Since that time, everyone inventing yet another synchronization primitive has felt obligated to demonstrate how wonderful the new

3

MEMORY MANAGEMENT

Main memory (RAM) is an important resource that must be very carefully managed. While the average home computer nowadays has 10,000 times more memory than the IBM 7094, the largest computer in the world in the early 1960s, programs are getting bigger faster than memories. To paraphrase Parkinson's Law, "Programs expand to fill the memory available to hold them." In this chapter we will study how operating systems create abstractions from memory and how they manage them.

What every programmer would like is a private, infinitely large, infinitely fast memory that is also nonvolatile, that is, does not lose its contents when the electric power is switched off. While we are at it, why not make it inexpensive, too? Unfortunately, technology does not provide such memories at present. Maybe you will discover how to do it.

What is the second choice? Over the years, people discovered the concept of a **memory hierarchy**, in which computers have a few megabytes of very fast, expensive, volatile cache memory, a few gigabytes of medium-speed, medium-priced, volatile main memory, and a few terabytes of slow, cheap, nonvolatile magnetic or solid-state disk storage, not to mention removable storage, such as DVDs and USB sticks. It is the job of the operating system to abstract this hierarchy into a useful model and then manage the abstraction.

The part of the operating system that manages (part of) the memory hierarchy is called the **memory manager**. Its job is to efficiently manage memory: keep track of which parts of memory are in use, allocate memory to processes when they need it, and deallocate it when they are done.

In this chapter we will investigate several different memory management models, ranging from very simple to highly sophisticated. Since managing the lowest level of cache memory is normally done by the hardware, the focus of this chapter will be on the programmer's model of main memory and how it can be managed. The abstractions for, and the management of, permanent storage—the disk—are the subject of the next chapter. We will first look at the simplest possible schemes and then gradually progress to more and more elaborate ones.

3.1 NO MEMORY ABSTRACTION

The simplest memory abstraction is to have no abstraction at all. Early mainframe computers (before 1960), early minicomputers (before 1970), and early personal computers (before 1980) had no memory abstraction. Every program simply saw the physical memory. When a program executed an instruction like

```
MOV REGISTER1,1000
```

the computer just moved the contents of physical memory location 1000 to *REGISTER1*. Thus, the model of memory presented to the programmer was simply physical memory, a set of addresses from 0 to some maximum, each address corresponding to a cell containing some number of bits, commonly eight.

Under these conditions, it was not possible to have two running programs in memory at the same time. If the first program wrote a new value to, say, location 2000, this would erase whatever value the second program was storing there. Nothing would work and both programs would crash almost immediately.

Even with the model of memory being just physical memory, several options are possible. Three variations are shown in Fig. 3-1. The operating system may be at the bottom of memory in RAM (Random Access Memory), as shown in Fig. 3-1(a), or it may be in ROM (Read-Only Memory) at the top of memory, as shown in Fig. 3-1(b), or the device drivers may be at the top of memory in a ROM and the rest of the system in RAM down below, as shown in Fig. 3-1(c). The first model was formerly used on mainframes and minicomputers but is rarely used any more. The second model is used on some handheld computers and embedded systems. The third model was used by early personal computers (e.g., running MS-DOS), where the portion of the system in the ROM is called the **BIOS** (Basic Input Output System). Models (a) and (c) have the disadvantage that a bug in the user program can wipe out the operating system, possibly with disastrous results.

When the system is organized in this way, generally only one process at a time can be running. As soon as the user types a command, the operating system copies the requested program from disk to memory and executes it. When the process finishes, the operating system displays a prompt character and waits for a user new command. When the operating system receives the command, it loads a new program into memory, overwriting the first one.

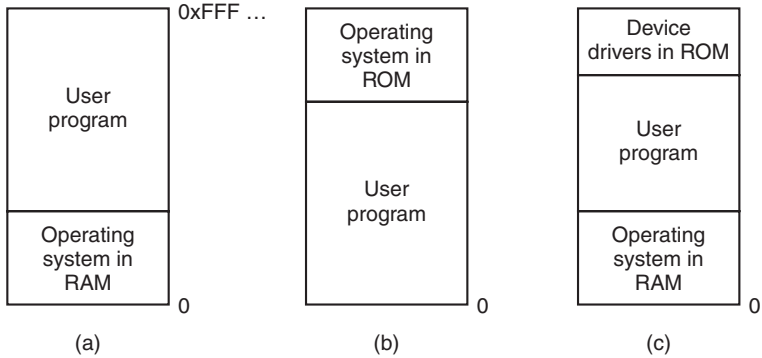


Figure 3-1. Three simple ways of organizing memory with an operating system and one user process. Other possibilities also exist.

One way to get some parallelism in a system with no memory abstraction is to program with multiple threads. Since all threads in a process are supposed to see the same memory image, the fact that they are forced to is not a problem. While this idea works, it is of limited use since what people often want is *unrelated* programs to be running at the same time, something the threads abstraction does not provide. Furthermore, any system that is so primitive as to provide no memory abstraction is unlikely to provide a threads abstraction.

Running Multiple Programs Without a Memory Abstraction

However, even with no memory abstraction, it is possible to run multiple programs at the same time. What the operating system has to do is save the entire contents of memory to a disk file, then bring in and run the next program. As long as there is only one program at a time in memory, there are no conflicts. This concept (swapping) will be discussed below.

With the addition of some special hardware, it is possible to run multiple programs concurrently, even without swapping. The early models of the IBM 360 solved the problem as follows. Memory was divided into 2-KB blocks and each was assigned a 4-bit protection key held in special registers inside the CPU. A machine with a 1-MB memory needed only 512 of these 4-bit registers for a total of 256 bytes of key storage. The PSW (Program Status Word) also contained a 4-bit key. The 360 hardware trapped any attempt by a running process to access memory with a protection code different from the PSW key. Since only the operating system could change the protection keys, user processes were prevented from interfering with one another and with the operating system itself.

Nevertheless, this solution had a major drawback, depicted in Fig. 3-2. Here we have two programs, each 16 KB in size, as shown in Fig. 3-2(a) and (b). The former is shaded to indicate that it has a different memory key than the latter. The

first program starts out by jumping to address 24, which contains a MOV instruction. The second program starts out by jumping to address 28, which contains a CMP instruction. The instructions that are not relevant to this discussion are not shown. When the two programs are loaded consecutively in memory starting at address 0, we have the situation of Fig. 3-2(c). For this example, we assume the operating system is in high memory and thus not shown.

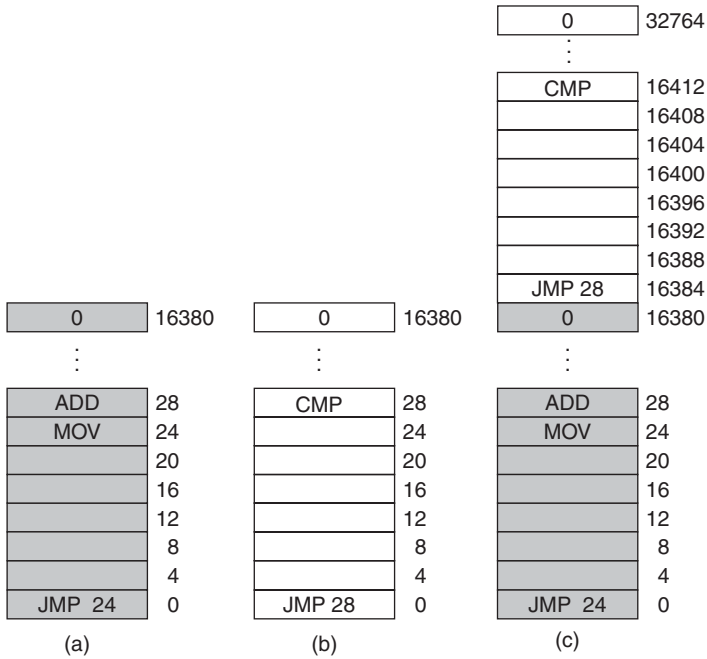


Figure 3-2. Illustration of the relocation problem. (a) A 16-KB program. (b) Another 16-KB program. (c) The two programs loaded consecutively into memory.

After the programs are loaded, they can be run. Since they have different memory keys, neither one can damage the other. But the problem is of a different nature. When the first program starts, it executes the JMP 24 instruction, which jumps to the instruction, as expected. This program functions normally.

However, after the first program has run long enough, the operating system may decide to run the second program, which has been loaded above the first one, at address 16,384. The first instruction executed is JMP 28, which jumps to the ADD instruction in the first program, instead of the CMP instruction it is supposed to jump to. The program will most likely crash in well under 1 sec.

The core problem here is that the two programs both reference absolute physical memory. That is not what we want at all. What we want is that each program

can reference a private set of addresses local to it. We will show how this can be acomplished shortly. What the IBM 360 did as a stop-gap solution was modify the second program on the fly as it loaded it into memory using a technique known as **static relocation**. It worked like this. When a program was loaded at address 16,384, the constant 16,384 was added to every program address during the load process (so "JMP 28" became "JMP 16,412", etc.).While this mechanism works if done right, it is not a very general solution and slows down loading. Furthermore, it requires extra information in all executable programs to indicate which words contain (relocatable) addresses and which do not. After all, the "28" in Fig. 3-2(b) has to be relocated but an instruction like

```
MOV REGISTER1,28
```

which moves the number 28 to *REGISTER1* must not be relocated. The loader needs some way to tell what is an address and what is a constant.

Finally, as we pointed out in Chap. 1, history tends to repeat itself in the computer world. While direct addressing of physical memory is but a distant memory on mainframes, minicomputers, desktop computers, notebooks, and smartphones, the lack of a memory abstraction is still common in embedded and smart card systems. Devices such as radios, washing machines, and microwave ovens are all full of software (in ROM) these days, and in most cases the software addresses absolute memory. This works because all the programs are known in advance and users are not free to run their own software on their toaster.

While high-end embedded systems (such as smartphones) have elaborate operating systems, simpler ones do not. In some cases, there is an operating system, but it is just a library that is linked with the application program and provides system calls for performing I/O and other common tasks. The **e-Cos** operating system is a common example of an operating system as library.

3.2 A MEMORY ABSTRACTION: ADDRESS SPACES

All in all, exposing physical memory to processes has several major drawbacks. First, if user programs can address every byte of memory, they can easily trash the operating system, intentionally or by accident, bringing the system to a grinding halt (unless there is special hardware like the IBM 360's lock-and-key scheme). This problem exists even if only one user program (application) is running. Second, with this model, it is difficult to have multiple programs running at once (taking turns, if there is only one CPU). On personal computers, it is common to have several programs open at once (a word processor, an email program, a Web browser), one of them having the current focus, but the others being reactivated at the click of a mouse. Since this situation is difficult to achieve when there is no abstraction from physical memory, something had to be done.

3.2.1 The Notion of an Address Space

Two problems have to be solved to allow multiple applications to be in memory at the same time without interfering with each other: protection and relocation. We looked at a primitive solution to the former used on the IBM 360: label chunks of memory with a protection key and compare the key of the executing process to that of every memory word fetched. However, this approach by itself does not solve the latter problem, although it can be solved by relocating programs as they are loaded, but this is a slow and complicated solution.

A better solution is to invent a new abstraction for memory: the address space. Just as the process concept creates a kind of abstract CPU to run programs, the address space creates a kind of abstract memory for programs to live in. An **address space** is the set of addresses that a process can use to address memory. Each process has its own address space, independent of those belonging to other processes (except in some special circumstances where processes want to share their address spaces).

The concept of an address space is very general and occurs in many contexts. Consider telephone numbers. In the United States and many other countries, a local telephone number is usually a 7-digit number. The address space for telephone numbers thus runs from 0,000,000 to 9,999,999, although some numbers, such as those beginning with 000 are not used. With the growth of smartphones, modems, and fax machines, this space is becoming too small, in which case more digits have to be used. The address space for I/O ports on the x86 runs from 0 to 16383. IPv4 addresses are 32-bit numbers, so their address space runs from 0 to $2^{32} - 1$ (again, with some reserved numbers).

Address spaces do not have to be numeric. The set of *.com* Internet domains is also an address space. This address space consists of all the strings of length 2 to 63 characters that can be made using letters, numbers, and hyphens, followed by *.com*. By now you should get the idea. It is fairly simple.

Somewhat harder is how to give each program its own address space, so address 28 in one program means a different physical location than address 28 in another program. Below we will discuss a simple way that used to be common but has fallen into disuse due to the ability to put much more complicated (and better) schemes on modern CPU chips.

Base and Limit Registers

This simple solution uses a particularly simple version of **dynamic relocation**. What it does is map each process' address space onto a different part of physical memory in a simple way. The classical solution, which was used on machines ranging from the CDC 6600 (the world's first supercomputer) to the Intel 8088 (the heart of the original IBM PC), is to equip each CPU with two special hardware registers, usually called the **base** and **limit** registers. When these registers are used,

programs are loaded into consecutive memory locations wherever there is room and without relocation during loading, as shown in Fig. 3-2(c). When a process is run, the base register is loaded with the physical address where its program begins in memory and the limit register is loaded with the length of the program. In Fig. 3-2(c), the base and limit values that would be loaded into these hardware registers when the first program is run are 0 and 16,384, respectively. The values used when the second program is run are 16,384 and 32,768, respectively. If a third 16-KB program were loaded directly above the second one and run, the base and limit registers would be 32,768 and 16,384.

Every time a process references memory, either to fetch an instruction or read or write a data word, the CPU hardware automatically adds the base value to the address generated by the process before sending the address out on the memory bus. Simultaneously, it checks whether the address offered is equal to or greater than the value in the limit register, in which case a fault is generated and the access is aborted. Thus, in the case of the first instruction of the second program in Fig. 3-2(c), the process executes a

```
JMP 28
```

instruction, but the hardware treats it as though it were

```
JMP 16412
```

so it lands on the CMP instruction as expected. The settings of the base and limit registers during the execution of the second program of Fig. 3-2(c) are shown in Fig. 3-3.

Using base and limit registers is an easy way to give each process its own private address space because every memory address generated automatically has the base-register contents added to it before being sent to memory. In many implementations, the base and limit registers are protected in such a way that only the operating system can modify them. This was the case on the CDC 6600, but not on the Intel 8088, which did not even have the limit register. It did have multiple base registers, allowing program text and data, for example, to be independently relocated, but offered no protection from out-of-range memory references.

A disadvantage of relocation using base and limit registers is the need to perform an addition and a comparison on every memory reference. Comparisons can be done fast, but additions are slow due to carry-propagation time unless special addition circuits are used.

3.2.2 Swapping

If the physical memory of the computer is large enough to hold all the processes, the schemes described so far will more or less do. But in practice, the total amount of RAM needed by all the processes is often much more than can fit in memory. On a typical Windows, OS X, or Linux system, something like 50–100

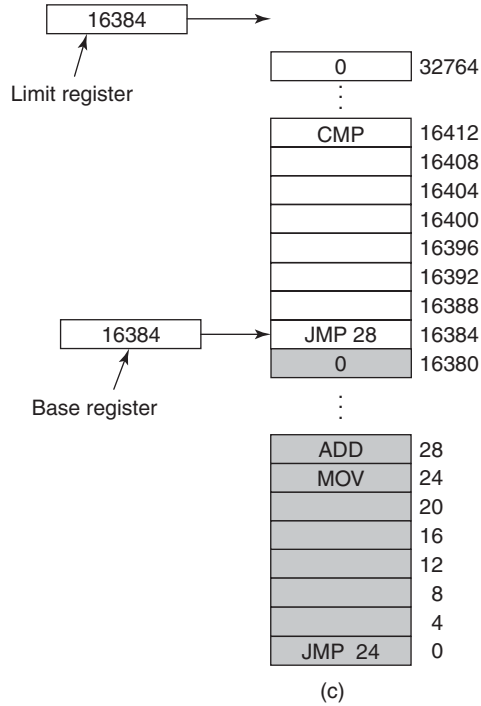


Figure 3-3. Base and limit registers can be used to give each process a separate address space.

processes or more may be started up as soon as the computer is booted. For example, when a Windows application is installed, it often issues commands so that on subsequent system boots, a process will be started that does nothing except check for updates to the application. Such a process can easily occupy 5–10 MB of memory. Other background processes check for incoming mail, incoming network connections, and many other things. And all this is before the first user program is started. Serious user application programs nowadays, like Photoshop, can easily require 500 MB just to boot and many gigabytes once they start processing data. Consequently, keeping all processes in memory all the time requires a huge amount of memory and cannot be done if there is insufficient memory.

Two general approaches to dealing with memory overload have been developed over the years. The simplest strategy, called **swapping**, consists of bringing in each process in its entirety, running it for a while, then putting it back on the disk. Idle processes are mostly stored on disk, so they do not take up any memory when they are not running (although some of them wake up periodically to do their work, then go to sleep again). The other strategy, called **virtual memory**, allows programs to run even when they are only partially in main memory. Below we will study swapping; in Sec. 3.3 we will examine virtual memory.

The operation of a swapping system is illustrated in Fig. 3-4. Initially, only process *A* is in memory. Then processes *B* and *C* are created or swapped in from disk. In Fig. 3-4(d) *A* is swapped out to disk. Then *D* comes in and *B* goes out. Finally *A* comes in again. Since *A* is now at a different location, addresses contained in it must be relocated, either by software when it is swapped in or (more likely) by hardware during program execution. For example, base and limit registers would work fine here.

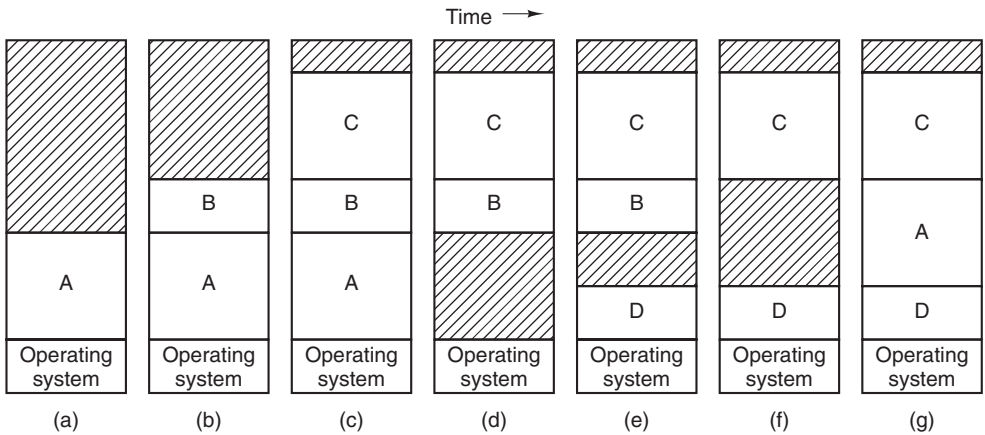


Figure 3-4. Memory allocation changes as processes come into memory and leave it. The shaded regions are unused memory.

When swapping creates multiple holes in memory, it is possible to combine them all into one big one by moving all the processes downward as far as possible. This technique is known as **memory compaction**. It is usually not done because it requires a lot of CPU time. For example, on a 16-GB machine that can copy 8 bytes in 8 nsec, it would take about 16 sec to compact all of memory.

A point that is worth making concerns how much memory should be allocated for a process when it is created or swapped in. If processes are created with a fixed size that never changes, then the allocation is simple: the operating system allocates exactly what is needed, no more and no less.

If, however, processes' data segments can grow, for example, by dynamically allocating memory from a heap, as in many programming languages, a problem occurs whenever a process tries to grow. If a hole is adjacent to the process, it can be allocated and the process allowed to grow into the hole. On the other hand, if the process is adjacent to another process, the growing process will either have to be moved to a hole in memory large enough for it, or one or more processes will have to be swapped out to create a large enough hole. If a process cannot grow in memory and the swap area on the disk is full, the process will have to suspended until some space is freed up (or it can be killed).

If it is expected that most processes will grow as they run, it is probably a good idea to allocate a little extra memory whenever a process is swapped in or moved, to reduce the overhead associated with moving or swapping processes that no longer fit in their allocated memory. However, when swapping processes to disk, only the memory actually in use should be swapped; it is wasteful to swap the extra memory as well. In Fig. 3-5(a) we see a memory configuration in which space for growth has been allocated to two processes.

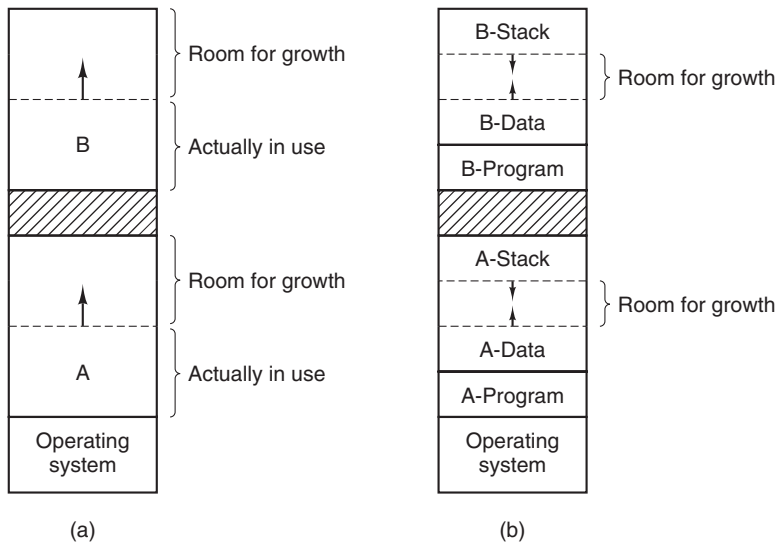


Figure 3-5. (a) Allocating space for a growing data segment. (b) Allocating space for a growing stack and a growing data segment.

If processes can have two growing segments—for example, the data segment being used as a heap for variables that are dynamically allocated and released and a stack segment for the normal local variables and return addresses—an alternative arrangement suggests itself, namely that of Fig. 3-5(b). In this figure we see that each process illustrated has a stack at the top of its allocated memory that is growing downward, and a data segment just beyond the program text that is growing upward. The memory between them can be used for either segment. If it runs out, the process will either have to be moved to a hole with sufficient space, swapped out of memory until a large enough hole can be created, or killed.

3.2.3 Managing Free Memory

When memory is assigned dynamically, the operating system must manage it. In general terms, there are two ways to keep track of memory usage: bitmaps and free lists. In this section and the next one we will look at these two methods. In

Chapter 10, we will look at some specific memory allocators used in Linux (like buddy and slab allocators) in more detail.

Memory Management with Bitmaps

With a bitmap, memory is divided into allocation units as small as a few words and as large as several kilobytes. Corresponding to each allocation unit is a bit in the bitmap, which is 0 if the unit is free and 1 if it is occupied (or vice versa). Figure 3-6 shows part of memory and the corresponding bitmap.

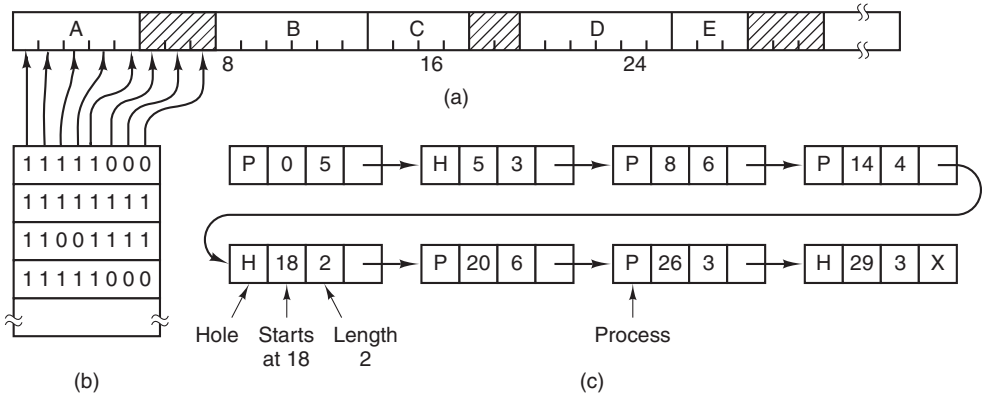


Figure 3-6. (a) A part of memory with five processes and three holes. The tick marks show the memory allocation units. The shaded regions (0 in the bitmap) are free. (b) The corresponding bitmap. (c) The same information as a list.

The size of the allocation unit is an important design issue. The smaller the allocation unit, the larger the bitmap. However, even with an allocation unit as small as 4 bytes, 32 bits of memory will require only 1 bit of the map. A memory of $32n$ bits will use n map bits, so the bitmap will take up only 1/32 of memory. If the allocation unit is chosen large, the bitmap will be smaller, but appreciable memory may be wasted in the last unit of the process if the process size is not an exact multiple of the allocation unit.

A bitmap provides a simple way to keep track of memory words in a fixed amount of memory because the size of the bitmap depends only on the size of memory and the size of the allocation unit. The main problem is that when it has been decided to bring a k-unit process into memory, the memory manager must search the bitmap to find a run of k consecutive 0 bits in the map. Searching a bitmap for a run of a given length is a slow operation (because the run may straddle word boundaries in the map); this is an argument against bitmaps.

Memory Management with Linked Lists

Another way of keeping track of memory is to maintain a linked list of allocated and free memory segments, where a segment either contains a process or is an empty hole between two processes. The memory of Fig. 3-6(a) is represented in Fig. 3-6(c) as a linked list of segments. Each entry in the list specifies a hole (H) or process (P), the address at which it starts, the length, and a pointer to the next item.

In this example, the segment list is kept sorted by address. Sorting this way has the advantage that when a process terminates or is swapped out, updating the list is straightforward. A terminating process normally has two neighbors (except when it is at the very top or bottom of memory). These may be either processes or holes, leading to the four combinations shown in Fig. 3-7. In Fig. 3-7(a) updating the list requires replacing a P by an H. In Fig. 3-7(b) and Fig. 3-7(c), two entries are coalesced into one, and the list becomes one entry shorter. In Fig. 3-7(d), three entries are merged and two items are removed from the list.

Since the process table slot for the terminating process will normally point to the list entry for the process itself, it may be more convenient to have the list as a double-linked list, rather than the single-linked list of Fig. 3-6(c). This structure makes it easier to find the previous entry and to see if a merge is possible.

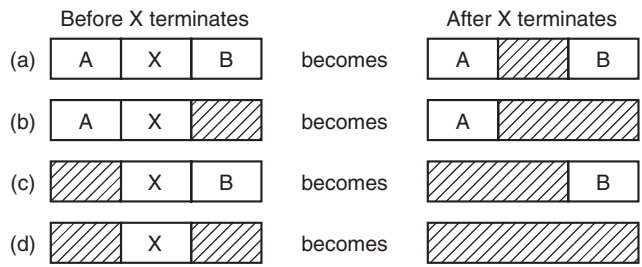


Figure 3-7. Four neighbor combinations for the terminating process, *X*.

When the processes and holes are kept on a list sorted by address, several algorithms can be used to allocate memory for a created process (or an existing process being swapped in from disk). We assume that the memory manager knows how much memory to allocate. The simplest algorithm is **first fit**. The memory manager scans along the list of segments until it finds a hole that is big enough. The hole is then broken up into two pieces, one for the process and one for the unused memory, except in the statistically unlikely case of an exact fit. First fit is a fast algorithm because it searches as little as possible.

A minor variation of first fit is **next fit**. It works the same way as first fit, except that it keeps track of where it is whenever it finds a suitable hole. The next time it is called to find a hole, it starts searching the list from the place where it left off last time, instead of always at the beginning, as first fit does. Simulations by Bays (1977) show that next fit gives slightly worse performance than first fit.

Another well-known and widely used algorithm is **best fit**. Best fit searches the entire list, from beginning to end, and takes the smallest hole that is adequate. Rather than breaking up a big hole that might be needed later, best fit tries to find a hole that is close to the actual size needed, to best match the request and the available holes.

As an example of first fit and best fit, consider Fig. 3-6 again. If a block of size 2 is needed, first fit will allocate the hole at 5, but best fit will allocate the hole at 18.

Best fit is slower than first fit because it must search the entire list every time it is called. Somewhat surprisingly, it also results in more wasted memory than first fit or next fit because it tends to fill up memory with tiny, useless holes. First fit generates larger holes on the average.

To get around the problem of breaking up nearly exact matches into a process and a tiny hole, one could think about **worst fit**, that is, always take the largest available hole, so that the new hole will be big enough to be useful. Simulation has shown that worst fit is not a very good idea either.

All four algorithms can be speeded up by maintaining separate lists for processes and holes. In this way, all of them devote their full energy to inspecting holes, not processes. The inevitable price that is paid for this speedup on allocation is the additional complexity and slowdown when deallocating memory, since a freed segment has to be removed from the process list and inserted into the hole list.

If distinct lists are maintained for processes and holes, the hole list may be kept sorted on size, to make best fit faster. When best fit searches a list of holes from smallest to largest, as soon as it finds a hole that fits, it knows that the hole is the smallest one that will do the job, hence the best fit. No further searching is needed, as it is with the single-list scheme. With a hole list sorted by size, first fit and best fit are equally fast, and next fit is pointless.

When the holes are kept on separate lists from the processes, a small optimization is possible. Instead of having a separate set of data structures for maintaining the hole list, as is done in Fig. 3-6(c), the information can be stored in the holes. The first word of each hole could be the hole size, and the second word a pointer to the following entry. The nodes of the list of Fig. 3-6(c), which require three words and one bit (P/H), are no longer needed.

Yet another allocation algorithm is **quick fit**, which maintains separate lists for some of the more common sizes requested. For example, it might have a table with *n* entries, in which the first entry is a pointer to the head of a list of 4-KB holes, the second entry is a pointer to a list of 8-KB holes, the third entry a pointer to 12-KB holes, and so on. Holes of, say, 21 KB, could be put either on the 20-KB list or on a special list of odd-sized holes.

With quick fit, finding a hole of the required size is extremely fast, but it has the same disadvantage as all schemes that sort by hole size, namely, when a process terminates or is swapped out, finding its neighbors to see if a merge with them

is possible is quite expensive. If merging is not done, memory will quickly fragment into a large number of small holes into which no processes fit.

3.3 VIRTUAL MEMORY

While base and limit registers can be used to create the abstraction of address spaces, there is another problem that has to be solved: managing bloatware. While memory sizes are increasing rapidly, software sizes are increasing much faster. In the 1980s, many universities ran a timesharing system with dozens of (more-or-less satisfied) users running simultaneously on a 4-MB VAX. Now Microsoft recommends having at least 2 GB for 64-bit Windows 8. The trend toward multimedia puts even more demands on memory.

As a consequence of these developments, there is a need to run programs that are too large to fit in memory, and there is certainly a need to have systems that can support multiple programs running simultaneously, each of which fits in memory but all of which collectively exceed memory. Swapping is not an attractive option, since a typical SATA disk has a peak transfer rate of several hundreds of MB/sec, which means it takes seconds to swap out a 1-GB program and the same to swap in a 1-GB program.

The problem of programs larger than memory has been around since the beginning of computing, albeit in limited areas, such as science and engineering (simulating the creation of the universe or even simulating a new aircraft takes a lot of memory). A solution adopted in the 1960s was to split programs into little pieces, called **overlays**. When a program started, all that was loaded into memory was the overlay manager, which immediately loaded and ran overlay 0. When it was done, it would tell the overlay manager to load overlay 1, either above overlay 0 in memory (if there was space for it) or on top of overlay 0 (if there was no space). Some overlay systems were highly complex, allowing many overlays in memory at once. The overlays were kept on the disk and swapped in and out of memory by the overlay manager.

Although the actual work of swapping overlays in and out was done by the operating system, the work of splitting the program into pieces had to be done manually by the programmer. Splitting large programs up into small, modular pieces was time consuming, boring, and error prone. Few programmers were good at this. It did not take long before someone thought of a way to turn the whole job over to the computer.

The method that was devised (Fotheringham, 1961) has come to be known as **virtual memory**. The basic idea behind virtual memory is that each program has its own address space, which is broken up into chunks called **pages**. Each page is a contiguous range of addresses. These pages are mapped onto physical memory, but not all pages have to be in physical memory at the same time to run the program. When the program references a part of its address space that is in physical

memory, the hardware performs the necessary mapping on the fly. When the program references a part of its address space that is *not* in physical memory, the operating system is alerted to go get the missing piece and re-execute the instruction that failed.

In a sense, virtual memory is a generalization of the base-and-limit-register idea. The 8088 had separate base registers (but no limit registers) for text and data. With virtual memory, instead of having separate relocation for just the text and data segments, the entire address space can be mapped onto physical memory in fairly small units. We will show how virtual memory is implemented below.

Virtual memory works just fine in a multiprogramming system, with bits and pieces of many programs in memory at once. While a program is waiting for pieces of itself to be read in, the CPU can be given to another process.

3.3.1 Paging

Most virtual memory systems use a technique called **paging**, which we will now describe. On any computer, programs reference a set of memory addresses. When a program executes an instruction like

```
MOV REG,1000
```

it does so to copy the contents of memory address 1000 to REG (or vice versa, depending on the computer). Addresses can be generated using indexing, base registers, segment registers, and other ways.

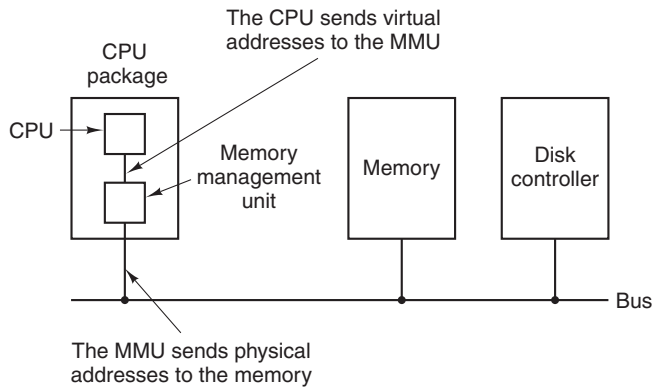


Figure 3-8. The position and function of the MMU. Here the MMU is shown as being a part of the CPU chip because it commonly is nowadays. However, logically it could be a separate chip and was years ago.

These program-generated addresses are called **virtual addresses** and form the **virtual address space**. On computers without virtual memory, the virtual address

is put directly onto the memory bus and causes the physical memory word with the same address to be read or written. When virtual memory is used, the virtual addresses do not go directly to the memory bus. Instead, they go to an **MMU** (**Memory Management Unit**) that maps the virtual addresses onto the physical memory addresses, as illustrated in Fig. 3-8.

A very simple example of how this mapping works is shown in Fig. 3-9. In this example, we have a computer that generates 16-bit addresses, from 0 up to 64K − 1. These are the virtual addresses. This computer, however, has only 32 KB of physical memory. So although 64-KB programs can be written, they cannot be loaded into memory in their entirety and run. A complete copy of a program's core image, up to 64 KB, must be present on the disk, however, so that pieces can be brought in as needed.

The virtual address space consists of fixed-size units called pages. The corresponding units in the physical memory are called **page frames**. The pages and page frames are generally the same size. In this example they are 4 KB, but page sizes from 512 bytes to a gigabyte have been used in real systems. With 64 KB of virtual address space and 32 KB of physical memory, we get 16 virtual pages and 8 page frames. Transfers between RAM and disk are always in whole pages. Many processors support multiple page sizes that can be mixed and matched as the operating system sees fit. For instance, the x86-64 architecture supports 4-KB, 2-MB, and 1-GB pages, so we could use 4-KB pages for user applications and a single 1-GB page for the kernel. We will see later why it is sometimes better to use a single large page, rather than a large number of small ones.

The notation in Fig. 3-9 is as follows. The range marked 0K–4K means that the virtual or physical addresses in that page are 0 to 4095. The range 4K–8K refers to addresses 4096 to 8191, and so on. Each page contains exactly 4096 addresses starting at a multiple of 4096 and ending one shy of a multiple of 4096.

When the program tries to access address 0, for example, using the instruction

```
MOV REG,0
```

virtual address 0 is sent to the MMU. The MMU sees that this virtual address falls in page 0 (0 to 4095), which according to its mapping is page frame 2 (8192 to 12287). It thus transforms the address to 8192 and outputs address 8192 onto the bus. The memory knows nothing at all about the MMU and just sees a request for reading or writing address 8192, which it honors. Thus, the MMU has effectively mapped all virtual addresses between 0 and 4095 onto physical addresses 8192 to 12287.

Similarly, the instruction

```
MOV REG,8192
```

is effectively transformed into

```
MOV REG,24576
```

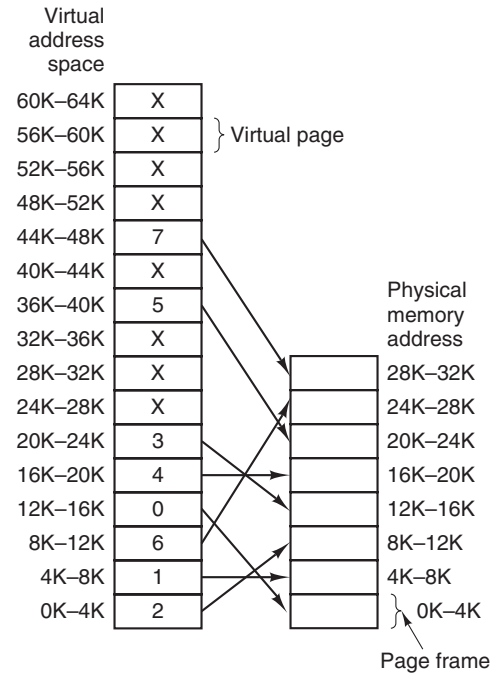


Figure 3-9. The relation between virtual addresses and physical memory addresses is given by the **page table**. Every page begins on a multiple of 4096 and ends 4095 addresses higher, so 4K–8K really means 4096–8191 and 8K to 12K means 8192–12287.

because virtual address 8192 (in virtual page 2) is mapped onto 24576 (in physical page frame 6). As a third example, virtual address 20500 is 20 bytes from the start of virtual page 5 (virtual addresses 20480 to 24575) and maps onto physical address 12288 + 20 = 12308.

By itself, this ability to map the 16 virtual pages onto any of the eight page frames by setting the MMU's map appropriately does not solve the problem that the virtual address space is larger than the physical memory. Since we have only eight physical page frames, only eight of the virtual pages in Fig. 3-9 are mapped onto physical memory. The others, shown as a cross in the figure, are not mapped. In the actual hardware, a **Present/absent bit** keeps track of which pages are physically present in memory.

What happens if the program references an unmapped address, for example, by using the instruction

```
MOV REG,32780
```

which is byte 12 within virtual page 8 (starting at 32768)? The MMU notices that the page is unmapped (indicated by a cross in the figure) and causes the CPU to

trap to the operating system. This trap is called a **page fault**. The operating system picks a little-used page frame and writes its contents back to the disk (if it is not already there). It then fetches (also from the disk) the page that was just referenced into the page frame just freed, changes the map, and restarts the trapped instruction.

For example, if the operating system decided to evict page frame 1, it would load virtual page 8 at physical address 4096 and make two changes to the MMU map. First, it would mark virtual page 1's entry as unmapped, to trap any future accesses to virtual addresses between 4096 and 8191. Then it would replace the cross in virtual page 8's entry with a 1, so that when the trapped instruction is reexecuted, it will map virtual address 32780 to physical address 4108 (4096 + 12).

Now let us look inside the MMU to see how it works and why we have chosen to use a page size that is a power of 2. In Fig. 3-10 we see an example of a virtual address, 8196 (0010000000000100 in binary), being mapped using the MMU map of Fig. 3-9. The incoming 16-bit virtual address is split into a 4-bit page number and a 12-bit offset. With 4 bits for the page number, we can have 16 pages, and with 12 bits for the offset, we can address all 4096 bytes within a page.

The page number is used as an index into the **page table**, yielding the number of the page frame corresponding to that virtual page. If the *Present/absent* bit is 0, a trap to the operating system is caused. If the bit is 1, the page frame number found in the page table is copied to the high-order 3 bits of the output register, along with the 12-bit offset, which is copied unmodified from the incoming virtual address. Together they form a 15-bit physical address. The output register is then put onto the memory bus as the physical memory address.

3.3.2 Page Tables

In a simple implementation, the mapping of virtual addresses onto physical addresses can be summarized as follows: the virtual address is split into a virtual page number (high-order bits) and an offset (low-order bits). For example, with a 16-bit address and a 4-KB page size, the upper 4 bits could specify one of the 16 virtual pages and the lower 12 bits would then specify the byte offset (0 to 4095) within the selected page. However a split with 3 or 5 or some other number of bits for the page is also possible. Different splits imply different page sizes.

The virtual page number is used as an index into the page table to find the entry for that virtual page. From the page table entry, the page frame number (if any) is found. The page frame number is attached to the high-order end of the offset, replacing the virtual page number, to form a physical address that can be sent to the memory.

Thus, the purpose of the page table is to map virtual pages onto page frames. Mathematically speaking, the page table is a function, with the virtual page number as argument and the physical frame number as result. Using the result of this

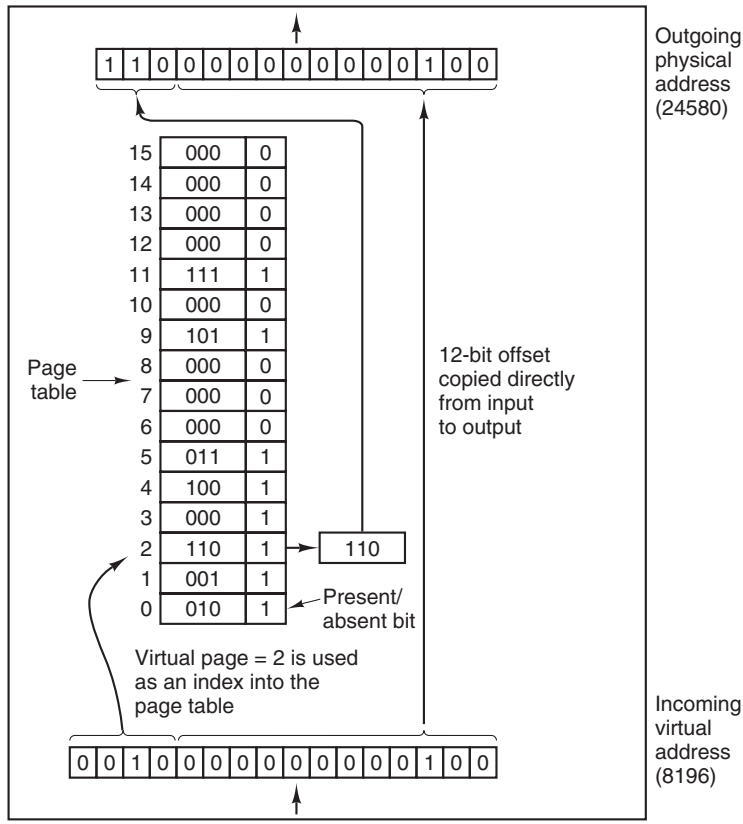


Figure 3-10. The internal operation of the MMU with 16 4-KB pages.

function, the virtual page field in a virtual address can be replaced by a page frame field, thus forming a physical memory address.

In this chapter, we worry only about virtual memory and not full virtualization. In other words: no virtual machines yet. We will see in Chap. 7 that each virtual machine requires its own virtual memory and as a result the page table organization becomes much more complicated—involving shadow or nested page tables and more. Even without such arcane configurations, paging and virtual memory are fairly sophisticated, as we shall see.

Structure of a Page Table Entry

Let us now turn from the structure of the page tables in the large, to the details of a single page table entry. The exact layout of an entry in the page table is highly machine dependent, but the kind of information present is roughly the same from machine to machine. In Fig. 3-11 we present a sample page table entry. The size

varies from computer to computer, but 32 bits is a common size. The most important field is the *Page frame number*. After all, the goal of the page mapping is to output this value. Next to it we have the *Present/absent* bit. If this bit is 1, the entry is valid and can be used. If it is 0, the virtual page to which the entry belongs is not currently in memory. Accessing a page table entry with this bit set to 0 causes a page fault.

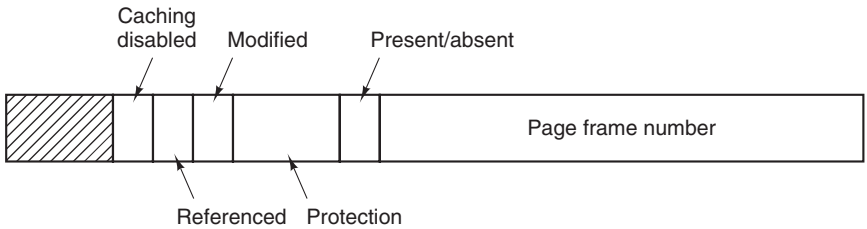


Figure 3-11. A typical page table entry.

The *Protection* bits tell what kinds of access are permitted. In the simplest form, this field contains 1 bit, with 0 for read/write and 1 for read only. A more sophisticated arrangement is having 3 bits, one bit each for enabling reading, writing, and executing the page.

The *Modified* and *Referenced* bits keep track of page usage. When a page is written to, the hardware automatically sets the *Modified* bit. This bit is of value when the operating system decides to reclaim a page frame. If the page in it has been modified (i.e., is "dirty"), it must be written back to the disk. If it has not been modified (i.e., is "clean"), it can just be abandoned, since the disk copy is still valid. The bit is sometimes called the **dirty bit**, since it reflects the page's state.

The *Referenced* bit is set whenever a page is referenced, either for reading or for writing. Its value is used to help the operating system choose a page to evict when a page fault occurs. Pages that are not being used are far better candidates than pages that are, and this bit plays an important role in several of the page replacement algorithms that we will study later in this chapter.

Finally, the last bit allows caching to be disabled for the page. This feature is important for pages that map onto device registers rather than memory. If the operating system is sitting in a tight loop waiting for some I/O device to respond to a command it was just given, it is essential that the hardware keep fetching the word from the device, and not use an old cached copy. With this bit, caching can be turned off. Machines that have a separate I/O space and do not use memory-mapped I/O do not need this bit.

Note that the disk address used to hold the page when it is not in memory is not part of the page table. The reason is simple. The page table holds only that information the hardware needs to translate a virtual address to a physical address.

Information the operating system needs to handle page faults is kept in software tables inside the operating system. The hardware does not need it.

Before getting into more implementation issues, it is worth pointing out again that what virtual memory fundamentally does is create a new abstraction—the address space—which is an abstraction of physical memory, just as a process is an abstraction of the physical processor (CPU). Virtual memory can be implemented by breaking the virtual address space up into pages, and mapping each one onto some page frame of physical memory or having it (temporarily) unmapped. Thus this section is bassically about an abstraction created by the operating system and how that abstraction is managed.

3.3.3 Speeding Up Paging

We have just seen the basics of virtual memory and paging. It is now time to go into more detail about possible implementations. In any paging system, two major issues must be faced:

1. The mapping from virtual address to physical address must be fast.
2. If the virtual address space is large, the page table will be large.

The first point is a consequence of the fact that the virtual-to-physical mapping must be done on every memory reference. All instructions must ultimately come from memory and many of them reference operands in memory as well. Consequently, it is necessary to make one, two, or sometimes more page table references per instruction. If an instruction execution takes, say, 1 nsec, the page table lookup must be done in under 0.2 nsec to avoid having the mapping become a major bottleneck.

The second point follows from the fact that all modern computers use virtual addresses of at least 32 bits, with 64 bits becoming the norm for desktops and laptops. With, say, a 4-KB page size, a 32-bit address space has 1 million pages, and a 64-bit address space has more than you want to contemplate. With 1 million pages in the virtual address space, the page table must have 1 million entries. And remember that each process needs its own page table (because it has its own virtual address space).

The need for large, fast page mapping is a very significant constraint on the way computers are built. The simplest design (at least conceptually) is to have a single page table consisting of an array of fast hardware registers, with one entry for each virtual page, indexed by virtual page number, as shown in Fig. 3-10. When a process is started up, the operating system loads the registers with the process' page table, taken from a copy kept in main memory. During process execution, no more memory references are needed for the page table. The advantages of this method are that it is straightforward and requires no memory references during mapping. A disadvantage is that it is unbearably expensive if the page table is

large; it is just not practical most of the time. Another one is that having to load the full page table at every context switch would completely kill performance.

At the other extreme, the page table can be entirely in main memory. All the hardware needs then is a single register that points to the start of the page table. This design allows the virtual-to-physical map to be changed at a context switch by reloading one register. Of course, it has the disadvantage of requiring one or more memory references to read page table entries during the execution of each instruction, making it very slow.

Translation Lookaside Buffers

Let us now look at widely implemented schemes for speeding up paging and for handling large virtual address spaces, starting with the former. The starting point of most optimization techniques is that the page table is in memory. Potentially, this design has an enormous impact on performance. Consider, for example, a 1-byte instruction that copies one register to another. In the absence of paging, this instruction makes only one memory reference, to fetch the instruction. With paging, at least one additional memory reference will be needed, to access the page table. Since execution speed is generally limited by the rate at which the CPU can get instructions and data out of the memory, having to make two memory references per memory reference reduces performance by half. Under these conditions, no one would use paging.

Computer designers have known about this problem for years and have come up with a solution. Their solution is based on the observation that most programs tend to make a large number of references to a small number of pages, and not the other way around. Thus only a small fraction of the page table entries are heavily read; the rest are barely used at all.

The solution that has been devised is to equip computers with a small hardware device for mapping virtual addresses to physical addresses without going through the page table. The device, called a **TLB** (**Translation Lookaside Buffer**) or sometimes an **associative memory**, is illustrated in Fig. 3-12. It is usually inside the MMU and consists of a small number of entries, eight in this example, but rarely more than 256. Each entry contains information about one page, including the virtual page number, a bit that is set when the page is modified, the protection code (read/write/execute permissions), and the physical page frame in which the page is located. These fields have a one-to-one correspondence with the fields in the page table, except for the virtual page number, which is not needed in the page table. Another bit indicates whether the entry is valid (i.e., in use) or not.

An example that might generate the TLB of Fig. 3-12 is a process in a loop that spans virtual pages 19, 20, and 21, so that these TLB entries have protection codes for reading and executing. The main data currently being used (say, an array being processed) are on pages 129 and 130. Page 140 contains the indices used in the array calculations. Finally, the stack is on pages 860 and 861.

Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

Figure 3-12. A TLB to speed up paging.

Let us now see how the TLB functions. When a virtual address is presented to the MMU for translation, the hardware first checks to see if its virtual page number is present in the TLB by comparing it to all the entries simultaneously (i.e., in parallel). Doing so requires special hardware, which all MMUs with TLBs have. If a valid match is found and the access does not violate the protection bits, the page frame is taken directly from the TLB, without going to the page table. If the virtual page number is present in the TLB but the instruction is trying to write on a read-only page, a protection fault is generated.

The interesting case is what happens when the virtual page number is not in the TLB. The MMU detects the miss and does an ordinary page table lookup. It then evicts one of the entries from the TLB and replaces it with the page table entry just looked up. Thus if that page is used again soon, the second time it will result in a TLB hit rather than a miss. When an entry is purged from the TLB, the modified bit is copied back into the page table entry in memory. The other values are already there, except the reference bit. When the TLB is loaded from the page table, all the fields are taken from memory.

Software TLB Management

Up until now, we have assumed that every machine with paged virtual memory has page tables recognized by the hardware, plus a TLB. In this design, TLB management and handling TLB faults are done entirely by the MMU hardware. Traps to the operating system occur only when a page is not in memory.

In the past, this assumption was true. However, many RISC machines, including the SPARC, MIPS, and (the now dead) HP PA, do nearly all of this page management in software. On these machines, the TLB entries are explicitly loaded by the operating system. When a TLB miss occurs, instead of the MMU going to the page tables to find and fetch the needed page reference, it just generates a TLB fault and tosses the problem into the lap of the operating system. The system must find the page, remove an entry from the TLB, enter the new one, and restart the

instruction that faulted. And, of course, all of this must be done in a handful of instructions because TLB misses occur much more frequently than page faults.

Surprisingly enough, if the TLB is moderately large (say, 64 entries) to reduce the miss rate, software management of the TLB turns out to be acceptably efficient. The main gain here is a much simpler MMU, which frees up a considerable amount of area on the CPU chip for caches and other features that can improve performance. Software TLB management is discussed by Uhlig et al. (1994).

Various strategies were developed long ago to improve performance on machines that do TLB management in software. One approach attacks both reducing TLB misses and reducing the cost of a TLB miss when it does occur (Bala et al., 1994). To reduce TLB misses, sometimes the operating system can use its intuition to figure out which pages are likely to be used next and to preload entries for them in the TLB. For example, when a client process sends a message to a server process on the same machine, it is very likely that the server will have to run soon. Knowing this, while processing the trap to do the send, the system can also check to see where the server's code, data, and stack pages are and map them in before they get a chance to cause TLB faults.

The normal way to process a TLB miss, whether in hardware or in software, is to go to the page table and perform the indexing operations to locate the page referenced. The problem with doing this search in software is that the pages holding the page table may not be in the TLB, which will cause additional TLB faults during the processing. These faults can be reduced by maintaining a large (e.g., 4-KB) software cache of TLB entries in a fixed location whose page is always kept in the TLB. By first checking the software cache, the operating system can substantially reduce TLB misses.

When software TLB management is used, it is essential to understand the difference between different kinds of misses. A **soft miss** occurs when the page referenced is not in the TLB, but is in memory. All that is needed here is for the TLB to be updated. No disk I/O is needed. Typically a soft miss takes 10–20 machine instructions to handle and can be completed in a couple of nanoseconds. In contrast, a **hard miss** occurs when the page itself is not in memory (and of course, also not in the TLB). A disk access is required to bring in the page, which can take several milliseconds, depending on the disk being used. A hard miss is easily a million times slower than a soft miss. Looking up the mapping in the page table hierarchy is known as a **page table walk**.

Actually, it is worse that that. A miss is not just soft or hard. Some misses are slightly softer (or slightly harder) than other misses. For instance, suppose the page walk does not find the page in the process' page table and the program thus incurs a page fault. There are three possibilities. First, the page may actually be in memory, but not in this process' page table. For instance, the page may have been brought in from disk by another process. In that case, we do not need to access the disk again, but merely map the page appropriately in the page tables. This is a pretty soft miss that is known as a **minor page fault**. Second, a **major page fault**

occurs if the page needs to be brought in from disk. Third, it is possible that the program simply accessed an invalid address and no mapping needs to be added in the TLB at all. In that case, the operating system typically kills the program with a **segmentation fault**. Only in this case did the program do something wrong. All other cases are automatically fixed by the hardware and/or the operating system—at the cost of some performance.

3.3.4 Page Tables for Large Memories

TLBs can be used to speed up virtual-to-physical address translation over the original page-table-in-memory scheme. But that is not the only problem we have to tackle. Another problem is how to deal with very large virtual address spaces. Below we will discuss two ways of dealing with them.

Multilevel Page Tables

As a first approach, consider the use of a **multilevel page table**. A simple example is shown in Fig. 3-13. In Fig. 3-13(a) we have a 32-bit virtual address that is partitioned into a 10-bit *PT1* field, a 10-bit *PT2* field, and a 12-bit *Offset* field. Since offsets are 12 bits, pages are 4 KB, and there are a total of 2^{20} of them.

The secret to the multilevel page table method is to avoid keeping all the page tables in memory all the time. In particular, those that are not needed should not be kept around. Suppose, for example, that a process needs 12 megabytes: the bottom 4 megabytes of memory for program text, the next 4 megabytes for data, and the top 4 megabytes for the stack. In between the top of the data and the bottom of the stack is a gigantic hole that is not used.

In Fig. 3-13(b) we see how the two-level page table works. On the left we see the top-level page table, with 1024 entries, corresponding to the 10-bit *PT1* field. When a virtual address is presented to the MMU, it first extracts the *PT1* field and uses this value as an index into the top-level page table. Each of these 1024 entries in the top-level page table represents 4M because the entire 4-gigabyte (i.e., 32-bit) virtual address space has been chopped into chunks of 4096 bytes.

The entry located by indexing into the top-level page table yields the address or the page frame number of a second-level page table. Entry 0 of the top-level page table points to the page table for the program text, entry 1 points to the page table for the data, and entry 1023 points to the page table for the stack. The other (shaded) entries are not used. The *PT2* field is now used as an index into the selected second-level page table to find the page frame number for the page itself.

As an example, consider the 32-bit virtual address 0x00403004 (4,206,596 decimal), which is 12,292 bytes into the data. This virtual address corresponds to *PT1* = 1, *PT2* = 3, and *Offset* = 4. The MMU first uses *PT1* to index into the top-

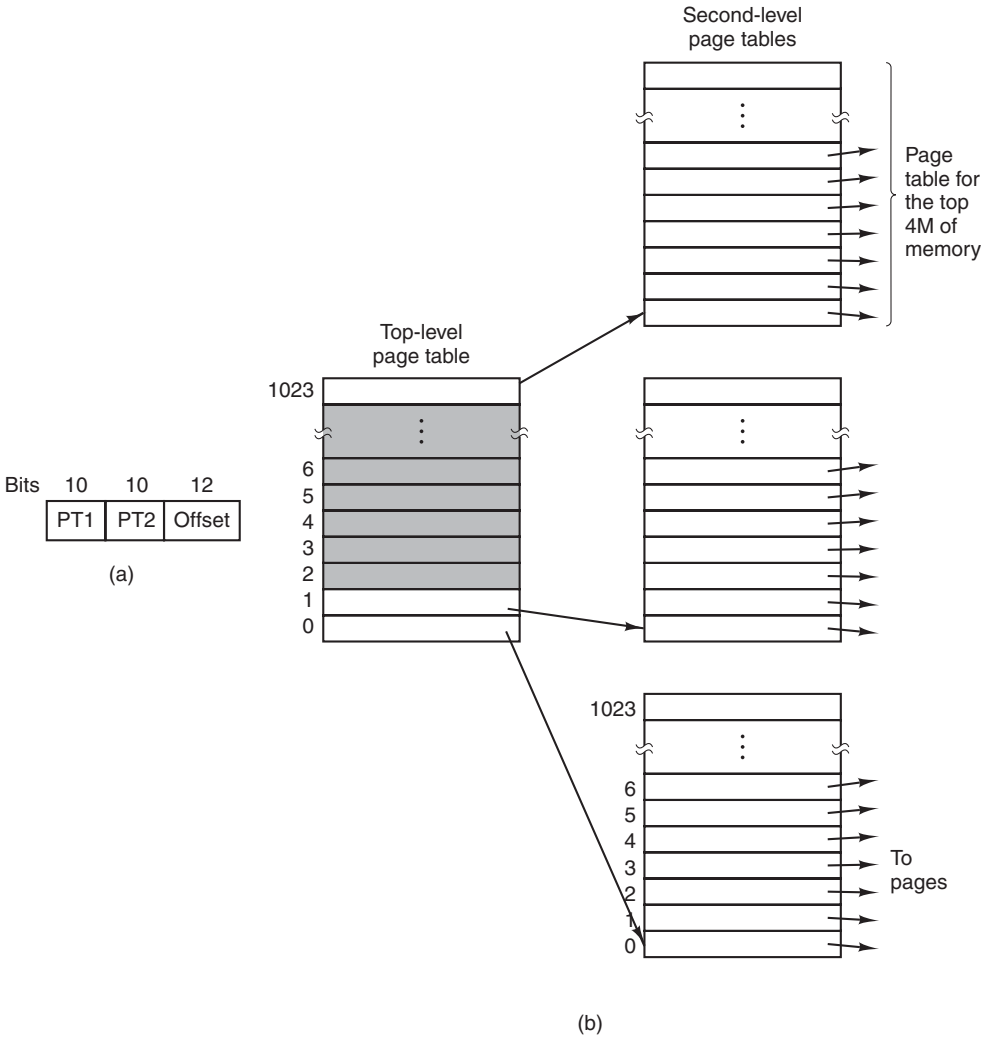


Figure 3-13. (a) A 32-bit address with two page table fields. (b) Two-level page tables.

level page table and obtain entry 1, which corresponds to addresses 4M to 8M – 1. It then uses *PT2* to index into the second-level page table just found and extract entry 3, which corresponds to addresses 12288 to 16383 within its 4M chunk (i.e., absolute addresses 4,206,592 to 4,210,687). This entry contains the page frame number of the page containing virtual address 0x00403004. If that page is not in memory, the *Present/absent* bit in the page table entry will have the value zero, causing a page fault. If the page is present in memory, the page frame number

taken from the second-level page table is combined with the offset (4) to construct the physical address. This address is put on the bus and sent to memory.

The interesting thing to note about Fig. 3-13 is that although the address space contains over a million pages, only four page tables are needed: the top-level table, and the second-level tables for 0 to 4M (for the program text), 4M to 8M (for the data), and the top 4M (for the stack). The *Present/absent* bits in the remaining 1021 entries of the top-level page table are set to 0, forcing a page fault if they are ever accessed. Should this occur, the operating system will notice that the process is trying to reference memory that it is not supposed to and will take appropriate action, such as sending it a signal or killing it. In this example we have chosen round numbers for the various sizes and have picked *PT1* equal to *PT2*, but in actual practice other values are also possible, of course.

The two-level page table system of Fig. 3-13 can be expanded to three, four, or more levels. Additional levels give more flexibility. For instance, Intel's 32 bit 80386 processor (launched in 1985) was able to address up to 4-GB of memory, using a two-level page table that consisted of a **page directory** whose entries pointed to page tables, which, in turn, pointed to the actual 4-KB page frames. Both the page directory and the page tables each contained 1024 entries, giving a total of $2^{10} \times 2^{10} \times 2^{12} = 2^{32}$ addressable bytes, as desired.

Ten years later, the Pentium Pro introduced another level: the **page directory pointer table**. In addition, it extended each entry in each level of the page table hierarchy from 32 bits to 64 bits, so that it could address memory above the 4-GB boundary. As it had only 4 entries in the page directory pointer table, 512 in each page directory, and 512 in each page table, the total amount of memory it could address was still limited to a maximum of 4 GB. When proper 64-bit support was added to the x86 family (originally by AMD), the additional level *could* have been called the "page directory pointer table pointer" or something equally horri. That would have been perfectly in line with how chip makers tend to name things. Mercifully, they did not do this. The alternative they cooked up, "**page map level 4**," may not be a terribly catchy name either, but at least it is short and a bit clearer. At any rate, these processors now use all 512 entries in all tables, yielding an amount of addressable memory of $2^9 \times 2^9 \times 2^9 \times 2^9 \times 2^{12} = 2^{48}$ bytes. They could have added another level, but they probably thought that 256 TB would be sufficient for a while.

Inverted Page Tables

An alternative to ever-increasing levels in a paging hierarchy is known as **inverted page tables**. They were first used by such processors as the PowerPC, the UltraSPARC, and the Itanium (sometimes referred to as "Itanic," as it was not nearly the success Intel had hoped for). In this design, there is one entry per page frame in real memory, rather than one entry per page of virtual address space. For

example, with 64-bit virtual addresses, a 4-KB page size, and 4 GB of RAM, an inverted page table requires only 1,048,576 entries. The entry keeps track of which (process, virtual page) is located in the page frame.

Although inverted page tables save lots of space, at least when the virtual address space is much larger than the physical memory, they have a serious downside: virtual-to-physical translation becomes much harder. When process *n* references virtual page *p*, the hardware can no longer find the physical page by using *p* as an index into the page table. Instead, it must search the entire inverted page table for an entry (*n*, *p*). Furthermore, this search must be done on every memory reference, not just on page faults. Searching a 256K table on every memory reference is not the way to make your machine blindingly fast.

The way out of this dilemma is to make use of the TLB. If the TLB can hold all of the heavily used pages, translation can happen just as fast as with regular page tables. On a TLB miss, however, the inverted page table has to be searched in software. One feasible way to accomplish this search is to have a hash table hashed on the virtual address. All the virtual pages currently in memory that have the same hash value are chained together, as shown in Fig. 3-14. If the hash table has as many slots as the machine has physical pages, the average chain will be only one entry long, greatly speeding up the mapping. Once the page frame number has been found, the new (virtual, physical) pair is entered into the TLB.

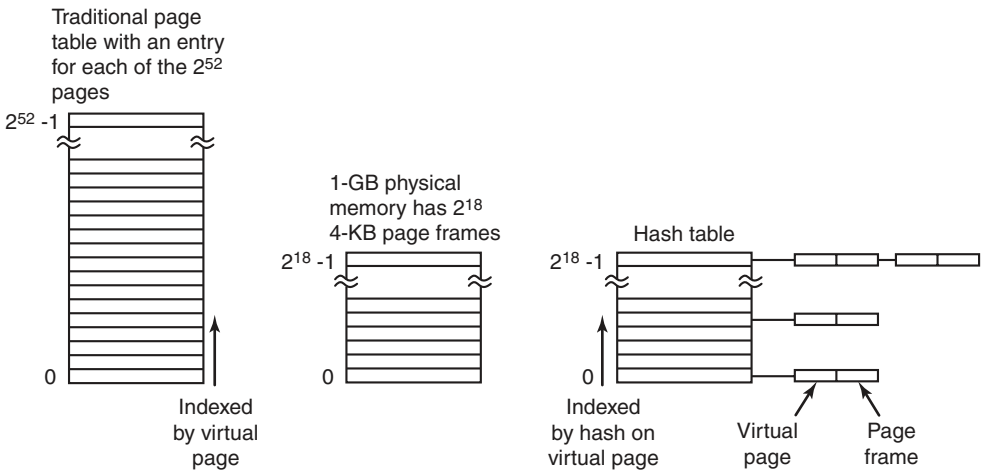


Figure 3-14. Comparison of a traditional page table with an inverted page table.

Inverted page tables are common on 64-bit machines because even with a very large page size, the number of page table entries is gigantic. For example, with 4-MB pages and 64-bit virtual addresses, 2^{42} page table entries are needed. Other approaches to handling large virtual memories can be found in Talluri et al. (1995).

3.4 PAGE REPLACEMENT ALGORITHMS

When a page fault occurs, the operating system has to choose a page to evict (remove from memory) to make room for the incoming page. If the page to be removed has been modified while in memory, it must be rewritten to the disk to bring the disk copy up to date. If, however, the page has not been changed (e.g., it contains program text), the disk copy is already up to date, so no rewrite is needed. The page to be read in just overwrites the page being evicted.

While it would be possible to pick a random page to evict at each page fault, system performance is much better if a page that is not heavily used is chosen. If a heavily used page is removed, it will probably have to be brought back in quickly, resulting in extra overhead. Much work has been done on the subject of page replacement algorithms, both theoretical and experimental. Below we will describe some of the most important ones.

It is worth noting that the problem of "page replacement" occurs in other areas of computer design as well. For example, most computers have one or more memory caches consisting of recently used 32-byte or 64-byte memory blocks. When the cache is full, some block has to be chosen for removal. This problem is precisely the same as page replacement except on a shorter time scale (it has to be done in a few nanoseconds, not milliseconds as with page replacement). The reason for the shorter time scale is that cache block misses are satisfied from main memory, which has no seek time and no rotational latency.

A second example is in a Web server. The server can keep a certain number of heavily used Web pages in its memory cache. However, when the memory cache is full and a new page is referenced, a decision has to be made which Web page to evict. The considerations are similar to pages of virtual memory, except that the Web pages are never modified in the cache, so there is always a fresh copy "on disk." In a virtual memory system, pages in main memory may be either clean or dirty.

In all the page replacement algorithms to be studied below, a certain issue arises: when a page is to be evicted from memory, does it have to be one of the faulting process' own pages, or can it be a page belonging to another process? In the former case, we are effectively limiting each process to a fixed number of pages; in the latter case we are not. Both are possibilities. We will come back to this point in Sec. 3.5.1.

3.4.1 The Optimal Page Replacement Algorithm

The best possible page replacement algorithm is easy to describe but impossible to actually implement. It goes like this. At the moment that a page fault occurs, some set of pages is in memory. One of these pages will be referenced on the very next instruction (the page containing that instruction). Other pages may not

be referenced until 10, 100, or perhaps 1000 instructions later. Each page can be labeled with the number of instructions that will be executed before that page is first referenced.

The optimal page replacement algorithm says that the page with the highest label should be removed. If one page will not be used for 8 million instructions and another page will not be used for 6 million instructions, removing the former pushes the page fault that will fetch it back as far into the future as possible. Computers, like people, try to put off unpleasant events for as long as they can.

The only problem with this algorithm is that it is unrealizable. At the time of the page fault, the operating system has no way of knowing when each of the pages will be referenced next. (We saw a similar situation earlier with the shortest-job-first scheduling algorithm—how can the system tell which job is shortest?) Still, by running a program on a simulator and keeping track of all page references, it is possible to implement optimal page replacement on the *second* run by using the page-reference information collected during the *first* run.

In this way, it is possible to compare the performance of realizable algorithms with the best possible one. If an operating system achieves a performance of, say, only 1% worse than the optimal algorithm, effort spent in looking for a better algorithm will yield at most a 1% improvement.

To avoid any possible confusion, it should be made clear that this log of page references refers only to the one program just measured and then with only one specific input. The page replacement algorithm derived from it is thus specific to that one program and input data. Although this method is useful for evaluating page replacement algorithms, it is of no use in practical systems. Below we will study algorithms that *are* useful on real systems.

3.4.2 The Not Recently Used Page Replacement Algorithm

In order to allow the operating system to collect useful page usage statistics, most computers with virtual memory have two status bits, *R* and *M*, associated with each page. *R* is set whenever the page is referenced (read or written). *M* is set when the page is written to (i.e., modified). The bits are contained in each page table entry, as shown in Fig. 3-11. It is important to realize that these bits must be updated on every memory reference, so it is essential that they be set by the hardware. Once a bit has been set to 1, it stays 1 until the operating system resets it.

If the hardware does not have these bits, they can be simulated using the operating system's page fault and clock interrupt mechanisms. When a process is started up, all of its page table entries are marked as not in memory. As soon as any page is referenced, a page fault will occur. The operating system then sets the *R* bit (in its internal tables), changes the page table entry to point to the correct page, with mode READ ONLY, and restarts the instruction. If the page is subsequently modified, another page fault will occur, allowing the operating system to set the *M* bit and change the page's mode to READ/WRITE.

The *R* and *M* bits can be used to build a simple paging algorithm as follows. When a process is started up, both page bits for all its pages are set to 0 by the operating system. Periodically (e.g., on each clock interrupt), the *R* bit is cleared, to distinguish pages that have not been referenced recently from those that have been.

When a page fault occurs, the operating system inspects all the pages and divides them into four categories based on the current values of their *R* and *M* bits:

Class 0: not referenced, not modified.
Class 1: not referenced, modified.
Class 2: referenced, not modified.
Class 3: referenced, modified.

Although class 1 pages seem, at first glance, impossible, they occur when a class 3 page has its *R* bit cleared by a clock interrupt. Clock interrupts do not clear the *M* bit because this information is needed to know whether the page has to be rewritten to disk or not. Clearing *R* but not *M* leads to a class 1 page.

The **NRU** (**Not Recently Used**) algorithm removes a page at random from the lowest-numbered nonempty class. Implicit in this algorithm is the idea that it is better to remove a modified page that has not been referenced in at least one clock tick (typically about 20 msec) than a clean page that is in heavy use. The main attraction of NRU is that it is easy to understand, moderately efficient to implement, and gives a performance that, while certainly not optimal, may be adequate.

3.4.3 The First-In, First-Out (FIFO) Page Replacement Algorithm

Another low-overhead paging algorithm is the **FIFO** (**First-In, First-Out**) algorithm. To illustrate how this works, consider a supermarket that has enough shelves to display exactly *k* different products. One day, some company introduces a new convenience food—instant, freeze-dried, organic yogurt that can be reconstituted in a microwave oven. It is an immediate success, so our finite supermarket has to get rid of one old product in order to stock it.

One possibility is to find the product that the supermarket has been stocking the longest (i.e., something it began selling 120 years ago) and get rid of it on the grounds that no one is interested any more. In effect, the supermarket maintains a linked list of all the products it currently sells in the order they were introduced. The new one goes on the back of the list; the one at the front of the list is dropped.

As a page replacement algorithm, the same idea is applicable. The operating system maintains a list of all pages currently in memory, with the most recent arrival at the tail and the least recent arrival at the head. On a page fault, the page at the head is removed and the new page added to the tail of the list. When applied to stores, FIFO might remove mustache wax, but it might also remove flour, salt, or butter. When applied to computers the same problem arises: the oldest page may still be useful. For this reason, FIFO in its pure form is rarely used.

3.4.6 The Least Recently Used (LRU) Page Replacement Algorithm

A good approximation to the optimal algorithm is based on the observation that pages that have been heavily used in the last few instructions will probably be heavily used again soon. Conversely, pages that have not been used for ages will probably remain unused for a long time. This idea suggests a realizable algorithm: when a page fault occurs, throw out the page that has been unused for the longest time. This strategy is called **LRU** (**Least Recently Used**) paging.

Although LRU is theoretically realizable, it is not cheap by a long shot. To fully implement LRU, it is necessary to maintain a linked list of all pages in memory, with the most recently used page at the front and the least recently used page at the rear. The difficulty is that the list must be updated on every memory reference. Finding a page in the list, deleting it, and then moving it to the front is a very time consuming operation, even in hardware (assuming that such hardware could be built).

However, there are other ways to implement LRU with special hardware. Let us consider the simplest way first. This method requires equipping the hardware with a 64-bit counter, C, that is automatically incremented after each instruction. Furthermore, each page table entry must also have a field large enough to contain the counter. After each memory reference, the current value of C is stored in the

page table entry for the page just referenced. When a page fault occurs, the operating system examines all the counters in the page table to find the lowest one. That page is the least recently used.

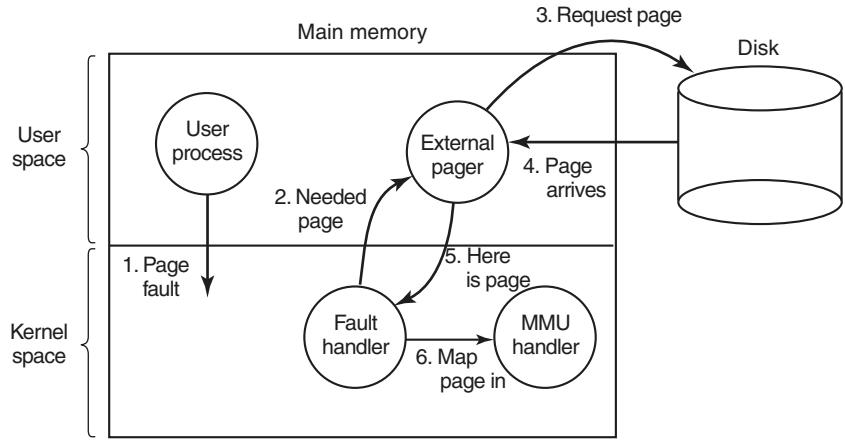


Figure 3-29. Page fault handling with an external pager.

This implementation leaves open where the page replacement algorithm is put. It would be cleanest to have it in the external pager, but there are some problems with this approach. Principal among these is that the external pager does not have access to the *R* and *M* bits of all the pages. These bits play a role in many of the paging algorithms. Thus, either some mechanism is needed to pass this information up to the external pager, or the page replacement algorithm must go in the kernel. In the latter case, the fault handler tells the external pager which page it has selected for eviction and provides the data, either by mapping it into the external pager's address space or including it in a message. Either way, the external pager writes the data to disk.

The main advantage of this implementation is more modular code and greater flexibility. The main disadvantage is the extra overhead of crossing the user-kernel boundary several times and the overhead of the various messages being sent between the pieces of the system. At the moment, the subject is highly controversial, but as computers get faster and faster, and the software gets more and more complex, in the long run sacrificing some performance for more reliable software will probably be acceptable to most implementers.

3.7 SEGMENTATION

The virtual memory discussed so far is one-dimensional because the virtual addresses go from 0 to some maximum address, one address after another. For many problems, having two or more separate virtual address spaces may be much better than having only one. For example, a compiler has many tables that are built up as compilation proceeds, possibly including

1. The source text being saved for the printed listing (on batch systems).
2. The symbol table, containing the names and attributes of variables.
3. The table containing all the integer and floating-point constants used.
4. The parse tree, containing the syntactic analysis of the program.
5. The stack used for procedure calls within the compiler.

Each of the first four tables grows continuously as compilation proceeds. The last one grows and shrinks in unpredictable ways during compilation. In a one-dimensional memory, these five tables would have to be allocated contiguous chunks of virtual address space, as in Fig. 3-30.

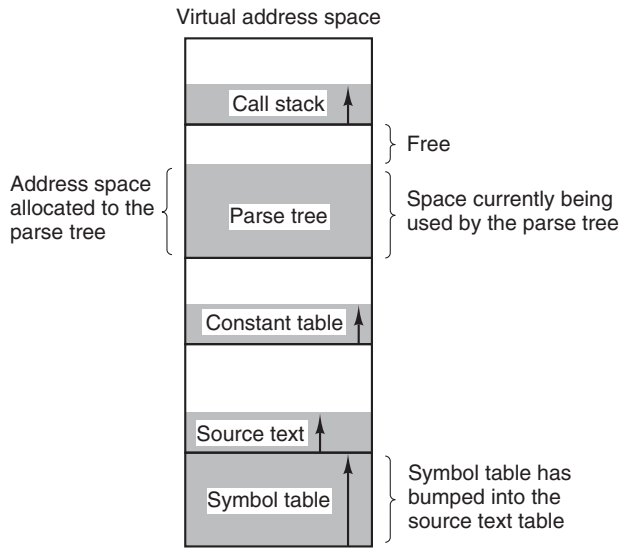


Figure 3-30. In a one-dimensional address space with growing tables, one table may bump into another.

Consider what happens if a program has a much larger than usual number of variables but a normal amount of everything else. The chunk of address space allocated for the symbol table may fill up, but there may be lots of room in the other tables. What is needed is a way of freeing the programmer from having to manage the expanding and contracting tables, in the same way that virtual memory eliminates the worry of organizing the program into overlays.

A straightforward and quite general solution is to provide the machine with many completely independent address spaces, which are called **segments**. Each segment consists of a linear sequence of addresses, starting at 0 and going up to some maximum value. The length of each segment may be anything from 0 to the

maximum address allowed. Different segments may, and usually do, have different lengths. Moreover, segment lengths may change during execution. The length of a stack segment may be increased whenever something is pushed onto the stack and decreased whenever something is popped off the stack.

Because each segment constitutes a separate address space, different segments can grow or shrink independently without affecting each other. If a stack in a certain segment needs more address space to grow, it can have it, because there is nothing else in its address space to bump into. Of course, a segment can fill up, but segments are usually very large, so this occurrence is rare. To specify an address in this segmented or two-dimensional memory, the program must supply a two-part address, a segment number, and an address within the segment. Figure 3-31 illustrates a segmented memory being used for the compiler tables discussed earlier. Five independent segments are shown here.

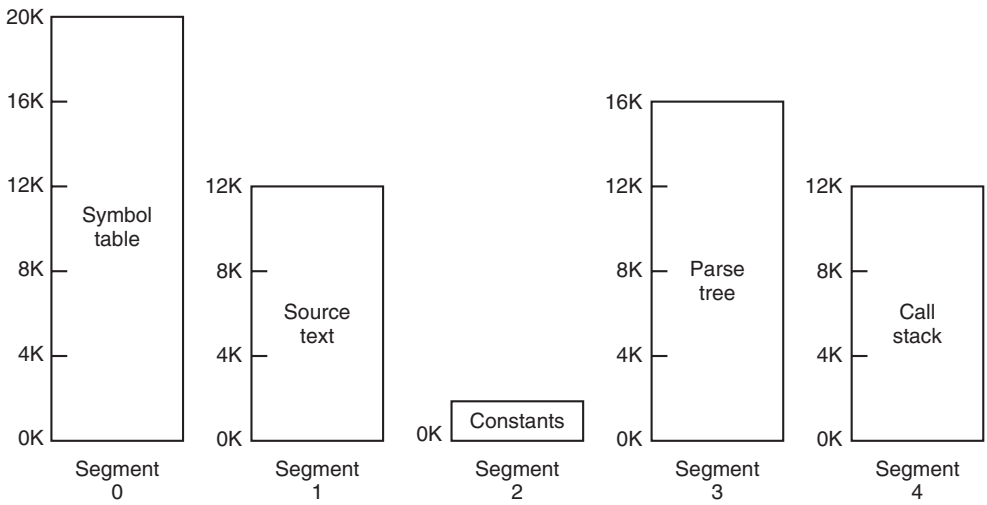


Figure 3-31. A segmented memory allows each table to grow or shrink independently of the other tables.

We emphasize here that a segment is a logical entity, which the programmer is aware of and uses as a logical entity. A segment might contain a procedure, or an array, or a stack, or a collection of scalar variables, but usually it does not contain a mixture of different types.

A segmented memory has other advantages besides simplifying the handling of data structures that are growing or shrinking. If each procedure occupies a separate segment, with address 0 as its starting address, the linking of procedures compiled separately is greatly simplified. After all the procedures that constitute a program have been compiled and linked up, a procedure call to the procedure in segment n will use the two-part address $(n, 0)$ to address word 0 (the entry point).

If the procedure in segment *n* is subsequently modified and recompiled, no other procedures need be changed (because no starting addresses have been modified), even if the new version is larger than the old one. With a one-dimensional memory, the procedures are packed tightly right up next to each other, with no address space between them. Consequently, changing one procedure's size can affect the starting address of all the other (unrelated) procedures in the segment. This, in turn, requires modifying all procedures that call any of the moved procedures, in order to incorporate their new starting addresses. If a program contains hundreds of procedures, this process can be costly.

Segmentation also facilitates sharing procedures or data between several processes. A common example is the shared library. Modern workstations that run advanced window systems often have extremely large graphical libraries compiled into nearly every program. In a segmented system, the graphical library can be put in a segment and shared by multiple processes, eliminating the need for having it in every process' address space. While it is also possible to have shared libraries in pure paging systems, it is more complicated. In effect, these systems do it by simulating segmentation.

Since each segment forms a logical entity that programmers know about, such as a procedure, or an array, different segments can have different kinds of protection. A procedure segment can be specified as execute only, prohibiting attempts to read from or store into it. A floating-point array can be specified as read/write but not execute, and attempts to jump to it will be caught. Such protection is helpful in catching bugs. Paging and segmentation are compared in Fig. 3-32.

3.7.1 Implementation of Pure Segmentation

The implementation of segmentation differs from paging in an essential way: pages are of fixed size and segments are not. Figure 3-33(a) shows an example of physical memory initially containing five segments. Now consider what happens if segment 1 is evicted and segment 7, which is smaller, is put in its place. We arrive at the memory configuration of Fig. 3-33(b). Between segment 7 and segment 2 is an unused area—that is, a hole. Then segment 4 is replaced by segment 5, as in Fig. 3-33(c), and segment 3 is replaced by segment 6, as in Fig. 3-33(d). After the system has been running for a while, memory will be divided up into a number of chunks, some containing segments and some containing holes. This phenomenon, called **checkerboarding** or **external fragmentation**, wastes memory in the holes. It can be dealt with by compaction, as shown in Fig. 3-33(e).

3.7.2 Segmentation with Paging: MULTICS

If the segments are large, it may be inconvenient, or even impossible, to keep them in main memory in their entirety. This leads to the idea of paging them, so that only those pages of a segment that are actually needed have to be around.

Consideration	Paging	Segmentation
Need the programmer be aware that this technique is being used?	No	Yes
How many linear address spaces are there?	1	Many
Can the total address space exceed the size of physical memory?	Yes	Yes
Can procedures and data be distinguished and separately protected?	No	Yes
Can tables whose size fluctuates be accommodated easily?	No	Yes
Is sharing of procedures between users facilitated?	No	Yes
Why was this technique invented?	To get a large linear address space without having to buy more physical memory	To allow programs and data to be broken up into logically independent address spaces and to aid sharing and protection

Figure 3-32. Comparison of paging and segmentation.

Several significant systems have supported paged segments. In this section we will describe the first one: MULTICS. In the next one we will discuss a more recent one: the Intel x86 up until the x86-64.

The MULTICS operating system was one of the most influential operating systems ever, having had a major influence on topics as disparate as UNIX, the x86 memory architecture, TLBs, and cloud computing. It was started as a research project at M.I.T. and went live in 1969. The last MULTICS system was shut down in 2000, a run of 31 years. Few other operating systems have lasted more-or-less unmodified anywhere near that long. While operating systems called Windows have also have be around that long, Windows 8 has absolutely nothing in common with Windows 1.0 except the name and the fact that it was written by Microsoft. Even more to the point, the ideas developed in MULTICS are as valid and useful now as they were in 1965, when the first paper was published (Corbató and Vyssotsky, 1965). For this reason, we will now spend a little bit of time looking at the most innovative aspect of MULTICS, the virtual memory architecture. More information about MULTICS can be found at *www.multicians.org*.

MULTICS ran on the Honeywell 6000 machines and their descendants and provided each program with a virtual memory of up to 2^{18} segments, each of which

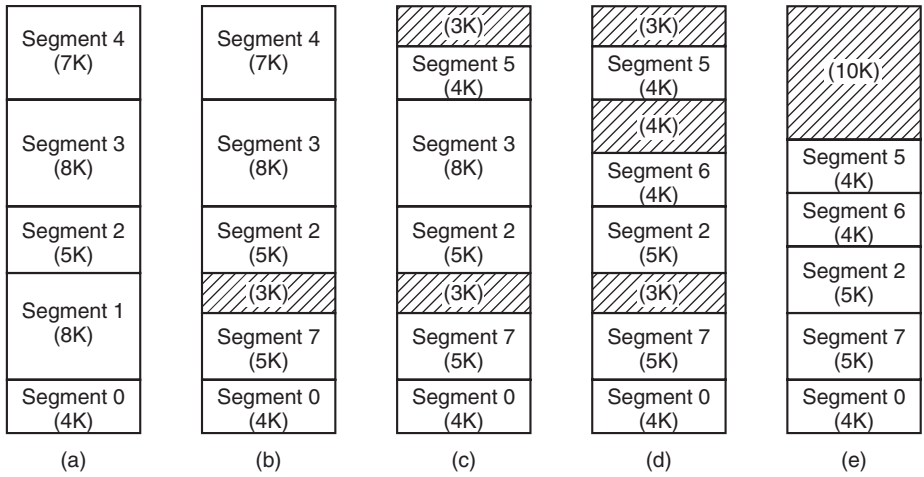


Figure 3-33. (a)-(d) Development of checkerboarding. (e) Removal of the checkerboarding by compaction.

was up to 65,536 (36-bit) words long. To implement this, the MULTICS designers chose to treat each segment as a virtual memory and to page it, combining the advantages of paging (uniform page size and not having to keep the whole segment in memory if only part of it was being used) with the advantages of segmentation (ease of programming, modularity, protection, sharing).

Each MULTICS program had a segment table, with one descriptor per segment. Since there were potentially more than a quarter of a million entries in the table, the segment table was itself a segment and was paged. A segment descriptor contained an indication of whether the segment was in main memory or not. If any part of the segment was in memory, the segment was considered to be in memory, and its page table was in memory. If the segment was in memory, its descriptor contained an 18-bit pointer to its page table, as in Fig. 3-34(a). Because physical addresses were 24 bits and pages were aligned on 64-byte boundaries (implying that the low-order 6 bits of page addresses were 000000), only 18 bits were needed in the descriptor to store a page table address. The descriptor also contained the segment size, the protection bits, and other items. Figure 3-34(b) illustrates a segment descriptor. The address of the segment in secondary memory was not in the segment descriptor but in another table used by the segment fault handler.

Each segment was an ordinary virtual address space and was paged in the same way as the nonsegmented paged memory described earlier in this chapter. The normal page size was 1024 words (although a few small segments used by MULTICS itself were not paged or were paged in units of 64 words to save physical memory).

An address in MULTICS consisted of two parts: the segment and the address within the segment. The address within the segment was further divided into a page

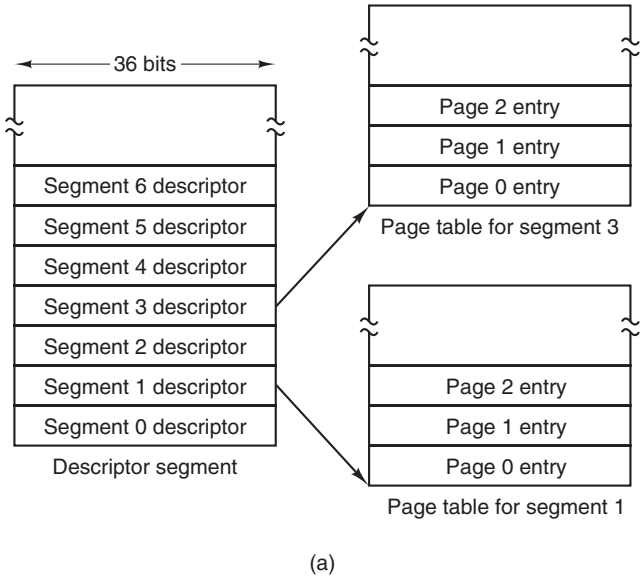


Figure 3-34. The MULTICS virtual memory. (a) The descriptor segment pointed to the page tables. (b) A segment descriptor. The numbers are the field lengths.

number and a word within the page, as shown in Fig. 3-35. When a memory reference occurred, the following algorithm was carried out.

1. The segment number was used to find the segment descriptor.
2. A check was made to see if the segment's page table was in memory. If it was, it was located. If it was not, a segment fault occurred. If there was a protection violation, a fault (trap) occurred.

3. The page table entry for the requested virtual page was examined. If the page itself was not in memory, a page fault was triggered. If it was in memory, the main-memory address of the start of the page was extracted from the page table entry.
4. The offset was added to the page origin to give the main memory address where the word was located.
5. The read or store finally took place.

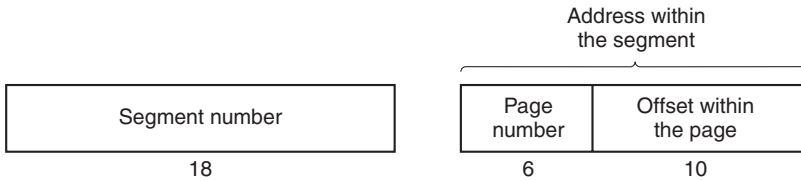


Figure 3-35. A 34-bit MULTICS virtual address.

This process is illustrated in Fig. 3-36. For simplicity, the fact that the descriptor segment was itself paged has been omitted. What really happened was that a register (the descriptor base register) was used to locate the descriptor segment's page table, which, in turn, pointed to the pages of the descriptor segment. Once the descriptor for the needed segment was been found, the addressing proceeded as shown in Fig. 3-36.

As you have no doubt guessed by now, if the preceding algorithm were actually carried out by the operating system on every instruction, programs would not run very fast. In reality, the MULTICS hardware contained a 16-word high-speed TLB that could search all its entries in parallel for a given key. This was the first system to have a TLB, something used in all modern architectures. It is illustrated in Fig. 3-37. When an address was presented to the computer, the addressing hardware first checked to see if the virtual address was in the TLB. If so, it got the page frame number directly from the TLB and formed the actual address of the referenced word without having to look in the descriptor segment or page table.

The addresses of the 16 most recently referenced pages were kept in the TLB. Programs whose working set was smaller than the TLB size came to equilibrium with the addresses of the entire working set in the TLB and therefore ran efficiently; otherwise, there were TLB faults.

3.7.3 Segmentation with Paging: The Intel x86

Up until the x86-64, the virtual memory system of the x86 resembled that of MULTICS in many ways, including the presence of both segmentation and paging. Whereas MULTICS had 256K independent segments, each up to 64K 36-bit words, the x86 has 16K independent segments, each holding up to 1 billion 32-bit

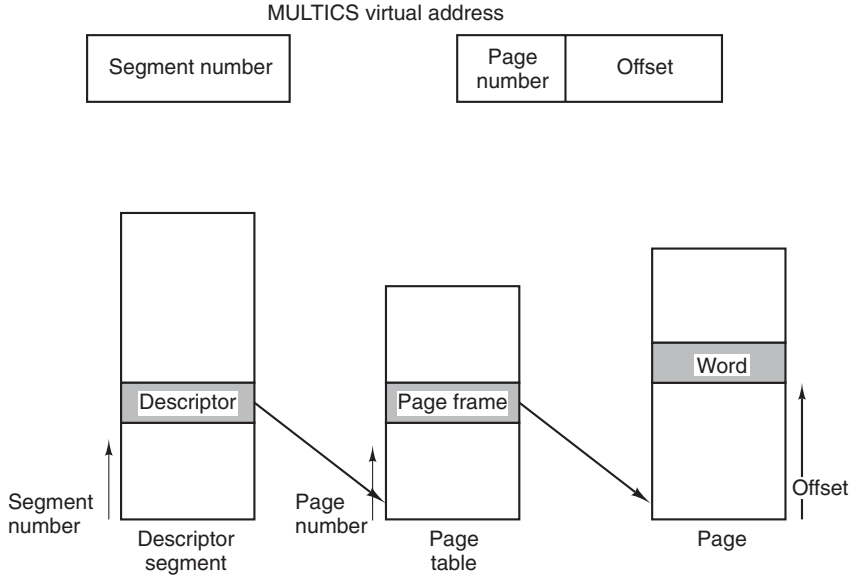


Figure 3-36. Conversion of a two-part MULTICS address into a main memory address.

Comparison field: Segment number	Virtual page	Page frame	Protection	Age	Is this entry used?
4	1	7	Read/write	13	1
6	0	2	Read only	10	1
12	3	1	Read/write	2	1
					0
2	1	0	Execute only	7	1
2	2	12	Execute only	9	1

Figure 3-37. A simplified version of the MULTICS TLB. The existence of two page sizes made the actual TLB more complicated.

words. Although there are fewer segments, the larger segment size is far more important, as few programs need more than 1000 segments, but many programs need large segments. As of x86-64, segmentation is considered obsolete and is no longer supported, except in legacy mode. Although some vestiges of the old segmentation

mechanisms are still available in x86-64's native mode, mostly for compatibility, they no longer serve the same role and no longer offer true segmentation. The x86-32, however, still comes equipped with the whole shebang and it is the CPU we will discuss in this section.

The heart of the x86 virtual memory consists of two tables, called the **LDT** (**Local Descriptor Table**) and the **GDT** (**Global Descriptor Table**). Each program has its own LDT, but there is a single GDT, shared by all the programs on the computer. The LDT describes segments local to each program, including its code, data, stack, and so on, whereas the GDT describes system segments, including the operating system itself.

To access a segment, an x86 program first loads a selector for that segment into one of the machine's six segment registers. During execution, the CS register holds the selector for the code segment and the DS register holds the selector for the data segment. The other segment registers are less important. Each selector is a 16-bit number, as shown in Fig. 3-38.

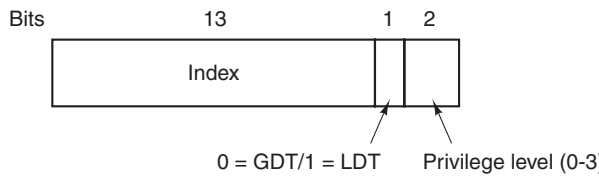


Figure 3-38. An x86 selector.

One of the selector bits tells whether the segment is local or global (i.e., whether it is in the LDT or GDT). Thirteen other bits specify the LDT or GDT entry number, so these tables are each restricted to holding 8K segment descriptors. The other 2 bits relate to protection, and will be described later. Descriptor 0 is forbidden. It may be safely loaded into a segment register to indicate that the segment register is not currently available. It causes a trap if used.

At the time a selector is loaded into a segment register, the corresponding descriptor is fetched from the LDT or GDT and stored in microprogram registers, so it can be accessed quickly. As depicted in Fig. 3-39, a descriptor consists of 8 bytes, including the segment's base address, size, and other information.

The format of the selector has been cleverly chosen to make locating the descriptor easy. First either the LDT or GDT is selected, based on selector bit 2. Then the selector is copied to an internal scratch register, and the 3 low-order bits set to 0. Finally, the address of either the LDT or GDT table is added to it, to give a direct pointer to the descriptor. For example, selector 72 refers to entry 9 in the GDT, which is located at address GDT + 72.

Let us now trace the steps by which a (selector, offset) pair is converted to a physical address. As soon as the microprogram knows which segment register is

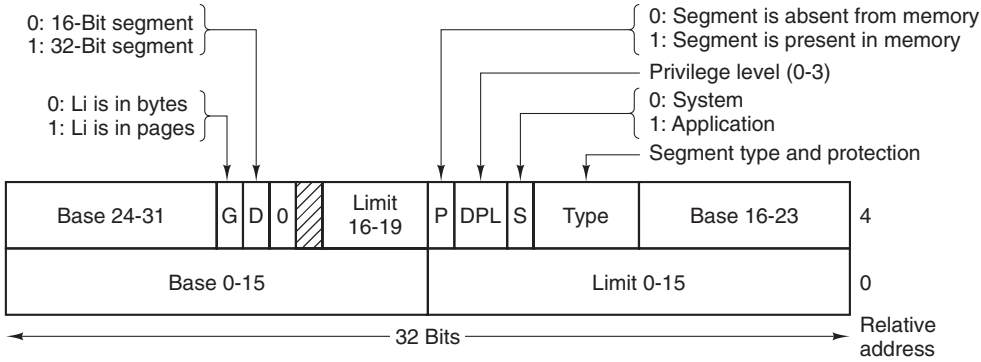


Figure 3-39. x86 code segment descriptor. Data segments differ slightly.

being used, it can find the complete descriptor corresponding to that selector in its internal registers. If the segment does not exist (selector 0), or is currently paged out, a trap occurs.

The hardware then uses the *Limit* field to check if the offset is beyond the end of the segment, in which case a trap also occurs. Logically, there should be a 32-bit field in the descriptor giving the size of the segment, but only 20 bits are available, so a different scheme is used. If the *Gbit* (Granularity) field is 0, the *Limit* field is the exact segment size, up to 1 MB. If it is 1, the *Limit* field gives the segment size in pages instead of bytes. With a page size of 4 KB, 20 bits are enough for segments up to 2^{32} bytes.

Assuming that the segment is in memory and the offset is in range, the x86 then adds the 32-bit *Base* field in the descriptor to the offset to form what is called a **linear address**, as shown in Fig. 3-40. The *Base* field is broken up into three pieces and spread all over the descriptor for compatibility with the 286, in which the *Base* is only 24 bits. In effect, the *Base* field allows each segment to start at an arbitrary place within the 32-bit linear address space.

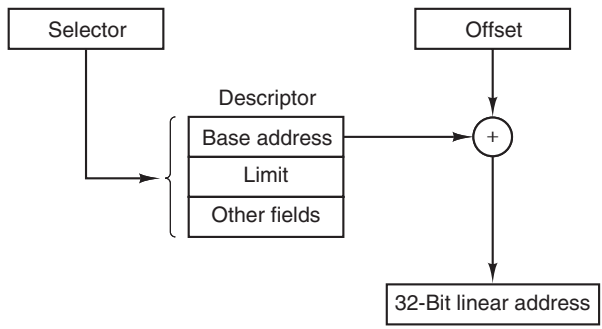


Figure 3-40. Conversion of a (selector, offset) pair to a linear address.

If paging is disabled (by a bit in a global control register), the linear address is interpreted as the physical address and sent to the memory for the read or write. Thus with paging disabled, we have a pure segmentation scheme, with each segment's base address given in its descriptor. Segments are not prevented from overlapping, probably because it would be too much trouble and take too much time to verify that they were all disjoint.

On the other hand, if paging is enabled, the linear address is interpreted as a virtual address and mapped onto the physical address using page tables, pretty much as in our earlier examples. The only real complication is that with a 32-bit virtual address and a 4-KB page, a segment might contain 1 million pages, so a two-level mapping is used to reduce the page table size for small segments.

Each running program has a page directory consisting of 1024 32-bit entries. It is located at an address pointed to by a global register. Each entry in this directory points to a page table also containing 1024 32-bit entries. The page table entries point to page frames. The scheme is shown in Fig. 3-41.

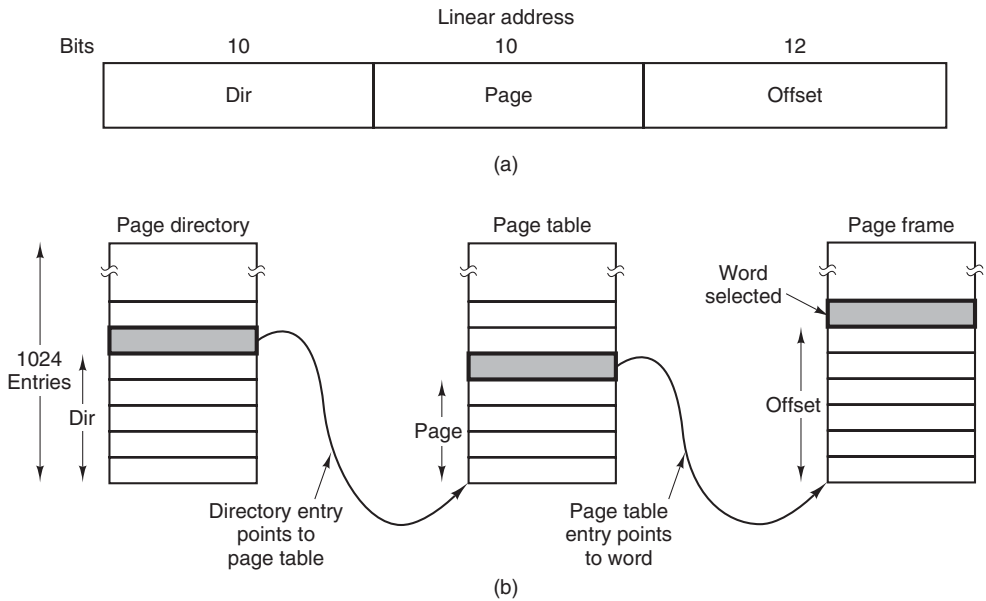


Figure 3-41. Mapping of a linear address onto a physical address.

In Fig. 3-41(a) we see a linear address divided into three fields, *Dir*, *Page*, and *Offset*. The *Dir* field is used to index into the page directory to locate a pointer to the proper page table. Then the *Page* field is used as an index into the page table to find the physical address of the page frame. Finally, *Offset* is added to the address of the page frame to get the physical address of the byte or word needed.

The page table entries are 32 bits each, 20 of which contain a page frame number. The remaining bits contain access and dirty bits, set by the hardware for the benefit of the operating system, protection bits, and other utility bits.

Each page table has entries for 1024 4-KB page frames, so a single page table handles 4 megabytes of memory. A segment shorter than 4M will have a page directory with a single entry, a pointer to its one and only page table. In this way, the overhead for short segments is only two pages, instead of the million pages that would be needed in a one-level page table.

To avoid making repeated references to memory, the x86, like MULTICS, has a small TLB that directly maps the most recently used *Dir-Page* combinations onto the physical address of the page frame. Only when the current combination is not present in the TLB is the mechanism of Fig. 3-41 actually carried out and the TLB updated. As long as TLB misses are rare, performance is good.

It is also worth noting that if some application does not need segmentation but is simply content with a single, paged, 32-bit address space, that model is possible. All the segment registers can be set up with the same selector, whose descriptor has *Base* = 0 and *Limit* set to the maximum. The instruction offset will then be the linear address, with only a single address space used—in effect, normal paging. In fact, all current operating systems for the x86 work this way. OS/2 was the only one that used the full power of the Intel MMU architecture.

So why did Intel kill what was a variant of the perfectly good MULTICS memory model that it supported for close to three decades? Probably the main reason is that neither UNIX nor Windows ever used it, even though it was quite efficient because it eliminated system calls, turning them into lightning-fast procedure calls to the relevant address within a protected operating system segment. None of the developers of any UNIX or Windows system wanted to change their memory model to something that was x86 specific because it would break portability to other platforms. Since the software was not using the feature, Intel got tired of wasting chip area to support it and removed it from the 64-bit CPUs.

All in all, one has to give credit to the x86 designers. Given the conflicting goals of implementing pure paging, pure segmentation, and paged segments, while at the same time being compatible with the 286, and doing all of this efficiently, the resulting design is surprisingly simple and clean.

3.8 RESEARCH ON MEMORY MANAGEMENT

Traditional memory management, especially paging algorithms for uniprocessor CPUs, was once a fruitful area for research, but most of that seems to have largely died off, at least for general-purpose systems, although there are some people who never say die (Moruz et al., 2012) or are focused on some application, such as online transaction processing, that has specialized requirements (Stoica and Ailamaki, 2013). Even on uniprocessors, paging to SSDs rather than to hard disks brings up new issues and requires new algorithms (Chen et al., 2012). Paging to the up-and-coming nonvolatile phase-change memories also requires rethinking

Unit 4: File Systems

Every computer program or application must store and retrieve the necessary information. However, here are some of the issues with how and why operating systems implement files. Let's start with the problems and then move on to the solution.

- **Problem No.1**: However, when the process is running on the computer system, the application can only store a limited amount of data in its own address space. The storage capacity is limited in this case by the size of the virtual address space.
- **Problem No.2**: Another significant issue is that when the process is terminated, the information is lost. And for many computer programs, the information stored must be kept for an extended period of time, if not forever.
- **Problem No.3**: Another major issue is that multiple processes must access the information at the same time. Consider the following scenario: if we have an online telephone directory that is stored within the address space of a single process and can only be accessed by that process.

Solution: As a result, in order to solve this type of large problem, information must be made independent of any single process. Following the resolution of the aforementioned major issues, we now have three essential requirements for the long-term storage of critical information stored in computer applications:

Problem No.	Solution
1	A large amount of data must be able to be stored.
2	The information must survive the end of the process that is using it.
3	Multiple processes must have concurrent access to the information.

The solution to all three of the aforementioned issues is to store information on discs and other external media in units known as files.

What is a file?

A file can be defined as a data structure that stores a sequence of records. The file is stored in the file system and can reside on a disk or in the main memory. The file can be simple (text only) or complex (specially formatted). A collection of files is called a directory. The collection of directories at different levels is called the file system.

4.1.1 File Naming

- Files are an abstract mechanism; the most important characteristic of any abstract mechanism is the way that the objects being managed are named.
- Every time a process in a computer system generates a new file, that file is subsequently assigned a name by the process. You could say that the file continues to exist even after the process has ended or stopped running; in other words, you can say that the file continues to exist and that it can also be accessed by some other processes simply by using the name of the file.
- The rule for file naming varies from system to system, but all the current OSs allow strings as file names. Therefore, myfile, file1, my_file, etc. are the three legal file names.
- Digits and special characters can also be used to name a file, such as myfile, file-2-1, file_21, 32, etc., which are also legal file names.
- The naming of files is supported by many file systems as long as the names contain all 255 possible characters. While other file systems do not, some do make a distinction between the capitalization of words and the case of their initial letters.

The following is a list of the names of the three files:

- myfile
- Myfile
- MYFILE

- In a UNIX operating system, each of the above-mentioned three files has its own identity, whereas in MS-DOS, each of the above-mentioned three files is identical.
- Almost every OS supports two-part file names. These two parts of the file names are separated by a period or dot (.).
- The part that comes after the period (.) is known as the file extension, and it typically provides some kind of information regarding the file.
- The following is a table that describes the most common file extensions and what files they can be used to open:

File Extension	File Meaning
myfile.bak	This indicates the existence of a backup file.
myfile.c	This denotes a source file for the C programming language.
myfile.gif	This denotes an image file in GIF format.
myfile.hlp	This indicates the presence of a help file.
myfile.html	This is an HTML (HyperText Markup Language) file.
myfile.jpg	This denotes an image file in JPG format.
myfile.mp3	This denotes an MP3 music or audio file in which the music has been encoded in MPEG Layer 3 audio format.
myfile.mpg	This is an MPEG video file containing a movie encoded with the MPEG standard.
myfile.o	This indicates the presence of an object file.
myfile.pdf	This indicates a file in Portable Document Format (PDF).
myfile.ps	This denotes a PostScript file.
myfile.tex	This denotes TEX formatting programme input.
myfile.txt	This denotes a standard text file.
myfile.zip	This indicates that the archive has been compressed.

4.1.2 File System Structure

A File Structure should be according to a required format that the operating system can understand.

• A file has a certain defined structure according to its type.

• A text file is a sequence of characters organized into lines.

• A source file is a sequence of procedures and functions.

• An object file is a sequence of bytes organized into blocks that are understandable by the machine.

• When operating system defines different file structures, it also contains the code to support these file structure. Unix, MS-DOS support minimum number of file structure.

Files can be structured in any of several ways. Three common possibilities are depicted in **Fig.1, Fig.2 Fig.3**

File Structure 1

• Here, as you can see from the figure 1, the file is an unstructured sequence of bytes.

• Therefore, the OS doesn't care about what is in the file, as all it sees are bytes.

File Structure 2

• Now, as you can see from the figure 2 that shows the second structure of a file, where a file is a sequence of fixed-length records where each with some internal structure.

• Central to the idea about a file being a sequence of records is the idea that read operation returns a record and write operation just appends a record.

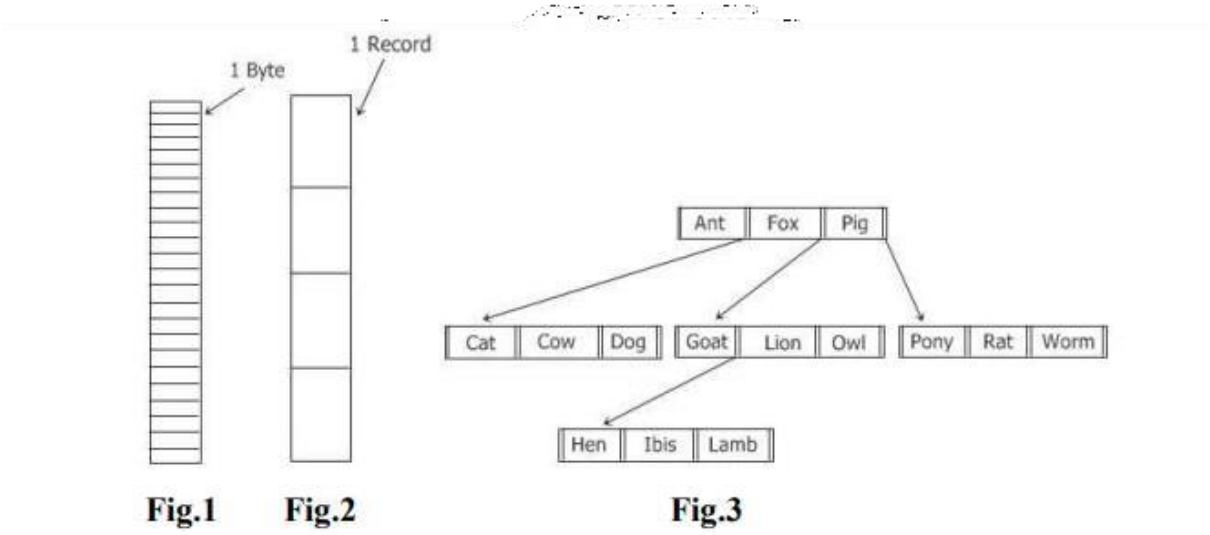


Fig.1 **Fig.2** **Fig.3**

File Structure 3

• Now in the last structure of a file that you can see in the figure 3, a file basically consists of a tree of records, not necessarily all the same length, each containing a key field in a fixed position in the record. The tree is stored on the field, just to allow the rapid searching for a specific key.

4.1.3 File Types

There are several types of files supported by many operating systems. For example, a Windows-based operating system supports the following types of files:

- Regular files
- Directories

And UNIX-based operating systems support the following types of files:

- Directories
- Regular files

- Character special files
- Block special files

Because "regular files" are available as a file type in both Windows and Unix systems, let us first briefly describe them. Then we'll talk about "directories," which are available in both Windows and Unix-based operating systems. The "character special files" and "block special files" will then be discussed.

Directories

Directories are the system files that are used to maintain the structure of the file system.

To keep track of files, file systems typically use directories. Directories are files in and of themselves in many systems.

Regular Files

Regular files contain user information. Generally, regular files are either ASCII files or binary files. These two types of files are described below.

- **ASCII Files**: ASCII files basically consist of lines of text. The big plus point or advantage of ASCII files is that they can be displayed and printed as is, and they can also be edited using any text editor such as Notepad, Wordpad, Notepad+, etc.
- **Binary Files**: The binary files listed on the printer produce random garbage. Generally, binary files have some internal structure that is only known to the programs that use them. Technically, binary files are just a sequence of bytes, and the OS will only execute a file if it has the proper format.

Character Special Files

Character special files are mostly about I/O and are used to model serial input/output devices like printers, networks, and so on.

Block Special Files

Basically, block-specific files are used to model the disks.

4.1.4 File Access

- There is only one type of file access that was supported by the early OS, which is called sequential access.
- In sequential access file systems, a process could read all bytes and records in a file in order, starting at the beginning, but couldn't skip around and read them out of order.
- Sequential files could be read as often as needed. Sequential files were very convenient when magnetic tape was the storage medium.
- However, when discs were used as the storage medium for the required files, it became possible to read the bytes or records of a file out of order or to access any record by key rather than by position.

- Files whose bytes or records can be read in any order are called random access files. Random access files are required and essential for many computer applications, e.g., database systems.

4.1.5 File Attribute

- Every file has a name and its data. In addition, all operating systems associate other information with each file, for example, the date and time the file was last modified and the file's size. We will call these extra items the file's attributes. Some people call them metadata.
- The list of attributes varies considerably from system to system. The table of Fig. 4-4 shows some of the possibilities, but other ones also exist.

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file was last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

Figure 4-4. Some possible file attributes.

- The first four attributes relate to the file's protection and tell who may access it and who may not. In some systems the user must present a password to access a file, in which case the password must be one of the attributes.
- The flags are bits or short fields that control or enable some specific property. The archive flag is a bit that keeps track of whether the file has been backed up recently.

- The temporary flag allows a file to be marked for automatic deletion when the process that created it terminates.
- The record-length, key-position, and key-length fields are only present in files whose records can be looked up using a key. They provide the information required to find the keys.
- The various times keep track of when the file was created, most recently accessed, and most recently modified. These are useful for a variety of purposes.
- The current size tells how big the file is at present. Some old mainframe operating systems required the maximum size to be specified when the file was created, in order to let the operating system, reserve the maximum amount of storage in advance.

4.1.5 File Operations

Files exist to store information and allow it to be retrieved later. Different systems provide different operations to allow storage and retrieval. Below is a discussion of the most common system calls relating to files.

1. Create: The file is created with no data. The purpose of the call is to announce that the file is coming and to set some of the attributes.

2. Delete: When the file is no longer needed, it has to be deleted to free up disk space. There is always a system call for this purpose.

3. Open: Before using a file, a process must open it. The purpose of the open call is to allow the system to fetch the attributes and list of disk addresses into main memory for rapid access on later calls.

4. Close: When all the accesses are finished, the attributes and disk addresses are no longer needed, so the file should be closed to free up internal table space.

5. Read: The file read operation is only used to read the data stored in the specified file.

6. Write: Data are written to the file again, usually at the current position. If the current position is the end of the file, the file's size increases. If the current position is in the middle of the file, existing data are overwritten and lost forever.

7. Append: The file append operation is similar to the file write operation, except that it only adds data to the end of the file.

8. Seek: For random-access files, a method is needed to specify from where to take the data. One common approach is a system call, seek, that repositions the file pointer to a specific place in the file. After this call has completed, data can be read from, or written to, that position.

9. Get attributes: The file get attributes operation is performed by processes when they need to read the file attributes in order to complete their tasks.

10. Set attributes: The file set attribute operation is used to set some of the attributes (user settable attributes) after the file has been created.

11. Rename: It frequently happens that a user needs to change the name of an existing file. This system call makes that possible. It is not always strictly necessary, because the file can usually be copied to a new file with the new name, and the old file then deleted.

4.2 Directories:

Directories are the system files that are used to maintain the structure of the file system. To keep track of files, file systems typically use directories. Directories are files in and of themselves in many systems

4.2.1 Single-Level Directory Systems

The simplest form of directory system is having one directory containing all the files.

Sometimes it is called the root directory, but since it is the only one, the name does not matter much. On early personal computers, this system was common, in part because there was only one user. Interestingly enough, the world's first supercomputer, the CDC 6600, also had only a single directory for all files, even though it was used by many users at once. This decision was no doubt made to keep the software design simple.

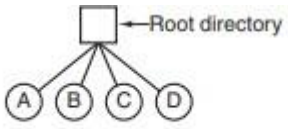


Figure 4-6. A single-level directory system containing four files.

An example of a system with one directory is given in Fig. 4-6. Here the directory contains four files.

The advantages of this scheme are its simplicity and the ability to locate files quickly.

4.2.2 Hierarchical Directory Systems:

- Hierarchical directory systems are used for users with a large number of files, as the single-level and two-level directory systems are not satisfactory.
- The term "hierarchy" refers to a tree structure made up of directories.

- A hierarchical approach allows for each user on a computer system to have access to as many directories as they require. This makes it possible for files to be organized in a manner that is more intuitive.
- The following hierarchical directory structure is depicted in the figure that follows:

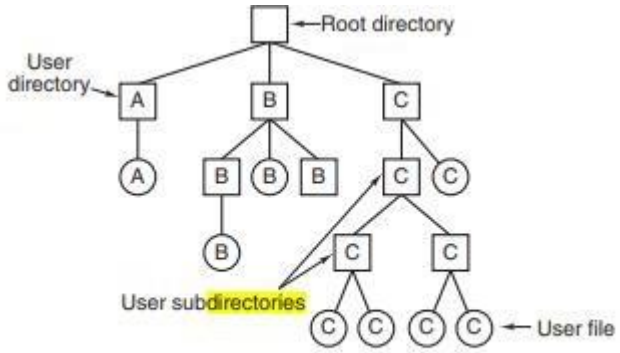


Figure 4-7. A hierarchical directory system.

- In hierarchical directory there can be as many directories as are needed to group the files in natural ways. Furthermore, if multiple users share a common file server, as is the case on many company networks, each user can have a private root directory for his or her own hierarchy. This approach is shown in Fig. 4-7. Here, the directories A, B, and C contained in the root directory each belong to a different user, two of whom have created subdirectories for projects they are working on.
- The ability for users to create an arbitrary number of subdirectories provides a powerful structuring tool for users to organize their work. For this reason, nearly all modern file systems are organized in this manner.

4.2.3 Path Names

When the file system is organized as a directory tree, a method for specifying the names of the files is required, and this method must be discovered. As a result, the file names are typically specified using one of the two methods that are listed below:

- Absolute path name
- Relative path name

Absolute Path Name

The path from the top-level directory all the way down to the file is what the "absolute path name" refers to. As an example, the path

usr/mcaos/myfile

means that the root directory has a subdirectory called usr, which has another subdirectory called mcaos, which has a file called myfile inside of it. The file is located within the mcaos directory.

On the Windows platform, the above-mentioned path name can be specified using the following format:

usr\mcaos\myfile

Relative Path Name

The relative path name is used in conjunction with the concept of the working directory or current directory.

A user can designate one directory as the current working directory, in which case all the path names not beginning at the root directory are taken relative to the working directory.

If the current working directory is /usr/mcaos, for example, the file whose absolute path is /usr/mcaos/myfile can now be called myfile.

4.2.4 Directory Operations

The allowed system calls for managing directories exhibit more variation from system to system than system calls for files. To give an impression of what they are and how they work, we will give a sample (taken from UNIX).

1. Create: A directory is created. It is empty except for dot and dotdot, which are put there automatically by the system (or in a few cases, by the mkdir program).

2. Delete: A directory is deleted. Only an empty directory can be deleted. A directory containing only dot and dotdot is considered empty as these cannot be deleted.

3. Opendir: Directories can be read. For example, to list all the files in a directory, a listing program opens the directory to read out the names of all the files it contains. Before a directory can be read, it must be opened, analogous to opening and reading a file.

4. Closedir: When a directory has been read, it should be closed to free up internal table space.

5. Readdir: This call returns the next entry in an open directory. Formerly, it was possible to read directories using the usual read system call, but that approach has the disadvantage of forcing the programmer to know and deal with the internal structure of directories. In contrast, readdir always returns one entry in a standard format, no matter which of the possible directory structures is being used.

6. Rename: In many respects, directories are just like files and can be renamed the same way

files can be.

7. Link: Linking is a technique that allows a file to appear in more than one directory. This system call specifies an existing file and a path name, and creates a link from the existing file to the name specified by the path. In this way, the same file may appear in multiple directories. A link of this kind, which increments the counter in the file's i-node (to keep track of the number of directory entries containing the file), is sometimes called a hard link.

8. Unlink: A directory entry is removed. If the file being unlinked is only present in one directory (the normal case), it is removed from the file system. If it is present in multiple directories, only the path name specified is removed. The others remain. In UNIX, the system call for deleting files (discussed earlier) is, in fact, unlink.

name, and creates a link from the existing file to the name specified by the path. In this way, the same file may appear in multiple directories. A link of this kind, which increments the counter in the file's i-node (to keep track of the number of directory entries containing the file), is sometimes called a **hard link**.

8. Unlink. A directory entry is removed. If the file being unlinked is only present in one directory (the normal case), it is removed from the file system. If it is present in multiple directories, only the path name specified is removed. The others remain. In UNIX, the system call for deleting files (discussed earlier) is, in fact, unlink.

The above list gives the most important calls, but there are a few others as well, for example, for managing the protection information associated with a directory.

A variant on the idea of linking files is the **symbolic link**. Instead, of having two names point to the same internal data structure representing a file, a name can be created that points to a tiny file naming another file. When the first file is used, for example, opened, the file system follows the path and finds the name at the end. Then it starts the lookup process all over using the new name. Symbolic links have the advantage that they can cross disk boundaries and even name files on remote computers. Their implementation is somewhat less efficient than hard links though.

4.3 FILE-SYSTEM IMPLEMENTATION

Now it is time to turn from the user's view of the file system to the implementor's view. Users are concerned with how files are named, what operations are allowed on them, what the directory tree looks like, and similar interface issues. Implementors are interested in how files and directories are stored, how disk space is managed, and how to make everything work efficiently and reliably. In the following sections we will examine a number of these areas to see what the issues and trade-offs are.

4.3.1 File-System Layout

File systems are stored on disks. Most disks can be divided up into one or more partitions, with independent file systems on each partition. Sector 0 of the disk is called the **MBR** (**Master Boot Record**) and is used to boot the computer. The end of the MBR contains the partition table. This table gives the starting and ending addresses of each partition. One of the partitions in the table is marked as active. When the computer is booted, the BIOS reads in and executes the MBR. The first thing the MBR program does is locate the active partition, read in its first block, which is called the **boot block**, and execute it. The program in the boot block loads the operating system contained in that partition. For uniformity, every

partition starts with a boot block, even if it does not contain a bootable operating system. Besides, it might contain one in the future.

Other than starting with a boot block, the layout of a disk partition varies a lot from file system to file system. Often the file system will contain some of the items shown in Fig. 4-9. The first one is the **superblock**. It contains all the key parameters about the file system and is read into memory when the computer is booted or the file system is first touched. Typical information in the superblock includes a magic number to identify the file-system type, the number of blocks in the file system, and other key administrative information.

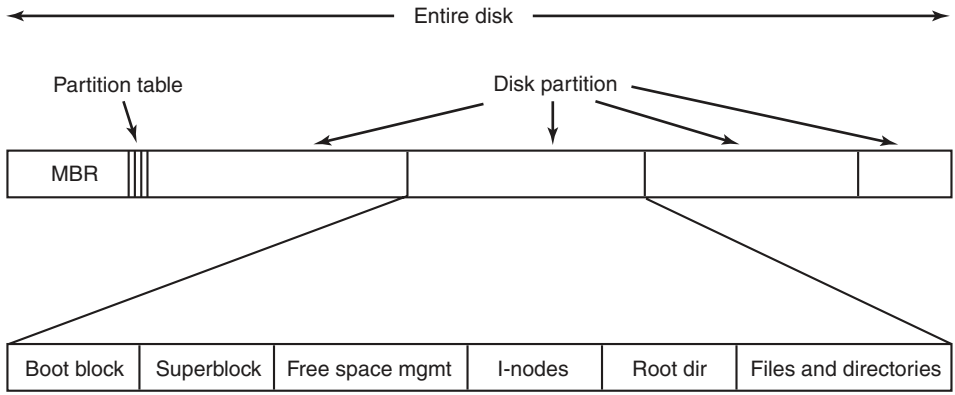


Figure 4-9. A possible file-system layout.

Next might come information about free blocks in the file system, for example in the form of a bitmap or a list of pointers. This might be followed by the i-nodes, an array of data structures, one per file, telling all about the file. After that might come the root directory, which contains the top of the file-system tree. Finally, the remainder of the disk contains all the other directories and files.

4.3.2 Implementing Files

Probably the most important issue in implementing file storage is keeping track of which disk blocks go with which file. Various methods are used in different operating systems. In this section, we will examine a few of them.

Contiguous Allocation

The simplest allocation scheme is to store each file as a contiguous run of disk blocks. Thus on a disk with 1-KB blocks, a 50-KB file would be allocated 50 consecutive blocks. With 2-KB blocks, it would be allocated 25 consecutive blocks.

We see an example of contiguous storage allocation in Fig. 4-10(a). Here the first 40 disk blocks are shown, starting with block 0 on the left. Initially, the disk

was empty. Then a file *A*, of length four blocks, was written to disk starting at the beginning (block 0). After that a six-block file, *B*, was written starting right after the end of file *A*.

Note that each file begins at the start of a new block, so that if file *A* was really 3½ blocks, some space is wasted at the end of the last block. In the figure, a total of seven files are shown, each one starting at the block following the end of the previous one. Shading is used just to make it easier to tell the files apart. It has no actual significance in terms of storage.

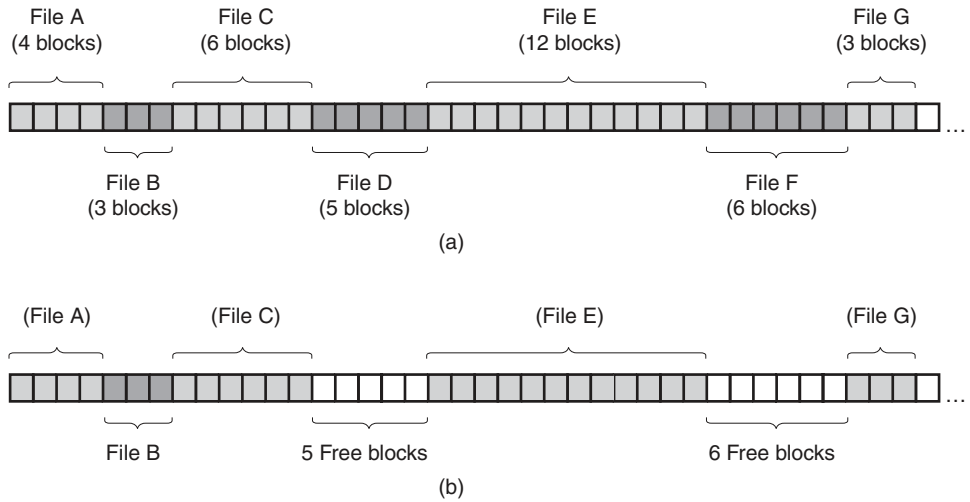


Figure 4-10. (a) Contiguous allocation of disk space for seven files. (b) The state of the disk after files *D* and *F* have been removed.

Contiguous disk-space allocation has two significant advantages. First, it is simple to implement because keeping track of where a file's blocks are is reduced to remembering two numbers: the disk address of the first block and the number of blocks in the file. Given the number of the first block, the number of any other block can be found by a simple addition.

Second, the read performance is excellent because the entire file can be read from the disk in a single operation. Only one seek is needed (to the first block). After that, no more seeks or rotational delays are needed, so data come in at the full bandwidth of the disk. Thus contiguous allocation is simple to implement and has high performance.

Unfortunately, contiguous allocation also has a very serious drawback: over the course of time, the disk becomes fragmented. To see how this comes about, examine Fig. 4-10(b). Here two files, *D* and F, have been removed. When a file is removed, its blocks are naturally freed, leaving a run of free blocks on the disk. The disk is not compacted on the spot to squeeze out the hole, since that would involve copying all the blocks following the hole, potentially millions of blocks, which

would take hours or even days with large disks. As a result, the disk ultimately consists of files and holes, as illustrated in the figure.

Initially, this fragmentation is not a problem, since each new file can be written at the end of disk, following the previous one. However, eventually the disk will fill up and it will become necessary to either compact the disk, which is prohibitively expensive, or to reuse the free space in the holes. Reusing the space requires maintaining a list of holes, which is doable. However, when a new file is to be created, it is necessary to know its final size in order to choose a hole of the correct size to place it in.

Imagine the consequences of such a design. The user starts a word processor in order to create a document. The first thing the program asks is how many bytes the final document will be. The question must be answered or the program will not continue. If the number given ultimately proves too small, the program has to terminate prematurely because the disk hole is full and there is no place to put the rest of the file. If the user tries to avoid this problem by giving an unrealistically large number as the final size, say, 1 GB, the editor may be unable to find such a large hole and announce that the file cannot be created. Of course, the user would be free to start the program again and say 500 MB this time, and so on until a suitable hole was located. Still, this scheme is not likely to lead to happy users.

However, there is one situation in which contiguous allocation is feasible and, in fact, still used: on CD-ROMs. Here all the file sizes are known in advance and will never change during subsequent use of the CD-ROM file system.

The situation with DVDs is a bit more complicated. In principle, a 90-min movie could be encoded as a single file of length about 4.5 GB, but the file system used, **UDF** (**Universal Disk Format**), uses a 30-bit number to represent file length, which limits files to 1 GB. As a consequence, DVD movies are generally stored as three or four 1-GB files, each of which is contiguous. These physical pieces of the single logical file (the movie) are called **extents**.

As we mentioned in Chap. 1, history often repeats itself in computer science as new generations of technology occur. Contiguous allocation was actually used on magnetic-disk file systems years ago due to its simplicity and high performance (user friendliness did not count for much then). Then the idea was dropped due to the nuisance of having to specify final file size at file-creation time. But with the advent of CD-ROMs, DVDs, Blu-rays, and other write-once optical media, suddenly contiguous files were a good idea again. It is thus important to study old systems and ideas that were conceptually clean and simple because they may be applicable to future systems in surprising ways.

Linked-List Allocation

The second method for storing files is to keep each one as a linked list of disk blocks, as shown in Fig. 4-11. The first word of each block is used as a pointer to the next one. The rest of the block is for data.

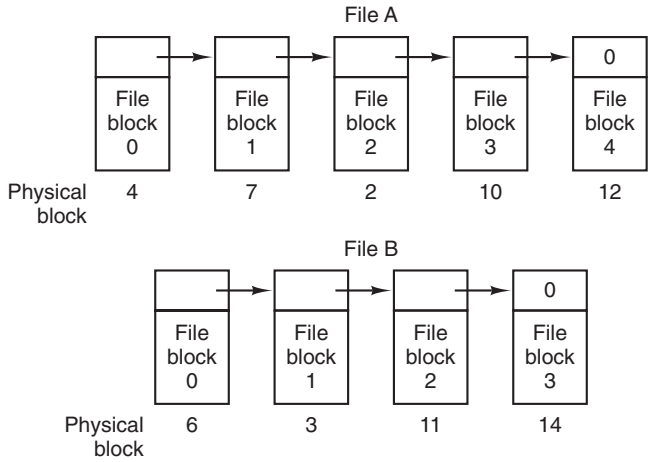


Figure 4-11. Storing a file as a linked list of disk blocks.

Unlike contiguous allocation, every disk block can be used in this method. No space is lost to disk fragmentation (except for internal fragmentation in the last block). Also, it is sufficient for the directory entry to merely store the disk address of the first block. The rest can be found starting there.

On the other hand, although reading a file sequentially is straightforward, random access is extremely slow. To get to block n, the operating system has to start at the beginning and read the $n - 1$ blocks prior to it, one at a time. Clearly, doing so many reads will be painfully slow.

Also, the amount of data storage in a block is no longer a power of two because the pointer takes up a few bytes. While not fatal, having a peculiar size is less efficient because many programs read and write in blocks whose size is a power of two. With the first few bytes of each block occupied by a pointer to the next block, reads of the full block size require acquiring and concatenating information from two disk blocks, which generates extra overhead due to the copying.

Linked-List Allocation Using a Table in Memory

Both disadvantages of the linked-list allocation can be eliminated by taking the pointer word from each disk block and putting it in a table in memory. Figure 4-12 shows what the table looks like for the example of Fig. 4-11. In both figures, we have two files. File *A* uses disk blocks 4, 7, 2, 10, and 12, in that order, and file *B* uses disk blocks 6, 3, 11, and 14, in that order. Using the table of Fig. 4-12, we can start with block 4 and follow the chain all the way to the end. The same can be done starting with block 6. Both chains are terminated with a special marker (e.g., −1) that is not a valid block number. Such a table in main memory is called a **FAT** (**File Allocation Table**).

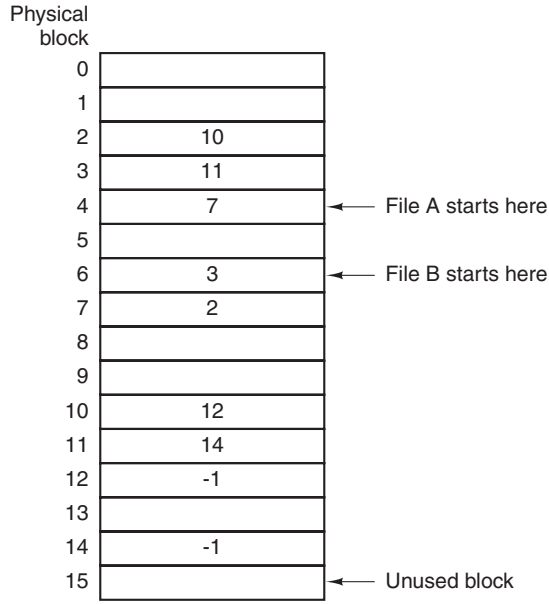


Figure 4-12. Linked-list allocation using a file-allocation table in main memory.

Using this organization, the entire block is available for data. Furthermore, random access is much easier. Although the chain must still be followed to find a given offset within the file, the chain is entirely in memory, so it can be followed without making any disk references. Like the previous method, it is sufficient for the directory entry to keep a single integer (the starting block number) and still be able to locate all the blocks, no matter how large the file is.

The primary disadvantage of this method is that the entire table must be in memory all the time to make it work. With a 1-TB disk and a 1-KB block size, the table needs 1 billion entries, one for each of the 1 billion disk blocks. Each entry has to be a minimum of 3 bytes. For speed in lookup, they should be 4 bytes. Thus the table will take up 3 GB or 2.4 GB of main memory all the time, depending on whether the system is optimized for space or time. Not wildly practical. Clearly the FAT idea does not scale well to large disks. It was the original MS-DOS file system and is still fully supported by all versions of Windows though.

I-nodes

Our last method for keeping track of which blocks belong to which file is to associate with each file a data structure called an **i-node** (**index-node**), which lists the attributes and disk addresses of the file's blocks. A simple example is depicted in Fig. 4-13. Given the i-node, it is then possible to find all the blocks of the file.

The big advantage of this scheme over linked files using an in-memory table is that the i-node need be in memory only when the corresponding file is open. If each i-node occupies n bytes and a maximum of k files may be open at once, the total memory occupied by the array holding the i-nodes for the open files is only kn bytes. Only this much space need be reserved in advance.

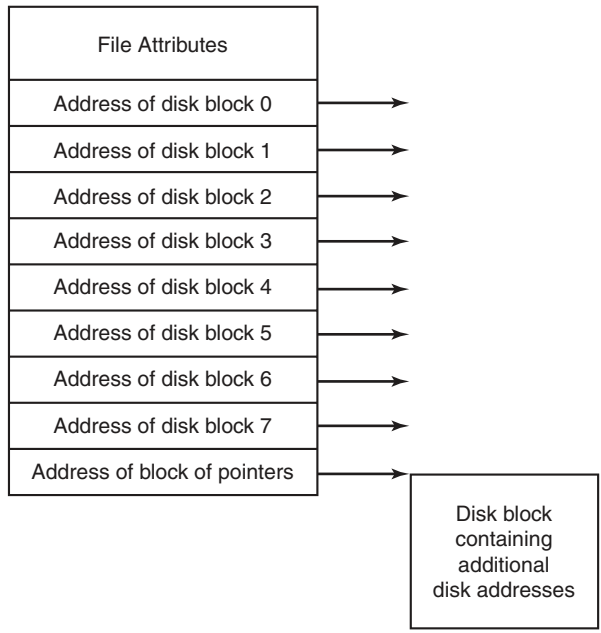


Figure 4-13. An example i-node.

This array is usually far smaller than the space occupied by the file table described in the previous section. The reason is simple. The table for holding the linked list of all disk blocks is proportional in size to the disk itself. If the disk has n blocks, the table needs n entries. As disks grow larger, this table grows linearly with them. In contrast, the i-node scheme requires an array in memory whose size is proportional to the maximum number of files that may be open at once. It does not matter if the disk is 100 GB, 1000 GB, or 10,000 GB.

One problem with i-nodes is that if each one has room for a fixed number of disk addresses, what happens when a file grows beyond this limit? One solution is to reserve the last disk address not for a data block, but instead for the address of a block containing more disk-block addresses, as shown in Fig. 4-13. Even more advanced would be two or more such blocks containing disk addresses or even disk blocks pointing to other disk blocks full of addresses. We will come back to i-nodes when studying UNIX in Chap. 10. Similarly, the Windows NTFS file system uses a similar idea, only with bigger i-nodes that can also contain small files.

4.3.3 Implementing Directories

Before a file can be read, it must be opened. When a file is opened, the operating system uses the path name supplied by the user to locate the directory entry on the disk. The directory entry provides the information needed to find the disk blocks. Depending on the system, this information may be the disk address of the entire file (with contiguous allocation), the number of the first block (both linked-list schemes), or the number of the i-node. In all cases, the main function of the directory system is to map the ASCII name of the file onto the information needed to locate the data.

A closely related issue is where the attributes should be stored. Every file system maintains various file attributes, such as each file's owner and creation time, and they must be stored somewhere. One obvious possibility is to store them directly in the directory entry. Some systems do precisely that. This option is shown in Fig. 4-14(a). In this simple design, a directory consists of a list of fixed-size entries, one per file, containing a (fixed-length) file name, a structure of the file attributes, and one or more disk addresses (up to some maximum) telling where the disk blocks are.

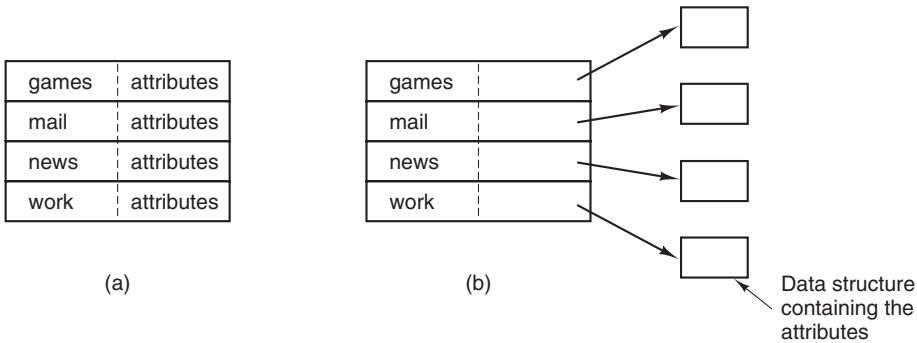


Figure 4-14. (a) A simple directory containing fixed-size entries with the disk addresses and attributes in the directory entry. (b) A directory in which each entry just refers to an i-node.

For systems that use i-nodes, another possibility for storing the attributes is in the i-nodes, rather than in the directory entries. In that case, the directory entry can be shorter: just a file name and an i-node number. This approach is illustrated in Fig. 4-14(b). As we shall see later, this method has some advantages over putting them in the directory entry.

So far we have made the assumption that files have short, fixed-length names. In MS-DOS files have a 1–8 character base name and an optional extension of 1–3 characters. In UNIX Version 7, file names were 1–14 characters, including any extensions. However, nearly all modern operating systems support longer, variable-length file names. How can these be implemented?

The simplest approach is to set a limit on file-name length, typically 255 characters, and then use one of the designs of Fig. 4-14 with 255 characters reserved for each file name. This approach is simple, but wastes a great deal of directory space, since few files have such long names. For efficiency reasons, a different structure is desirable.

One alternative is to give up the idea that all directory entries are the same size. With this method, each directory entry contains a fixed portion, typically starting with the length of the entry, and then followed by data with a fixed format, usually including the owner, creation time, protection information, and other attributes. This fixed-length header is followed by the actual file name, however long it may be, as shown in Fig. 4-15(a) in big-endian format (e.g., SPARC). In this example we have three files, *project-budget*, *personnel*, and *foo*. Each file name is terminated by a special character (usually 0), which is represented in the figure by a box with a cross in it. To allow each directory entry to begin on a word boundary, each file name is filled out to an integral number of words, shown by shaded boxes in the figure.

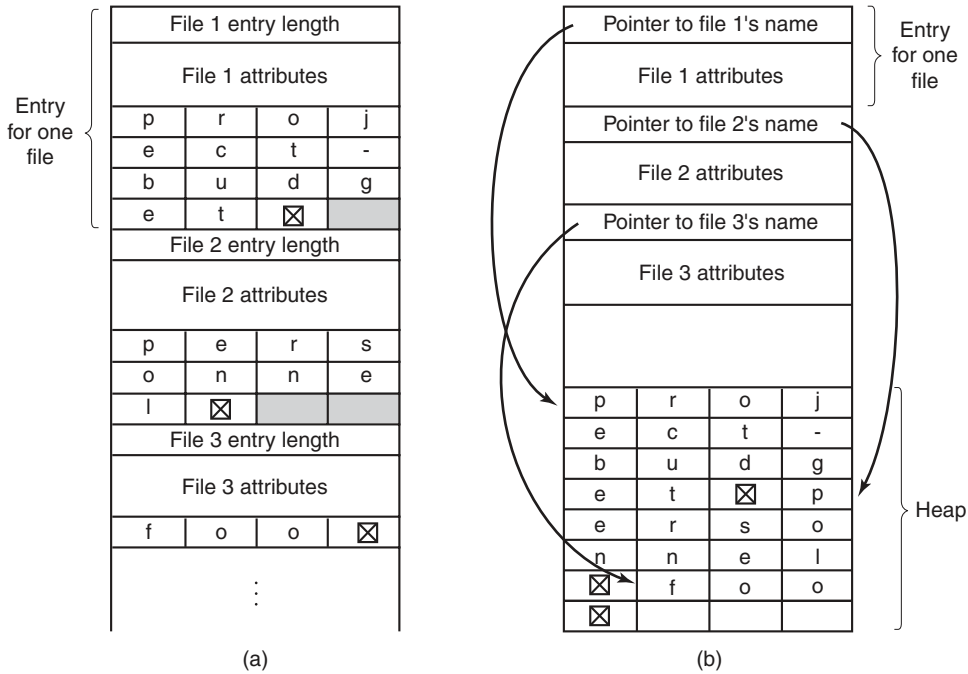


Figure 4-15. Two ways of handling long file names in a directory. (a) In-line. (b) In a heap.

A disadvantage of this method is that when a file is removed, a variable-sized gap is introduced into the directory into which the next file to be entered may not fit. This problem is essentially the same one we saw with contiguous disk files,

only now compacting the directory is feasible because it is entirely in memory. Another problem is that a single directory entry may span multiple pages, so a page fault may occur while reading a file name.

Another way to handle variable-length names is to make the directory entries themselves all fixed length and keep the file names together in a heap at the end of the directory, as shown in Fig. 4-15(b). This method has the advantage that when an entry is removed, the next file entered will always fit there. Of course, the heap must be managed and page faults can still occur while processing file names. One minor win here is that there is no longer any real need for file names to begin at word boundaries, so no filler characters are needed after file names in Fig. 4-15(b) as they are in Fig. 4-15(a).

In all of the designs so far, directories are searched linearly from beginning to end when a file name has to be looked up. For extremely long directories, linear searching can be slow. One way to speed up the search is to use a hash table in each directory. Call the size of the table n. To enter a file name, the name is hashed onto a value between 0 and $n - 1$, for example, by dividing it by n and taking the remainder. Alternatively, the words comprising the file name can be added up and this quantity divided by n, or something similar.

Either way, the table entry corresponding to the hash code is inspected. If it is unused, a pointer is placed there to the file entry. File entries follow the hash table. If that slot is already in use, a linked list is constructed, headed at the table entry and threading through all entries with the same hash value.

Looking up a file follows the same procedure. The file name is hashed to select a hash-table entry. All the entries on the chain headed at that slot are checked to see if the file name is present. If the name is not on the chain, the file is not present in the directory.

Using a hash table has the advantage of much faster lookup, but the disadvantage of more complex administration. It is only really a serious candidate in systems where it is expected that directories will routinely contain hundreds or thousands of files.

A different way to speed up searching large directories is to cache the results of searches. Before starting a search, a check is first made to see if the file name is in the cache. If so, it can be located immediately. Of course, caching only works if a relatively small number of files comprise the majority of the lookups.

4.3.4 Shared Files

When several users are working together on a project, they often need to share files. As a result, it is often convenient for a shared file to appear simultaneously in different directories belonging to different users. Figure 4-16 shows the file system of Fig. 4-7 again, only with one of *C*'s files now present in one of *B*'s directories as well. The connection between *B*'s directory and the shared file is called a

link. The file system itself is now a **Directed Acyclic Graph**, or **DAG**, rather than a tree. Having the file system be a DAG complicates maintenance, but such is life.

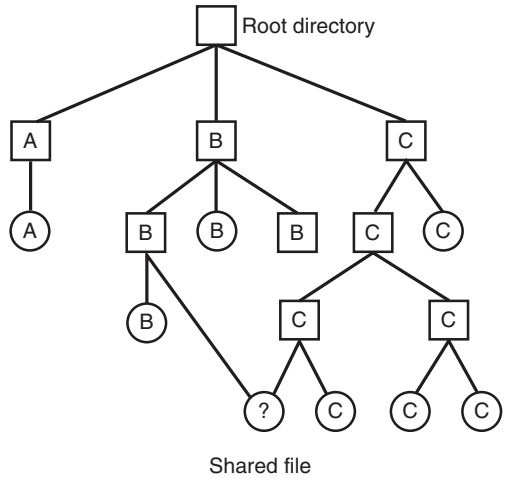


Figure 4-16. File system containing a shared file.

Sharing files is convenient, but it also introduces some problems. To start with, if directories really do contain disk addresses, then a copy of the disk addresses will have to be made in *B*'s directory when the file is linked. If either *B* or *C* subsequently appends to the file, the new blocks will be listed only in the directory of the user doing the append. The changes will not be visible to the other user, thus defeating the purpose of sharing.

This problem can be solved in two ways. In the first solution, disk blocks are not listed in directories, but in a little data structure associated with the file itself. The directories would then point just to the little data structure. This is the approach used in UNIX (where the little data structure is the i-node).

In the second solution, *B* links to one of *C*'s files by having the system create a new file, of type LINK, and entering that file in *B*'s directory. The new file contains just the path name of the file to which it is linked. When *B* reads from the linked file, the operating system sees that the file being read from is of type LINK, looks up the name of the file, and reads that file. This approach is called **symbolic linking**, to contrast it with traditional (hard) linking.

Each of these methods has its drawbacks. In the first method, at the moment that *B* links to the shared file, the i-node records the file's owner as *C*. Creating a link does not change the ownership (see Fig. 4-17), but it does increase the link count in the i-node, so the system knows how many directory entries currently point to the file.

If *C* subsequently tries to remove the file, the system is faced with a problem. If it removes the file and clears the i-node, *B* will have a directory entry pointing to

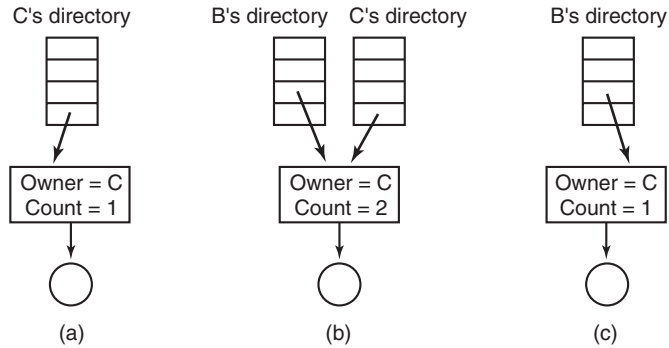


Figure 4-17. (a) Situation prior to linking. (b) After the link is created. (c) After the original owner removes the file.

an invalid i-node. If the i-node is later reassigned to another file, *B*'s link will point to the wrong file. The system can see from the count in the i-node that the file is still in use, but there is no easy way for it to find all the directory entries for the file, in order to erase them. Pointers to the directories cannot be stored in the i-node because there can be an unlimited number of directories.

The only thing to do is remove *C*'s directory entry, but leave the i-node intact, with count set to 1, as shown in Fig. 4-17(c). We now have a situation in which *B* is the only user having a directory entry for a file owned by *C*. If the system does accounting or has quotas, *C* will continue to be billed for the file until *B* decides to remove it, if ever, at which time the count goes to 0 and the file is deleted.

With symbolic links this problem does not arise because only the true owner has a pointer to the i-node. Users who have linked to the file just have path names, not i-node pointers. When the *owner* removes the file, it is destroyed. Subsequent attempts to use the file via a symbolic link will fail when the system is unable to locate the file. Removing a symbolic link does not affect the file at all.

The problem with symbolic links is the extra overhead required. The file containing the path must be read, then the path must be parsed and followed, component by component, until the i-node is reached. All of this activity may require a considerable number of extra disk accesses. Furthermore, an extra i-node is needed for each symbolic link, as is an extra disk block to store the path, although if the path name is short, the system could store it in the i-node itself, as a kind of optimization. Symbolic links have the advantage that they can be used to link to files on machines anywhere in the world, by simply providing the network address of the machine where the file resides in addition to its path on that machine.

There is also another problem introduced by links, symbolic or otherwise. When links are allowed, files can have two or more paths. Programs that start at a given directory and find all the files in that directory and its subdirectories will locate a linked file multiple times. For example, a program that dumps all the files

7

VIRTUALIZATION AND THE CLOUD

In some situations, an organization has a multicomputer but does not actually want it. A common example is where a company has an email server, a Web server, an FTP server, some e-commerce servers, and others. These all run on different computers in the same equipment rack, all connected by a high-speed network, in other words, a multicomputer. One reason all these servers run on separate machines may be that one machine cannot handle the load, but another is reliability: management simply does not trust the operating system to run 24 hours a day, 365 or 366 days a year, with no failures. By putting each service on a separate computer, if one of the servers crashes, at least the other ones are not affected. This is good for security also. Even if some malevolent intruder manages to compromise the Web server, he will not immediately have access to sensitive emails also—a property sometimes referred to as **sandboxing**. While isolation and fault tolerance are achieved this way, this solution is expensive and hard to manage because so many machines are involved.

Mind you, these are just two out of many reasons for keeping separate machines. For instance, organizations often depend on more than one operating system for their daily operations: a Web server on Linux, a mail server on Windows, an e-commerce server for customers running on OS X, and a few other services running on various flavors of UNIX. Again, this solution works, but cheap it is definitely not.

What to do? A possible (and popular) solution is to use virtual machine technology, which sounds very hip and modern, but the idea is old, dating back to the

1960s. Even so, the way we use it today is definitely new. The main idea is that a **VMM** (**Virtual Machine Monitor**) creates the illusion of multiple (virtual) machines on the same physical hardware. A VMM is also known as a **hypervisor**. As discussed in Sec. 1.7.5, we distinguish between type 1 hypervisors which run on the bare metal, and type 2 hypervisors that may make use of all the wonderful services and abstractions offered by an underlying operating system. Either way, **virtualization** allows a single computer to host multiple virtual machines, each potentially running a completely different operating system.

The advantage of this approach is that a failure in one virtual machine does not bring down any others. On a virtualized system, different servers can run on different virtual machines, thus maintaining the partial-failure model that a multicomputer has, but at a lower cost and with easier maintainability. Moreover, we can now run multiple different operating systems on the same hardware, benefit from virtual machine isolation in the face of attacks, and enjoy other good stuff.

Of course, consolidating servers like this is like putting all your eggs in one basket. If the server running all the virtual machines fails, the result is even more catastrophic than the crashing of a single dedicated server. The reason virtualization works, however, is that most service outages are due not to faulty hardware, but to ill-designed, unreliable, buggy and poorly configured software, emphatically including operating systems. With virtual machine technology, the only software running in the highest privilege mode is the hypervisor, which has two orders of magnitude fewer lines of code than a full operating system, and thus two orders of magnitude fewer bugs. A hypervisor is simpler than an operating system because it does only one thing: emulate multiple copies of the bare metal (most commonly the Intel x86 architecture).

Running software in virtual machines has other advantages in addition to strong isolation. One of them is that having fewer physical machines saves money on hardware and electricity and takes up less rack space. For a company such as Amazon or Microsoft, which may have hundreds of thousands of servers doing a huge variety of different tasks at each data center, reducing the physical demands on their data centers represents a huge cost savings. In fact, server companies frequently locate their data centers in the middle of nowhere—just to be close to, say, hydroelectric dams (and cheap energy). Virtualization also helps in trying out new ideas. Typically, in large companies, individual departments or groups think of an interesting idea and then go out and buy a server to implement it. If the idea catches on and hundreds or thousands of servers are needed, the corporate data center expands. It is often hard to move the software to existing machines because each application often needs a different version of the operating system, its own libraries, configuration files, and more. With virtual machines, each application can take its own environment with it.

Another advantage of virtual machines is that checkpointing and migrating virtual machines (e.g., for load balancing across multiple servers) is much easier than migrating processes running on a normal operating system. In the latter case, a fair

amount of critical state information about every process is kept in operating system tables, including information relating to open files, alarms, signal handlers, and more. When migrating a virtual machine, all that have to be moved are the memory and disk images, since all the operating system tables move, too.

Another use for virtual machines is to run legacy applications on operating systems (or operating system versions) no longer supported or which do not work on current hardware. These can run at the same time and on the same hardware as current applications. In fact, the ability to run at the same time applications that use different operating systems is a big argument in favor of virtual machines.

Yet another important use of virtual machines is for software development. A programmer who wants to make sure his software works on Windows 7, Windows 8, several versions of Linux, FreeBSD, OpenBSD, NetBSD, and OS X, among other systems no longer has to get a dozen computers and install different operating systems on all of them. Instead, he merely creates a dozen virtual machines on a single computer and installs a different operating system on each one. Of course, he could have partitioned the hard disk and installed a different operating system in each partition, but that approach is more difficult. First of all, standard PCs support only four primary disk partitions, no matter how big the disk is. Second, although a multiboot program could be installed in the boot block, it would be necessary to reboot the computer to work on a new operating system. With virtual machines, all of them can run at once, since they are really just glorified processes.

Perhaps the most important and buzzword-compliant use case for virtualization nowadays is found in the **cloud**. The key idea of a cloud is straightforward: outsource your computation or storage needs to a well-managed data center run by a company specializing in this and staffed by experts in the area. Because the data center typically belongs to someone else, you will probably have to pay for the use of the resources, but at least you will not have to worry about the physical machines, power, cooling, and maintenance. Because of the isolation offered by virtualizaton, cloud-providers can allow multiple clients, even competitors, to share a single physical machine. Each client gets a piece of the pie. At the risk of stretching the cloud metaphor, we mention that early critics maintained that the pie was only in the sky and that real organizations would not want to put their sensitive data and computations on someone else's resources. By now, however, virtualized machines in the cloud are used by countless organization for countless applications, and while it may not be for all organizations and all data, there is no doubt that cloud computing has been a success.

7.1 HISTORY

With all the hype surrounding virtualizaton in recent years, we sometimes forget that by Internet standards virtual machines are ancient. As early as the 1960s. IBM experimented with not just one but two independently developed hypervisors:

SIMMON and **CP-40**. While CP-40 was a research project, it was reimplemented as **CP-67** to form the control program of **CP/CMS**, a virtual machine operating system for the IBM System/360 Model 67. Later, it was reimplemented again and released as **VM/370** for the System/370 series in 1972. The System/370 line was replaced by IBM in the 1990s by the System/390. This was basically a name change since the underlying architecture remained the same for reasons of backward compatibility. Of course, the hardware technology was improved and the newer machines were bigger and faster than the older ones, but as far as virtualization was concerned, nothing changed. In 2000, IBM released the z-series, which supported 64-bit virtual address spaces but was otherwise backward compatible with the System/360. All of these systems supported virtualization decades before it became popular on the x86.

In 1974, two computer scientists at UCLA, Gerald Popek and Robert Goldberg, published a seminal paper ("Formal Requirements for Virtualizable Third Generation Architectures") that listed exactly what conditions a computer architecture should satisfy in order to support virtualization efficiently (Popek and Goldberg, 1974). It is impossible to write a chapter on virtualization without referring to their work and terminology. Famously, the well-known x86 architecture that also originated in the 1970s did not meet these requirements for decades. It was not the only one. Nearly every architecture since the mainframe also failed the test. The 1970s were very productive, seeing also the birth of UNIX, Ethernet, the Cray-1, Microsoft, and Apple—so, despite what your parents may say, the 1970s were not just about disco!

In fact, the real **Disco** revolution started in the 1990s, when researchers at Stanford University developed a new hypervisor by that name and went on to found **VMware**, a virtualization giant that offers type 1 and type 2 hypervisors and now rakes in billions of dollars in revenue (Bugnion et al., 1997, Bugnion et al., 2012). Incidentally, the distinction between "type 1" and "type 2" hypervisors is also from the seventies (Goldberg, 1972). VMware introduced its first virtualization solution for x86 in 1999. In its wake other products followed: **Xen**, **KVM**, **VirtualBox, Hyper-V, Parallels**, and many others. It seems the time was right for virtualization, even though the theory had been nailed down in 1974 and for decades IBM had been selling computers that supported—and heavily used—virtualization. In 1999, it became popular among the masses, but new it was not, despite the massive attention it suddenly gained.

7.2 REQUIREMENTS FOR VIRTUALIZATION

It is important that virtual machines act just like the real McCoy. In particular, it must be possible to boot them like real machines and install arbitrary operating systems on them, just as can be done on the real hardware. It is the task of the

hypervisor to provide this illusion and to do it efficiently. Indeed, hypervisors should score well in three dimensions:

1. **Safety**: the hypervisor should have full control of the virtualized resources.
2. **Fidelity**: the behavior of a program on a virtual machine should be identical to that of the same program running on bare hardware.
3. **Efficiency**: much of the code in the virtual machine should run without intervention by the hypervisor.

An unquestionably safe way to execute the instructions is to consider each instruction in turn in an **interpreter** (such as Bochs) and perform exactly what is needed for that instruction. Some instructions can be executed directly, but not too many. For instance, the interpreter may be able to execute an INC (increment) instruction simply as is, but instructions that are not safe to execute directly must be simulated by the interpreter. For instance, we cannot really allow the guest operating system to disable interrupts for the entire machine or modify the page-table mappings. The trick is to make the operating system on top of the hypervisor think that it has disabled interrupts, or changed the machine's page mappings. We will see how this is done later. For now, we just want to say that the interpreter may be safe, and if carefully implemented, perhaps even hi-fi, but the performance sucks. To also satisfy the performance criterion, we will see that VMMs try to execute most of the code directly.

Now let us turn to fidelity. Virtualization has long been a problem on the x86 architecture due to defects in the Intel 386 architecture that were slavishly carried forward into new CPUs for 20 years in the name of backward compatibility. In a nutshell, every CPU with kernel mode and user mode has a set of instructions that behave differently when executed in kernel mode than when executed in user mode. These include instructions that do I/O, change the MMU settings, and so on. Popek and Goldberg called these **sensitive instructions**. There is also a set of instructions that cause a trap if executed in user mode. Popek and Goldberg called these **privileged instructions**. Their paper stated for the first time that a machine is virtualizable only if the sensitive instructions are a subset of the privileged instructions. In simpler language, if you try to do something in user mode that you should not be doing in user mode, the hardware should trap. Unlike the IBM/370, which had this property, Intel's 386 did not. Quite a few sensitive 386 instructions were ignored if executed in user mode or executed with different behavior. For example, the POPF instruction replaces the flags register, which changes the bit that enables/disables interrupts. In user mode, this bit is simply not changed. As a consequence, the 386 and its successors could not be virtualized, so they could not support a hypervisor directly.

Actually, the situation is even worse than sketched. In addition to the problems with instructions that fail to trap in user mode, there are instructions that can read

sensitive state in user mode without causing a trap. For example, on x86 processors prior to 2005, a program can determine whether it is running in user mode or kernel mode by reading its code-segment selector. An operating system that did this and discovered that it was actually in user mode might make an incorrect decision based on this information.

This problem was finally solved when Intel and AMD introduced virtualization in their CPUs starting in 2005 (Uhlig, 2005). On the Intel CPUs it is called **VT** (**Virtualization Technology**); on the AMD CPUs it is called **SVM** (**Secure Virtual Machine**). We will use the term VT in a generic sense below. Both were inspired by the IBM VM/370 work, but they are slightly different. The basic idea is to create containers in which virtual machines can be run. When a guest operating system is started up in a container, it continues to run there until it causes an exception and traps to the hypervisor, for example, by executing an I/O instruction. The set of operations that trap is controlled by a hardware bitmap set by the hypervisor. With these extensions the classical **trap-and-emulate** virtual machine approach becomes possible.

The astute reader may have noticed an apparent contradiction in the description thus far. On the one hand, we have said that x86 was not virtualizable until the architecture extensions introduced in 2005. On the other hand, we saw that VMware launched its first x86 hypervisor in 1999. How can both be true at the same time? The answer is that the hypervisors before 2005 did not really run the original guest operating system. Rather, they *rewrote* part of the code on the fly to replace problematic instructions with safe code sequences that emulated the original instruction. Suppose, for instance, that the guest operating system performed a privileged I/O instruction, or modified one of the CPU's privileged control registers (like the CR3 register which contains a pointer to the page directory). It is important that the consequences of such instructions are limited to this virtual machine and do not affect other virtual machines, or the hypervisor itself. Thus, an unsafe I/O instruction was replaced by a trap that, after a safety check, performed an equivalent instruction and returned the result. Since we are rewriting, we can use the trick to replace instructions that are sensitive, but not privileged. Other instructions execute natively. The technique is known as **binary translation**; we will discuss it more detail in Sec. 7.4.

There is no need to rewrite all sensitive instructions. In particular, user processes on the guest can typically run without modification. If the instruction is nonprivileged but sensitive and behaves differently in user processes than in the kernel, that is fine. We are running it in userland anyway. For sensitive instructions that are privileged, we can resort to the classical trap-and-emulate, as usual. Of course, the VMM must ensure that it receives the corresponding traps. Typically, the VMM has a module that executes in the kernel and redirects the traps to its own handlers.

A different form of virtualization is known as **paravirtualization**. It is quite different from **full virtualization**, because it never even aims to present a virtual machine that looks just like the actual underlying hardware. Instead, it presents a

machine-like software interface that explicitly exposes the fact that it is a virtualized environment. For instance, it offers a set of **hypercalls**, which allow the guest to send explicit requests to the hypervisor (much as a system call offers kernel services to applications). Guests use hypercalls for privileged sensitive operations like updating the page tables, but because they do it explicitly in cooperation with the hypervisor, the overall system can be simpler and faster.

It should not come as a surprise that paravirtualization is nothing new either. IBM's VM operating system has offered such a facility, albeit under a different name, since 1972. The idea was revived by the Denali (Whitaker et al., 2002) and Xen (Barham et al., 2003) virtual machine monitors. Compared to full virtualization, the drawback of paravirtualization is that the guest has to be aware of the virtual machine API. Typically, this means it should be customized explicitly for the hypervisor.

Before we delve more deeply into type 1 and type 2 hypervisors, it is important to mention that not all virtualization technology tries to trick the guest into believing that it has the entire system. Sometimes, the aim is simply to allow a process to run that was originally written for a different operating system and/or architecture. We therefore distinguish between full system virtualization and **process-level virtualization**. While we focus on the former in the remainder of this chapter, process-level virtualization technology is used in practice also. Well-known examples include the WINE compatibility layer that allows Windows application to run on POSIX-compliant systems like Linux, BSD, and OS X, and the process-level version of the QEMU emulator that allows applications for one architecture to run on another.

7.3 TYPE 1 AND TYPE 2 HYPERVISORS

Goldberg (1972) distinguished between two approaches to virtualization. One kind of hypervisor, dubbed a **type 1 hypervisor** is illustrated in Fig. 7-1(a). Technically, it is like an operating system, since it is the only program running in the most privileged mode. Its job is to support multiple copies of the actual hardware, called **virtual machines**, similar to the processes a normal operating system runs.

In contrast, a **type 2 hypervisor**, shown in Fig. 7-1(b), is a different kind of animal. It is a program that relies on, say, Windows or Linux to allocate and schedule resources, very much like a regular process. Of course, the type 2 hypervisor still pretends to be a full computer with a CPU and various devices. Both types of hypervisor must execute the machine's instruction set in a safe manner. For instance, an operating system running on top of the hypervisor may change and even mess up its own page tables, but not those of others.

The operating system running on top of the hypervisor in both cases is called the **guest operating system**. For a type 2 hypervisor, the operating system running on the hardware is called the **host operating system**. The first type 2 hypervisor

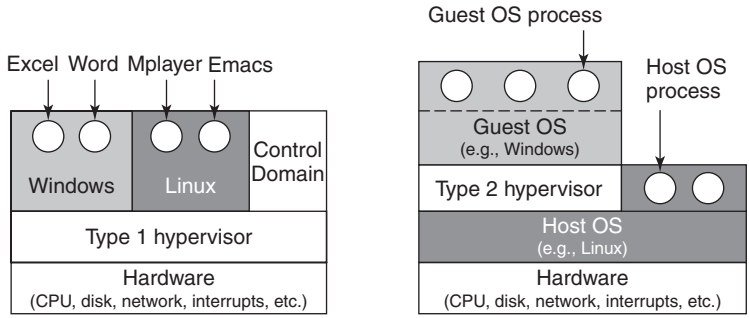


Figure 7-1. Location of type 1 and type 2 hypervisors.

on the x86 market was **VMware Workstation** (Bugnion et al., 2012). In this section, we introduce the general idea. A study of VMware follows in Sec. 7.12.

Type 2 hypervisors, sometimes referred to as **hosted hypervisors**, depend for much of their functionality on a host operating system such as Windows, Linux, or OS X. When it starts for the first time, it acts like a newly booted computer and expects to find a DVD, USB drive, or CD-ROM containing an operating system in the drive. This time, however, the drive could be a virtual device. For instance, it is possible to store the image as an ISO file on the hard drive of the host and have the hypervisor pretend it is reading from a proper DVD drive. It then installs the operating system to its **virtual disk** (again really just a Windows, Linux, or OS X file) by running the installation program found on the DVD. Once the guest operating system is installed on the virtual disk, it can be booted and run.

The various categories of virtualization we have discussed are summarized in the table of Fig. 7-2 for both type 1 and type 2 hypervisors. For each combination of hypervisor and kind of virtualization, some examples are given.

Virtualizaton method	Type 1 hypervisor	Type 2 hypervisor
Virtualization without HW support	ESX Server 1.0	VMware Workstation 1
Paravirtualization	Xen 1.0	
Virtualization with HW support	vSphere, Xen, Hyper-V	VMware Fusion, KVM, Parallels
Process virtualization		Wine

Figure 7-2. Examples of hypervisors. Type 1 hypervisors run on the bare metal whereas type 2 hypervisors use the services of an existing host operating system.

7.4 TECHNIQUES FOR EFFICIENT VIRTUALIZATION

Virtualizability and performance are important issues, so let us examine them more closely. Assume, for the moment, that we have a type 1 hypervisor supporting one virtual machine, as shown in Fig. 7-3. Like all type 1 hypervisors, it

runs on the bare metal. The virtual machine runs as a user process in user mode, and as such is not allowed to execute sensitive instructions (in the Popek-Goldberg sense). However, the virtual machine runs a guest operating system that thinks it is in kernel mode (although, of course, it is not). We will call this **virtual kernel mode**. The virtual machine also runs user processes, which think they are in user mode (and really are in user mode).

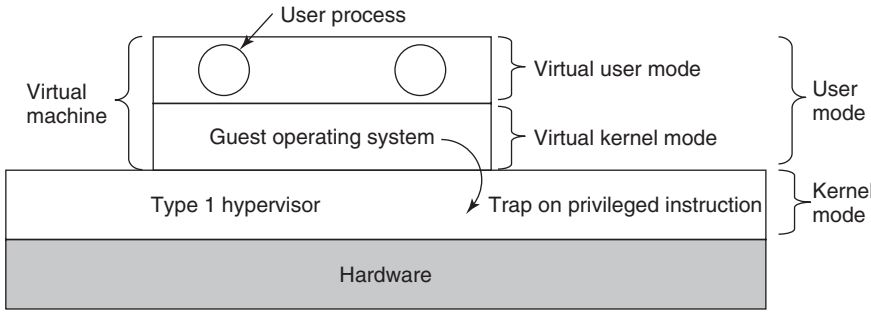


Figure 7-3. When the operating system in a virtual machine executes a kernel-only instruction, it traps to the hypervisor if virtualization technology is present.

What happens when the guest operating system (which thinks it is in kernel mode) executes an instruction that is allowed only when the CPU really is in kernel mode? Normally, on CPUs without VT, the instruction fails and the operating system crashes. On CPUs with VT, when the guest operating system executes a sensitive instruction, a trap to the hypervisor does occur, as illustrated in Fig. 7-3. The hypervisor can then inspect the instruction to see if it was issued by the guest operating system in the virtual machine or by a user program in the virtual machine. In the former case, it arranges for the instruction to be carried out; in the latter case, it emulates what the real hardware would do when confronted with a sensitive instruction executed in user mode.

7.4.1 Virtualizing the Unvirtualizable

Building a virtual machine system is relatively straightforward when VT is available, but what did people do before that? For instance, VMware released a hypervisor well before the arrival of the virtualization extensions on the x86. Again, the answer is that the software engineers who built such systems made clever use of **binary translation** and hardware features that *did* exist on the x86, such as the processor's **protection rings**.

For many years, the x86 has supported four protection modes or rings. Ring 3 is the least privileged. This is where normal user processes execute. In this ring, you cannot execute privileged instructions. Ring 0 is the most privileged ring that allows the execution of any instruction. In normal operation, the kernel runs in

ring 0. The remaining two rings are not used by any current operating system. In other words, hypervisors were free to use them as they pleased. As shown in Fig. 7-4, many virtualization solutions therefore kept the hypervisor in kernel mode (ring 0) and the applications in user mode (ring 3), but put the guest operating system in a layer of intermediate privilege (ring 1). As a result, the kernel is privileged relative to the user processes and any attempt to access kernel memory from a user program leads to an access violation. At the same time, the guest operating system's privileged instructions trap to the hypervisor. The hypervisor does some sanity checks and then performs the instructions on the guest's behalf.

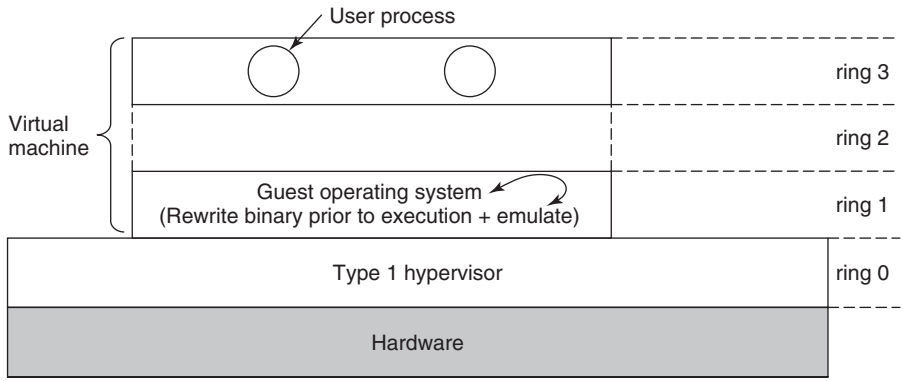


Figure 7-4. The binary translator rewrites the guest operating system running in ring 1, while the hypervisor runs in ring 0.

As for the sensitive instructions in the guest's kernel code: the hypervisor makes sure they no longer exist. To do so, it rewrites the code, one basic block at a time. A **basic block** is a short, straight-line sequence of instructions that ends with a branch. By definition, a basic block contains no jump, call, trap, return, or other instruction that alters the flow of control, except for the very last instruction which does precisely that. Just prior to executing a basic block, the hypervisor first scans it to see if it contains sensitive instructions (in the Popek and Goldberg sense), and if so, replaces them with a call to a hypervisor procedure that handles them. The branch on the last instruction is also replaced by a call into the hypervisor (to make sure it can repeat the procedure for the next basic block). Dynamic translation and emulation sound expensive, but typically are not. Translated blocks are cached, so no translation is needed in the future. Also, most code blocks do not contain sensitive or privileged instructions and thus can executes natively. In particular, as long as the hypervisor configures the hardware carefully (as is done, for instance, by VMware), the binary translator can ignore all user processes; they execute in nonprivileged mode anyway.

After a basic block has completed executing, control is returned to the hypervisor, which then locates its successor. If the successor has already been translated,

it can be executed immediately. Otherwise, it is first translated, cached, then executed. Eventually, most of the program will be in the cache and run at close to full speed. Various optimizations are used, for example, if a basic block ends by jumping to (or calling) another one, the final instruction can be replaced by a jump or call directly to the translated basic block, eliminating all overhead associated with finding the successor block. Again, there is no need to replace sensitive instructions in user programs; the hardware will just ignore them anyway.

On the other hand, it is common to perform binary translation on all the guest operating system code running in ring 1 and replace even the privileged sensitive instructions that, in principle, could be made to trap also. The reason is that traps are very expensive and binary translation leads to better performance.

So far we have described a type 1 hypervisor. Although type 2 hypervisors are conceptually different from type 1 hypervisors, they use, by and large, the same techniques. For instance, VMware ESX Server (a type 1 hypervisor first shipped in 2001) used exactly the same binary translation as the first VMware Workstation (a type 2 hypervisor released two years earlier).

However, to run the guest code natively and use exactly the same techniques requires the type 2 hypervisor to manipulate the hardware at the lowest level, which cannot be done from user space. For instance, it has to set the segment descriptors to exactly the right value for the guest code. For faithful virtualization, the guest operating system should also be tricked into thinking that it is the true and only king of the mountain with full control of all the machine's resources and with access to the entire address space (4 GB on 32-bit machines). When the king finds another king (the host kernel) squatting in its address space, the king will not be amused.

Unfortunately, this is exactly what happens when the guest runs as a user process on a regular operating system. For instance, in Linux a user process has access to just 3 GB of the 4-GB address space, as the remaining 1 GB is reserved for the kernel. Any access to the kernel memory leads to a trap. In principle, it is possible to take the trap and emulate the appropriate actions, but doing so is expensive and typically requires installing the appropriate trap handler in the host kernel. Another (obvious) way to solve the two-kings problem, is to reconfigure the system to remove the host operating system and actually give the guest the entire address space. However, doing so is clearly not possible from user space either.

Likewise, the hypervisor needs to handle the interrupts to do the right thing, for instance when the disk sends an interrupt or a page fault occurs. Also, if the hypervisor wants to use trap-and-emulate for privileged instructions, it needs to receive the traps. Again, installing trap/interrupt handlers in the kernel is not possible for user processes.

Most modern type 2 hypervisors therefore have a kernel module operating in ring 0 that allows them to manipulate the hardware with privileged instructions. Of course, manipulating the hardware at the lowest level and giving the guest access to the full address space is all well and good, but at some point the hypervisor

needs to clean it up and restore the original processor context. Suppose, for instance, that the guest is running when an interrupt arrives from an external device. Since a type 2 hypervisor depends on the host's device drivers to handle the interrupt, it needs to reconfigure the hardware completely to run the host operating system code. When the device driver runs, it finds everything just as it expected it to be. The hypervisor behaves just like teenagers throwing a party while their parents are away. It is okay to rearrange the furniture completely, as long as they put it back exactly as they found it before the parents come home. Going from a hardware configuration for the host kernel to a configuration for the guest operating system is known as a **world switch**. We will discuss it in detail when we discuss VMware in Sec. 7.12.

It should now be clear why these hypervisors work, even on unvirtualizable hardware: sensitive instructions in the guest kernel are replaced by calls to procedures that emulate these instructions. No sensitive instructions issued by the guest operating system are ever executed directly by the true hardware. They are turned into calls to the hypervisor, which then emulates them.

7.4.2 The Cost of Virtualization

One might naively expect that CPUs with VT would greatly outperform software techniques that resort to translation, but measurements show a mixed picture (Adams and Agesen, 2006). It turns out that the trap-and-emulate approach used by VT hardware generates a lot of traps, and traps are very expensive on modern hardware because they ruin CPU caches, TLBs, and branch prediction tables internal to the CPU. In contrast, when sensitive instructions are replaced by calls to hypervisor procedures within the executing process, none of this context-switching overhead is incurred. As Adams and Agesen show, depending on the workload, sometimes software beats hardware. For this reason, some type 1 (and type 2) hypervisors do binary translation for performance reasons, even though the software will execute correctly without it.

With binary translation, the translated code itself may be either slower or faster than the original code. Suppose, for instance, that the guest operating system disables hardware interrupts using the CLI instruction ("clear interrupts"). Depending on the architecture, this instruction can be very slow, taking many tens of cycles on certain CPUs with deep pipelines and out-of-order execution. It should be clear by now that the guest's wanting to turn off interrupts does not mean the hypervisor should really turn them off and affect the entire machine. Thus, the hypervisor must turn them off for the guest without really turning them off. To do so, it may keep track of a dedicated **IF** (**Interrupt Flag**) in the virtual CPU data structure it maintains for each guest (making sure the virtual machine does not get any interrupts until the interrupts are turned off again). Every occurrence of CLI in the guest will be replaced by something like "VirtualCPU.IF = 0", which is a very cheap move

instruction that may take as little as one to three cycles. Thus, the translated code is faster. Still, with modern VT hardware, usually the hardware beats the software.

On the other hand, if the guest operating system modifies its page tables, this is very costly. The problem is that each guest operating system on a virtual machine thinks it "owns" the machine and is at liberty to map any virtual page to any physical page in memory. However, if one virtual machine wants to use a physical page that is already in use by another virtual machine (or the hypervisor), something has to give. We will see in Section 7.6 that the solution is to add an extra level of page tables to map "guest physical pages" to the actual physical pages on the host. Not surprisingly, mucking around with multiple levels of page tables is not cheap.

7.5 ARE HYPERVISORS MICROKERNELS DONE RIGHT?

Both type 1 and type 2 hypervisors work with unmodified guest operating systems, but have to jump through hoops to get good performance. We have seen that **paravirtualization** takes a different approach by modifying the source code of the guest operating system instead. Rather than performing sensitive instructions, the paravirtualized guest executes **hypercalls**. In effect the guest operating system is acting like a user program making system calls to the operating system (the hypervisor). When this route is taken, the hypervisor must define an interface consisting of a set of procedure calls that guest operating systems can use. This set of calls forms what is effectively an **API** (**Application Programming Interface**) even though it is an interface for use by guest operating systems, not application programs.

Going one step further, by removing all the sensitive instructions from the operating system and just having it make hypercalls to get system services like I/O, we have turned the hypervisor into a microkernel, like that of Fig. 1-26. The idea, explored in paravirtualization, is that emulating peculiar hardware instructions is an unpleasant and time-consuming task. It requires a call into the hypervisor and then emulating the exact semantics of a complicated instruction. It is far better just to have the guest operating system call the hypervisor (or microkernel) to do I/O, and so on.

Indeed, some researchers have argued that we should perhaps consider hypervisors as "microkernels done right" (Hand et al., 2005). The first thing to mention is that this is a highly controversial topic and some researchers have vocally opposed the notion, arguing that the difference between the two is not fundamental to begin with (Heiser et al., 2006). Others suggest that compared to microkernels, hypervisors may not even be that well suited for building secure systems, and advocate that they be extended with kernel functionality like message passing and memory sharing (Hohmuth et al., 2004). Finally, some researchers argue that perhaps hypervisors are not even "operating systems research done right" (Roscoe et al., 2007). Since nobody said anything about operating system textbooks done right

(or wrong)—yet—we think we do right by exploring the similarity between hypervisors and microkernels a bit more.

The main reason the first hypervisors emulated the complete machine was the lack of availability of source code for the guest operating system (e.g., for Windows) or the vast number of variants (e.g., for Linux). Perhaps in the future the hypervisor/microkernel API will be standardized, and subsequent operating systems will be designed to call it instead of using sensitive instructions. Doing so would make virtual machine technology easier to support and use.

The difference between true virtualization and paravirtualization is illustrated in Fig. 7-5. Here we have two virtual machines being supported on VT hardware. On the left is an unmodified version of Windows as the guest operating system. When a sensitive instruction is executed, the hardware causes a trap to the hypervisor, which then emulates it and returns. On the right is a version of Linux modified so that it no longer contains any sensitive instructions. Instead, when it needs to do I/O or change critical internal registers (such as the one pointing to the page tables), it makes a hypervisor call to get the work done, just like an application program making a system call in standard Linux.

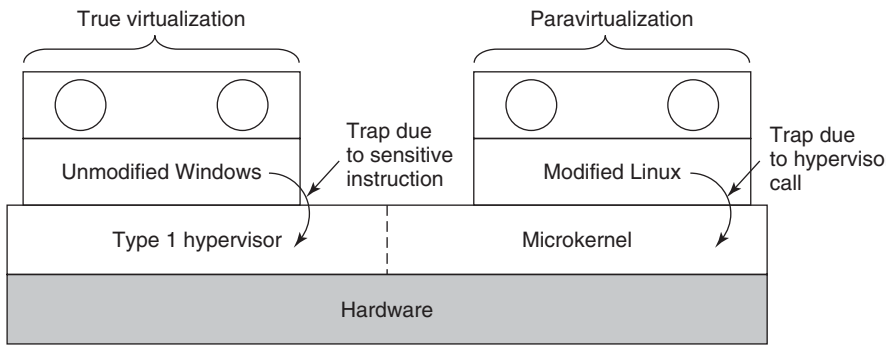


Figure 7-5. True virtualization and paravirtualization

In Fig. 7-5 we have shown the hypervisor as being divided into two parts separated by a dashed line. In reality, only one program is running on the hardware. One part of it is responsible for interpreting trapped sensitive instructions, in this case, from Windows. The other part of it just carries out hypercalls. In the figure the latter part is labeled "microkernel." If the hypervisor is intended to run only paravirtualized guest operating systems, there is no need for the emulation of sensitive instructions and we have a true microkernel, which just provides very basic services such as process dispatching and managing the MMU. The boundary between a type 1 hypervisor and a microkernel is vague already and will get even less clear as hypervisors begin acquiring more and more functionality and hypercalls, as seems likely. Again, this subject is controversial, but it is increasingly clear that the program running in kernel mode on the bare hardware should be small and reliable and consist of thousands, not millions, of lines of code.

Paravirtualizing the guest operating system raises a number of issues. First, if the sensitive instructions are replaced with calls to the hypervisor, how can the operating system run on the native hardware? After all, the hardware does not understand these hypercalls. And second, what if there are multiple hypervisors available in the marketplace, such as VMware, the open source Xen originally from the University of Cambridge, and Microsoft's Hyper-V, all with somewhat different hypervisor APIs? How can the kernel be modified to run on all of them?

Amsden et al. (2006) have proposed a solution. In their model, the kernel is modified to call special procedures whenever it needs to do something sensitive. Together these procedures, called the **VMI** (**Virtual Machine Interface**), form a low-level layer that interfaces with the hardware or hypervisor. These procedures are designed to be generic and not tied to any specific hardware platform or to any particular hypervisor.

An example of this technique is given in Fig. 7-6 for a paravirtualized version of Linux they call VMI Linux (VMIL). When VMI Linux runs on the bare hardware, it has to be linked with a library that issues the actual (sensitive) instruction needed to do the work, as shown in Fig. 7-6(a). When running on a hypervisor, say VMware or Xen, the guest operating system is linked with different libraries that make the appropriate (and different) hypercalls to the underlying hypervisor. In this way, the core of the operating system remains portable yet is hypervisor friendly and still efficient.

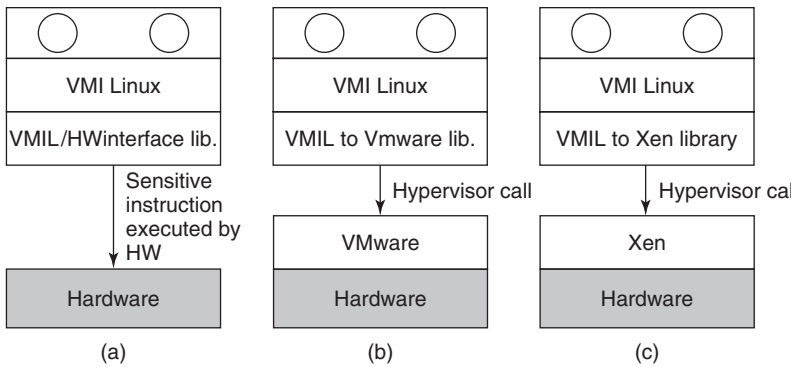


Figure 7-6. VMI Linux running on (a) the bare hardware, (b) VMware, (c) Xen

Other proposals for a virtual machine interface have also been made. Another popular one is called **paravirt ops**. The idea is conceptually similar to what we described above, but different in the details. Essentially, a group of Linux vendors that included companies like IBM, VMware, Xen, and Red Hat advocated a hypervisor-agnostic interface for Linux. The interface, included in the mainline kernel from version 2.6.23 onward, allows the kernel to talk to whatever hypervisor is managing the physical hardware.

7.6 MEMORY VIRTUALIZATION

So far we have addressed the issue of how to virtualize the CPU. But a computer system has more than just a CPU. It also has memory and I/O devices. They have to be virtualized, too. Let us see how that is done.

Modern operating systems nearly all support virtual memory, which is basically a mapping of pages in the virtual address space onto pages of physical memory. This mapping is defined by (multilevel) page tables. Typically the mapping is set in motion by having the operating system set a control register in the CPU that points to the top-level page table. Virtualization greatly complicates memory management. In fact, it took hardware manufacturers two tries to get it right.

Suppose, for example, a virtual machine is running, and the guest operating system in it decides to map its virtual pages 7, 4, and 3 onto physical pages 10, 11, and 12, respectively. It builds page tables containing this mapping and loads a hardware register to point to the top-level page table. This instruction is sensitive. On a VT CPU, it will trap; with dynamic translation it will cause a call to a hypervisor procedure; on a paravirtualized operating system, it will generate a hypercall. For simplicity, let us assume it traps into a type 1 hypervisor, but the problem is the same in all three cases.

What does the hypervisor do now? One solution is to actually allocate physical pages 10, 11, and 12 to this virtual machine and set up the actual page tables to map the virtual machine's virtual pages 7, 4, and 3 to use them. So far, so good.

Now suppose a second virtual machine starts and maps its virtual pages 4, 5, and 6 onto physical pages 10, 11, and 12 and loads the control register to point to its page tables. The hypervisor catches the trap, but what should it do? It cannot use this mapping because physical pages 10, 11, and 12 are already in use. It can find some free pages, say 20, 21, and 22, and use them, but it first has to create new page tables mapping the virtual pages 4, 5, and 6 of virtual machine 2 onto 20, 21, and 22. If another virtual machine starts and tries to use physical pages 10, 11, and 12, it has to create a mapping for them. In general, for each virtual machine the hypervisor needs to create a **shadow page table** that maps the virtual pages used by the virtual machine onto the actual pages the hypervisor gave it.

Worse yet, every time the guest operating system changes its page tables, the hypervisor must change the shadow page tables as well. For example, if the guest OS remaps virtual page 7 onto what it sees as physical page 200 (instead of 10), the hypervisor has to know about this change. The trouble is that the guest operating system can change its page tables by just writing to memory. No sensitive operations are required, so the hypervisor does not even know about the change and certainly cannot update the shadow page tables used by the actual hardware.

A possible (but clumsy) solution is for the hypervisor to keep track of which page in the guest's virtual memory contains the top-level page table. It can get this information the first time the guest attempts to load the hardware register that points to it because this instruction is sensitive and traps. The hypervisor can create

a shadow page table at this point and also map the top-level page table and the page tables it points to as read only. A subsequent attempts by the guest operating system to modify any of them will cause a page fault and thus give control to the hypervisor, which can analyze the instruction stream, figure out what the guest OS is trying to do, and update the shadow page tables accordingly. It is not pretty, but it is doable in principle.

Another, equally clumsy, solution is to do exactly the opposite. In this case, the hypervisor simply allows the guest to add new mappings to its page tables at will. As this is happening, nothing changes in the shadow page tables. In fact, the hypervisor is not even aware of it. However, as soon as the guest tries to access any of the new pages, a fault will occur and control reverts to the hypervisor. The hypervisor inspects the guest's page tables to see if there is a mapping that it should add, and if so, adds it and reexecutes the faulting instruction. What if the guest removes a mapping from its page tables? Clearly, the hypervisor cannot wait for a page fault to happen, because it will not happen. Removing a mapping from a page table happens by way of the INVLPG instruction (which is really intended to invalidate a TLB entry). The hypervisor therefore intercepts this instruction and removes the mapping from the shadow page table also. Again, not pretty, but it works.

Both of these techniques incur many page faults, and page faults are expensive. We typically distinguish between "normal" page faults that are caused by guest programs that access a page that has been paged out of RAM, and page faults that are related to ensuring the shadow page tables and the guest's page tables are in sync. The former are known as **guest-induced page faults**, and while they are intercepted by the hypervisor, they must be reinjected into the guest. This is not cheap at all. The latter are known as **hypervisor-induced page faults** and they are handled by updating the shadow page tables.

Page faults are always expensive, but especially so in virtualized environments, because they lead to so-called VM exits. A **VM exit** is a situation in which the hypervisor regains control. Consider what the CPU needs to do for such a VM exit. First, it records the cause of the VM exit, so the hypervisor knows what to do. It also records the address of the guest instruction that caused the exit. Next, a context switch is done, which includes saving all the registers. Then, it loads the hypervisor's processor state. Only then can the hypervisor start handling the page fault, which was expensive to begin with. Oh, and when it is all done, it should reverse these steps. The whole process may take tens of thousands of cycles, or more. No wonder people bend over backward to reduce the number of exits.

In a paravirtualized operating system, the situation is different. Here the paravirtualized OS in the guest knows that when it is finished changing some process' page table, it had better inform the hypervisor. Consequently, it first changes the page table completely, then issues a hypervisor call telling the hypervisor about the new page table. Thus, instead of a protection fault on every update to the page table, there is one hypercall when the whole thing has been updated, obviously a more efficient way to do business.

Hardware Support for Nested Page Tables

The cost of handling shadow page tables led chip makers to add hardware support for **nested page tables**. Nested page tables is the term used by AMD. Intel refers to them as **EPT** (**Extended Page Tables**). They are similar and aim to remove most of the overhead by handling the additional page-table manipulation all in hardware, all without any traps. Interestingly, the first virtualization extensions in Intel's x86 hardware did not include support for memory virtualization at all. While these VT-extended processors removed many bottlenecks concerning CPU virtualization, poking around in page tables was as expensive as ever. It took a few years for AMD and Intel to produce the hardware to virtualize memory efficiently.

Recall that even without virtualization, the operating system maintains a mapping between the virtual pages and the physical page. The hardware "walks" these page tables to find the physical address that corresponds to a virtual address. Adding more virtual machines simply adds an extra mapping. As an example, suppose we need to translate a virtual address of a Linux process running on a type 1 hypervisor like Xen or VMware ESX Server to a physical address. In addition to the **guest virtual addresses**, we now also have **guest physical addresses** and subsequently **host physical addresses** (sometimes referred to as **machine physical addresses**). We have seen that without EPT, the hypervisor is responsible for maintaining the shadow page tables explicitly. With EPT, the hypervisor still has an additional set of page tables, but now the CPU is able to handle much of the intermediate level in hardware also. In our example, the hardware first walks the "regular" page tables to translate the guest virtual address to a guest physical address, just as it would do without virtualization. The difference is that it also walks the extended (or nested) page tables without software intervention to find the host physical address, and it needs to do this every time a guest physical address is accessed. The translation is illustrated in Fig. 7-7.

Unfortunately, the hardware may need to walk the nested page tables more frequently then you might think. Let us suppose that the guest virtual address was not cached and requires a full page-table lookup. Every level in paging hierarchy incurs a lookup in the nested page tables. In other words, the number of memory references grows quadratically with the depth of the hierarchy. Even so, EPT dramatically reduces the number of VM exits. Hypervisors no longer need to map the guest's page table read only and can do away with shadow page-table handling. Better still, when switching virtual machines, it just changes this mapping, the same way an operating system changes the mapping when switching processes.

Reclaiming Memory

Having all these virtual machines on the same physical hardware all with their own memory pages and all thinking they are the king of the mountain is great—until we need the memory back. This is particularly important in the event of

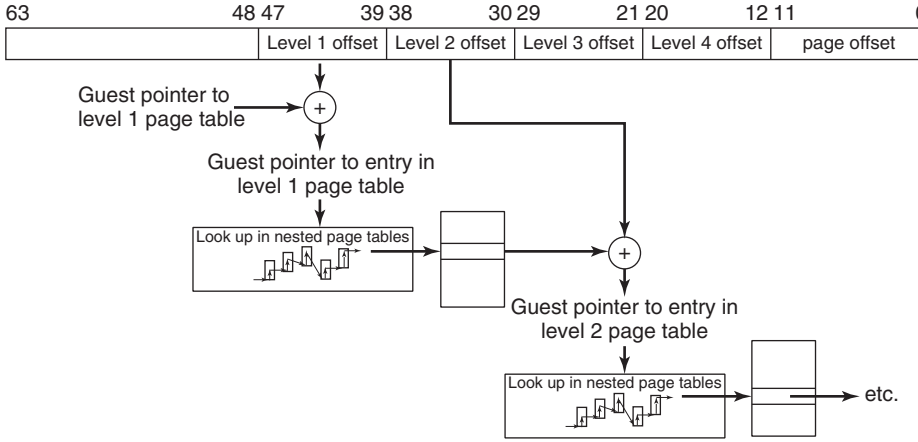


Figure 7-7. Extended/nested page tables are walked every time a guest physical address is accessed—including the accesses for each level of the guest's page tables.

overcommitment of memory, where the hypervisor pretends that the total amount of memory for all virtual machines combined is more than the total amount of physical memory present on the system. In general, this is a good idea, because it allows the hypervisor to admit more and more beefy virtual machines at the same time. For instance, on a machine with 32 GB of memory, it may run three virtual machines each thinking it has 16 GB of memory. Clearly, this does not fit. However, perhaps the three machines do not really need the maximum amount of physical memory at the same time. Or perhaps they share pages that have the same content (such as the Linux kernel) in different virtual machines in an optimization known as **deduplication**. In that case, the three virtual machines use a total amount of memory that is less than 3 times 16 GB. We will discuss deduplication later; for the moment the point is that what looks like a good distribution now, may be a poor distribution as the workloads change. Maybe virtual machine 1 needs more memory, while virtual machine 2 could do with fewer pages. In that case, it would be nice if the hypervisor could transfer resources from one virtual machine to another and make the system as a whole benefit. The question is, how can we take away memory pages safely if that memory is given to a virtual machine already?

In principle, we could use yet another level of paging. In case of memory shortage, the hypervisor would then page out some of the virtual machine's pages, just as an operating system may page out some of an application's pages. The drawback of this approach is that the hypervisor should do this, and the hypervisor has no clue about which pages are the most valuable to the guest. It is very likely to page out the wrong ones. Even if it does pick the right pages to swap (i.e., the pages that the guest OS would also have picked), there is still more trouble ahead.

For instance, suppose that the hypervisor pages out a page *P*. A little later, the guest OS also decides to page out this page to disk. Unfortunately, the hypervisor's swap space and the guest's swap space are not the same. In other words, the hypervisor must first page the contents back into memory, only to see the guest write it back out to disk immediately. Not very efficient.

A common solution is to use a trick known as **ballooning**, where a small balloon module is loaded in each VM as a pseudo device driver that talks to the hypervisor. The balloon module may inflate at the hypervisor's request by allocating more and more pinned pages, and deflate by deallocating these pages. As the balloon inflates, memory scarcity in the guest increases. The guest operating system will respond by paging out what it believes are the least valuable pages—which is just what we wanted. Conversely, as the balloon deflates, more memory becomes available for the guest to allocate. In other words, the hypervisor tricks the operating system into making tough decisions for it. In politics, this is known as passing the buck (or the euro, pound, yen, etc.).

7.7 I/O VIRTUALIZATION

Having looked at CPU and memory virtualization, we next examine I/O virtualization. The guest operating system will typically start out probing the hardware to find out what kinds of I/O devices are attached. These probes will trap to the hypervisor. What should the hypervisor do? One approach is for it to report back that the disks, printers, and so on are the ones that the hardware actually has. The guest will then load device drivers for these devices and try to use them. When the device drivers try to do actual I/O, they will read and write the device's hardware device registers. These instructions are sensitive and will trap to the hypervisor, which could then copy the needed values to and from the hardware registers, as needed.

But here, too, we have a problem. Each guest OS could think it owns an entire disk partition, and there may be many more virtual machines (hundreds) than there are actual disk partitions. The usual solution is for the hypervisor to create a file or region on the actual disk for each virtual machine's physical disk. Since the guest OS is trying to control a disk that the real hardware has (and which the hypervisor understands), it can convert the block number being accessed into an offset into the file or disk region being used for storage and do the I/O.

It is also possible for the disk that the guest is using to be different from the real one. For example, if the actual disk is some brand-new high-performance disk (or RAID) with a new interface, the hypervisor could advertise to the guest OS that it has a plain old IDE disk and let the guest OS install an IDE disk driver. When this driver issues IDE disk commands, the hypervisor converts them into commands to drive the new disk. This strategy can be used to upgrade the hardware without changing the software. In fact, this ability of virtual machines to remap

hardware devices was one of the reasons VM/370 became popular: companies wanted to buy new and faster hardware but did not want to change their software. Virtual machine technology made this possible.

Another interesting trend related to I/O is that the hypervisor can take the role of a virtual switch. In this case, each virtual machine has a MAC address and the hypevisor switches frames from one virtual machine to another—just like an Ethernet switch would do. Virtual switches have several advantages. For instance, it is very easy to reconfigure them. Also, it is possible to augment the switch with additional functionality, for instance for additional security.

I/O MMUs

Another I/O problem that must be solved somehow is the use of DMA, which uses absolute memory addresses. As might be expected, the hypervisor has to intervene here and remap the addresses before the DMA starts. However, hardware already exists with an **I/O MMU**, which virtualizes the I/O the same way the MMU virtualizes the memory. I/O MMU exists in different forms and shapes for many processor architectures. Even if we limit ourselves to the x86, Intel and AMD have slightly different technology. Still, the idea is the same. This hardware eliminates the DMA problem.

Just like regular MMUs, the I/O MMU uses page tables to map a memory address that a device wants to use (the device address) to a physical address. In a virtual environment, the hypervisor can set up the page tables in such a way that a device performing DMA will not trample over memory that does not belong to the virtual machine on whose behalf it is working.

I/O MMUs offer different advantages when dealing with a device in a virtualized world. **Device pass through** allows the physical device to be directly assigned to a particular virtual machine. In general, it would be ideal if device address space were exactly the same as the guest's physical address space. However, this is unlikely—unless you have an I/O MMU. The MMU allows the addresses to remapped transparently, and both the device and the virtual machine are blissfully unaware of the address translation that takes place under the hood.

Device isolation ensures that a device assigned to a virtual machine can directly access that virtual machine without jeopardizing the integrity of the other guests. In other words, the I/O MMU prevents rogue DMA traffic, just as a normal MMU prevents rogue memory accesses from processes—in both cases accesses to unmapped pages result in faults.

DMA and addresses are not the whole I/O story, unfortunately. For completeness, we also need to virtualize interrupts, so that the interrupt generated by a device arrives at the right virtual machine, with the right interrupt number. Modern I/O MMUs therefore support **interrupt remapping**. Say, a device sends a message signaled interrupt with number 1. This message first hits the I/O MMU that will use the interrupt remapping table to translate to a new message destined for

the CPU that currently runs the virtual machine and with the vector number that the VM expects (e.g., 66).

Finally, having an I/O MMU also helps 32-bit devices access memory above 4 GB. Normally, such devices are unable to access (e.g., DMA to) addresses beyond 4 GB, but the I/O MMU can easily remap the device's lower addresses to any address in the physical larger address space.

Device Domains

A different approach to handling I/O is to dedicate one of the virtual machines to run a standard operating system and reflect all I/O calls from the other ones to it. This approach is enhanced when paravirtualization is used, so the command being issued to the hypervisor actually says what the guest OS wants (e.g., read block 1403 from disk 1) rather than being a series of commands writing to device registers, in which case the hypervisor has to play Sherlock Holmes and figure out what it is trying to do. Xen uses this approach to I/O, with the virtual machine that does I/O called **domain 0**.

I/O virtualization is an area in which type 2 hypervisors have a practical advantage over type 1 hypervisors: the host operating system contains the device drivers for all the weird and wonderful I/O devices attached to the computer. When an application program attempts to access a strange I/O device, the translated code can call the existing device driver to get the work done. With a type 1 hypervisor, the hypervisor must either contain the driver itself, or make a call to a driver in domain 0, which is somewhat similar to a host operating system. As virtual machine technology matures, future hardware is likely to allow application programs to access the hardware directly in a secure way, meaning that device drivers can be linked directly with application code or put in separate user-mode servers (as in MINIX3), thereby eliminating the problem.

Single Root I/O Virtualization

Directly assigning a device to a virtual machine is not very scalable. With four physical networks you can support no more than four virtual machines that way. For eight virtual machines you need eight network cards, and to run 128 virtual machines—well, let's just say that it may be hard to find your computer buried under all those network cables.

Sharing devices among multiple hypervisors in software is possible, but often not optimal because an emulation layer (or device domain) interposes itself between hardware and the drivers and the guest operating systems. The emulated device frequently does not implement all the advanced functions supported by the hardware. Ideally, the virtualization technology would offer the equivalence of device pass through of a single device to multiple hypervisors, without any overhead. Virtualizing a single device to trick every virtual machine into believing that it has

exclusive access to its own device is much easier if the hardware actually does the virtualization for you. On PCIe, this is known as single root I/O virtualization.

Single root I/O virtualization (**SR-IOV**) allows us to bypass the hypervisor's involvement in the communication between the driver and the device. Devices that support SR-IOV provide an independent memory space, interrupts and DMA streams to each virtual machine that uses it (Intel, 2011). The device appears as multiple separate devices and each can be configured by separate virtual machines. For instance, each will have a separate base address register and address space. A virtual machine maps one of these memory areas (used for instance to configure the device) into its address space.

SR-IOV provides access to the device in two flavors: **PF** (**Physical Functions**) and (**Virtual Functions**). PFs are full PCIe functions and allow the device to be configured in whatever way the administrator sees fit. Physical functions are not accessible to guest operating systems. VFs are lightweight PCIe functions that do not offer such configuration options. They are ideally suited for virtual machines. In summary, SR-IOV allows devices to be virtualized in (up to) hundreds of virtual functions that trick virtual machines into believing they are the sole owner of a device. For example, given an SR-IOV network interface, a virtual machine is able to handle its virtual network card just like a physical one. Better still, many modern network cards have separate (circular) buffers for sending and receiving data, dedicated to this virtual machines. For instance, the Intel I350 series of network cards has eight send and eight receive queues

7.8 VIRTUAL APPLIANCES

Virtual machines offer an interesting solution to a problem that has long plagued users, especially users of open source software: how to install new application programs. The problem is that many applications are dependent on numerous other applications and libraries, which are themselves dependent on a host of other software packages, and so on. Furthermore, there may be dependencies on particular versions of the compilers, scripting languages, and the operating system.

With virtual machines now available, a software developer can carefully construct a virtual machine, load it with the required operating system, compilers, libraries, and application code, and freeze the entire unit, ready to run. This virtual machine image can then be put on a CD-ROM or a Website for customers to install or download. This approach means that only the software developer has to understand all the dependencies. The customers get a complete package that actually works, completely independent of which operating system they are running and which other software, packages, and libraries they have installed. These "shrink-wrapped" virtual machines are often called **virtual appliances**. As an example, Amazon's EC2 cloud has many pre-packaged virtual appliances available for its clients, which it offers as convenient software services ("Software as a Service").

7.9 VIRTUAL MACHINES ON MULTICORE CPUS

The combination of virtual machines and multicore CPUs creates a whole new world in which the number of CPUs available can be set by the software. If there are, say, four cores, and each can run, for example, up to eight virtual machines, a single (desktop) CPU can be configured as a 32-node multicomputer if need be, but it can also have fewer CPUs, depending on the software. Never before has it been possible for an application designer to first choose how many CPUs he wants and then write the software accordingly. This is clearly a new phase in computing.

Moreover, virtual machines can share memory. A typical example where this is useful is a single server hosting multiple instances of the same operating systems. All that has to be done is map physical pages into the address spaces of multiple virtual machines. Memory sharing is already available in deduplication solutions. **Deduplication** does exactly what you think it does: avoids storing the same data twice. It is a fairly common technique in storage systems, but is now appearing in virtualization as well. In Disco, it was known as **transparent page sharing** (which requires modification to the guest), while VMware calls it **content-based page sharing** (which does not require any modification). In general, the technique revolves around scanning the memory of each of the virtual machines on a host and hashing the memory pages. Should some pages produce an identical hash, the system has to first check to see if they really are the same, and if so, deduplicate them, creating one page with the actual content and two references to that page. Since the hypervisor controls the nested (or shadow) page tables, this mapping is straightforward. Of course, when either of the guests modifies a shared page, the change should not be visible in the other virtual machine(s). The trick is to use **copy on write** so the modified page will be private to the writer.

If virtual machines can share memory, a single computer becomes a virtual multiprocessor. Since all the cores in a multicore chip share the same RAM, a single quad-core chip could easily be configured as a 32-node multiprocessor or a 32-node multicomputer, as needed.

The combination of multicore, virtual machines, hypervisor, and microkernels is going to radically change the way people think about computer systems. Current software cannot deal with the idea of the programmer determining how many CPUs are needed, whether they should be a multicomputer or a multiprocessor, and how minimal kernels of one kind or another fit into the picture. Future software will have to deal with these issues. If you are a computer science or engineering student or professional, you could be the one to sort out all this stuff. Go for it!

7.10 LICENSING ISSUES

Some software is licensed on a per-CPU basis, especially software for companies. In other words, when they buy a program, they have the right to run it on just one CPU. What's a CPU, anyway? Does this contract give them the right to run

the software on multiple virtual machines all running on the same physical machine? Many software vendors are somewhat unsure of what to do here.

The problem is much worse in companies that have a license allowing them to have *n* machines running the software at the same time, especially when virtual machines come and go on demand.

In some cases, software vendors have put an explicit clause in the license forbidding the licensee from running the software on a virtual machine or on an unauthorized virtual machine. For companies that run all their software exclusively on virtual machines, this could be a real problem. Whether any of these restrictions will hold up in court and how users respond to them remains to be seen.

7.11 CLOUDS

Virtualization technology played a crucial role in the dizzying rise of cloud computing. There are many clouds. Some clouds are public and available to anyone willing to pay for the use of resources, others are private to an organization. Likewise, different clouds offer different things. Some give their users access to physical hardware, but most virtualize their environments. Some offer the bare machines, virtual or not, and nothing more, but others offer software that is ready to use and can be combined in interesting ways, or platforms that make it easy for their users to develop new services. Cloud providers typically offer different categories of resources, such as "big machines" versus "little machines," etc.

For all the talk about clouds, few people seem really sure about what they are exactly. The National Institute of Standards and Technology, always a good source to fall back on, lists five essential characteristics:

1. **On-demand self-service.** Users should be able to provision resources automatically, without requiring human interaction.
2. **Broad network access.** All these resources should be available over the network via standard mechanisms so that heterogeneous devices can make use of them.
3. **Resource pooling.** The computing resource owned by the provider should be pooled to serve multiple users and with the ability to assign and reassign resources dynamically. The users generally do not even know the exact location of "their" resources or even which country they are located in.
4. **Rapid elasticity.** It should be possible to acquire and release resources elastically, perhaps even automatically, to scale immediately with the users' demands.
5. **Measured service.** The cloud provider meters the resources used in a way that matches the type of service agreed upon.

7.11.1 Clouds as a Service

In this section, we will look at clouds with a focus on virtualization and operating systems. Specifically, we consider clouds that offer direct access to a virtual machine, which the user can use in any way he sees fit. Thus, the same cloud may run different operating systems, possibly on the same hardware. In cloud terms, this is known as **IAAS** (**Infrastructure As A Service**), as opposed to **PAAS** (**Platform As A Service**, which delivers an environment that includes things such as a specific OS, database, Web server, and so on), **SAAS** (**Software As A Service**, which offers access to specific software, such as Microsoft Office 365, or Google Apps), and many other types of as-a-service. One example of an IAAS cloud is Amazon EC2, which happens to be based on the Xen hypervisor and counts multiple hundreds of thousands of physical machines. Provided you have the cash, you can have as much computing power as you need.

Clouds can transform the way companies do computing. Overall, consolidating the computing resources in a small number of places (conveniently located near a power source and cheap cooling) benefits from economy of scale. Outsourcing your processing means that you need not worry so much about managing your IT infrastructure, backups, maintenance, depreciation, scalability, reliability, performance, and perhaps security. All of that is done in one place and, assuming the cloud provider is competent, done well. You would think that IT managers are happier today than ten years ago. However, as these worries disappeared, new ones emerged. Can you really trust your cloud provider to keep your sensitive data safe? Will a competitor running on the same infrastructure be able to infer information you wanted to keep private? What law(s) apply to your data (for instance, if the cloud provider is from the United States, is your data subject to the PATRIOT Act, even if your company is in Europe)? Once you store all your data in cloud X, will you be able to get them out again, or will you be tied to that cloud and its provider forever, something known as **vendor lock-in**?

7.11.2 Virtual Machine Migration

Virtualization technology not only allows IAAS clouds to run multiple different operating systems on the same hardware at the same time, it also permits clever management. We have already discussed the ability to overcommit resources, especially in combination with deduplication. Now we will look at another management issue: what if a machine needs servicing (or even replacement) while it is running lots of important machines? Probably, clients will not be happy if their systems go down because the cloud provider wants to replace a disk drive.

Hypervisors decouple the virtual machine from the physical hardware. In other words, it does not really matter to the virtual machine if it runs on this machine or that machine. Thus, the administrator could simply shut down all the virtual machines and restart them again on a shiny new machine. Doing so, however, results

in significant downtime. The challenge is to move the virtual machine from the hardware that needs servicing to the new machine without taking it down at all.

A slightly better approach might be to pause the virtual machine, rather than shut it down. During the pause, we copy over the memory pages used by the virtual machine to the new hardware as quickly as possible, configure things correctly in the new hypervisor and then resume execution. Besides memory, we also need to transfer storage and network connectivity, but if the machines are close, this can be relatively fast. We could make the file system network-based to begin with (like NFS, the network file system), so that it does not matter whether your virtual machine is running on hardware in server rack 1 or 3. Likewise, the IP address can simply be switched to the new location. Nevertheless, we still need to pause the machine for a noticeable amount of time. Less time perhaps, but still noticeable.

Instead, what modern virtualization solutions offer is something known as **live migration**. In other words, they move the virtual machine while it is still operational. For instance, they employ techniques like **pre-copy memory migration**. This means that they copy memory pages while the machine is still serving requests. Most memory pages are not written much, so copying them over is safe. Remember, the virtual machine is still running, so a page may be modified after it has already been copied. When memory pages are modified, we have to make sure that the latest version is copied to the destination, so we mark them as dirty. They will be recopied later. When most memory pages have been copied, we are left with a small number of dirty pages. We now pause very briefly to copy the remaining pages and resume the virtual machine at the new location. While there is still a pause, it is so brief that applications typically are not affected. When the downtime is not noticeable, it is known as a **seamless live migration**.

7.11.3 Checkpointing

Decoupling of virtual machine and physical hardware has additional advantages. In particular, we mentioned that we can pause a machine. This in itself is useful. If the state of the paused machine (e.g., CPU state, memory pages, and storage state) is stored on disk, we have a snapshot of a running machine. If the software makes a royal mess of the still-running virtual machine, it is possible to just roll back to the snapshot and continue as if nothing happened.

The most straightforward way to make a snapshot is to copy everything, including the full file system. However, copying a multiterabyte disk may take a while, even if it is a fast disk. And again, we do not want to pause for long while we are doing it. The solution is to use **copy on write** solutions, so that data is copied only when absolutely necessary.

Snapshotting works quite well, but there are issues. What to do if a machine is interacting with a remote computer? We can snapshot the system and bring it up again at a later stage, but the communicating party may be long gone. Clearly, this is a problem that cannot be solved.

MCA -402 Project Management (4 Credits)

Unit-1 Fundamentals of Project Management
Definition, Characteristics of Project, Types of Project, Project Phases, Project management Process, Project life cycle

Unit-2 Project formulation
Significance of project formulation, Project formulation, Feasibility analysis, technical analysis and financial analysis, cost of project, financing and estimates of sales of project.

Unit-3 Risk and Uncertainty Decisions
Project risk, types of project risk, identifying the risk, risk category, methods using risk identification,, Project risk Analysis, qualitative analysis and quantitative analysis, sensitivity analysis, break even analysis, cost of capital and capital budgeting.

Unit-4 Project scheduling and Control
Definition of Project scheduling, Project controls and importance, Network techniques of Project Management: Gantt chart, CPM, PERT

Unit-5 Software Project Management
Nature of Software and Software Development, software between computers and people, software quality, Software quality assurance, relation between software quality and software productivity, risk reduction, Role of project manager in software development

Unit-6 Project Management Information System (PMIS)
Concept of PMIS, components of PMIs, choosing and implementing PMIS, benefit of PMIS. Common errors managing PMIS

Unit 1 Fundamentals of Project Management

1.1 <u>Definition</u>:- Projects can be of any size and duration. They can be simple, like planning a party, or complex like launching a space shuttle.

A project can be defined in many ways :

- A project is “a temporary endeavor undertaken to create a unique product, service, or result.” Operations, on the other hand, is work done in organizations to sustain the business. Projects are different from operations in that they end when their objectives have been reached or the project has been terminated.

- A project is *temporary*. A project’s duration might be just one week or it might go on for years, but every project has an end date. You might not know that end date when the project begins, but it’s there somewhere in the future. Projects are not the same as ongoing operations, although the two have a great deal in common.

- A project is an *endeavor*. Resources, such as people and equipment, need to do work. The endeavor is undertaken by a team or an organization, and therefore projects have a sense of being intentional, planned events. Successful projects do not happen spontaneously; some amount of preparation and planning happens first.

1.2 <u>Characteristics of Project:-</u>

- Projects are unique.
- Projects are temporary in nature and have a definite beginning and ending date.
- Projects are completed when the project goals are achieved or it’s determined the project is no longer viable.

Projects come in all shapes and sizes. The following Attributes (Characteristics) help us to define a project further:

1.2.1 <u>Specific / Unique</u>:-

A project has a unique purpose. Every project should have a well-defined objective. For example, many people hire firms to design and build a new house, but each house, like each person, is unique. A project is always specific and unique, so the goal / purpose of the project must be specific. Since each and every project is unique and one different than that of the other; the project must be clearly defined in terms of its goals and purpose.

For example – Building the 10^{th} building will be different than that of the 9^{th} building already built or 11^{th} building to be built. That is one building to be built will be different than that of all the buildings built previously and to be built in future; the location is different, the design is different, the people living or would be living in the building will be different. Therefore each project must be specific.

1.2.2 Measurable:-

Right from the recourses to the time required for the completion of the project is uniquely defined in the initial stage of the project that makes the project measurable.

A project must be quantifiable in terms of its resources required and measurable in terms of time duration it requires for completion. The quantifiable resources may include accessories, men power, etc.

1.2.3 Achievable :-

A project must be achievable means that if you think of taking the project, u must have all the knowledge and resources needed to complete that particular project, or at least you must have capability to prepare such arrangements and environment required for the particular project's completion.

For example – If a software developer team who works in java environment, tends to take a .net or PHP project, then there would be many obstacles like, need a new developer, need to install .net php environment in the systems, etc; which makes that project of this particular team not achievable.

1.2.4 Relevant :-

The relevance of a project describes how efficient the outcome of the project is expected to be with respect to a given goal, to be specified by the evaluation or by the project being submitted. This means that the relevance of a project has always to be related to some goal and, for general research projects, that goal is most often the increase of our scientific knowledge, although it is sometimes also related to more direct social or environmental benefits for our society. In this last sense a project on renewable energy will be considered more relevant than a study of medieval music, although both may be scientifically relevant.

If a research project is set up in order to solve a specific problem, one may specifically judge the relevance of the proposed methodology with respect to solving this problem. This contains some aspect of feasibility but it is more than this: it may be quite feasible to perform a proposed project, whereas its result will never be able to solve the problem at hand because one started from a false assumption.

1.2.5 Time bound :-

The project are of two types on the basis of time it takes let's say a finite and an infinite project. As stated before a project can take a day or two or can take years or decades to complete.

But, in a project at its initial stage time bound is decided, the dead line is fixed, whereas during the progress of the project the due date is always considered in the process and accordingly the project is completed step by step. Therefore the time bound is again a very important aspect of the project development.

1.3 Types of Project :-

Projects can be of any size and duration. They can be simple, like planning a party, or complex like launching a space shuttle. Therefore we can broadly classify projects in two categories as:

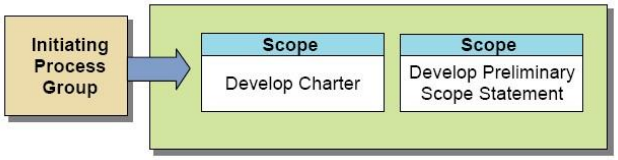
- Personal Projects &
- Industrial Projects.

For example, Personal projects like : obtaining an MCA degree, writing a report, planning a party, planting a garden, etc…
And Industrial projects like : Construction of a building, provide electricity to an industrial estate, building a bridge, designing a new airplane, etc…

1.4 Project Phases:-

1.4.1 Initial Phase :-

The initiating processes determine the nature and scope of the project. If this stage is not performed well, it is unlikely that the project will be successful in meeting the business' needs. The key project controls needed here are an understanding of the business environment and making sure that all necessary controls are incorporated into the project. Any deficiencies should be reported and a recommendation should be made to fix them.



The initiating stage should include a plan that encompasses the following areas:

- analyzing the business needs/requirements in measurable goals
- reviewing of the current operations
- financial analysis of the costs and benefits including a budget
- stakeholder analysis, including users, and support personnel for the project
- project charter including costs, tasks, deliverables, and schedules

1.4.2 Planning Phase :-

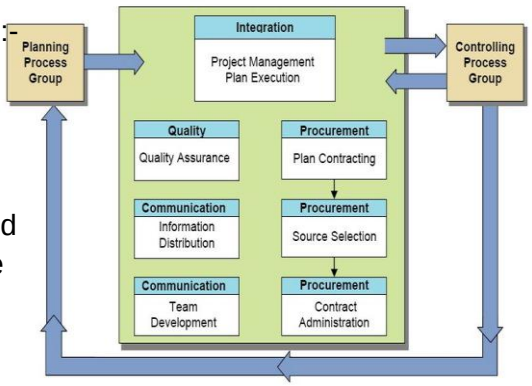
After the initiation stage, the project is planned to an appropriate level of detail. The main purpose is to plan time, cost and resources adequately to estimate the work needed and to effectively manage risk during project execution. As with the Initiation process group, a failure to adequately plan greatly reduces the project's chances of successfully accomplishing its goals. Project planning generally consists of

- determining how to plan (e.g. by level of detail or Rolling Wave planning);
- developing the scope statement;
- selecting the planning team;
- identifying deliverables and creating the work breakdown structure;
- identifying the activities needed to complete those deliverables and networking the activities in their logical sequence;
- estimating the resource requirements for the activities;
- estimating time and cost for activities;
- developing the schedule;
- developing the budget;
- risk planning;
- Gaining formal approval to begin work.

Additional processes, such as planning for communications and for scope management, identifying roles and responsibilities, determining what to purchase for the project and holding a kick-off meeting are also generally advisable are also included.

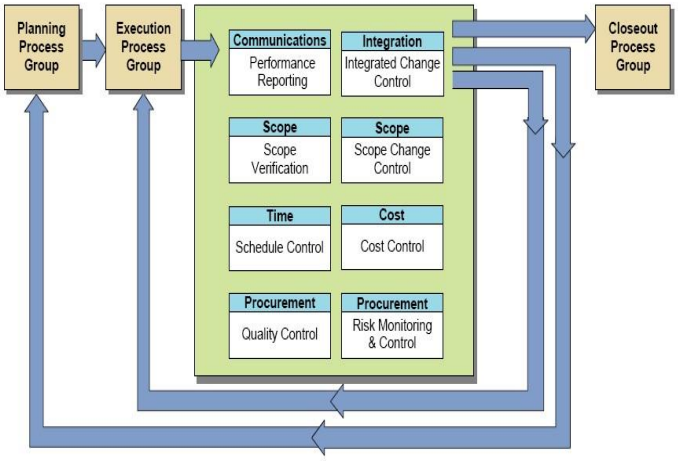
For new product development projects, conceptual design of the operation of the final product may be performed concurrent with the project planning activities, and may help to inform the planning team when identifying deliverables and planning activities.

1.4.3 Implementation / Execution / Production Phase :-



The execution/implementation phase ensures that the project management plan's deliverables are executed accordingly. This phase involves proper allocation, co-ordination and management of human resources and any other resources such as material and budgets. The output of this phase is the project deliverables.

1.4.4 Monitoring and Controlling Phase :-



Monitoring and controlling consists of those processes performed to observe project execution so that potential problems can be identified in a timely manner and corrective action can be taken, when necessary, to control the execution of the project. The key benefit is that project performance is observed and measured regularly to identify variances from the project management plan.

Monitoring and controlling includes:

- Measuring the ongoing project activities.
- Monitoring the project variables (cost, effort, scope, etc.) against the project management plan and the project performance baseline (where we should be);
- Identify corrective actions to address issues and risks properly (How can we get on track again);
- Influencing the factors that could circumvent integrated change control so only approved changes are implemented.

In multi-phase projects, the monitoring and control process also provides feedback between project phases, in order to implement corrective or preventive actions to bring the project into compliance with the project management plan.

Project maintenance is an ongoing process, and it includes:

- Continuing support of end-users
- Correction of errors
- Updates to the product over time

In this stage, auditors should pay attention to how effectively and quickly user problems are solved.

1.4.5 Closing Phase :-



Closing includes the formal acceptance of the project and the ending thereof. Administrative activities include the archiving of the files and documenting lessons learned.

This phase consists of:

- Contract closure: Complete and settle each contract (including the resolution of any open items) and close each contract applicable to the project or project phase.
- Project close: Finalize all activities across all of the process groups to formally close the project or a project phase

Also included in this phase is the Post Implementation Review. This is a vital phase of the project for the project team to learn from experiences and apply to future projects. Normally a Post Implementation Review consists of looking at things that went well and analyzing things that went badly on the project to come up with lessons learned

1.5 Project management Process:-

Project initiation is the starting point of any project. In this process, all the activities related to winning a project takes place. Usually, the main activity of this phase is the pre-sale. During the pre-sale period, the service provider proves the eligibility and ability of completing the project to the client and eventually wins the business. Then, it is the detailed requirements gathering which comes next.

During the requirements gathering activity, all the client requirements are gathered and analyzed for implementation. In this activity, negotiations may take place to change certain requirements or remove certain requirements altogether.

Usually, project initiation process ends with requirements sign-off.

1.5.1 Project Planning :-

Project planning is one of the main project management processes. If the project management team gets this step wrong, there could be heavy negative consequences during the next phases of the project. Therefore, the project management team will have to pay detailed attention to this process of the project.

In this process, the project plan is derived in order to address the project requirements such as, requirements scope, budget and timelines. Once the project plan is derived, then the project schedule is developed. Depending on the budget and the schedule, the resources are then allocated to the project. This phase is the most important phase when it comes to project cost and effort.

1.5.2 Project Execution :-

After all paperwork is done, in this phase, the project management executes the project in order to achieve project objectives.

When it comes to execution, each member of the team carries out their own assignments within the given deadline for each activity. The detailed project schedule will be used for tracking the project progress. During the project execution, there are many reporting activities to be done. The senior management of the company will require daily or weekly status updates on the project progress.

In addition to that, the client may also want to track the progress of the project. During the project execution, it is a must to track the effort and cost of the project in order to determine whether the project is progressing in the right direction or not.

In addition to reporting, there are multiple deliveries to be made during the project execution. Usually, project deliveries are not one time deliveries made at the end of the project. Instead, the deliveries are scattered throughout the project execution period and delivered upon agreed timelines.

1.5.3 Controll and Validation :-

During the project life cycle, the project activities should be thoroughly controlled and validated. The controlling can be mainly done by adhering to the initial protocols such as project plan, quality assurance test plan and communication plan for the project. Sometimes, there can be instances that are not covered by such protocols. In such cases, the project manager should use adequate and necessary measurements in order to control such situations.

Validation is a supporting activity that runs from first day to the last day of a project. Each and every activity and delivery should have its own validation criteria in order to verify the successful outcome or the successful completion.

When it comes to project deliveries and requirements, a separate team called 'quality assurance team' will assist the project team for validation and verification functions.

1.5.4 Closeout and Evalution :-

Once all the project requirements are achieved, it is time to hand over the implemented system and closeout the project. If the project deliveries are in par with the acceptance criteria defined by the client, the project will be duly accepted and paid by the customer.

Once the project closeout takes place, it is time to evaluate the entire project. In this evaluation, the mistakes made by the project team will be identified and will take necessary steps to avoid them in the future projects.

During the project evaluation process, the service provider may notice that they haven't gained the expected margins for the project and may have exceeded the timelines planned at the beginning. In such cases, the project is not a 100% success to the service provider. Therefore, such instances should be studied carefully and should take necessary actions to avoid in the future.

Conclusion :-

Project management is a responsible process. The project management process connects all other project activities together and creates the harmony in the project.

Therefore, the project management team should have a detailed understanding on all the project management processes and the tools that they can make use for each project management process.

1.6 Project life cycle :-

Project lifecycle is spread over a period of time. ~~There is an unavoidable gestation period for the complex of activities involved to attain the objectives in view. This gestation period, however, varies from project to project but it is possible to describe, in general term, the time phasing of project planning activities common to most projects.~~
The principal stages in the life of a project are :

- Identification
- Initial formulation
- Evaluation (selection or rejection)
- Final formulation (or selection)
- Implementation
- Completion and operation

1.6.1 Initial Formulation:-

Identification is only the beginning in the lifecycle of a project. Having identified the prospective projects, the details of each project will have to be worked out and analysed in order to determine which of them could be reckoned as suitable for inclusion in the plan, allocate funds and put into execution. ~~As a follow up to the finding of techno-economic surveys, and number of feasibility study group are set up, as the name implies to examine the possibility of formulating suitable projects and to put concrete proposals in sufficient detail to enable authorities concerned to consider the feasibility of the proposal submitted.~~

1.6.2 Evaluation or Project Appraisal:-

After the socio-economic problems of an economy have been determined and developments objectives and strategies agreed, concrete steps have to be taken. The main form this takes is that of formulating appropriate development projects to achieve plan objectives and meet the development needs of the economy. Proposals relating to them are then put to the plan authorities for consideration and inclusion in the plan. These proposals as pointed out above take. the following forms of feasibility studies :

- Commercial viability
- Economic feasibility
- Financial feasibility
- Technical feasibility
- Management

~~The scope for scrutiny under each of these five heads would necessarily render their careful assessment and the examination of all possible alternative approaches. The process almost invariably involves making decision relating to technology, scale, location, costs and benefits, time of completion (gestation period), degree of risk and uncertainty, financial viability, organisation and management, availability of inputs, know-how, labour etc. The detailed analysis is set down in what is called a feasibility report.~~

1.6.3 Formulation :-

Once a project has been appraised and approved, next step would logically, appear to that of implementation. This is, however, not necessarily true, if the approval is conditional to certain modifications being affected or for other reasons, such as availability of funds, etc. The implementation stage will be reached only after these pre-conditions have been

fulfilled. Project formulation divides the process of project development into eight distinct and sequential stages. These stages are

- General information
- Project description
- Market potential
- Capital costs and sources of finance
- Assessment of working capital requirement
- Other financial aspect
- Economic and social variables.

1.6.4 Project Implementation :-

Last but not the least, every entrepreneur should draw an implementation time table for his project. The network having been prepared, the project authorities are now ready to embark on the main task of implementation the project. To begin with successful implementation will depend on how well the network has been designed. ~~However, during the course of implementation, many factors arise which cannot be anticipated or adequately taken note of in advance and built into the initial network. A number of network techniques have been developed for project implementation. Some of them are PERT, CPM, Graphical Evaluation and Review Technique (GERT), Workshop Analysis Scheduling Programmed (WRSP) and Line of Balance (LOB).~~

1.6.5 Project Completion :-

It is often debated as to the point at which the project life cycle is completed. The cycle is completed only when the development objectives are realized.

Unit-2 Project formulation

2.1 Significance of project formulation :-

Project formulation divides the process of project development into eight distinct and sequential stages. These stages are

- General information
- Project description
- Market potential
- Capital costs and sources of finance
- Assessment of working capital requirement
- Other financial aspect
- Economic and social variables.

2.2 Project formulation :-

Project Formulation is a concise, exact statement of a project to set the boundaries or limits of work to be performed by the project. It is a formal document that gives a distinctive identity of the project and precise meaning of project work to prevent conflict, confusion, or overlap.

Project Formulation is defined as taking of first look carefully and critically a project idea by an entrepreneur to build up an all round beneficial to project after carefully weighing its various componants. Project formulation can be also defined as one of the stages in the lifecycle of a project. The formulation stage is also called Initiation, Conceptualization, Definition, Pre-Project. This stage aims to:

- Carefully identify and weight various components of project work
- Analyze project feasibility and cost-effectiveness
- Examine and approve project inputs and outputs
- Identify stakeholders and their involvement and contribution
- Define benefits and expectations
- Estimate resources needed
- Perform a preliminary analysis of risks
- Make an outline of project schedule

A project lifecycle itself consists of formulation such as;

Initial Formulation:- Identification is only the beginning in the lifecycle of a project. Having identified the prospective projects, the details of each project will have to be worked out and analyzed in order to determine which of them could be reckoned as suitable for inclusion in the plan, allocate funds and put into execution. As a follow up to the finding of techno-economic surveys, and number of feasibility study group are set up, as the name implies to examine the possibility of formulating suitable projects and to put concrete proposals in sufficient detail to enable authorities concerned to consider the feasibility of the proposal submitted.

2.3 Feasibility analysis :-

A feasibility study aims to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects for success In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained.

A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies,

financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based.
The acronym TELOS refers to the five areas of feasibility - Technical, Economic, Legal, Operational, and Scheduling.

2.3.1 Technical feasibility :-

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. When writing a feasibility report, the following should be taken to consideration:

- A brief description of the business to assess more possible factors which could affect the study,
- The part of the business being examined,
- The human and economic factor,
- The possible solutions to the problem.

At this level, the concern is whether the proposal is both technically and legally feasible (assuming moderate cost). The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

2.3.2 Economic feasibility :-

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

2.3.3 Legal feasibility **:-**

This test determines whether the proposed system conflicts with legal requirements, e.g. a data processing system must comply with the local data protection regulations.

2.3.4 Operational feasibility :-

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture, and existing business processes.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters such as reliability, maintainability, supportability, usability, productibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

2.3.5 Schedule feasibility **:-**

A project will fail if it takes too long to be completed before it is useful. Typically this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable.

2.4 Technical analysis and financial analysis :-

2.4.1 Technical Analysis :-

Technical analysis implies the adequacy of the proposed plant and equipment to prescribed norms. It should be ensured whether the required know how is available with the entrepreneur. The following inputs concerned in the project should also be taken into consideration.

- Availability of Land and site
- Availability of Water Power, transport, communication facilities.
- Availability of servicing facilities like machine shop, electric repair shop etc.
- Coping with anti pollution law
- Availability of work force
- Availability of required raw material as per quantity and quality.

2.4.2 Financial Analysis :-

Finance is one of the most important prerequisites to establish an enterprise. It is finance only that facilitates an entrepreneur to bring together the labor, machines and raw materials to combine them to produce goods. In order to adjudge the financial viability of the project, the following aspects need to be carefully analyzed:

- Cost of capital
- Means of finance
- Estimates of sales and production
- Cost of production
- Working capital requirement and its financing
- Estimates of working results
- Break-even point
- Projected cash flow
- Projected balance sheet.

The activity level of an enterprise expressed as capacity utilization needs to be well spelled out, However the enterprise sometimes fails to achieve the targeted level of capacity due to various business vicissitudes like unforeseen shortage of raw

material, unexpected disruption in power supply, instability to penetrate the market mechanism etc.

2.5 Cost of project :-

The cost of project is the sum of design fee, material and resources cost, financing, production cost etc.

The cost associated with a project can be divided into two components: direct cost and indirect cost. Direct costs are incurred on direct material and direct labor. Indirect costs consists of overhead items like indirect supplies, rent, insurance, managerial services etc.

2.6 Financing and estimates of sales of project :-

What is Project Finance?

Project Finance can be characterized in a variety of ways and there is no universally adopted definition but as a financing technique, the author's definition is:
"The raising of finance on a Limited Recourse basis, for the purposes of developing a large capital- intensive infrastructure project, where the borrower is a special purpose vehicle and repayment of the financing by the borrower will be dependent on the internally generated cash flows of the project"

Finance Arrangements

The funds required for the project is to meet the capital expenditure for the plants and machineries, initial working capital and Pre-operating expenses.

To be able to access the correctly the fund requirement and plan for the same, the first step would be to identify what are the various capital, working, and operating costs that are to be financed for the project.

1. Capital Cost

All the costs incurred in the project before it becomes ready, to start commercial production will be treated as capital costs. Therefore capital cost will include not only expenditure on assets such as land, licensor's fee, management and supervision and pre-operative expenses.

2. Working Cost

The funds required for maintaining various inventories in the form of raw materials, operating supplies, intermediate products, finished products and, meeting miscellaneous cash requirements for maintenance of a leave of production is treated as working capital.

3. Operating Cost

These are the costs which will be incurred on a recurring basis for production, maintenance and marketing. Raw materials, labor, utilities, repair and maintenance, selling expenses and any other expenses to be incurred year after year once the project goes into commercial production are treated as operating expenses.

Source Financing

There are basically two sources available for financing the fund requirement of the project— Internal source and external source. Unless a project is very small, the fund requirements will have to be financed from external sources.

Capital cost of the project will require to be financed basically through equity and long term debts. What need to be decided now is what portion of the total funds should be from equity and what should be from equity and what should be from long term debt.

Financial Institutions

Most projects are primarily financed from debts or term loans. There are several institutions functioning at the all India level as also at state levels which have been set up not only to finance the projects but also to promote establishments of new projects. Besides there are several foreign financial institutions which provide funds to finance projects in developing countries, these are as follows:

1. **National Financial Institutions :**
   i. Industrial Development bank of India (IDBI) :-

This is the principal financial institution of India. It coordinates the financial activities of other financial institutions, supplements their resources to plan and promote industries.

   ii. Industrial Financial Corporation of India (IFCI) :-

It provides long term loans under various schemes to industrial concerns both in the public and the private sector.

   iii. Industrial Credit and Investment Corporation of India (ICICI) :-

This is the financial institution which is mainly concerned with providing foreign currency loans to industrial concerns in the private sector.

   iv. United Trust of India :-

This institution mobilizes the savings of the general public and invests them in general industrial units.

   v. Life Insurance Corporation (LIC) :-

It has now acquired the status of a development financial institution. It provides long term finance to the industrial units.

2. **Foreign Financial Institutions :**
   i. World Bank (International Bank for Reconstruction and Development) :-

World Bank provides funds to the less developed member countries to develop infrastructure, Schools, irrigation dams, Power Plants, roads, water supplies and sewerages, etc. are the special projects which have been aided by the World Bank.

   ii. International Finance Corporation (IFC) :-

It is the subsidiary of the World Bank and provides funds especially for the private sector.

   iii. International Development Association (IDA) :-

This is also the subsidiary to the World Bank. It provides soft loans to under-developed countries

   iv. International Monitory Fund (IMF) :-

This is a part of United Nations. It compliments World Bank’s efforts to promote economic growth.

v. Asian Development Bank (ADB) :-

This is the development bank for the Asian continent. This institution finances infrastructure projects and new industrial units.

Preparation of cost estimate

A project cost estimate is required not only for accessing fund requirement but also for ascertaining the economy viability of the project. The estimate should be accurate. If the cost of the project is not measured accurately the fund plan consequently will be inaccurate.

Types of Estimates

1. **Order of magnitude estimate**

This estimate is made when a project has been identified and the entrepreneur wants to get a rough idea of the investment so as to decide whether to pursue then project or not.

2. **Study estimate**

This estimate is for studying the economic viability of the project and also for arranging funds for the project. At this stage, preliminary flow sheets, listing of major process equipment giving size and material of construction are obtained from the process licenser.

3. **Preliminary estimate**

The preliminary estimate is prepared when the technology package is frozen and a firm implementation schedule is available.

4. **Definitive estimate**

This estimate is prepared after the zero date when the detailed engineering of a project is in an advanced stage.

5. **Detailed estimate**

This estimate may be made on completion of engineering, ordering of equipment and machineries and award of major field contracts. At this stage head office work for a project is mostly completed.

Unit-3 Risk and Uncertainty Decisions

3.1 Project risk :-

Definition: - Project risk is defined by PMI as 'an uncertain event or condition that, if it occurs, has a positive or negative effect on a project's objectives'

According to the Project Management Institute's PMBOK, Risk management is one of the ten knowledge areas in which a project manager must be competent.

Assessing risks on a project, in a way that will add value and help you manage the project better, is not easy. Much of what will happen on a project is uncertain anyway so when do uncertainties become risks? We have, after all, merely estimated the resources we think we need or we may not fully understand the business needs so projects are inherently risky anyway. For some projects, such as ones using new, untried technology the whole project may be a risk.

Let's take, for example, estimating the time to build a new prototype car. Unlike building a standard car, building a prototype will involve new designs and technology so the estimate will not be entirely accurate. A bit of over-run may not be expected but if there is a tight deadline because the manufacturer wants the car ready for a particular motor show then the uncertainty combined with the deadline creates a real risk to the project's success.

3.2 Types of project risk :-

Complex projects are always fraught with a variety of risks ranging from scope risk to cost overruns. One of the main duties of a project manager is to manage these risks and prevent them from ruining the project. In this post, I will cover the major risks involved in a typical project.

3.2.1 Scope Risk :-

This risk includes changes in scope caused by the following factors:

- Scope creep – the project grows in complexity as clients add to the requirements and developers start gold plating.
- Integration issues
- Hardware & Software defects
- Change in dependencies

Defining what is required is not always easy. However, so as to ensure that scope risk is minimized, the deliverables, the objectives, the project charter, and of course, the scope needs to be clearly defined.

All scope risks, be they quantifiable or not, needs to recognized. Scope creep, hardware defects, software defects, an insufficiently defined scope, unexpected changes in the legal or regulatory framework and integration defects can all be classified under the broad umbrella of scope risk.

There are a variety of methods that help stakeholders identify the scope of the project. The risk framework analyses the project's dependency on technology and the market and then assesses how changes in each would affect the outcome of the project.

Similarly, the risk complexity index looks at the technical aspects of the projects, which can be easily quantified and allocated a number between 0 and 99 to indicate the risk of the project.

Risk assessment, on the other hand, uses a grid of technology, structure and magnitude to assess the proposed risk of the project.

A work breakdown structure, commonly abbreviated as WBS, also considers the risks of projects, which are ill defined and where the stated objectives are ambiguous.

Scope risks can be minimized and managed with savvy planning. Defining the project clearly, managing the changes in scope throughout the duration of the project, making use of risk registers to better manage risks, identifying the causative factors, and the appropriate responses to risky situations and developing greater risk tolerance in collaboration with the customer, would pay great dividends in the long run.

3.2.2 **Schedule Risk :-**

There are a number of reasons why the project might not proceed in the way you scheduled. These include unexpected delays at an external vendor, natural factors, errors in estimation and delays in acquisition of parts. For instance, the test team cannot begin the work until the developers finish their milestone deliverables and a delay in those can cause cascading delays.

To reduce scheduling risks use tools such as a Work Breakdown Structure (WBS) and RACI matrix (Responsibilities, Accountabilities, Consulting and Information) and Gantt charts to help you in scheduling.

Keeping to timelines and agreed critical paths is one of the most difficult situations that project managers now face.

An extensive reliance on external parties whose output is not within the project's scope of control, estimation errors, which most often are too optimistic, hardware delays and putting off decision making, all tend to delay the project at hand.

To minimize schedule risks, there are a few time-tested methods that can be put to good use. The process flow of the project should be broken down into small, clearly defined components where the allocated timeframe for each process is relatively short in duration (this makes it easy to identify things when tasks veer off schedule, at its earliest).

Be wary of team members or external parties, who hesitate to give estimates or whose estimates seem unrealistic based on historical data and previous experience.

When formulating the critical path, ensure that any holidays that arise are in-built into the equation, so that realistic expectations are created, right from inception. Defining re-work loops too is also recommended, wherever possible.

3.2.3 **Resource Risk :-**

This risk mainly arises from outsourcing and personnel related issues. A big project might involve dozens or even hundreds of employees and it is essential to manage the attrition issues and leaving of key personnel. Bringing in a new worker at a later stage in the project can significantly slow down the project.

Apart from attrition, there is a skill related risk too. For instance, if the project requires a lot of website front end work and your team doesn’t have a designer skilled in HTML/CSS, you could face unexpected delays there.

Another source of the risk includes lack of availability of funds. This could happen if you are relying on an external source of funding (such as a client who pays per milestone) and the client suddenly faces a cash crunch.

People and funds are any project's main resource base. If the people are unskilled or incompetent to perform the task at hand, if the project is under-staffed

from the beginning, or if key project members come on aboard far after the inception of the project, there is an obvious project risk that has ill-planned human resources as its base.

Similarly, from a financial perspective, if insufficient funds are provided to carry out the necessary tasks, be it relevant training programs for the people in question or be it inadequate investments in technology or required machinery, the project is doomed to fail from inception.

Estimating project costs accurately, allocating a suitable budget to meet these costs, not placing undue expectations on the capacity of the staff in question and avoiding burn-out at a later date are all factors that help minimize the project resource risk.

Outsourced functions merit even more attention to detail, as it is for the most part, it is away from the direct purview of the project manager. Clearly defined contracts and regular monitoring would lessen this risk substantially.

Conflict management, which too generally arises with the progression of a project, should also be handled in a skilful manner, so that the project has a smooth run throughout its entire duration.

3.2.4 **Technology Risk :-**

This risk includes delays arising out of software & hardware defects or the failure of an underlying service or a platform. For instance, halfway through the project you might realize the cloud service provider you are using doesn't satisfy your performance benchmarks. Apart from this, there could be issues in the platform used to build your software or a software update of a critical tool that no longer supports some of your functions.

3.2.5 **Customer Risk :-**

1. Customer insists on new requirements.
2. Customer review/decision cycles for plans, prototypes, and specifications are slower than expected.
3. Customer insists on technical decisions that lengthen the schedule.
4. Customer will not accept the software as delivered even though it meets all specifications.
5. Customer has expectations for development speed that developers cannot meet.

As is readily apparent, all projects do run the risk of failure due to unplanned contingencies and inaccurate estimates.

Yet, careful planning, constraint management, successful recovery from mistakes if and when they do arise will minimize most risks. True, luck too, does play a part in the success of a project, but hard work and savvy management practices will override most such difficulties.

3.3 **Identifying the risk :-**

Project Risk Management includes the processes of conducting risk management planning, identification, analysis, response planning, and controlling risk on a project. The objectives of project risk management are to increase the likelihood and impact of positive events, and decrease the likelihood and impact of negative events in the project.

Project Risk identification is the most important process in the Risk Management Planning. Risk Identification determines which risks might affect the project and documents their characteristics. However, as recommended by [Donna Ritter], we should not spend too much time in identifying risks. After the list is made, qualitative and quantitative analysis is done to figure out which risks you spend time and/or money on.

There are some specific tools and techniques for identifying risk as listed below:

1. Documentation Reviews
2. Information Gathering Techniques - Brainstorming, Delphi Technique, Interviewing, Root cause analysis
3. Checklist analysis - previous similar project, lowest level RBS
4. Assumption analysis
5. Diagramming Techniques - cause and effect diagram, system and process flow chart, influence diagrams
6. SWOT Analysis
7. Expert Judgment

The risk identification method suggested in this article is to compliment the existing tools and techniques recommended by PMBOK.

The common Project Risk List Reference below which are divided into a number of risk categories are samples of potential risks of a project may be exposed to and should only be used by the Project Team as a reference and starting point for risk identification during the project risk management planning.

3.4 Risk category :-

3.4.1 SCHEDULE RISK :-

1. Schedule not realistic, only "best case".
2. Important task missing from the schedule.
3. A delay in one task causes cascading delays in dependent tasks.
4. Unfamiliar areas of the product take more time than expected to design and implement

3.4.2 REQUIREMENT RISK :-

1. Requirements have been base lined but continue to change.
2. Requirements are poorly defined, and further definition expands the scope of the project
3. Specified areas of the product are more time-consuming than expected.
4. Requirements are only partly known at project start
5. The total features requested may be beyond what the development team can deliver in the time available.

3.4.3 **PROJECT MANAGEMENT RISK :-**

1. PM has little authority in the organization structure and little personal power to influence decision-making and resources
2. Priorities change on existing program
3. Project key success criteria not clearly defined to verify the successful completion of each project phase.
4. Projects within the program often need the same resources at the same time
5. Date is being totally driven by need to meet marketing demo, trade show, or other mandate; little consideration of project team estimates

3.4.4 **PRODUCT/TECHNOLOGY RISK :-**

1. Development of the wrong user interface results in redesign and implementation.
2. Development of extra software functions that are not required (gold plating) extends the schedule.
3. Requirements for interfacing with other systems are not under the team's scope.
4. Dependency on a technology that is still under development lengthens the schedule.
5. Selected technology is a poor match to the problem or customer

3.4.5 **CUSTOMER RISK :-**

1. Customer insists on new requirements.
2. Customer review/decision cycles for plans, prototypes, and specifications are slower than expected.
3. Customer insists on technical decisions that lengthen the schedule.
4. Customer will not accept the software as delivered even though it meets all specifications.
5. Customer has expectations for development speed that developers cannot meet.

3.4.6 **HUMAN RESOURCES & CONTRACTORS RISK :-**

1. Critical development work is being performed by one developer
2. Some developers may leave the project before it is finished.
3. Hiring process takes longer than expected.
4. Personnel need extra time to learn unfamiliar software tools, hardware and programming language.
5. Contract personnel leave before project is complete.

6. Conflicts among team members result in poor communication, poor designs, interface errors and extra rework.
7. Personnel with critical skills needed for the project cannot be found.
8. Contractor does not deliver components when promised.

3.5 Methods using risk identification :-

Risk is inevitable in a business organization when undertaking projects. However, the project manager needs to ensure that risks are kept to a minimal. Risks can be mainly divided between two types, negative impact risk and positive impact risk.

Not all the time would project managers be facing negative impact risks as there are positive impact risks too. Once the risk has been identified, project managers need to come up with a mitigation plan or any other solution to counter attack the risk.

Managers can plan their strategy based on four steps of risk management which prevails in an organization. Following are the steps to manage risks effectively in an organization:

a. Risk Identification
b. Risk Quantification
c. Risk Response
d. Risk Monitoring and Control

a. Risk Identification :-

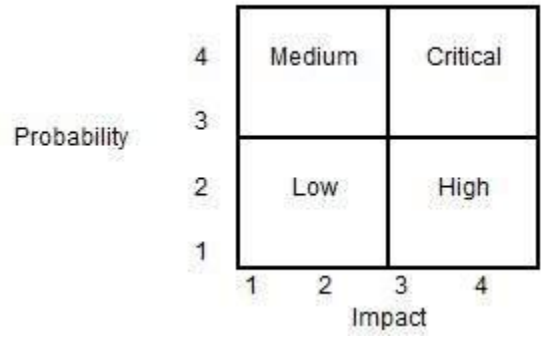
Managers face many difficulties when it comes to identifying and naming the risks that occur when undertaking projects. These risks could be resolved through structured or unstructured brainstorming or strategies. It's important to understand that risks pertaining to the project can only be handled by the project manager and other stakeholders of the project.

Risks, such as operational or business risks will be handled by the relevant teams. The risks that often impact a project are supplier risk, resource risk and budget risk. Supplier risk would refer to risks that can occur in case the supplier is not meeting the timeline to supply the resources required.

Resource risk occurs when the human resource used in the project is not enough or not skilled enough. Budget risk would refer to risks that can occur if the costs are more than what was budgeted.

b. Risk Quantification :-

Risks can be evaluated based on quantity. Project managers need to analyze the likely chances of a risk occurring with the help of a matrix.



Using the matrix, the project manager can categorize the risk into four categories as Low, Medium, High and Critical. The probability of occurrence and the impact on the project are the two parameters used for placing the risk in the matrix categories. As an example, if a risk occurrence is low (probability = 2) and it has the highest impact (impact = 4), the risk can be categorized as 'High'.

c. Risk Response :-

When it comes to risk management, it depends on the project manager to choose strategies that will reduce the risk to minimal. Project managers can choose between the four risk response strategies, which are outlined below.

- Risks can be avoided
- Pass on the risk
- Take corrective measures to reduce the impact of risks
- Acknowledge the risk

d. Risk Monitoring and Control :-

Risks can be monitored on a continuous basis to check if any change is made. New risks can be identified through the constant monitoring and assessing mechanisms.

3.6 Project risk Analysis :-

Risk analysis is the review of the risks associated with a particular event or action. It is applied to projects, information technology, security issues and any action where risks may be analyzed on a quantitative and qualitative basis. Risk analysis is a component of risk management.

Risk analysis can be defined in many different ways, and much of the definition depends on how risk analysis relates to other concepts. Risk analysis can be "broadly defined to include risk assessment, risk characterization, risk communication, risk, and policy relating to risk, in the context of risks of concern to individuals, to public- and private-sector organizations, and to society at a local, regional, national, or global level. A useful construct is to divide risk analysis into two components:

(1) risk assessment (identifying, evaluating, and measuring the probability and severity of risks)
(2) risk management (deciding what to do about risks).

Some books take a slightly different approach and define risk management as the overarching concept, where risk analysis is the component that seeks to identify and measure the risks and risk mitigation is determining what to do about the risks.

The risk management process involves a few key steps. First, potential threats are identified. For example, risks are associated with individuals using a computer either incorrectly or inappropriately, which creates security risks. Risks are also related to projects that are not completed in a timely manner, resulting in significant costs.

Next, quantitative and/or qualitative risk analysis is applied to study identified risks. Quantitative risk analysis measures expected risk probability to forecast estimated financial losses from potential risks. Qualitative risk analysis does not use numbers but reviews threats, and determines and establishes risk mitigation methods and solutions.

A contingency plan may be used during risk analysis. If a risk is presented, contingency plans help minimize damage.

Risk analysis is the process of defining and analyzing the dangers to individuals, businesses and government agencies posed by potential natural and human-caused adverse events. In IT, a risk analysis report can be used to align technology-related objectives with a company's business objectives. A risk analysis report can be either quantitative or qualitative.

In quantitative risk analysis, an attempt is made to numerically determine the probabilities of various adverse events and the likely extent of the losses if a particular event takes place.

Qualitative risk analysis, which is used more often, does not involve numerical probabilities or predictions of loss. Instead, the qualitative method involves defining the various threats, determining the extent of vulnerabilities and devising countermeasures should an attack occur.

3.7 Qualitative analysis and quantitative analysis sensitivity analysis :-

Risk analysis is the process of defining and analyzing the dangers to individuals, businesses and government agencies posed by potential natural and human-caused adverse events. In IT, a risk analysis report can be used to align technology-related objectives with a company's business objectives. A risk analysis report can be either quantitative or qualitative.

In quantitative risk analysis, an attempt is made to numerically determine the probabilities of various adverse events and the likely extent of the losses if a particular event takes place.

Qualitative risk analysis, which is used more often, does not involve numerical probabilities or predictions of loss. Instead, the qualitative method involves defining the various threats, determining the extent of vulnerabilities and devising countermeasures should an attack occur.

3.8 Break even analysis :-

The break-even analysis lets you determine what you need to sell, monthly or annually, to cover your costs of doing business—your break-even point.

The break-even point (BEP) is the point at which total cost and total revenue are equal: there is no net loss or gain, and one has "broken even." A profit or a loss has not been made, although opportunity costs have been "paid", and capital has received the risk-adjusted, expected return. In short, all costs that needs to be paid are paid by the firm but the profit is equal to 0.

A break-even analysis is an analysis to determine the point at which revenue received equals the costs associated with receiving the revenue. Break-even analysis calculates what is known as a margin of safety, the amount that revenues exceed the break-even point. This is the amount that revenues can fall while still staying above the break-even point.

Break-even analysis is a supply-side analysis; that is, it only analyzes the costs of the sales. It does not analyze how demand may be affected at different price levels.

For example, if it costs $50 to produce a widget, and there are fixed costs of $1,000, the break-even point for selling the widgets would be:

If selling for $100: 20 Widgets (Calculated as 1000/(100-50)=20)
If selling for $200: 7 Widgets (Calculated as 1000/(200-50)=6.7)

In this example, if someone sells the product for a higher price, the break-even point will come faster. What the analysis does not show is that it may be easier to sell 20 widgets at $100 each than 7 widgets at $200 each. A demand-side analysis would give the seller that information.

3.9 Cost of capital and capital budgeting. :-

3.9.1 Cost of capital :-

The cost of funds used for financing a business. Cost of capital depends on the mode of financing used – it refers to the cost of equity if the business is financed solely through equity or to the cost of debt if it is financed solely through debt. Many companies use a combination of debt and equity to finance their businesses, and for such companies, their overall cost of capital is derived from a weighted average of all capital sources, widely known as the weighted average cost of capital (WACC).

Since the cost of capital represents a hurdle rate that a company must overcome before it can generate value, it is extensively used in the capital budgeting process to determine whether the company should proceed with a project.

The cost of various capital sources varies from company to company, and depends on factors such as its operating history, profitability, credit worthiness, etc. In general, newer enterprises with limited operating histories will have higher costs of capital than established companies with a solid track record, since lenders and investors will demand a higher risk premium for the former.

Every company has to chart out its game plan for financing the business at an early stage. The cost of capital thus becomes a critical factor in deciding which financing track to follow – debt, equity or a combination of the two. Early-stage companies seldom have sizable assets to pledge as collateral for debt financing, so equity financing becomes the default mode of funding for most of them.

Companies strive to attain the optimal financing mix, based on the cost of capital for various funding sources. Debt financing has the advantage of being more tax-efficient than equity financing, since interest expenses are tax-deductible and dividends on common shares have to be paid with after-tax dollars. However, too much debt can result in dangerously high leverage, resulting in higher interest rates sought by lenders to offset the higher default risk.

3.9.2 Capital budgeting :-

Capital budgeting, or investment appraisal, is the planning process used to determine whether an organization's long term investments such as new machinery, replacement machinery, new plants, new products, and research development projects are worth the funding of cash through the firm's capitalization structure (debt, equity or retained earnings). It is the process of allocating resources for major capital, or investment, expenditures.[1] One of the primary goals of capital budgeting investments is to increase the value of the firm to the shareholders.

Many formal methods are used in capital budgeting, including the techniques such as

- Accounting rate of return
- Average accounting return
- Payback period
- Net present value
- Profitability index
- Internal rate of return
- Modified internal rate of return
- Equivalent annual cost
- Real options valuation

These methods use the incremental cash flows from each potential investment, or project. Techniques based on accounting earnings and accounting rules are sometimes used - though economists consider this to be improper - such as the accounting rate of return, and "return on investment." Simplified and hybrid methods are used as well, such as payback period and discounted payback period.

Unit-4 Project scheduling and Control

4.1 Definition of Project scheduling,
4.2 Project controls and importance,
4.3 Network techniques of Project Management: Gantt chart, CPM, PERT .

4.1 Definition of Project scheduling :-

The **project schedule** is the tool that communicates what work needs to be performed, which resources of the organization will perform the work and the timeframes in which that work needs to be performed.

The project schedule should reflect all of the work associated with delivering the project on time. Without a full and complete schedule, the project manager will be unable to communicate the complete effort, in terms of cost and resources, necessary to deliver the project.

Online project management software allows project managers to track project schedules, resources, budgets and project related assets in real time. The project schedule can be viewed and updated by team members associated with the project, keeping everyone well informed on the overall project status.

In project management, a **schedule** is a listing of a project's milestones, activities, and deliverables, usually with intended start and finish dates. Those items are often estimated by other information included in the project schedule of resource allocation, budget, task duration, and linkages of dependencies and scheduled events. A schedule is commonly used in the project planning and project portfolio management parts of project management. Elements on a schedule may be closely related to the work breakdown structure(WBS) terminal elements, the Statement of work, or a Contract Data Requirements List.

In many industries, such as engineering and construction, the development and maintenance of the project schedule is the responsibility of a full-time scheduler or team of schedulers, depending on the size of the project. Though the techniques of scheduling are well developed, they are inconsistently applied throughout industry. Standardization and promotion of scheduling best practices are being pursued by the Association for the Advancement of Cost Engineering (AACE), the Project Management Institute (PMI), and the US Government for acquisition and accounting purposes.

It should be noted that project management is not limited to industry; the average person can use it to organize their own life. Some examples are:

- Homeowner renovation project
- Keeping track of all the family activities
- Coaching a team
- Planning a vacation
- Planning a wedding

Some project management software programs provide templates, lists, and example schedules to help their users with creating their schedule.

Before a project schedule can be created, the schedule maker should have a work breakdown structure (WBS), an effort estimate for each task, and a resource list with availability for each resource. If these components for the schedule are not available, they can be created with a consensus-driven estimation method like Wideband Delphi. The reason for this is that a schedule itself is an estimate: each date in the schedule is estimated, and if those dates do not have the buy-in of the people who are going to do the work, the schedule will be inaccurate.In order for a project schedule to be healthy, the following criteria must be met:

- The schedule must be constantly (weekly works best) updated.
- The EAC (Estimation at Completion) value must be equal to the baseline value.
- The remaining effort must be appropriately distributed among team members (taking vacations into consideration).

The schedule structure may closely follow and include citations to the index of work breakdown structure or deliverables, using decomposition or templates to describe the activities needed to produce the deliverables defined in the WBS.[7]

4.2 Project controls and importance :-

Definition of Project Controls:

There are many definitions of Project Controls used across industries and indeed across companies within industries. For the purposes of this portal, the field of project controls is defined as follows:

Project controls are the data gathering, management and analytical processes used to predict, understand and constructively influence the time and cost outcomes of a project or program; through the communication of information in formats that assist effective management and decision making. This definition encompasses all stages of a project or program's lifecycle from the initial estimating needed to 'size' a proposed project, through to reflective learning (lessons learned) and the forensic analysis needed to understand the causes of failure (and develop claims). Consequently, the project controls discipline can be seen as encompassing:

- Project strategy, undertaking planning and methods studies to help the PM optimize future outcomes;
- Scheduling including development, updating and maintenance;
- Cost estimation, cost engineering/control and value engineering;
- Risk management, including maintaining the risk register and risk analysis/assessment;
- Earned Value Management and Earned Schedule, including WBS, OBS and other breakdown structures,
- Document control;
- Forensic Assessment for required diagnosis of schedule and cost

- Supplier performance measurement / oversight (but excluding contract administration);
- The elements of a project management methodology that integrate these disciplines both within the 'controls' domain and with other project management functions;

Put simply, Project Controls encompass the people, processes and tools used to plan, manage and mitigate cost and schedule issues and any risk events that may impact a project. In other words, Project control is essentially equivalent to the project management process stripped of its facilitating sub-processes for safety, quality, organizational, behavioral, and communications management.

Importance of Project Controls:

The successful performance of a project depends on appropriate planning. The PMBOK Guide defines the use of 21 processes that relate to planning out of the 39 processes for project management.

The execution of a project is based on a robust project plan and can only be achieved through an effective schedule control methodology.

The development of a suitable Project Control system is an important part of the project management effort.

Furthermore, it is widely recognized that planning and monitoring plays a major role as the cause of project failures. Despite the continuous evolution in the project management field, it appears evident that the traditional approach still shows a lack of utilization of Project Controls and there have been a number of articles published to support the importance of control in the achievement of project objectives.

It has been proved time and again that Project performance can be improved if dedicated Project Controls systems are in place. An IBC 2000 Project Control Best Practice Study carried out by IPA identified that good Project Control practices reduce execution schedule slip by 15%. Project Controls cost range from 0.5% to 3% of total project, (including cost accounting), therefore, to break even, Project Control needs to improve cost effectiveness by around 2%. A sample study carried out by the IBC Cost Engineering Committee (CEC) in 1999, showed cost improvements for the projects in the study, was more than 10%. It is noted also that NPV (Net Project Value) also benefits from schedule improvements. Success factors are based on good Project Control practices, which result in good cost and schedule outcomes.

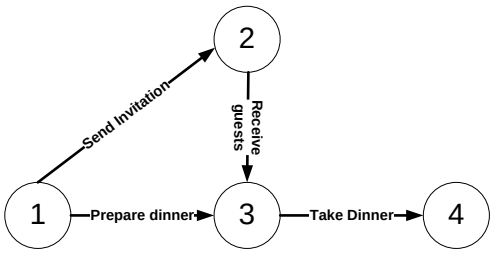
"The fact that one failed project can potentially wipe out an entire year's profit helps put the value of Project Controls into perspective."

4.3 Network techniques of Project Management: Gantt chart, CPM, PERT :-

Once the project is selected, the effort is shifted to the project implementation. In this Unit, we will see the network and scheduling techniques. They facilitate a proper planning, scheduling and the control of the project activities.

The network Diagram shows the activities and events of the project and their logical relationship and also referred as the project graph. The network Diagram is constructed into:

- An activity: task, job or function to be performed in the project.
- An event: is a specific point in time indicating the beginning or ending of one or more activities.



xample: A dinner Project :-

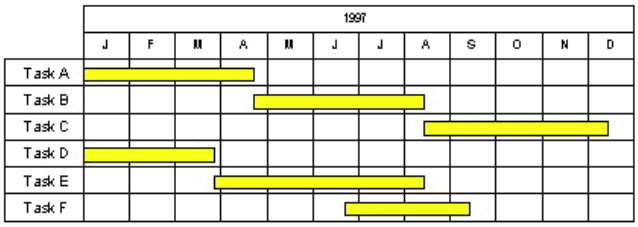
- Activities:
  - (1-2): Send Invitation.
  - (1-3): Prepare Dinner.
  - (2-3): Receive guests.
  - (3-4): Take dinner
- Events: (1), (2), (3), (4)

A network diagram is a graphical representation of all the activities of a project, placing them in their proper sequence and with all interdependencies clearly established. The network diagram provides a complete picture of the project.

4.3.1 Gantt Chart :-

Gantt charts are also called Bar charts. The use of Gantt charts started during the industrial revolution of the late 1800's. An early industrial engineer named Henry Gantt developed these charts to improve factory efficiency.

Gantt chart is now commonly used for scheduling the tasks and tracking the progress of energy management projects. Gantt charts are developed using bars to represent each task. The length of the bar shows how long the task is expected to take to complete. Duration is easily shown on Gantt charts. Sequence is not well shown on Gantt Charts

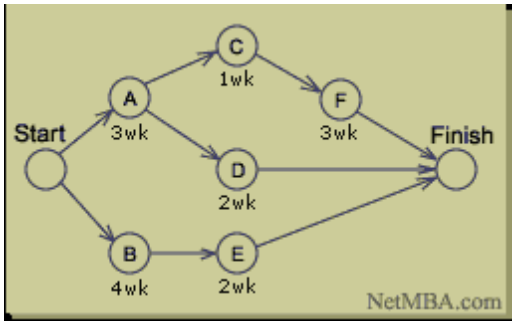


If, for example, the start of Task C depends on both Activity B and Activity E, then any delay to Task E will also delay Task C. We just don't have enough information on the Gantt chart to know this information.

4.3.2 CPM - Critical Path Method :-

DuPont developed a **Critical Path Method** (CPM) designed to address the challenge of shutting down chemical plants for maintenance and then restarting the plants once the maintenance had been completed.

Complex project, like the above example, require a series of activities, some of which must be performed sequentially and others that can be performed in parallel with other activities. This collection of series and parallel tasks can be modeled as a network.



CPM models the activities and events of a project as a network. Activities are shown as nodes on the network and events that signify the beginning or ending of activities are shown as arcs or lines between the nodes. The Figure 7.3 shows an example of a CPM network diagram:

Steps in CPM Project Planning:-

1. Specify the individual activities.
2. Determine the sequence of those activities.
3. Draw a network diagram.
4. Estimate the completion time for each activity.
5. Identify the critical path (longest path through the network)
6. Update the CPM diagram as the project progresses.

1. Specify the individual activities

All the activities in the project are listed. This list can be used as the basis for adding sequence and duration information in later steps.

2. Determine the sequence of the activities

Some activities are dependent on the completion of other activities. A list of the immediate predecessors of each activity is useful for constructing the CPM network diagram.

3. Draw the Network Diagram

Once the activities and their sequences have been defined, the CPM diagram can be drawn. CPM originally was developed as an *activity on node* network.

4. Estimate activity completion time

The time required to complete each activity can be estimated using past experience. CPM does not take into account variation in the completion time.

5. Identify the Critical Path

The critical path is the longest-duration path through the network. The significance of the critical path is that the activities that lie on it cannot be delayed without delaying the project. Because of its impact on the entire project, critical path analysis is an important aspect of project planning.

- ES - earliest start time: the earliest time at which the activity can start given that its precedent activities must be completed first.
- EF - earliest finish time, equal to the earliest start time for the activity plus the time required to complete the activity.
- LF - latest finish time: the latest time at which the activity can be completed without delaying the project.
- LS - latest start time, equal to the latest finish time minus the time required to complete the activity.

The *slack time* for an activity is the time between its earliest and latest start time, or between its earliest and latest finish time. Slack is the amount of time that an activity can be delayed past its earliest start or earliest finish without delaying the project.

The critical path is the path through the project network in which none of the activities have slack, that is, the path for which ES=LS and EF=LF for all activities in the path. A delay in the critical path delays the project. Similarly, to accelerate reduce the total time required for the activities in the critical path.

6. Update CPM diagram

As the project progresses, the actual task completion times will be known and the network diagram can be updated to include this information. A new critical path may emerge, and structural changes may be made in the network if project requirements change

CPM Benefits

• Provides a graphical view of the project.

• Predicts the time required to complete the project.

• Shows which activities are critical to maintaining the schedule and which are not.

CPM Limitations

While CPM is easy to understand and use, it does not consider the time variations that can have a great impact on the completion time of a complex project. CPM was developed for complex but fairly routine projects with minimum uncertainty in the project completion times. For less routine projects there is more uncertainty in the completion times, and this uncertainty limits its usefulness.

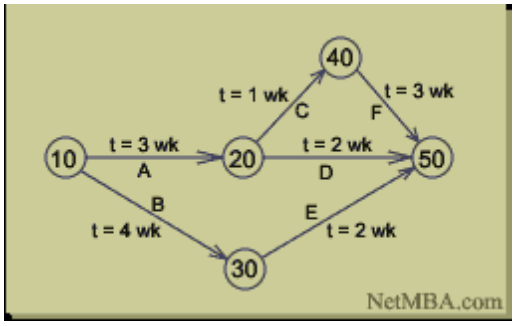
4.3.3 PERT :-

The *Program Evaluation and Review Technique* (PERT) is a network model that allows for randomness in activity completion times. PERT was developed in the late 1950's for the U.S. Navy's Polaris project having thousands of contractors. It has the potential to reduce both the time and cost required to complete a project.

The Network Diagram

In a project, an activity is a task that must be performed and an event is a milestone marking the completion of one or more activities. Before an activity can begin, all of its predecessor activities must be completed. Project network models represent activities and milestones by arcs and nodes.

PERT is typically represented as an *activity on arc* network, in which the activities are represented on the lines and milestones on the nodes. The Figure 7.4 shows a simple example of a PERT diagram.



The milestones generally are numbered so that the ending node of an activity has a higher number than the beginning node. Incrementing the numbers by 10 allows for new ones to be inserted without modifying the numbering of the entire diagram. The activities in the above diagram are labeled with letters along with the expected time required to complete the activity.

Steps in the PERT Planning Process

PERT planning involves the following steps:

1. Identify the specific activities and milestones.
2. Determine the proper sequence of the activities.
3. Construct a network diagram.
4. Estimate the time required for each activity.
5. Determine the critical path.
6. Update the PERT chart as the project progresses.

1. Identify activities and milestones

The activities are the tasks required to complete the project. The milestones are the events marking the beginning and end of one or more activities.

2. Determine activity sequence

This step may be combined with the activity identification step since the activity sequence is known for some tasks. Other tasks may require more analysis to determine the exact order in which they must be performed.

3. Construct the Network Diagram

Using the activity sequence information, a network diagram can be drawn showing the sequence of the serial and parallel activities.

4. Estimate activity times

Weeks are a commonly used unit of time for activity completion, but any consistent unit of time can be used. A distinguishing feature of PERT is its ability to deal with uncertainty in activity completion times. For each activity, the model usually includes three time estimates:

- Optimistic time (OT) - generally the shortest time in which the activity can be completed. (This is what an inexperienced manager believes!)
- Most likely time (MT) - the completion time having the highest probability. This is different from expected time. Seasoned managers have an amazing way of estimating very close to actual data from prior estimation errors.
- Pessimistic time (PT) - the longest time that an activity might require.

The expected time for each activity can be approximated using the following weighted average:

$$\text{Expected time} = (OT + 4 \times MT + PT) / 6$$

This expected time might be displayed on the network diagram.
Variance for each activity is given by:

$$[(PT - OT) / 6]^2$$

5. Determine the Critical Path

The critical path is determined by adding the times for the activities in each sequence and determining the longest path in the project. The critical path determines the total time required for the project.

Even if the activities outside the critical path speed up or slow down (within limits), the total project time does not change. The amount of time that a non-critical path activity can be delayed without delaying the project is referred to as slack time.

If the critical path is not immediately obvious, it may be helpful to determine the following four quantities for each activity:

- ES - Earliest Start time
- EF - Earliest Finish time
- LS - Latest Start time
- LF - Latest Finish time

These times are calculated using the expected time for the relevant activities. The ES and EF of each activity are determined by working forward through the network and determining the earliest time at which an activity can start and finish considering its predecessor activities.

The latest start and finish times are the latest times that an activity can start and finish without delaying the project. LS and LF are found by working backward through the network. The difference in the latest and earliest finish of each activity is that activity's slack. The critical path then is the path through the network in which none of the activities have slack.

The variance in the project completion time can be calculated by summing the variances in the completion times of the activities in the critical path. Given this variance, one can calculate the probability that the project will be completed by a certain date.

Since the critical path determines the completion date of the project, the project can be accelerated by adding the resources required to decrease the time for the activities in the critical path. Such a shortening of the project sometimes is referred to as project crashing.

6. Update as project progresses

Make adjustments in the PERT chart as the project progresses. As the project unfolds, the estimated times can be replaced with actual times. In cases where there are delays, additional resources may be needed to stay on schedule and the PERT chart may be modified to reflect the new situation.

Benefits of PERT

PERT is useful because it provides the following information:

- Expected project completion time.
- Probability of completion before a specified date.
- The critical path activities that directly impact the completion time.
- The activities that have slack time and that can lend resources to critical path activities.
- Activities start and end dates.

Limitations of PERT

The following are some of PERT's limitations:

- The activity time estimates are somewhat subjective and depend on judgment. In cases where there is little experience in performing an activity, the numbers may be only a guess. In other cases, if the person or group performing the activity estimates the time there may be bias in the estimate.
- The underestimation of the project completion time due to alternate paths becoming critical is perhaps the most serious.

Unit-5 Software Project Management

5.1 Nature of Software and Software Development,
5.2 Software between computers and people,
5.3 Software quality,
5.4 Software quality assurance,
5.5 Relation between software quality and software productivity,
5.6 Risk reduction,
5.7 Role of project manager in software development.

5.1 Nature of Software and Software Development :-

5.2 Software between computers and people :-

5.3 Software quality :-

5.4 Software quality assurance :-

5.5 Relation between software quality and software productivity :-

5.6 Risk reduction :-

5.7 Role of project manager in software development :-

Unit-6 Project Management Information System (PMIS)

6.1 Concept of PMIS,
6.2 Components of PMIs,
6.3 Choosing and implementing PMIS,
6.4 Benefit of PMIS,
6.5 Common errors managing PMIS.

6.1 Concept of PMIS :-

Project Management Information System is a tool used to document and store the project management plan, subsidiary plans and other documents / work products relevant for the project. It could be manual or automated and should support the change control procedures defined in the project.

A **project management information system** (PMIS) is the coherent organization of the information required for an organization to execute projects successfully. A PMIS is typically one or more software applications and a methodical process for collecting and using project information. These electronic systems "help [to] plan, execute, and close project management goals."[1] PMIS systems differ in scope, design and features depending upon an organisation's operational requirements.

At the center of any modern **PMIS** is a software. **Project management information system** can vary from something as simple as a File system containing Microsoft Excel documents, to a full blown enterprise PMIS software.

Characteristics of a PMIS Software[edit]

The methodological process used to collect and organize project information can match normalized methodologies such as PRINCE2.

A PMIS Software supports all Project management knowledge areas such as : Integration Management, Project Scope Management, Project Time Management, Project Cost Management, Project Quality Management, Project Human Resource Management, Project Communications Management, Project Risk Management, Project Procurement Management, and Project Stakeholders Management.[2]

A PMIS Software is a multi-user application, and can be cloud based or hosted on-premises.

6.2 Components of PMIs :-

Based on the above requirements the consultants developed five modules for assisting the project management team to improve the quality of the information used in management planning, control and reporting. The design principle using rapid prototyping techniques stressed the ease of use, use of existing formats and procedures while requiring the minimum of effort in maintaining the systems. The following paragraphs briefly describe the components that were developed and adopted. Efforts will be made in the future to integrate the components into a single system when the organization's local area network (LAN) is operational. A flowchart of the system is shown in the following fig

No fig

1. Project Performance Indicator Tracking System

The project managers are required to report to the World Bank the status of each Performance Indicator from the project's Hierarchy of Objectives and management actions taken towards their achievement. The report must be on a semi annual basis and in an agreed format. For the past few years, this report was compiled using a word processor. As the report has a column on comments by the project manager on the status of each indicator, the project manager had to devote considerable time in the production of this report.

The new Project Performance Indicator Tracking System (PPITS), stores the indicators in a database format, and codifies the status into five outcomes: Accomplished Successfully, Being Achieved (on course), Experiencing Minor Problems (being addressed), Experiencing Major Problems and Rescheduling of Target Date Required, and Not Yet Due. With a graphic interface in MS Access, the project manager updates the status of each indicator as events occur, inputs the comments, and records the management actions undertaken. At the end of each reporting period, the required report is produced automatically (an internal feature of the system) with all the up-to-date information.

2. Procurement Planning and Monitoring System (PPMS)

Procurement is a major part of the project activities. The procurement activity involves a number of discrete steps to be followed in sequence leading from development of specifications through bidding to contract signing with the suppliers who win the bid. The acquisition of goods and services essential to the implementation of the project will be delayed if the timetable of events is not followed. However, in many projects, procurement staffs simply follow the process in a step-by-step manner, finishing one step before tackling the next, with neither systematic planning nor tracking.

A good estimate of when project goods and services will become available can only be done when a contract is signed. The revised list of dates will be automatically updated if a constituent step for any item slips. Since the different methods of procurement for works, goods and services have different steps the PPMS uses different milestones for each type. It also produces a list of procurement activities for a specified time period, thus providing a calendar of all procurement activities required for the next month. This serves as a reminder of critical procurement tasks that the project staff has to perform on a day-to-day basis.

3. Disbursement Planning and Tracking System (DPTS)

With the recent introduction of the Loan Administration Change Initiative (LACI), project management units of World Bank supported projects have to furnish the Bank with accounting reports in a specified format. The tables in these reports require listing disbursements made in each quarter and the forecast of payments for the following quarter. The DPTS is a system designed to enable the planning of the payment schedule of each contract for works, goods and services and entering the dates of actual payments against this schedule. The system automatically analyzes the data and produces the reports in the required format. Together with the Procurement Planning and Monitoring System (PPMS), all the required LACI reports can be produced directly from the database.

4. Procurement Activity Tracking System (PATS)

Apart from the major contracts for the building of new schools and the major consultancies, each project management unit also undertakes a number of relatively small contracts for furnishing the new schools and for purchasing school supplies. These shopping activities include the following steps: Finalizing the initial specifications; contacting suppliers for price quotations; negotiating specification modifications, discounts and delivery dates; receiving shipments or verifying deliveries in terms of quantities and quality; and authorizing payments by the accountant.

Although the process for procuring an individual item is not complicated, when the number of items required for a particular date becomes large, there is a need for a database to keep track of the status of placement of orders and of deliveries so that suppliers can be paid promptly. Notification of authorization for payment is at present done off line, involving printing a list for payments authorized or passing a diskette to the Accountant.

5. Project Planning and Scheduling System (PP&SS)

A complete critical path based project plan and schedule was developed using MS project. A portion of this plan is shown in **Figure 3**. The first level of indenture is the WBS of the project. The schedule for the items of procurement, transferred from the PPMS, is presented on one line in the CPM chart using the rollup technique in MS Project.

6.3 Choosing and implementing PMIS :-

The system was designed and implemented in three phases. The first phase was requirements definition, basic training and design of a pilot system using rapid prototyping techniques. The second phase was fine-tuning of the system and on the job training. The third phase was higher-level training and assistance in updating the system after a period of a few months. Some additional modifications were also done at this time.

Conclusions

1. A simple Project Management Information System can be very useful in implementing a World Bank project. The discipline of a PMIS assists in avoiding many of the problems listed in **Figure 1**.
2. Procurement activities are better planned and managed using a Data Base Management System rather than a critical path plan and schedule since the activities are completely lineal.
3. Both formal and on-the-job training is needed in addition to systems development. Assistance in entering basic data the first time is also necessary.
4. New project managers placed in a matrix organization need assistance in team building to gain influence without authority in a functional ministry.

The following show additional screen shots of the system.

6.4 Benefit of PMIS :-

6.5 Common errors managing PMIS :-

I. Principles of Management

End of Semester Examination	40 Marks
Continuous Assessment	35 Marks
Total	75 Marks

Learning Objectives: - The objective of the course is to equip the students with the ability to analyse, interpret and apply the basic concepts and management thinker's theories in business management.
Utility: Student can acquire basic-level knowledge of management for beginners.
Prerequisite: Basic knowledge of management.

Unit. No.	Particulars	No. of Lectures
1	**Nature of Management: -** • Management**:** Meaning, Significance, objectives • Nature of Managements as science, as an Art and as a Profession. • Levels of Management: Top, Middle, Supervisor. • Contribution of Management thinkers - Taylor, Fayol and Elton Mayo.	**11**
2	**Planning** • Planning: Meaning, Features, Importance, planning process, types of planning, limitations of planning. •	**11**
3	**Organizing:** • Organizing: Meaning, Importance, Process, Principles, Structure of Organization. • Formal and Informal Organization, Types of Organization: Line, Functional, Line and Staff and Committee. • Delegation: Meaning, Elements of Delegation	**11**
4	**Staffing:** • Meaning, need, importance of staffing • Steps in Staffing Process. • Recruitment: Meaning & Sources. • Selection: Meaning & Procedure. • Training and Development: Meaning and Importance.	**11**
5	**Direction & Controlling:** • Meaning, Nature, Importance of Direction. • Types of Directing. • Supervision: Meaning & Importance. • Meaning and importance of controlling, • Steps in controlling process. • Budget as a control technique, quality control and cost control	**10**
	Total	**54**

Reference Books:

1. Principles of Management - Dr. S.C. Saxena – Sahitya Bhavan Publication
2. Principles of Management - T. Ramasamy – Himalaya Publication
2. Principles & Practices of Management -P. C. Shejwalkar & A. Ghanekar – Everest Publishing House
3. Managament - Koontz & Weihrich

Unit I

Nature of Management

Meaning of Management

Management is an art of getting things done through others and with formally organised groups

Management is art of knowing what you want to do and then seeing that they do it in the best and the cheapest manner

Management is distinct process consisting of planning, organising, actuating and controlling; utilizing in both science and arts and followed in order to accomplish pre-determined objectives

Management is to forecast, to plan, to organize, to command, to coordinate and control activities of others

Management is art of getting things done through people.

Management is defined as the process by which a cooperative group directs actions towards common goals.....

Management is a social and technical process which utilizes, resources, influences, human actions and facilitates changes in order to accomplish organisational goals.

1.1 Significance of Management

1. Optimum use of Resources

Management is process of making optimum utilization of available resources in order to achieve common goal of organization. In every organisation physical (material, machines, money, etc.) and human (manpower) resources are used. The function of management is nothing but to put them to optimum use or avoid wastage of resources. And it is possible only when a manager makes use of the principles of management.

2. Encourage the team work

Management encourages employees to give better performance. Team work productivity makes achievement of common goal.

Fulfilling Social Responsibility:

The principles of management increase the efficiency of the managers and enable them to fulfil their social responsibility. More efficient managers can make good quality products at reasonable prices available all the times. For example, the principle of 'Fair Remuneration' advocates appropriate remuneration to the employees and helps in fulfilling responsibility towards them

Meeting Changing Environmental Requirement:

In modern times only such a business unit can remain alive and develop which can keep pace with the rapidly changing environment. In every field of business activity (production, marketing, etc.) new techniques are emerging every day. Managers who can understand the changing situation well in time can alone face competition successfully. The principles of management enable the managers to face this challenge.

For example, the principle of 'unity of direction' tells us that all the activities connected with the attainment of the same objective should be completed under the direction of the same manager.

Decision Making

Principles of scientific management provide the necessary training to the managers so that they can take right decision making in business practices.

Effective Administration

The primary need of every organisation is that its administrative system should be effective. Only that administrative system is called effective where the managers can

take balanced and quick decisions. In order to take such decisions, sufficient information and authority should be available.

This is made possible with the help of the principles of management. For example, the principle of 'authority and responsibility' advocates granting of sufficient authority, while the principle of scalar chain advocates making adequate provision for providing information.

1.2 Objectives of Management

1. Maximize results with minimum efforts

   All management principles aim at ensuring the overall efficiency of the business. Management principles provide guidelines and directions as to how tasks are to be completed for increased efficiency with minimum efforts.

2. Coordination

   Coordination is an essential function of management. It allows for ensuring that the work done by the individual elements of the firm are combined for the furtherance of the collective objective of the firm. Most management principles aim at ensuring this function and facilitating coordination.

3. Increasing the Efficiency of factors of Production –

   Through proper utilization of various factors of production, their efficiency can be increased to a great extent which can be obtained by reducing spoilage, wastages and breakage of all kinds, this in turn leads to saving of time, effort and money which is essential for the growth & prosperity of the enterprise.

4. Maximum Prosperity for Employer & Employees

   - Management ensures smooth and coordinated functioning of the enterprise. This in turn helps in providing maximum benefits to the employee in the shape of good working condition, suitable wage system, incentive plans on the one hand and higher profits to the employer on the other hand.

5. Human betterment & Social Justice

   - Management serves as a tool for betterment of the society. Through increased productivity & employment, management ensures better standards of living for the society. It provides justice through its uniform policies

6. Optimum utilization of resources.
   The most important objectives of the management are to use various resources of the enterprise in a most economic way

7. Minimise the element of risk:
   Management involves the function of forecasting. Though the exact future can never be predicted yet on the basis of previous experience and existing circumstances, management can minimise the element of risk. Management always keeps its ears and eyes to the changing circumstances.

8. Planning for future:
   Another important purpose of management is to prepare a prospective plan. No management should feel satisfied with today's work. Future plans should take into consideration what is to be done next. Future performance will depend upon present planning. So, planning for future is essential to every organisation.

1.3 Nature of Management as Science

Its principles are exact and have universally accepted.

Scientific principles represent basic truth about a particular field of enquiry. These principles may be applied in all situations, at all time & at all places. E.g. - law of gravitation which can be applied in all countries irrespective of the time. Many Mathematical or statistical techniques are used by many organization across the world. Management also contains some fundamental principles which can be applied universally like the Principle of Unity of Command i.e. one man, one boss. This principle is applicable to all type of organization - business or non-business.

Its principles are evolved on the basis of continuous observation and experiment Scientific principles are derived through scientific investigation & researching
i.e. they are based on logic. E.g. the principle that earth goes round the sun has been scientifically proved.
Management principles are also based on scientific enquiry & observation and not only on the opinion of Henri Fayol. They have been developed through experiments & practical experiences of large no. of managers. E.g. it is observed that fair remuneration to personal helps in creating a satisfied work force.

Scientific principles are derived on the basis of logical and scientific observations: The scientists perform logical observation before deriving any principle or theory. They are

very objective while doing the observations. But when managers are observing they have to observe human beings and observation of human being cannot be purely logical and objective, some kind of subjectivity enters in the observations so this feature of science is not present in management. All the scientific principles have same effect, wherever we try them whereas effect of management principles varies from one situation to other

Systematic body of knowledge:
In science organised and systematic study material is available which is used to acquire the knowledge of science. Like science in management also there is availability of systematic and organised study material. So first feature of science is present in management

1.4 Nature of Management as Art

Art refers to the way of doing specific things; it indicates how an object can be achieved. In the words of George R. Terry, "Art is bringing about of a desired result through the application of skill." Art is, thus, skillful application of knowledge which entirely depends on the inherent capacity of a person which comes from within a person and is learned from practice and experience. In this sense, management is certainly an art as a manager uses his skill, knowledge and experience in solving various problems, both complicated and non-complicated that arise in the working of his enterprise successful.

Based on Practice and creativity:
The artist requires regular practice of art to become more fine and perfect. Without practice artists lose their perfection. Art requires creative practice, i.e., artist must add his creativity to the theoretical knowledge he has learned. Same way with experience managers also improves their managerial skills and efficiency. So this feature of art is also present in management.
Practical Knowledge
Knowledge refers to possession of facts and techniques of a particular field. Knowledge can be acquired through study and/or practical experience. Generally, in art, more emphasis is given on acquiring knowledge through practical experience. In management, knowledge is acquired both through study and experience. Thus, management is an art.

Thus the meaning of art is associated with the getting of a desired result through the application of skills. Art demands practical knowledge thus learning of theory is just not enough. It is crucial to know practical application of theoretical principles. Like any other art, management is also creative in nature. It fuses human and non-human resources in beneficial way in order to achieve desired results

1.5 Nature of Management as a Profession

The most characteristic feature of modern management is that it is gradually becoming a profession. A profession may be defined as an occupation that requires specialized knowledge and intensive academic preparations to which entry is regulated by a representative body. The essentials of a profession are:

1. Substantial body of accumulated knowledge / Specialized Knowledge – A profession must have a systematic body of knowledge that can be used for development of professionals. Every professional must make deliberate efforts to acquire expertise in the principles and techniques. Similarly, a manager must have devotion and involvement to acquire expertise in the science of management.
2. Use of specialised experts and professionals: in the modern age there is an increased tendency in favour of management by experts or professionals. Today there is a growing tendency to select and appoint highly qualified, trained and experienced persons to manage the business

3. Formal Education & Training – There are no. of institutes and universities to impart education & training for a profession. No one can practice a profession without going through a prescribed course. Many institutes of management have been set up for imparting education and training. For example, a CA cannot audit the A/C's unless he has acquired a degree or diploma for the same but no minimum qualifications and a course of study has been prescribed for managers by law. For example, MBA may be preferred but not necessary.

4 Social Obligations – Profession is a source of livelihood but professionals are primarily motivated by the desire to serve the society. Their actions are influenced by social norms and values. Similarly, a manager is responsible not only to its owners but also to the society and therefore he is expected to provide quality goods at reasonable prices to the society.

5. Code of Conduct – Members of a profession have to abide by a code of conduct which contains certain rules and regulations, norms of honesty, integrity and special ethics. A code of conduct is enforced by a representative association to ensure self discipline among its members. Any member violating the code of conduct can be punished and his membership can be withdrawn. The AIMA has prescribed a code of conduct for managers but it has no right to take legal action against any manager who violates it.

Almost management is belongs to profession but it is not fully fledged profession because of :

a. It does not restrict the entry in managerial jobs for account of one standard or other.

b. No minimum qualifications have been prescribed for managers.

c. No management association has the authority to grant a certificate of practice to various managers.

d. All managers are supposed to abide by the code formulated by AIMA,

f. Managers are responsible to many groups such as shareholders, employees and society. A regulatory code may curtail their freedom.

1.6 Levels of Management

Levels of Management can be classified into three levels

Top level management (Strategic)
Middle level management (Tactical)
Low level management (Operational)

1. Top Level Management(Strategic)

Top level is the head of the organisation. It consists of Board of Directors, Chief Executive, Managing Director or General Manager. They set the objective of organisation. Top level management involves-

i. Determine objectives: the common objective/goal of organisation is determined by top management considering survival, profit, business growth, prestige and social acceptance.
ii. Frame the Policies: top management frame the policies and chalk out plans to carry out the objectives and policies. Policies may relate to different aspects of the organisation. Production policy, marketing policy etc.

2. Middle Level Management(Tactical)

The Middle level Management is also referred to as the executory level, they are subordinates of the top-level management and are responsible for the organization and direction of the low-level management. They account for the top-level management for the activities of their departments.

The middle-level managers are semi- executives and are made up of the departmental managers and branch manager. They could be divided into senior and junior middle-level management if the organization is big. They coordinate the responsibilities of the sub-unit of the firm and access the efficiency of lower-level managers.

The middle-level managers are in charge of the employment and training of the lower levels. They are also the communicators between the top level and the lower level as they transfer information, reports, and other data of the enterprise to the top-level. Apart from these, there are three primary functions of the middle-level management in the organization briefed below:

- To carry out the plans of the organization according to policies and directives laid down by the top level management.
- To organize the division or departmental activities.
- To be an inspiration or create motivation for junior managers to improve their efficiency.

Low level Management(Operational)

The lower level Management is also referred to as the supervisory or the operative level of managers. They oversee and direct the operative employees. They spend most of their time addressing the functions of the firm, as instructed by the managers above them.

The lower level managers are the first line of managers as they feature at the base of operations, so they are essential personnel that communicates the fundamental problems of the firm to the higher levels. This management level is made up of the foreman, the line boss, the shift boss, the section chief, the head nurse, superintendents, and sergeants. They are the intermediary, they solve issues amidst the workers and are responsible for the maintenance of appropriate relationship within the organization. They are also responsible for training, supervising and directing the operative employees.

The lower level managers represent the management to the operative workers as they ensure discipline and efficiency in the organization. The duty of inspiration and encouragement falls to them, as they strengthened the workforce. They also organize the essential machines, tools and other materials required by the employees to get their job done.

Briefed below are the primary functions of the lower-level management:

- To allocate tasks and responsibilities to the operative employees.
- To ensure quality and be responsible for the production quantity.
- To communicate the goals and objective of the firm laid down by the higher level managers to the employees and also the suggestions, recommendations, appeals and information concerning employee problems to the higher level managers.
- To give instruction and guided direction to workers on their day to day jobs.
- To give periodic reports of the workers to the higher level managers.

1.7 F.W. Taylor's Principles of Scientific Management

1. Science, not Rule of Thumb:

This principle requires development and application of scientific methods. Taylor advocated that the traditional rule of thumb' methods should be replaced with the scientific methods.

Scientific methods should be used for the following purposes:

(a) To determine standard time required to do a job;

(b) To determine a fair day's work for the w

(c) To determine the best way of doing the work;

(d) To select standard tools and equipment, maintain standard working conditions, etc.

2. Scientific Selection, Training and Development of Workers:

The procedure for selection of workers should be designed scientifically. The errors committed at the time of selection may prove o be very costly later on. If we do not have right workers on the right job, the efficiency of the organisation will be reduced.

Therefore, every organisation should follow a scientific system of selection. The selected workers should be trained to avoid wrong methods of work. Management is responsible for the scientific education and training of workers. It must provide opportunities for the development of workers having better capabilities.

3. Harmony, not Discord (Conflict):

There should be harmony (not conflict) between the management and the workers. This requires change of mental attitudes of the workers and the management towards each other. Taylor called it mental revolution. When this mental revolution takes place, workers and management turn their attention towards increasing profits.

They do not quarrel about the distribution of profits.

4. Cooperation, not Individualism:

Scientific management is based on cooperation between management and workers, as also between workers themselves. Management can earn higher profits if the workers perform their jobs efficiently and thus ensure better quality, lower costs and larger sales. Workers on their part can earn higher wages if the management provides them with standard materials, standard tools, standardised working conditions, training in standard methods, etc.

Scientific management also promotes cooperation among workers and departments. As activities of all individuals and departments are directly or indirectly linked with one another, interruption of work at any stage would affect the work of many individuals and departments, resulting in lower production and lower wages. The fear of reduced earnings will force workers to cooperate with each other for the smooth working of their departments.

5. Maximum, not Restricted Output:

Both the management and workers should try to achieve maximum output in place of restricted output. This will be beneficial to both the parties. Maximum output will result in higher wages for the workers and greater profit for the management. Increased productivity is also in the interest of the consumers and the society at large.

6. **Equal Division of Responsibility between Management and Workers:** There must be equal division of responsibility between the managers and the workers. The management should assume responsibility for the work for which it is better suited. For instance, management should decide the method of work, working conditions, time for completion of work, etc., instead of leaving these to the discretion of workers.

Techniques of F. W. Taylor's Scientific Management

1. Functional Foremanship

F.W. Taylor has propounded the functional organisation. This form of organisation is totally based on the principle of specialisation and makes full utilisation of the expertise of various experts.

In a functional organization, work is divided into many small parts and each part is assigned to an expert. In this manner, all the benefits of specialization are availed of.

Taylor has suggested the division of the work of factory manager into two sub departments:

(i) Planning department, and (ii) production department. Four experts are appointed in each department. The experts in the planning department do planning and the experts in the production department help in production.

(i) Specialists of Planning Department and their Functions (a) Route Clerk:

This clerk ensures the sequence of completing a particular work, meaning thereby the stages it shall have to pass before being finalized. He also decides the job to be done for the day and where it is to be done.

(b) Instruction Card Clerk:

This clerk prepares the instruction cards for the workers and hands them over to the gang boss. These cards contain information about the nature of the work, procedure of doing it, material to be used and the details about machinery.

This clerk decides as to when a particular work is to be started and finished, meaning thereby as to what time the whole work will take place. It is also decided at the same time at what cost the product will be produced.

(d) Discipline Officer:

The discipline officer ensures that every work is being performed in a disciplined manner.

(ii) Specialists of Production Department and their Functions (a)

Gang Boss:

The workers are divided into various groups from the point of view of control. A group leader is selected who is known as the gang boss. He is expected to ensure that both the workers and the machines are fit enough for production and that the material required for their use has been made available to them.

(b) Speed Boss:

The main function of the speed boss is to ensure that all the workers are performing their job at the required or expected speed. If it is not so, the speed boss tries to find out the cause of slow speed and hence a solution for it.

(c) Repair Boss:

The main function of the repair boss is to keep the machines and tools in working condition.

(d) Inspector:

He inspects the things produced and compares their quality with the standard prescribed for them and tries to find out the difference. In case of unfavorable result he initiates corrective action.

2. Standardisation of Work

Standardisation means setting standards for different factors, after due deliberation. For example, the amount of work to be done by a worker in a day may be standardised. In other words, the worker is expected to do the standard amount of work every day. In the same manner standards may also be set for raw materials, machines and tools, techniques, conditions of work, etc. The following is a brief description of such standards:

4. Scientific Study of Work

It means to conduct the deep analysis of all the activities being performed in the organization with the aim of producing maximum possible quality output at minimum costs. Taylor has conducted the following studies:

(i) Method Study, (ii) Motion Study, (iii) Time Study, and (iv) Fatigue Study.

(i) Method Study:

It refers to identify the most suitable way to do a particular activity. To conduct this study, process chart and operation research techniques are used. The main objective of this study is to minimize the cost of production and maximize the quality and level of consumer satisfaction.

(ii) Motion Study:

It refers to conduct the study of motions being performed by workers and machines while doing the job. The movie camera is used to conduct this study. The main objective of this study is to eliminate the unnecessary motions.

For example, during an experiment it was found that while laying a brick, a mason was conducting 18 different activities, but after eliminating the unnecessary activities the number of activities could be reduced to five, and in certain cases even down to two activities.

(iii) Time Study:

It refers to determine the standard time required to complete a particular activity. The standard time is determined on the basis of average time taken by the several experiences of the same work.

This study is conducted with the help of a stopwatch. The main objectives of the study are (i) to get the estimated figure of labour costs, (ii) to determine the number of required workers and (iii) to decide about the suitable incentive plan.

(iv) Fatigue Study:

It refers to determine the duration and frequency of rest intervals to complete a particular job. The rest refreshes the workers. They work again with their full capacity.

The main objective of this study is to maintain the efficiency level of workers. There may be so many causes of fatigue, such as long working hours, poor working conditions, unsuitable work, unhappy relations with the boss, etc.

5. Differential Wage System/Differential Piece Rate

Taylor has advised the adoption of differential wage systems in order to motivate the employees. According to this system, wages are paid on the basis of work done and not on the basis of time spent in doing the work.

In this system two different wage rates are used: one is the high wage rate and the other is the low wage rate. Those workers who are able to produce the standard number of units within a fixed duration are paid as per the high wage rate, and those workers who are not able to produce the standard number of units within the same time are paid as per the lower wage rate.

6. Mental Revolution

Mental revolution calls for a change in the mindset of both employers and workers. As per Taylor, a revolution in mindset of both the employers and the workers is required because it will promote feeling of cooperation, and will be beneficial to both the parties.

Normally, it is seen that conflict between employers and workers results in division of profits, with both the parties demanding a larger share of profits.

1.8 Henri Fayol's 14 Principle of Management.

1. Division of labour

Henri fayol has stressed on the specialization of jobs

He recommended that work of all kinds must be divided and subdivided and allotted to various persons according to their expertise in a particular area.

Subdivision of work makes it simpler and results in efficiency

It also helps the individual in acquiring speed, accuracy in his performance.

2. **Authority and Responsibility**

If authority is given to a person, he/she should be able to complete responsibilities.
In a same way, if anyone is made responsible for any job, he should also have concerned authority. Authority refers to the right of superiors to complete the task from people exactly. Authority without responsibility leads to irresponsible behaviour whereas responsibility without authority makes the person ineffective.

3 .Principle of one Boss or Unity of Command

A subordinate should receive orders and be accountable to one and only one boss at a time.

In other words a subordinate should not receive instructions from more than one person because it create confusion, overlapping of efforts etc.

It creates harmonious relationship between superiors and sub-ordinates.

4 . Unity of Direction

Fayol advocates one head one plan which means that there should be one plan for a group of activities having similar objectives.

Related activities should be grouped together. There should be one plan of action for them and they should under the charge of a particular manager Without unity of direction, unity of action cannot be achieved.

In fact, unity of command is not possible without unity of direction.

5. Equity

It means combination of fairness, kindness and justice

Manager should be fair and impartial (no partiality) while dealing with the subordinates.

They should not discriminate with respect to age, caste, sex, religion, relation etc.

6. Order

It concerned with proper and systematic arrangement of things and people. Arrangement of things is called material order and placement of people is called social order.

Material order – there should be safe, appropriate and specific place for every article and every place to be effectively used for specific activity and commodity. Social order-

selection and appointment of most suitable person on the suitable job. There should be a specific place for everyone and everyone should have a specific place so that they can easily be contacted whenever need arises.

7. Discipline

Discipline means sincerity, obedience, respect of authority and observance of rules and regulations of the enterprise.

This principle applies that subordinate should respect their superiors and obey their order.

8. Initiative

Workers should be encouraged to take initiative in the work assigned to them.

It means without any instruction workers or employees should complete the task.

It helps in developing an atmosphere of trust and understanding.

They can be encouraged with help of monetary and non-monetary incentives.

9. Fair remuneration

This management principle of the 14 principles of management argues that the remuneration should be sufficient to keep employees motivated and productive. There are two types of remuneration namely non-monetary (a compliment, more responsibilities, credits) and monetary (compensation, bonus or other financial compensation). Ultimately, it is about rewarding the efforts that have been made.

10. Stability of tenure

Fayol says employees should not be moved frequently from one job position to another i.e. period of service in job should be fixed. Appointing of new employees causes the wastage of money and time spend on training. Employees should be retained for longer period of time.

11. Scalar Chain

Hierarchy presents itself in any given organization. This varies from senior management (executive board) to the lowest levels in the organization. Henri Fayol 's "hierarchy" management principle states that there should be a clear line in the area of authority (from top to bottom and all managers at all levels). This can be seen as a type of management structure. Each employee can contact a manager or a superior in an emergency situation without challenging the hierarchy. Especially, when it concerns reports about calamities to the immediate managers/superior

12. Sub-ordination of individual interest to general interest

There are always all kinds of interests in an organization. In order to have an organization function well, Henri Fayol indicated that personal interests are subordinate to the interests of the organization (ethics). The primary focus is on the organizational objectives and not on those of the individual. This applies to all levels of the entire organization, including the managers.

13. Centralisation and Decentralisation

Management and authority for decision-making process must be properly balanced in an organization. This depends on the volume and size of an organization including its hierarchy. Centralization implies the concentration of decision making authority at the top management (executive board). Sharing of authorities for the decision-making process with lower levels (middle and lower management), is referred to as decentralization by Henri Fayol. Henri Fayol indicated that an organization should strive for a good balance in this.

14. Espirit De Corps

It concerned with harmony in the work and mutual understanding among the employees. Managers are responsible for the development of morale in the workplace; individually and in the area of communication. Esprit de corps contributes to the development of the culture and creates an atmosphere of mutual trust and understanding.

1.9 Elton Mayo's Hawthorne Experiments

Elton Mayo conducted the research on worker's working conditions and then monitored how the working conditions affected the workers morale and productivity. The changes in working conditions included changes in working hours, rest brakes, lighting, humidity and temperature. The changes were explained to the workers prior to implementation.

Some of the major phases of Hawthorne experiments are as follows:

1. Illumination experiments
2. Relay Assembly Test Room Experiments
3. Mass Interviewing Programme
4. Bank Wiring Observation Room Experiment

1. Illumination Experiments:

Illumination experiments were undertaken to find out how varying levels of illumination(amount of light at the workplace, a physical factor) affected the productivity. The assumptions was that with higher illumination, productivity will increase. In the first series of experiments, a group of workers was chosen and placed in two separate groups. One group was exposed to varying intensities of illumination.

Since this group was subjected to experimental changes, it was termed as experimental group. Another group called as control group, continued to work under constant intensities of illumination. The researchers found that as they increased the illumination in the experimental group, both group increased production. When intensity of illumination decreased, the production continued to increase in both the groups.

The production in the experimental group decreased only when the illumination was decreased to the level of moonlight. The decrease was due to light falling much below the normal level.

Thus, it was concluded that illumination did not have any effect on productivity but something else was interfering with the productivity. At that time, it was concluded that human factor was important in determining productivity but which aspect was affecting. It was not sure. Therefore, another phase of experiments was undertaken

2. Relay Assembly Test Room Experiments

Relay assembly test room experiments were designed to determine the effect of changes in various job conditions on group productivity. The researchers setup a relay assembly test room and two girls were chosen.

These girls were asked to choose for more girls as co-workers. The work related to the assembly of telephone relays. Each relay consisted of number of parts which girls assembled into finished products. Output depended on the speed and continuity with which girls worked. The experiments started with introducing numerous changes in sequence with duration of each change ranging from four to twelve weeks.

An observer was associated with girls to supervise their work. Before each change was introduced, the girls were consulted. They were given opportunity to express their viewpoints and concerns to the supervisor. In some cases, they were allowed to take decision on matters concerning them.

Following are the changes and resultant outcomes:

1. The incentive system was changed so that each girl's extra pay was based on the other five rather than output of larger group. The productivity increase as compared to before
2. The rest period was reduced to five minutes but frequency was increased. The productivity decreased slightly and the girls complained that frequent rest intervals affected the rhythm of the work.

3. The number of rest was reduced to two of ten minutes of each, but in the morning, coffee or soup was served along with the sandwich and in the evening, snack was provided. The productivity increased.
4. Working time changed or reduced by half hour or one hour.

As each change was introduced, absenteeism decreased, morale increased, and less supervision was required. It was assumed that these positive factors were there because of the various factors being adjusted and making them more positive. At this time, the researchers decided to revert back to original position, that is , no rest and other benefits. Surprisingly, productivity increased further instead of going down. This experiment results as the physical factors were not reasons of productivity but changes made due to attitude of girls towards work and their work group.

3. Mass interviewing programme :

During the course of experiments, about 20000 interviews were conducted between 1928 to 1920 by determine employees attitudes towards company, supervision, increase plans promotions and wages. Initially, these interviews were conducted by means of direct questioning such as do you like your supervisor?

this method has disadvantage of stimulating antagonism or the oversimplified 'yes' or 'no' responses which could not get to the root of the problem, the method was changed to non- directive interviewing where interviewer was asked to listen to instead of talking, arguing or advising. The interview programme gave valuable insights about the human behaviour in the company.

4. Bank Wiring Observation Room Experiment:

These experiments were conducted to find out the impact of small groups on the individuals. In this experiment, a group of 14 male workers were formed into a small work group. The men were engaged in the assembly of terminal banks for the use in telephone exchanges.

The work involved attaching wire with switches for certain equipment used in telephone exchanges. Hourly wage for each worker was fixed on the basis of average output of each worker. Bonus as also payable on the basis of group effort.

It was expected that highly efficient workers would bring pressure on less efficient workers to increase output and take advantage of group incentive plan. However, the strategy did not work and workers established their own standard of output and this was enforced vigorously by various methods of social pressure. The workers cited various reasons for this behaviour viz fear of unemployment, fear of increase output.

Unit-II Planning

According to Theo Haimann, "Planning is deciding in advance what is to be done. When a manager plans, he projects a course of action for further attempting to achieve a consistent coordinate structure of operations aimed at the desired results.

2.2 Features/Characteristics of Planning

- **Goal-Oriented Approach:** The features of planning can put to be a goal oriented approach because it is directly related with goals of organization. In the absence of goals or targets we cannot assume about planning. Because within the organization all the various activities are done with a view to attain the primary goals of organization. In fact in the lack of goals there is no need to do plan towards any activity. On the whole we can say; planning is a goal oriented approach.

- **Forward Looking Process:** It is a salient feature of planning. Planning is forward looking process because it is directly concerned with forecasting of future events. The purpose of every plan is to determine the future events with a view to bring certainty in the present activities of the organization. Thus we can say that planning is forward looking process.

- **Choice Making Activity:** Planning can set to be a choice making activity because planning provides a large number of alternative courses to an organization to do a various activities in the most appropriate manner. It is matter of fact that an action may be performed by many ways but there is particular way to perform an action which is most appropriate to organization. Planning provides these alternatives to organization to perform its various actions in the most proper way. Thus, planning is a choice making activity.

- **Primary Function:** The feature of planning refers to primary function because planning is the first task of business management. All other functions of business management such as organizing, staffing, directing & controlling are done after completing the planning. Infect, whenever we are going to do any event or activity at first we make plan & accordingly we will perform to attain result oriented work. Thus we can say to plan as primary function.

- **Planning is a Continuous Process:** Planning can put to be a continuous process because it is not an event which is concerned with the point of time. It is needed every walk of life of an organization. Infect, in the absence of planning any organization cannot able to take even one step properly. Thus, it is clear that planning is a continuous process.

- **Pervasive Function:** It is a unique feature of planning. Planning is a pervasive function because it can universally applicable at all the levels of management either it may be like top level of management, middle level of management or lower level of management. Infect planning is the function of all managers. So that, planning is a pervasive function.

- **Planning is a Dynamic process:** Planning can be treated as a dynamic process because it is affected if any changes are made in the internal & external environment of organization.

Such as if the goals of organization will change then it also affects the plans of organization. Similarly, in external environment if any changes are made in the government policy, regarding business then it also affects the plans of organization. Therefore, it is clear that planning can put to be as a dynamic process.

2.2 Importance/Significance of Planning:

1. Planning provides Direction:

Planning is concerned with predetermined course of action. It provides the directions to the efforts of employees. Planning makes clear what employees have to do, how to do, etc. By stating in advance how work has to be done, planning provides direction for action. Employees know in advance in which direction they have to work. This leads to Unity of Direction also. If there were no planning, employees would be working in different directions and organization would not be able to achieve its desired goal.

2. Planning Reduces the risk of uncertainties:

Organizations have to face many uncertainties and unexpected situations every day. Planning helps the manager to face the uncertainty because planners try to foresee the future by making some assumptions regarding future keeping in mind their past experiences and scanning of business environments. The plans are made to overcome such uncertainties. The plans also include unexpected risks such as fire or some other calamities in the organization. The resources are kept aside in the plan to meet such uncertainties.

3. Planning reduces over lapping and wasteful activities:

The organizational plans are made keeping in mind the requirements of all the departments. The departmental plans are derived from main organizational plan. As a result there will be coordination in different departments. On the other hand, if the managers, non-managers and all the employees are following course of action according to plan then there will be integration in the activities. Plans ensure clarity of thoughts and action and work can be carried out smoothly.

4. Planning Promotes innovative ideas:

Planning requires high thinking and it is an intellectual process. So, there is a great scope of finding better ideas, better methods and procedures to perform a particular job. Planning process forces managers to think differently and assume the future conditions. So, it makes the managers innovative and creative.

5. Planning Facilitates Decision Making:

Planning helps the managers to take various decisions. As in planning goals are set in advance and predictions are made for future. These predictions and goals help the manager to take fast decisions.

6. Planning establishes standard for controlling:

Controlling means comparison between planned and actual output and if there is variation between both then find out the reasons for such deviations and taking measures to match the actual output with the planned. But in case there is no planned output then controlling manager will have no base to compare whether the actual output is adequate or not.

For example, if the planned output for a week is 100 units and actual output produced by employee is 80 units then the controlling manager must take measures to bring the 80 unit production upto 100 units but if the planned output, i.e., 100 units is not given by the planners then finding out whether 80 unit production is sufficient or not will be difficult to know. So, the base for comparison in controlling is given by planning function only.

7. Focuses attention on objectives of the company:

Planning function begins with the setting up of the objectives, policies, procedures, methods and rules, etc. which are made in planning to achieve these objectives only. When employees follow the plan they are leading towards the achievement of objectives. Through planning, efforts of all the employees are directed towards the achievement of organizational goals and objectives.

2.3 Planning Process/Steps in planning process

1. **Analysis of External environment**

   External environment analysis is related to analysis of the factors which are beyond the control of organization. Organization cannot control these factors. Before actual planning, the management must carefully analyse the external factors which influence the business. These factors are uncontrollable and unpredictable. It includes analysis of social factors, political factors(Govt. rules, tax policy etc.), economic factors(inflation, recession etc.)

2. **Analysis of internal environment**

   It is related to analysis of the factors which are under the control of organization. These controllable factors consist of plans, policies, strategies, budgets etc. are adjusted or implemented using internal resources of organization related to men, money, material, machines and managerial man power etc. It gives answers of following questions do we have enough resources? do we have up-to-date plant, equipments and work environment?
   Is our profitability?
   Is our managerial leadership effective?

3. **Define the organization mission**

   Mission of an organization gives the direction. It related to the fundamental reason for the existence of an organization. Mission of an organization directs and makes favourable environment to the planning

4. **Determination of objectives**

   Objectives are formulated according to various departments and sections. Objectives indicates what basically is to be done, where the primary emphasis is to be placed and what is to be accomplished by the network of policies, procedures, rules, budgets, programmes and strategy

5. **Forecasting planning premises**

   For preparing planning it is essential to forecast some major areas which has fundamental importance. These are General business forecasts, sales forecast, capital forecast, demand forecast etc. Forecasting is a systematic investigation of various known facts relating to the past and future for the prediction of future.

6. **Determining alternative course of actions or plans**

   It is related to formulating various courses of action that are suitable and help to achieve the objectives of organization. Management should design alternative courses of actions that belongs to planning premises.

7. **Evaluating alternative courses of actions**

   It concerned with measuring of various alternatives to know whether they are favourable and important for business objectives.

8. **Selecting the best course or developing strategies**

   This is the important point of decision making. Framing selecting best suitable alternatives for business is called as strategy development.

9. **Establishing the sequence of activities**

   After the best alternative selection, it is essential to formulate number of courses of actions for each department for each month, quarter, weak etc.

10. **Formulation of strategic or long range plan**

    It covers all important areas of business activities such as profit, capital expenditure for growth or diversification, organization structure managerial philosophy, pricing, leadership in the market finance personnel advertising industrial relations products production planning and development, R & D, CSR corporate social responsibilities.

11. **Formulation of tactical plan**

    Purchase, production, marketing, R & D. these are the areas where plans designed as per mission

12. **Formulation of action programmes**

    Allocation of tasks, responsibilities authorities to achieve objectives, design schedule for executives.

2.4 Types of Plans

Standing Plans

- Mission
- Objectives
- Policies
- Procedures
- Rules and Methods
- Strategies

Single Use Plans

- Programmes
- Projects
- Schedules
- Budgets
- Standards

Mission

Every organization must have a mission then only it becomes meaningful or purposeful. Mission or purpose, often used interchangeably, identifies the basic task of an organization for which it is created. For example, the mission of a University is to impart higher education. The mission of the garments factory is to produce and sell ready-made garments and so on

Policies

Policies are, in most cases, standing plans. As a matter of fact, policies provide guidelines for repetitive actions. They define an area or provide limits within which decisions are to be made and ensure that the decision will be consistent with, and contribute to, an objective. Policies are types of plans that allow decision-makers some discretion to carry out a plan. Otherwise, there will be no difference between policies and rules. Policies must allow for some discretion. Policies help decide issues before they become problems and make it unnecessary to analyze the same situation every time it comes up. It permits managers to delegate authority and still maintain control over subordinates about the matter

Strategy

The strategy is another type of broad-based standing plan which helps the determination of the basic long-term objectives of an enterprise and adoption of courses of action and allocation of resources necessary to achieve these objectives.

Objectives

This is the first step in planning the action plan of the organization. Objectives are the basics of every company and the desired objective/result that the company plans on achieving, so they are the endpoint of every planning activity. For example, one of the objectives of an organization could be to increase sales by 20%. So the manager will plan all activities of the organization with this end objective in mind. While framing the objectives of the organization some points should be kept in mind. **Rules**

Rules Like policies, rules, too, are standing plans that guide action. Rules spell out specifically what employees are supposed to do or not to do. For example, the no-smoking campaign launched by some organizations is supported by some organizational rules. As opposed to policies, rules do not permit the exercise of individual discretion. Instead, rules specify what actions will be taken (or not taken) and what behaviour is permitted or not. Policies, on the other hand, tell people how to think about decisions to be made about actions

Procedures

Procedures Like rules, procedures are standing plans that provide guidance for action rather than speculation. They are plans that establish a required method of handling future activities. Procedures establish customary ways for handling certain activities like hiring a clerk, promoting employees, obtaining a loan from a bank. The major characteristic of a procedure is that it represents a chronological sequencing of events. It specifies a series of steps that must be taken to accomplish a task. Specified series of steps that are required to be taken for admission into the MBA program of AUB is an example of the procedure.

Budgets

It covers a particular period of time, and once the period is over, a new budget comes into being. It not only a planning tool but also works as a controlling tool.

Programs

Programs are plans of action followed in proper sequence according to objectives, policies, and procedures. Thus a program lays down the major steps to be taken to achieve an objective and sets an approximate time frame for its fulfilment. Programs are usually supported by budgets. A program may be a major or a minor one or long, medium or short-term one. Since it is not used in the same form once its task is over it belongs to single-use plan category.

Projects

A project is a particular job that needs to be done in connection with a general program. So a single step in a program is set up as a project. A project has a distinct object and clear-cut termination. "Projects have the same characteristics as programs but are generally narrower in scope and less complex. Projects are frequently created to support or complement a program."

2.5 Limitations of Planning

Unreliability of forecasts:

Forecasting made on the basis of past records and some statistical methods. But number of environmental factors change during execution periods. Hence the forecast become unreliable and redundant.

Time consuming

Planning process involves a number of steps and forecast based on analysis and applying statistical methods on past data. Planning process requires lot of time to perform all these activities and to make planning effective.

High Cost

Planning process require more time and resources to make it effective. For forecasting and execution of various activities more personnel are required. Thus planning become more expensive.

Organizational politics

Though the planning process is a systematic and sequential process sometimes few of leaders in organization dominate planning process. Thus in organization politics can be possible.

Inflexibility

Liberalization and globalization are the major effective factors towards change. But sometimes due to planning process it is very difficult to make change as per external environment. Planning makes most of the organizational activities inflexible and static

Unit-III Organizing

Organizing is the process of arranging and allocating work, authority and resources among an organization's members so that they can achieve organization goal.

Stoner, freeman and gilbert

The process of arranging people and other resources to work together to accomplish a goal.

3.1 Importance of Organizing:

Promotes Effective Communication

Organizing is an important means of creating coordination and communication among the various departments of the organization. Different jobs and positions are interrelated by structural relationship. It specifies the channel and mode of communication among different members.

Efficient Administration:

It has generally been observed that there is always a condition of doubt about the authority of the managers among themselves. The process of organising makes a clear mention of each and every activity of every manager and also of their extent of authority

Expansion and Growth:

The process of organising allows the employees the freedom to take decisions which helps them to grow. They are always ready to face new challenges. This situation can help in the development of the enterprise. This helps in increasing the earning capacity of the enterprise which in turn helps its development.

Adaptation to Changes

Organising helps business enterprises to adjust to the environmental changes. In order to ensure smooth functioning of the enterprise, organising suitably modifies the organisational structure and various inter-relationships existing in the enterprise. Moreover, it also provides stability in the enterprise so that it continues to grow and survive in spite of changes.

Clarity in Working Relationship:

Organizing clarifies the working relations among employees. It specifies who is to report to whom. Therefore, communication becomes effective. It also helps in fixing accountability

Optimum Utilization of Resources:

Under the process of organizing the entire work is divided into various small activities. There is a different employee performing every different job.

By doing so, there is no possibility of any activity being left out or any possibility of unnecessary duplicating any job. Consequently, there is optimum utilization of all the available resources (e.g., material, machine, financial, human resource, etc.) in the organization.

3.2 Principles of organizing

Principle of objective: the organization is formed with the specific goal or mission, purpose and objective like constructing building for profit.

Principle of specialization and division of labor: the organization activities are divided into departments and units and task and each employee may be assigned with each task. Employees can specialise in that activity

Principle of coordination: all the activities, units and departments should be coordinated with the organisation's objective, purpose and mission.

Principle of co-equal authority and responsibility : Authority and responsibility are equally delegated. Otherwise subordinate may misuse authority, if more authority than required to discharge the responsibility is delegated. Alternatively, the subordinate can't discharge responsibility , if responsibility is more than the authority.

Span of control: span of control is number of subordinates, that can be managed efficiently by a superior. The span should be optimum depending upon organisational requirements.

Principle of continuity: organisation's should continue their operations, therefore, they should be flexible and responsive to the environmental changes.

Principle of teamwork: the purpose of organisation is to achieve it's mission. Achieving mission. Strategy and goal efficiently is possible through a collaborative work of a team of people even though the organisation is departmentalised.

Principle of efficiency: organisations should be designed in such a way that they should perform their activities with least cost and achieve highest return. Therefore, organisations should perform most efficiently.

3.3 Process of Organizing

1. Division of work
2. Grouping of Work
3. Delegation of Authority
4. Establishing Reporting Relationship

1. Division of Work

The division of work into smaller jobs leads to specialization because jobs are assigned to individuals according to their qualifications and capabilities. The division of work leads to systematic working. For example, in a bank every individual is assigned a job. One cashier accepts cash, one cashier makes payments, one person issues cheque books, one person receives cheques, etc. With division of work into jobs the banks work very smoothly and systematically.

2. Grouping of Work

Related and similar jobs are grouped together and put under one department. The departmentalization or grouping of jobs can be done by the organisation in different ways. But the most common two ways are:

(a) Functional departmentalization.

Under this method jobs related to common function are grouped under one department. For example, all the jobs related to production are grouped under production department; jobs related to sales are grouped under sales department and so on

(b) Divisional departmentalisation:

When an organisation is producing more than one type of products then they prefer divisional departmentalisation. Under this jobs related to one product are grouped under one department. For example, if an organisation is producing cosmetics, textile and medicines then jobs related to production, sale and marketing of cosmetics are grouped under one department, jobs related to textile under one and so on.

3. Delegation of Authority

Each individual working in different departments is assigned a duty matching to his skill and qualifications. The work is assigned according to the ability of individuals. Employees are assigned duties by giving them a document called job description. This document clearly defines the contents and responsibilities related to the job.

4. Establishing Reporting Relationship:

After grouping the activities in different departments the employees have to perform the job and to perform the job every individual needs some authority. So, in the fourth step of organising process all the individuals are assigned some authority matching to the job they have to perform.

The assignment of the authority results in creation of superior-subordinate relationship and the question of who reports to whom is clarified. The individual of higher authority becomes the superior and with less authority becomes the subordinate.

3.4 Structure of organisation

The system of tasks, workflows, reporting relationships, and communication channels that link together diverse individuals and groups.

Vertical organisation

Horizontal organisation

Vertical organisation

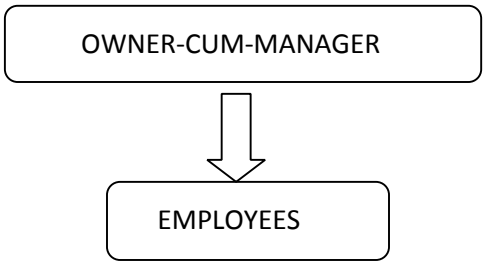
Vertical organisation refer to increase in the length of organisation's hierarchical chain of command. The hierarchical chain of command represents the company's authority-accountability relationship between superiors and subordinates. Authority and responsibility flows from the top to bottom through all the levels of the hierarchy. Accountability flows from the lowest level to the highest level.

Horizontal organisation

Horizontal organisation refers to an increase in breadth of an organisation's structure. The number of levels in the organisation's hierarchy are few. The span of control is relatively large. Authority is more decentralised in relatively flat structures.

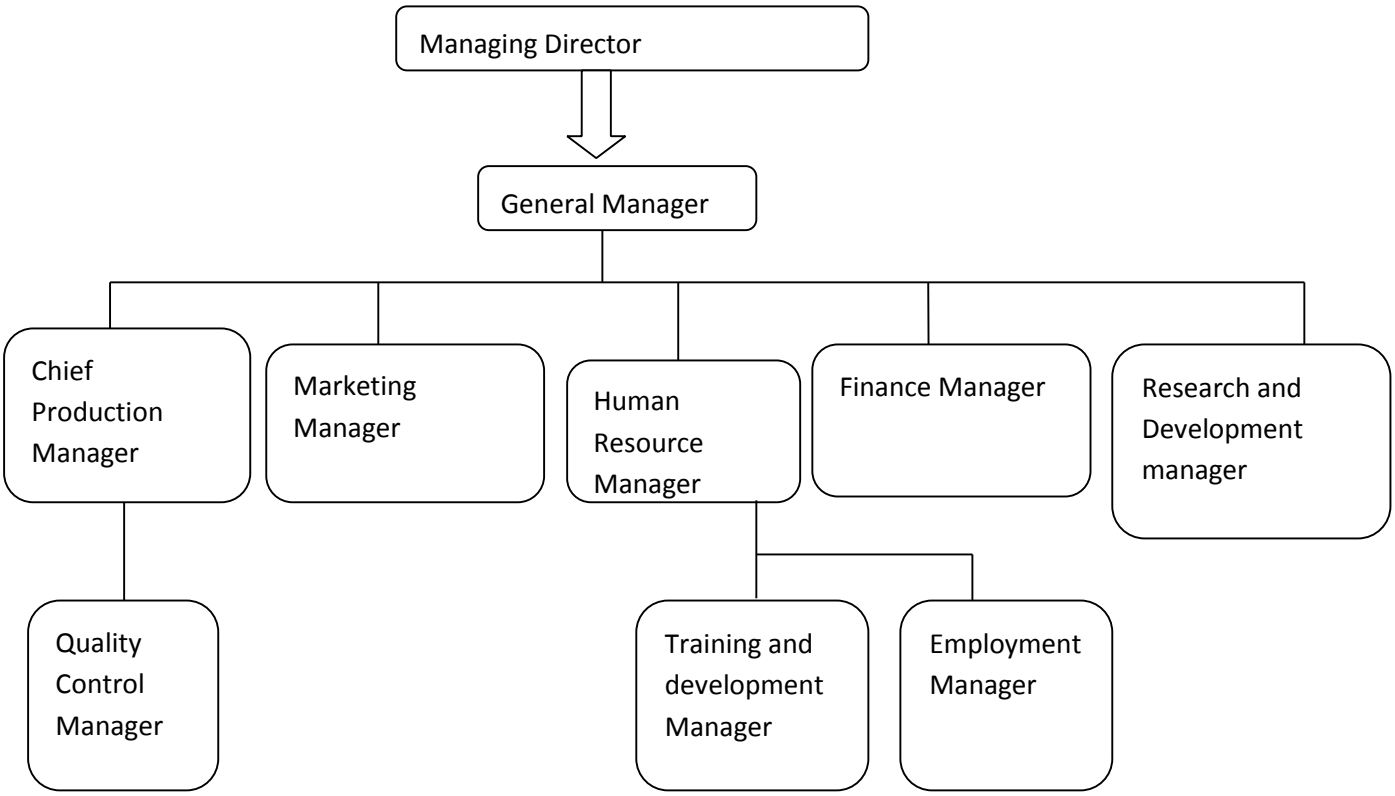
Types /Approaches of Organisation Structure

Entrepreneurial Structure



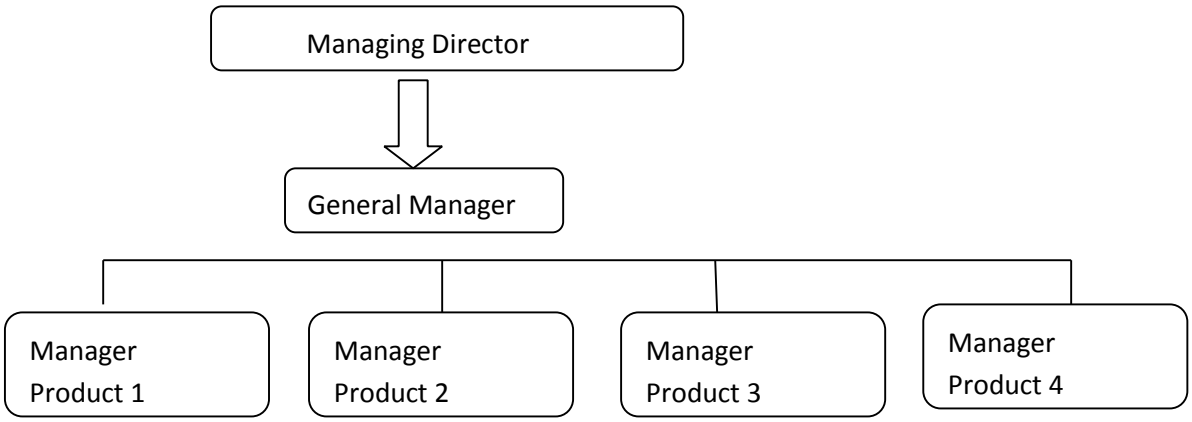
Generally, the small businesses when they are started consist of an owner-manager alone or also with few employees. These types of organisations do not require an organisational chart and formal assignment of responsibilities. It can refers to as start-ups .

Functional organisational Structure:



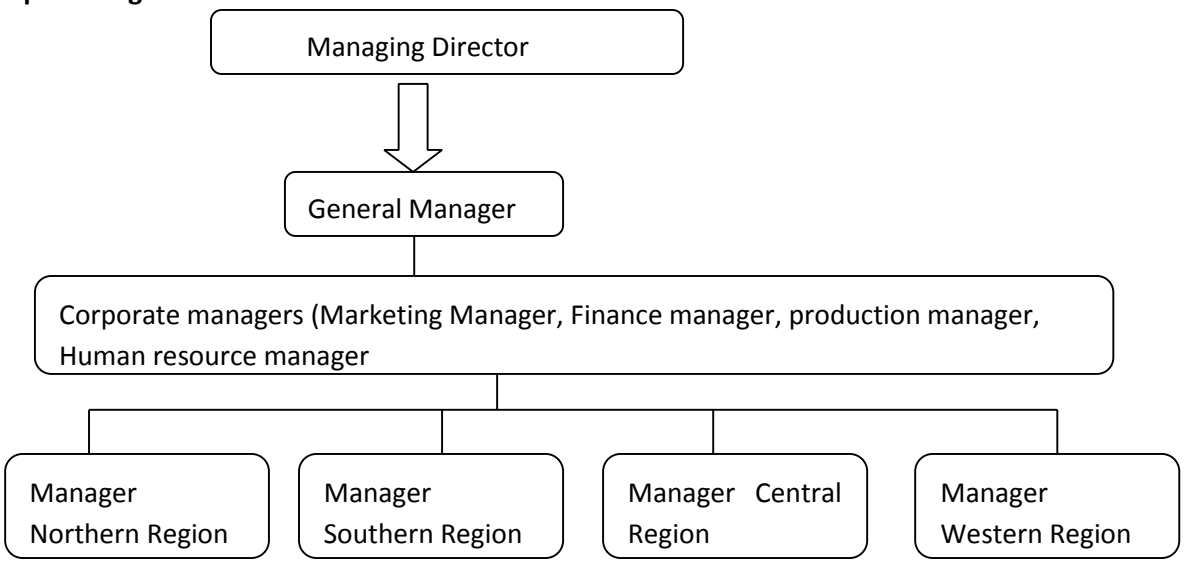
Functional organisation structure is most widely used structure. Each functional department consists of those jobs in which employees perform similar jobs at different levels. The commonly used functions are marketing, finance and accounting, human resources, manufacturing, research and development etc.

Product organisation structure



This structure is based on the products produced by company. Activities are divided on the basis of individual products, product line, services and are grouped into departments in product organisation structure. All important functions like marketing, production, finance and HRM are contained within each department. This type of organisation structure overcomes many of the major limitations of functional organisational structure.

Geographical organisation structure



Companies operating in various geographical regions of the country and world, structure of such organisation based on geographical structure. The activities or functions are grouped in to departments based on the activities performed in the geographical areas. Each geographical unit including all functions required to produce and market the products in a particular geographical area.

Decentralised business unit

Based on product lines the grouping activities has been a trend among diversified companies since 1920. In a diversified firm, the basic organisational building is its business units, each business is operated as a stand-alone profit centre.

Strategic business unit structure.

A single chief executive cannot control a number of decentralised units of a broadly diversified company. The business can effectively controlled, if the related businesses are grouped in to strategic units and the a efficient and senior executive is delegated with the authority and responsibility for its management. This arrangement will improve strategic planning and implementation, though it adds one layer in the organisational hierarchy.

Matrix structure

Combines functional and divisional structures to gain advantages and minimize disadvantages of each.
Used in:

- Manufacturing
- Service industries
- Professional fields
- Non-profit sector
- Multi-national corporations

3.5 Formal and informal organisation

Features of Formal organisation:

(1) The formal organisational structure is created intentionally by the process of organising.

(2) The purpose of formal organisation structure is achievement of organisational goal.

(3) In formal organisational structure each individual is assigned a specific job.

(4) In formal organisation every individual is assigned a fixed authority or decision-making power.

(5) Formal organisational structure results in creation of superior-subordinate relations.

(6) Formal organisational structure creates a scalar chain of communication in the organisation.

Features of informal organisation:

(1) Informal organisational structure gets created automatically without any intended efforts of managers.

(2) Informal organisational structure is formed by the employees to get psychological satisfaction.

(3) Informal organisational structure does not follow any fixed path of flow of authority or communication.

(4) Source of information cannot be known under informal structure as any person can communicate with anyone in the organisation.

(5) The existence of informal organisational structure depends on the formal organisation structure.

BASIS FOR COMPARISON	FORMAL ORGANIZATION	INFORMAL ORGANIZATION
Meaning	An organization type in which the job of each member is clearly defined, whose authority, responsibility and accountability are fixed is formal organization.	An organization formed within the formal organization as a network of interpersonal relationship, when people interact with each other, is known as informal communication.
Creation	Deliberately by top management.	Spontaneously by members.
Purpose	To fulfill, the ultimate objective of the organization.	To satisfy their social and psychological needs.
Nature	Stable, it continues for a long time.	Not stable
Communication	Official communication	Grapevine
Control mechanism	Rules and Regulations	Norms, values and beliefs
Focus on	Work performance	Interpersonal relationship
Authority	Members are bound by	All members are equal.

3.6 Types of Organization

Line and Staff organisation

Line

According to Allen, "Line refers to those positions and elements of organisation, which have the responsibility and authority and are accountable for the accomplishment of primary objectives.

- Line organisation is the simple and oldest type of organisation followed in an organisation. Under line organisation, each department is generally a complete self-contained unit.
- A separate person will look after the activities of the department and he has full control over the department.
- The same level executives do not give or receive orders amongst themselves. But they receive orders from their immediate boss and give orders to their subordinates. Hence, all the heads are responsible to the general manager, the general manager, in turn, is responsible to the shareholders who are the owners.
- The line of authority flows from the top to bottom vertically. So it is called line organisation.
  - It consists of direct vertical relationships. ○ Departmental heads are given full freedom to control their departments.
- Each member knows from whom he would get orders and to whom he should give his orders.
- The superior takes decisions within the scope of his authority

Staff

○ Staff elements are those which have responsibility and authority for providing advice and service to the line in the attainment of objectives.

- Line authority represents superior-subordinate relationships for functions, which are directly responsible for accomplishing the key objectives of the enterprise.
- The line identifies those positions that possess command and authority to make decisions. ○ The staff officers assist the line officers in the planning of business activity.
- The board of directors frames the policies of the business on the basis of recommendations given by the staff officers.
- The managers can get the advice from the staff officers regarding the selection. 'training' placement and remuneration fixation the personnel .
- Staff identifies those positions in an organization that do not command a decision making authority.

- These are support positions providing services, advice, assistance and support to the line authority to enable them to perform their duties.
- It is for the line managers to decide whether to take some decision with respect to the given advice.

Functional organisation:

Functional organisation structure is most widely used structure. Each functional department consists of those jobs in which employees perform similar jobs at different levels. The commonly used functions are marketing, finance and accounting, human resources, manufacturing, research and development etc

Committee Organization

A **committee organization** is group of people set up to solve specific problems in organization. The line people are given opportunities to discuss their problems in the committee. The committee organizational structure is not like line or functional organization, but is similar to staff organization. Its decisions are implemented, whereas staff decisions are not necessarily implemented. It is a formal part of the organizational structure wherein the members are specifically mentioned. The Board of Directors of a company is an example of a committee organisation.

Another example is, the Finance Committee will include all the functional managers, viz. Marketing Manager, Production Manager, Personnel Managers, etc. as members, and the Managing Director as the Chairman. It will decide the financial requirements of each and every department. The decisions taken by the committee are followed by the line people, as the committees are representatives of various functional departments.

3.7 Delegation of Authority

The process of giving authority to a subordinate in order to perform the assigned activities by a superior is known as delegation of authority.

Delegation is the process a manager follows in dividing the work assigned to him so that he performs only that part which he can perform effectively and so get others help him with the remaining work

Principles of Delegation

1. **Delegation of results expected:** managers have to clearly know the activities to be performed by the subordinates and the results to be achieved by the subordinates. Manager have to delegate the necessary authority and responsibility to produce the results.
2. **Co-equal Authority and responsibility:** managers should delegate equal authority and responsibility. It means that the amount of authority that is to be delegated should be enough

to complete the responsibility. If it is more than the responsibility, it may be misused. If authority is less than responsibility, the subordinate cannot complete the responsibility.

3. **Absoluteness of Responsibility:** the superior holds responsibility though he delegates it to the subordinates. Therefore, the responsibility of subordinates to their superiors is absolute.
4. **Creation of Accountability** : the subordinates who receive both authority and responsibility should be accountable for their responsibility of completing the activities.
5. **Unity of Command:** it indicate that the subordinate should receive orders, instructions and commands from one superior only.

3.8 Elements of Delegation

1. **Assignment of Duties or Responsibility**: The responsibility means, the assignment of work or duty to the subordinate i.e. delegation of authority. The managers assign certain responsibility to the subordinates for the completion of certain tasks on his behalf. An individual has to apply all his physical and mental ability to get the task completed efficiently.

   Duty is that which a person is bound by obligation to do things'. Duties may be expressed either in terms of functions or in terms of goals and objectives or results. For effective delegation, duties must be expressed in terms of goals and objectives; since this will provide a psychological satisfaction to the subordinates

2. **Grant of Authority:** To fulfil the responsibility, certain authority is delegated to the subordinate. Authority means the power to take decisions. Hence, the manager along with the responsibility also delegates authority to enable the subordinate to take decisions independently and accomplish the task efficiently.

   The authority must be equal to the responsibility, this means, a certain level of authority is delegated which is sufficient to complete the responsibility. The authority also flows upward, as we go up in the scalar chain, the authority increases.

3. **Accountability or Creation of Obligation**: Accountability means, to check whether the subordinates are performing their responsibilities in an expected manner or not. The Accountability cannot be delegated which means, in the case of non-completion of the task, the manager will only be held responsible for it, not the subordinates. The accountability also flows upward, i.e. subordinates will be accountable to the manager and the manager to his superior. When the subordinate knows his assigned responsibilities, he can concentrate his efforts in meeting the requirements of his position.

   Definite fixing of responsibility is advantageous because;

   1. It acts as an incentive to a subordinate;

   2. It aids in the general speed up 'of work; and

   3. It assists in developing discipline as a means of control.

Unit IV: Staffing

Staffing

The managerial function of staffing involves manning the organization structure through proper and effective selection, appraisal and development of the personnel to fill the roles assigned to the employers/workforce.

"Staffing means filling and keeping filled, positions in the organisation structure."

Harold Koonz

4.1 Nature of Staffing

- Staffing is an important managerial function.
- Staffing is a pervasive activity.
- Staffing is a continuous activity.
- The basis of staffing function is efficient management of personnel.
- Staffing helps in placing right men at the right job.
- Staffing is performed by all managers

4.2 Staffing Process

1. **Manpower Planning:** Manpower Planning which is also called as Human Resource Planning consists of putting right number of people, right kind of people at the right place, right time, doing the right things for which they are suited for the achievement of goals of the organization
2. **Recruitment:** it is the process involves attracting and obtaining as many applications as possible from eligible job seekers.

   Selection: It is a long process, commencing from the preliminary interview of the applicants and ending with the contract of employment. Following are the steps of selection process
3. **Orientation and Placement (Induction**): During Orientation employees are made aware about the mission and vision of the organization

   Generally, the information given during the orientation programme includes-

- Employee's layout
- Type of organizational structure
- Departmental goals
- Organizational layout
- General rules and regulations
- Standing Orders
- Grievance system or procedure Placement

- Once the candidates are selected for the required job, they have to be fitted as per the qualifications.

- Placement is said to be the process of fitting the selected person at the right job or place, i.e. fitting square pegs in square holes and round pegs in round holes.
- Once he is fitted into the job, he is given the activities he has to perform and also told about his duties.

4. **Training and Development**: Training is crucial for organizational development and success.

   It is fruitful to both employers and employees of an organization. An employee will become more efficient and productive if he is trained well.
   It also-
   Improves morale of employees
   Less supervision
   Chances of promotion
   Increased productivity

5. **Remuneration:** Employee Remuneration refers to the reward or compensation given to the employees for their work performances.

   Remuneration provides basic attraction to a employee to perform job efficiently and effectively.
   There are mainly two types of Employee Remuneration
   Time Rate Method
   Piece Rate Method

6. **Performance and Evaluation:** performance appraisal is the method that evaluate the performance of employee. It measures the productivity of employees.
7. **Promotion and Transfer:** Through performance and evaluation organisation come to know the employees who have more productivity. Promotion of such employees done on other position for which they are eligible. Sometimes it is essential to transfer the employees to the other branches which are in need of such employees.

4.3 Recruitment

Definition 1: It is a process of searching for prospective employees and stimulating and encouraging them to apply for jobs in an organisation. By Adwin B. Flippo Definition 2: it is the process involves attracting and obtaining as many applications as possible from eligible job seekers.

The term recruitment is used to describe the entire process of employee hiring. There are recruitment boards for railways, banks and other organisations. B schools talk about campus recruitment6. In newspapers too, recruitment is frequently used to denote the process of employment. But students of management should know that there is difference between the two terms – recruitment and selection.

Sources of Recruitment

There are several sources of recruitment that may be broadly categorised into (i) Internal, and (ii) External

(i) Internal sources of Recruitment

1. Present employees 2. Former employee
3. Previous employee
4. Employee referrals

1. **Present employees:** It is the easiest and effective source of recruitment. Promotion and transfer from among present employees can be a good source of recruitment. It has several advantages (1) it has good public relations (ii) it builds morals (iii) it encourages the employees (iv) it improves the probability of good selection (v) reduce cost of recruitment (vi) selected employees are familiar with organisation. Performance and potential of Present employees can easily analysed and improved.
2. **Former employees:** former employees are also an internal source of applicants. Some retired employees may be willing to come back to work on a part-time basis or may recommend someone who would by interested in working for the company. Sometimes, people who have left the company for some reason or the other are wiling to come back and work. Individuals who left for other jobs, might be willing to come back for higher emoluments. An advantage with this source is that the performance of these people is already known.
3. **Previous applicants**: those who have previously applied for jobs can be contacted by mail, a quick and inexpensive way to fill an unexpected opening. Although walk in are likely to be more suitable for filling unskilled and semiskilled jobs, some professional opening can be filled by applicants to previous job.
4. **Employee referrals:** this can be a good source of internal recruitment. Employees can develop good prospects for their families and friends by making them aware about the advantages of a job with the company, furnishing cards of introduction, and even encouraging them to apply. Employees are familiar with referrals which mean organisation is known about referred employees.

External Sources of Recruitment

1. **Professional or Trade Association:** Many associations provide placement services for their members. These services may consist of compiling job seekers list and providing access to members. Trade associations publishes classified advertisements from employers interested in recruiting their members. It is useful for attracting highly educated experienced or skilled personnel.
2. **Advertisements**: These constitute a popular method of seeking recruits as many recruiters prefer advertisements because of their wide reach. Many employers or companies are publishing their recruitment advertisements in order to attract eligible candidates as many as possible to fill the vacant posts. It is mostly used source for recruitment because many people have access of advertisements.

3. **Employment exchanges**: employment exchanges have been setup all over the country in deference to the provisions of the employment exchanges (compulsory notification of vacancies) Act. 1959. The act applies to all industrial establishments having 25 workers or more each. The act requires all the industrial establishments to notify the vacancies before they are filled. The major functions of the exchanges are to increase the pool of possible applicants and to do preliminary careening.
4. **Radio and Television**: this source for recruitment used by government department only. Companies in private sector are hesitant to use the media because of high cost and possibility to damage image.
5. Perhaps no method has ever had as revolutionary an effect on recruitment practices as the internet. Currently employers or companies are electronically screen candidate's soft attributes and other factors to judge whether the candidates are eligible or not.

4.4 Selection Process

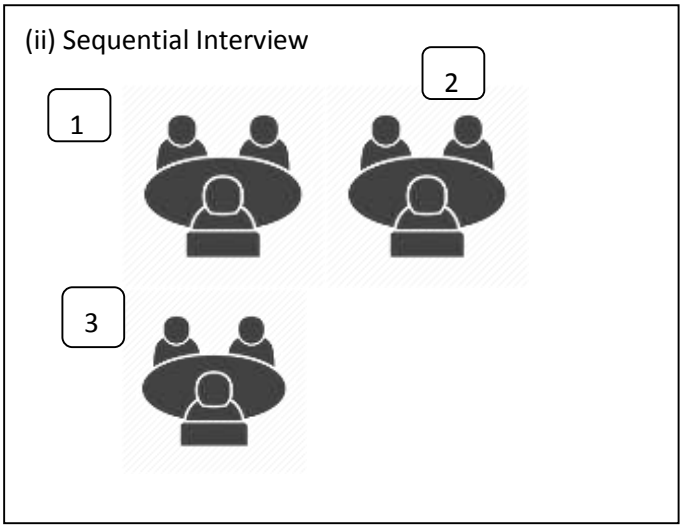
It is a long process, commencing from the preliminary interview of the applicants and ending with the contract of employment. Following are the steps of selection process

1. **Preliminary Interview**: this interview rejects the candidates who are not fit for the job. It is used as scrutiny so as to eliminate unqualified applicants. The preliminary interview helps reject misfits for reasons, which did not mention in application or resume.
2. **Selection Tests**: job seekers who pass the screening and the preliminary interview are called for tests. Different types tests may be administered, depending on the job and the company. Generally, tests are used to determine the applicant's ability, attitude and personality. Ability test assist in determining how well an individual can perform tasks related to the job. An aptitude test helps determine a person's potential to learn in a given area. An example of such a test is the General Management Aptitude Test(GMAT) which many business students take prior to gaining admission to a graduate business school programme. Personality tests are given to measure a prospective employee's motivation to function in a particular working environment. Such many tests are conducted in order to judge candidate's skills as per requirements of company.
3. **Employment Interview**: two interviews are taken in selection process. One at beginning and one at the end. More emphasis on the end interview. Interview is a formal, in-depth conversation conducted to evaluate the applicant's acceptability. It is considered to be an excellent selection device. The employment interview can by (i) one to one interview (ii) sequential interview (iii) panel interview.

(i) One to one interview:

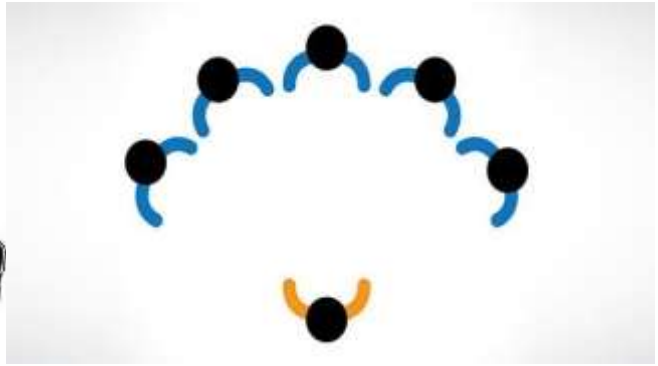


(ii) Sequential Interview



(ii) Panel Interview





4. Reference and Background check: many employers request names, addresses and telephone numbers or references for the purpose of verifying information and, perhaps, gaining additional background information on an applicant. Previous employers, known public figures, neighbours or friends can act as references. Previous employers are preferable because they are already aware of the applicant's performance

   Reference recheck covers:

- Criminal record check
- Character reference check Neighbourhood reference check
  Previous employment check etc.

5. Selection Decision: after obtaining information through the preceding steps, selection decision the most critical of all the steps must be made. The final selection haze to be made from the individuals who pass the tests, interviews and reference checks.
6. Physical Examination: it belongs to physically fitness of candidates. This tests are conducted by various govt. departments and MNCs.
7. Job Offer: it is made through a letter of appointment. Such letter contains a date by which appointee must report on duty and salary decided.

4.5 Training and Development

Training and development may be understood as any attempt to improve current or future employee performance by increasing an employee's ability to perform through learning, usually by changing the employee's attitude or increasing his or her skills and knowledge.

Importance of Training and Development

1. **Improve profitability:** as skills of the employees are improved by training and development, they will increase their productivity which can increase the profit of organisation.
2. **Improves the job knowledge and skills at all the levels:** top, middle and low level management employees can improve their knowledge and skills through training and development process.
3. **Employee familiar with goal**: trained employees are familiar with goal of organisation which makes integration of all the efforts that employee take to common goal.
4. **Create corporate image:** Productivity of organisation increased by means of training which can increase image of organisation through various CSR activities and offering good quality products for society.
5. **Productivity**: Training and Development helps in increasing the productivity of the employees that helps the organization further to achieve its long-term goal.
6. **Team spirit:** Training and Development helps in inculcating the sense of team work, team spirit, and inter-team collaborations. It helps in inculcating the zeal to learn within the employees.
7. **Organization Culture:** Training and Development helps to develop and improve the organizational health culture and effectiveness. It helps in creating the learning culture within the organization.
8. **Organization Climate**: Training and Development helps building the positive perception and feeling about the organization. The employees get these feelings from leaders, subordinates, and peers.
9. Quality Training and Development helps in improving upon the quality of work and work-life.
10. **Healthy work environment**: Training and Development helps in creating the healthy working environment. It helps to build good employee, relationship so that individual goals aligns with organizational goal.
11. **Health and Safety**: Training and Development helps in improving the health and safety of the organization thus preventing obsolescence.
12. **Optimum Utilization of Human Resources:** Training and Development helps in optimizing the utilization of human resource that further helps the employee to achieve the organizational goals as well as their individual goals.

Unit V: Direction and Controlling

Directing: it is a function that includes all activities which are designed to encourage subordinates and to work effectively and efficiently

It consists of the process and technology utilized in using instructions and making certain that operations are carried out on as originally planned.

5.1 Importance of Directing

1. Directing Initiates Action

Directing initiates Each and every action in an organization. It act as guidelines for subordinates. The managers direct the subordinates about what to do, how to do when to do and also see to it that their instructions are properly followed.

2. Ingrates all Efforts to achieve common goal

Directing integrates the efforts of all the employees and departments through persuasive leadership and effective communication towards the accomplishment of organizational goals.

3. Support to adopt change:

Employees have a tendency to resist any kind of change in the organization. But, adapting the environmental changes is necessary for the growth of the organization. A manager through motivation, proper communication and leadership can make the employees understand the nature and contents of change. This will help in a smooth adaptation of the changes without any friction between the management and employees.

4. Effective Utilization of Resources

It involves defining the duties and responsibilities of every subordinate clearly thereby avoiding wastages, duplication of efforts, etc. and utilizing the resources of men, machine, materials, and money in the maximum possible way. It helps in reducing costs and increasing profits.

5. Motivates Employees

A manager identifies the potential and abilities of its subordinates and helps them to give their best. He also motivates them by offering them financial and non-financial incentives to improve their performance.

6. Provides Stability

Stability is significant in the growth of any organization. Effective directing develops cooperation and commitment among the employees and creates a balance among various departments and groups.

5.2 Supervision:

It is the process which provides constant observation, monitoring, evaluation, guidance to workers to enable them to perform them

It is defined as a cooperative relationship between a leader and one or more persons to accomplish a particular purpose.

Importance of Supervision

Optimum Utilisation of Resources:

When the workers are constantly monitored or observed then they always use the resources in the best possible manner which leads to minimum wastage. But if there is no supervision or check on workers they may result in wastage of resources

Unity among Workers:

He plays an important role in integrating the activities of different workers under him by solving their problems among themselves only as for as possible.

Motivating Workers:

He has the responsibility for completion of a job; hence he motivates his workers to perform effectively for completing same.

Training of Workers:

A skilled and experienced supervisor makes arrangement for providing on the job training to his workers and also builds up an efficient team of workers.

Provides a Link between Workers and Management:

A supervisor is an individual engaged in supervision. He supervises the work of subordinates. He is the link between workers and management. He ensures that plans and procedures are implemented as per schedule.

2. Direct Contact with the Workers:

Supervisor maintains a direct contact with the workers; he acts as a friend, philosopher and guide to the workers.

Improves Communication:

It improves communication and hence quality of relations “with workers” as well as “among workers”

5.3 Controlling:

Meaning: Control consists of making something happen the way it was planned to happen. According to Henri Fayol, control consists in verifying whether everything occurs in conformity with the plan adopted, the instructions issued and principles established.

Managerial control implies the measurement of accomplishment against the standard and the correction of deviations to assure attainment of objectives according to plans"... Koontz And O'Donnell Objective of controlling is to point out weaknesses and errors in order to rectify them and prevent. It includes (i) measuring actual performance (ii) comparing actual performance to standards and (iii) taking corrective action to ensure that planned events actually occur.

Importance of Controlling

Ensures order and discipline:

Control creates an atmosphere of order and discipline in the organisation. Effective controlling system keeps the subordinates under check and makes sure they perform their functions efficiently. Sharp control can have a check over dishonesty and fraud of employees. Strict control monitor, employees work on computer monitor which brings more order and discipline in work environment.

Controlling helps in minimising the errors:

Small errors or small mistakes may not seriously affect the organisation. But if these errors are repeated again and again it will become a serious matter and can bring disaster for the organisation. An effective controlling system helps in minimising the errors by continuous monitoring and check. The managers try to detect the error on time and take remedial steps to minimise the effect of error.

Making Efficient Use of Resources:

Controlling makes it possible to use human and physical resources efficiently. Under controlling, it is ensured that no employee deliberately delays his work performance. In the same way, wastage in all the physical resources is checked.

Facilitating Coordination in Action:

Coordination among all the departments of the organisation is necessary in order to achieve the organisational objectives successfully. All the departments of the organisation are interdependent. For

example, the supply of orders by the sales department depends on the production of goods by the production department.

Improving Employee Motivation:

Employees know the standards against which their performance will be judged. Systematic evaluation of performance and consequent rewards in the form of increment, bonus, promotion etc. motivate the employees to put in their best efforts.

Achieving Goals:

Being goal oriented provides an entity let it be a business or not, the advantage of making it easier to achieve goals. When the employees understand the goals and are kept goal oriented their efforts will be directed towards achieving it. If the effort is in a different direction altogether, the process of controlling weeds it out. Thus the process ensures channelizing effort into achieving predetermined goals.

5.4 Controlling Process

Steps 1: Key areas to be monitored

Macro environment: As stated earlier, one of the key areas to be monitored is macroenvironment of the company. Macro environmental factors can influence company continuously so it is essential to monitor in order to prevent negative impact and frame policies.

Mission and Objectives: it includes modifying any one or more of the areas like company's mission, objectives, plans, goals, strategy formulation and implementation.

Internal operations: the manager has to evaluate the internal operations continuously in view of the changes in the macro-environment and industry environment.

Step 2: Establishing Standards:

Standards are criteria against which results are measured. They are norms to achieve the goals. Standards are usually measured in terms of output. They can also be measured in non-monetary terms like loyalty, customer attraction, goodwill etc. Some of the standards are as. **a. Time standards:**

The goal will be set on the basis of time lapse in performing a task.

b. Cost standards:

These indicate the financial expenditures involved per unit, e.g. material cost per unit, cost per person, etc.

c. Income standards:

These relate to financial rewards received due to a particular activity like sales volume per month, year etc.

d. Market share:

This relates to the share of the company's product in the market.

e. Productivity:

Productivity can be measured on the basis of units produced per man hour etc.

f. Profitability:

These goals will be set with the consideration of cost per unit, market share, etc.

Step3: Measuring Performance

Measurement involves comparison between what is accomplished and what was intended to be accomplished. The measurement of actual performance must be in the units similar to those of predetermined criterion. The unit or the yardstick thus chosen be clear, well-defined and easily identified, and should be uniform and homogenous throughout the measurement process.

The performance can be measured by the following steps: (a)
Strategic control points:

It is not possible to check everything that is being done. So it is necessary to pick strategic control points for measurement. Some of these points are:

(i) Income:

It is a significant control point and must be as much per unit of time as was expected. If the income is significantly off form the expectation then the reasons should be investigated and a corrective action taken.

(ii) Expenses:

Total and operational cost per unit must be computed and must be adhered to. Key expense data must be reviewed periodically.

(iii) Inventory:

Some minimum inventory of both the finished product as well as raw materials must be kept in stock as a buffer. Any change in inventory level would determine whether the production is to be increased or decreased.

(iv) Quality of the product:

Standards of established quality must be maintained especially in food processing, drug manufacturing, automobiles, etc. The process should be continuously observed for any deviations.

(v) Absenteeism:

Excessive absenteeism of personnel is a serious reflection on the environment and working conditions. Absenteeism in excess of chance expectations must be seriously investigated.

Step 4: Comparing the Actual Performance with Expected Performance

This is the active principle of the process. The previous two, setting the goals and the measurement format are the preparatory parts of the process. It is the responsibility of the management to compare the actual performance against the standards established.

This comparison is less complicate if the measurement units for the standards set and the performance measured are the same and quantified. The comparison becomes more difficult when these require subjective evaluations

Ralph C. Davis identifies four phases in the comparison.

1. Receiving the raw data.

2. Accumulation, classification and recording of this information.

3. Periodic evaluation of completed action to date.

4. Reporting the status of accomplishment to higher line authority.

At the third phase, deviations if any are noted between standards and performance. If clear cut deviations are there, then management must study the:-

(i) Causes for deviation

(ii) Effect of deviation

(iii) Size of deviation

(iv) Positive or negative deviation.

Step 5: Correcting Deviations:

The final element in the process is the taking corrective action. Measuring and comparing performance, detecting shortcomings, failures or deviations, from plans will be of no avail if it does point to the needed corrective action.

Thus controlling to be effective, should involve not only the detection of lapses but also probe into the failure spots, fixation of responsibility for the failures at the right quarters, recommendation of the best possible steps to correct them. These corrective actions must be applied when the work is in progress. The primary objective should be avoidance of such failures in future.

The required corrective action can be determined from the qualified data as per the standards laid out and the performance evaluation already done.

5.5 Budget as a control technique

Budget are plans for a specific period in numerical terms. They are statements of expected outcome in financial terms, physical terms, human resources terms etc. Budgets would be the basis for delegation of authority and responsibility without loss of control.

Types of Budgets

1. Capital Expenditure budgets: companies basically require capital for establishing manufacturing facilities, marketing network, and the like many activities. Capital expenditure budgets specify the estimates of the amount of capital required for land, buildings, plant, equipment, machinery, technology, minimum level of materials etc. It is used to control the deviations or errors.
2. Cash budgets: cash budgets provide the estimates of cash receipts and disbursements. The actual cash receipts and disbursements are compared with the estimates.
3. Production Budget: they specify the amount of the estimated output, estimated materials, human resources etc. These budgets help to control the production factors and ensure the production as ensured.
4. Sales budget: sales budgets specify the amount of estimated sales. They help to estimate the sales, identify the factors affects sales and control these factors.

In this way different budgets support and work as guidelines for action and control. Budget also assist for coordination , encourage team efforts and creativity.

Quality Control

Quality control (QC) is a procedure or set of procedures intended to ensure that a manufactured product or performed service adheres to a defined set of quality criteria or meets the requirements of the client or customer.

In order to implement an effective QC program, an enterprise must first decide which specific standards the product or service must meet. Then the extent of QC actions must be determined (for example, the percentage of units to be tested from each lot). Next, real-world data must be collected (for example, the percentage of units that fail) and the results reported to management personnel. After this, corrective action must be decided upon and taken (for example, defective units must be repaired or rejected and poor service repeated at no charge until the customer is satisfied). If too many unit failures or instances of poor service occur, a plan must be devised to improve the production or service process and then that plan must be put into action. Finally, the QC process must be ongoing to ensure that remedial efforts, if required, have produced satisfactory results and to immediately detect recurrences or new instances of trouble.

It brings standardization into the process. Most organizations have a quality control/assurance department that provides the set of standards to be followed for each product. Either an internal team or a third-party team is hired to determine whether the products that are delivered meet these standards. Quality control relies on testing of products, as product inspection gives a clearer picture of the quality of the end product.

Cost Control

Cost control is a series of steps that a business uses to maintain proper control over its costs. Implementing this level of control can have a profound positive impact on profits over the long term. The following four steps are associated with cost control:

1. **Create a baseline.** Establish a standard or baseline against which actual costs are to be compared. These standards may be based on historical results, a reasonable improvement on historical results, or the theoretically best attainable cost performance. The middle alternative is generally considered to yield the best results, since it sets an achievable standard.
2. **Calculate a variance.** Calculate the variance between actual results and the standard or baseline noted in the first step. Particular emphasis is placed on the detection of unfavourable variances, which are those actual costs that are higher than expected. If a variance is immaterial, it may not be worthwhile to report the item to management.
3. **Investigate variances**. Conduct a detailed drill-down into the actual cost information to ascertain the reason for an unfavourable variance.
4. **Take action.** Based on the information found in the preceding step, recommend to management whatever corrective actions are needed to reduce the risk of continued unfavourable cost variances.

The preceding steps are only recommended if a company routinely attempts to force its actual costs incurred to closely match its budgeted cost structure. If there is no budget, then an alternative way to practice cost control is to plot individual cost line items from the income statement on a trend line. If there is an unusual spike in the trend line, then the spike is investigated in relation to the average cost level, and corrective action is taken. Thus, operating without a budget eliminates the first two steps in the preceding list of activities, but cost control still requires investigatory work and recommendations to management for corrective action.

The shareholders of a publicly held company are particularly interested in a system of cost control, for they realize that tight cost control gives a company considerable influence over its cash flows and reported profits.

BBA FIRST YEAR

SUBJECT: COMPUTER APPLICATION IN BUSINESS II

DISCLAIMER

THESE EDUCATIONAL CONTENTS (NOTES) ARE COMPILED FROM VARIOUS BOOKS, JOURNALS AND INTERNET.

XIV.1 Computer Applications in Business II

End of Semester Examination	40 Marks
Continuous Assessment	35 Marks
Total	75 Marks

Learning Objectives: - The objective of the course is to equip the students with the ability to analyse, interpret and apply the basic concepts of information & communication technology in business decision-making.
Utility: Student can acquire basic-level knowledge of computer for business application.
Prerequisite: Basic knowledge of computer.

Unit. No.	Particulars	No. of Lectures
1	**Introduction to MS-Word**: • Introduction to MS-Word window, Introduction to Menus, working with menus, Understanding & working with toolbars, Creating & Saving Documents, Save, Open file, save file with new name, Quitting word.	11
2	**Editing Document** • Editing text, finding & replacing text, Working with Tools: spell check, grammar check, auto correct, Creating auto correct entry, auto format, Thesaurus	11
3	**Formatting Document** • Character formatting by using Font dialog box, Paragraph formatting, adding borders and shading, Page Setup, Adding page numbers. Headers & Footers	11
4	**Introduction to MS-Excel**: • Introduction to Excel working environment: the application window (with its elements), the document window (with its elements), Moving Around in a Worksheet: The workbook, worksheet, rows, columns, & cells, Working with Charts, what is Formula & Function, Entering & editing formula, guidelines for using functions, use of functions in formulas.	11
5	**Introduction to Power Point:** • Power point basics, creating presentation, working with graphics in power point, Show effect and Animation effects.	10
	Total	54

Reference Books:

1. Microsoft word 2000 – Varsha verma shekhar – Unistar publication
2. Microsoft excel 2000 – Varsha verma shekhar – Unistar publication
3. Microsoft power point 2000 – Varsha verma shekhar – Unistar publication

Unit 1:

Introduction To MS-Word

Microsoft Word or Word Processor

“A Word Processor is an application software that allows you to create any kind of text document.”

- ✓ Microsoft word is a word processor software developed by Microsoft in 1983. and is part of Microsoft Office Suite.
- ✓ Microsoft Word is used to create professional-quality documents, letters, reports, resumes, etc.
- ✓ It helps you to create a document, format it, correct your mistake, save, review, and print it.
- ✓ The word processor has many helpful, time-saving features that enable the user to design a document in a professional manner, quickly, and easily.
- ✓ MS Word file is always saved with the extension of .doc or .docx.

Features and characteristics of MS-Word

- ✓ Creating text documents
- ✓ Editing and formatting the existing documents
- ✓ Making a text document interactive with different features and tools
- ✓ Graphical documents, comprising images.
- ✓ Used by Authors and Researchers
- ✓ Detect grammatical errors in a text document.
- ✓ Cut, copy, and paste the text.
- ✓ Insert or delete text.
- ✓ Set page size and margins.
- ✓ Find and Replace
- ✓ Insert pictures, graphs, page numbers, date & time, and symbols.
- ✓ Mail merging
- ✓ Headers and footers
- ✓ Print

Starting Microsoft Word

Follow the simple steps given below:

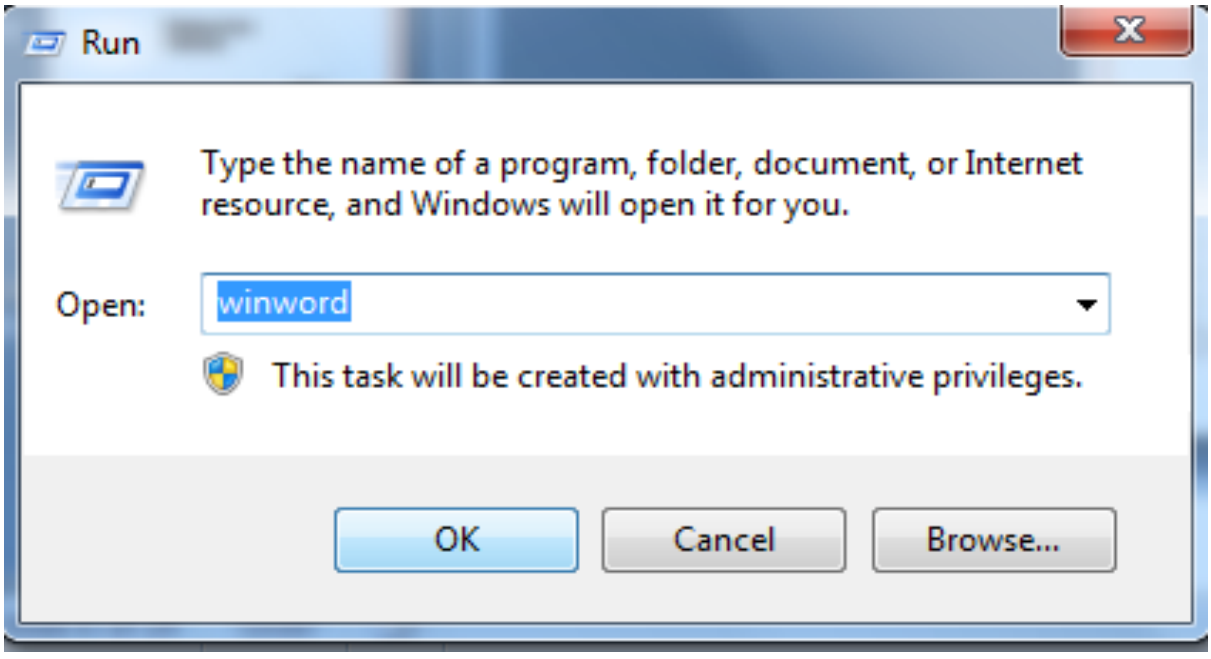
- Step 1: Click the start button which is located on the left-hand bottom corner of your Desktop or Laptop
- Step 2: Click the All-Programs button just above the Start button.
- Step 3: Find the group Microsoft Office. If you point at the Microsoft Office group icon, the sub-group will open.
- Step 4: Click the Microsoft Word

OR

Another method for opening MS Word is:

Using Run Command.

- Press Window Key + R, It will open the Run dialog box, type winword, and click ok. This will open MS word document.



Uses of MS Word

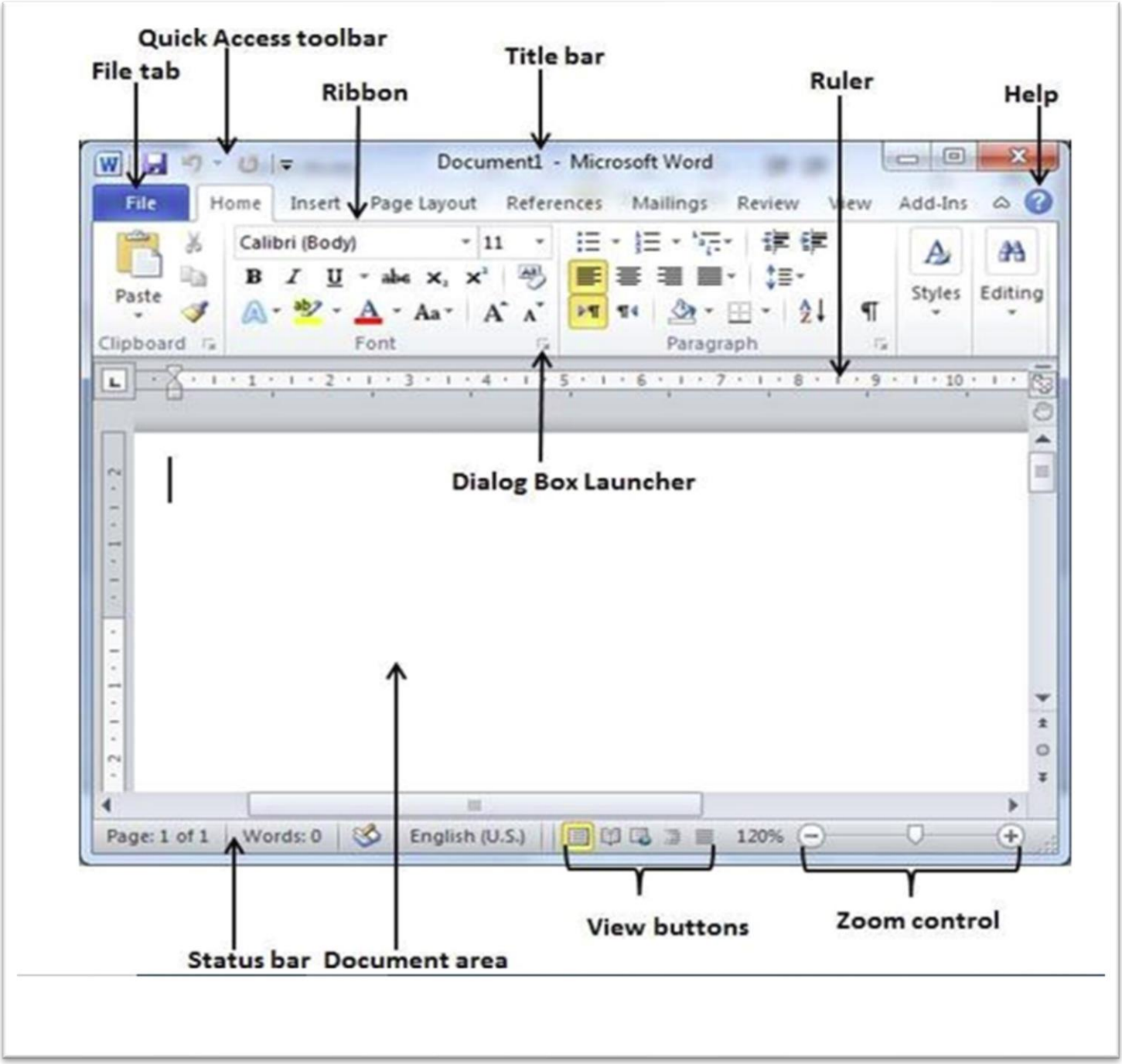
Given below are the different fields in which MS Word is used and simplifies the work of an individual:

- **In Education**: It is considered one of the simplest tools which can be used by both teachers and students. Creating notes is easier using MS Word as they can be made more interactive by adding shapes and images. It is also convenient to make assignments on MS Word and submit them online.
- **In the Workplace:** Submitting letters, bills, creating reports, letterheads, and sample documents, can all easily be done using MS Word
- **Creating & Updating Resume:** One of the best tools to create your resumes and is easy to edit and make changes in it as per your experience.
- **For Authors:** Since separate options are available for bibliography, table of contents, etc., it is the best tool that can be used by authors for writing books and adjusting it as per the layout and alignment of your choice.

Understanding the Word Screen

When MS Word is launched, two windows appear nested one within the other, the larger window, which covers the entire screen, is called the Word Application Window. The smaller window is called the document window and is a part of the application window when the word is loaded, the Word application window opens with a new word document named Document1(default name).

The word screen or the application window is shown in the figure below:



The main components of the Microsoft Word screen are:

1. Title Bar
2. Quick Access Toolbar
3. Ribbon/ Toolbar
4. File Tab
5. Work Area
6. Rulers
7. Status Bar
8. Zoom Control
9. View Button
10. Dialog Box Launcher
11. Help
12. Scroll Bars

1. Title Bar:

The title Bar lies in the middle and at the top of the window. The title bar shows the program and document titles, that is, the name of the program that is currently used. The title bar displays the document name.

2. Quick Access Toolbar

This you will find just above the File tab. Quick Access Toolbar contains frequently used commands.

It gives you quick access to commonly used commands such as Save, Undo, Redo, etc. The quick Access Toolbar is Customizable.

3. Ribbon/Toolbar

The Ribbon is a user interface element that was introduced by Microsoft in Microsoft Office 2007. It is located below the Quick Access Toolbar and the Title Bar. It comprises seven tabs; Home, Insert, Page Layout References, Mailing,

Review, and View. Each tab has specific groups of related commands. It gives you quick access to the commonly used commands that you need to complete a task.

The ribbon contains commands organized into three components:

- Tabs: These appear across the top of the Ribbon and contain groups of related commands. Home, Insert, and Page Layout are examples of ribbon tabs.

- Groups: They organize related commands; each group name appears below the group on the Ribbon. For example, a group of commands related to fonts, or a group of commands related to alignment, etc.

- Commands: Commands appear within each group as mentioned above.

4. **File Tab**

The File tab replaces the Office button from Word 2007. You can click it to check the Backstage view. This is where you come when you need to open or save files, create new documents, print a document, and do other file-related operations.

5. **Work area or Document Window**

This is the area where you type and work is done in the document. The flashing vertical bar is called the insertion point. It represents the location where text will appear when you type.

6. **Rulers**

   The word screen includes horizontal and vertical rulers, which can be used to make changes to margins and indents and set tabs and columns.

   The ruler helps the user create documents according to the specified dimensions.

7. **Status Bar**

The status bar is at the bottom of the Word window. Status Bar displays the document information as well as the insertion point location. From left to right, this bar contains the total number of pages and words in the document, language, etc. You can configure the status bar by right-clicking anywhere on it and by selecting or deselecting options from the provided list.

- Page count: The page count shows the total number of pages in the document and which page is currently visible in the window.
- Word count: The word count displays the total number of words in the document.

8. **Zoom Control**

Zoom control lets you zoom in for a closer look at your text. The zoom control consists of a slider that you can slide left or right to zoom in or out; you can click the + buttons to increase or decrease the zoom factor.

9. **View Buttons**

The group of five buttons located to the left of the Zoom control, near the bottom of the screen, lets you switch through Word's various document views.

i. Print Layout view
ii. Read Mode view
iii. Web Layout view
iv. Outline view
v. Draft view

Print Layout view: This displays pages exactly as they will appear when printed.

Read Mode view: This gives a full-screen view of the document.

Web Layout view: This shows how a document appears when viewed by a Web browser, such as Internet Explorer.

Outline view: This lets you work with outlines established using Word’s standard heading styles.

Draft view: This formats text as it appears on the printed page with a few exceptions. For example, headers and footers aren't shown.

10.Dialog Box Launcher

This appears as a very small arrow in the lower-right corner of many groups on the Ribbon. Clicking this button opens a dialog box or task pane that provides more options about the group.

11.Help

The Help Icon can be used to get word-related help anytime you like. This provides a nice tutorial on various subjects related to word.

12.Scroll Bars

The scroll bars are used to shift the window’s contents to view different parts of the worksheet. They enable the user to move within the document.

The vertical scrollbar enables the movement up or down, while the horizontal scrollbar helps move right or left in a window.

Introduction to Menus (Tabs)

The menu bar provides access to all of the word's command features organized into nine categories:

1. File
2. Home
3. Insert
4. Design
5. Layout
6. References
7. Mailing
8. Review
9. View

File Tab

The Backstage view was introduced in Word 2010. This acts as the central place for managing your documents. The backstage view helps in creating new documents, saving and opening documents, printing and sharing documents, and so on. Getting to the Backstage View is easy: Just click the File tab, located in the upper-left corner of the Word Ribbon.

This menu allows the user to perform file-related activities, including opening, saving, printing, closing, etc.

New (ctrl+n) – This option is used to insert a new blank document or template.

Open (ctrl+o) – This option is used to open any existing document saved in memory.

Save (ctrl+s) – This option is used to save the current document.

Save As – This option is used to re-save the edited document with a new name or save the document in other versions like 97, 2000, 2003 format etc.

Print (ctrl+p)— This option prepares the document for printing and print preview.

Share – This option is used to send the document to anyone using E-mail, but outlook must be configured for using this feature.

Export – This option is used to export the file in PDF or XPS document and we can change the file type as well.
Publish – This option is used to publish the document for blogs or a website.
Close – This option is used to close the currently active document.
Option – This option is used to manage the settings of the word application.

Home Tab

The home Tab in Microsoft Word is the default tab for editing and formatting the document in MS Word. The ribbon or toolbar of the Home Tab in Microsoft Word is divided into multiple sections with the name Clipboard, Font, Paragraph, Styles & Editing. A detailed explanation of the Home Tab in Microsoft Word is given below.

Clipboard

- Cut (ctrl+x) – This option is used to cut the selected data.
- Copy (ctrl+c) – This option is used to copy the selected data.
- Paste (ctrl+v) – This option is used to paste the cut or copied data.
- Paste Special – This option creates a link between copied and pasted data. By pasting data with the link, if source data (copied data) change or delete then target data (pasted data) will also change.
- Format Painter – This option is used to copy and paste text formatting.

Font

- Font – This font Box is used to select the different font styles for the selected paragraph or text.
- Font Size (ctrl+shift+p) – This option is used to change the size of the selected font.
- Bold (ctrl+b) – This option is used to make the selected font bold.
- Italic (ctrl+i) – This option is used to make the selected font italic.

- Underline (ctrl+u) – This option is used to draw a line under the selected text.
- Strikethrough – This option is used to draw a middle line for the selected text. Ex- ~~word processor~~
- Subscript (ctrl+=) – This option is used to make the selected text subscript form. Ex- X_2
- Superscript (ctrl+shift++) – This option is used to make the selected text superscript form. Ex- X^2
- Change Case – This option is used to change the font cases with UPPER, lower, Capitalize, Sentence and tOGGLE.
- Increase font size (Ctrl+Shift+>) – This option is used to increase the size of the selected font.
- Decrease font size (Ctrl+Shift+<) – This option is used to decrease the size of the selected font.
- Clear All Formatting – This option is used to clear all the formatting from the selected text and make the text plain.
- Text Highlight Color – This option is used to make the text highlighted with a color, look as marked with a highlight pen.
- Font Color – This option is used to change the color of selected text.

Paragraph

- Bullets – This option is used to create a list of data using bullets.
- Numbering – This option is used to create a list of data using the roman number, decimal number, alphabetic, etc.
- Multilevel List – This option is used to create a multilevel list using alphabetical, numerical, roman numbers etc.
- Align Left (ctrl+l) – This option moves the text left of the page.
- Align Center (ctrl+e) – This option moves the text center of the page.
- Align Right (ctrl+r) – This option moves the text right of the page.

- Justify (ctrl+j) – This option is used to adjust the text equal from the right and left margins.
- Line Spacing – This option is used to adjust the gap between lines and paragraphs.
- Increase Indent – This option is used to move the paragraph or line right, far from the left margin.
- Decrease Indent – This option is used to move the paragraph or line left, closer to the left margin.
- Sort – This option is used to sort the selected alphabetical or numerical data in a table in ascending or descending order.
- Shading – This option is used to add color in the background for a selected paragraph or text.
- Border – This option is used to create the border for a selected paragraph or text.

Styles

- MS Word provides different readymade sets of font styles including colors, fonts, and headings. Users can select anyone and apply it to a selected paragraph or text. Users can also create new styles by clicking the dropdown button.

Editing

- Find (ctrl+f)– This option is used to find any information in the document.
- Go To (ctrl+g) – This option is used to go any page or line directly by a given page or line number. Users can also jump to the sections, bookmarks, footnotes, comments, etc.
- Replace (ctrl+h) – This option is used to replace any text with another text. Replace all function is used to replace multiple similar text at a time.
- Select – This option is used to select all (ctrl+a) the data in the document, also select any object, etc.

Insert Tab

Insert Tab in Microsoft Word is used for inserting objects in the documents, such as – pages, charts, pictures, symbols, tables, shapes hyperlinks etc. The ribbon or toolbar of Insert Tab in Microsoft Word is divided into multiple sections with the name Pages, Tables, Illustrations, Links, Header & Footer, Text, and Symbols. A detailed explanation of the Insert Tab in Microsoft Word is given below.

Pages

- Cover page – The cover page feature can be utilized at the start of the document, or it will be used on the first page of the document. Such as students, teachers, professionals, and writers can insert cover pages in stories, assignments, notes, articles, etc. to highlight the title of the written story, notes, description, and author name. There are so many pre-defined cover page templates. This option is used to insert the fully formatted cover page template for a document, or any book, magazine, notes, or record file. Users can specify the title, author, date, and other information on the cover page.
- Blank Page –This option is used to inserting the blank page in the document. For example, if you want a blank page after writing the first paragraph on one page then you can click on this option. You can also use Ctrl + Enter to insert a blank page after the cursor.
- Page Break – Using this option user can break the page where the mouse cursor is present and move the remaining matter to the next page.

Table

- Table – This option is used to insert a table by a selection of blocks, the user can also specify rows and column numbers according to the need. There are many preformatted table options available in this option.

Illustrations

- Picture – This option is used to insert the picture in the current document from the computer memory or the internet.
- Clip art – With this option, the user can insert clip art in the document, including drawings, movies, sounds, or stock photography to illustrate a specific concept.
- Shapes – This option is used to insert ready-made shapes, such as rectangles, circles, lines and arrows, flow charts, symbols, etc.
- Smart Art – With this option user can insert the graphical ready-made charts to present data in graphical mode. There are many different types of charts available related to tree type charts, organization charts, hierarchy charts, cycle charts, pyramid charts, etc.
- Chart – With this option user can insert charts for illustrating and comparing data. Users can add many types of charts related to Pie chart, Bar chart, Line chart, Area and Surface etc.

Links

- Hyperlink (ctrl+k) – This option is used to create links for selected information with web pages, files, folders, Pictures, or programs.
- Bookmark – This option is used to create bookmarks by assigning a name to selected information. Later we can access them by using the bookmarks option.
- Cross-reference – This option is used to create cross-references to any information using tables, headings, figures, or bookmarks. We can quickly jump to that location by pressing the control key and mouse click.

Header & Footer

- Header – This option is used to edit the header of the document; it is appeared top of each page. Users can specify the header with any short

information such as page number, date and time, and any other text or graphical information.

- Footer – This option behaves the same as the header, but the footer is located at bottom of the page.
- Page Number – This option is used to insert the page numbers in the document. We can customize the area and select different types of page numbering styles.

Text

- Text Box – This option is used to insert a text box to put some text. The text box can be rotated and placed anywhere in the document.
- Quick Parts – This option is used to insert any preformatted snippets such as title, author, or date & time for the document property.
- Word Art – This option is used to insert the decorative text as an object in the document.
- Drop cap – This option is used to make the first letter large at the beginning of the paragraph.
- Signature Line – This option is used to insert the signature line that specifies the individual person who must sign.
- Date and Time – This option is used to insert the current date and time.
- Object – This option is used to insert the embedded object of any program or application.

Symbols

- Equation – This option is used to insert common mathematical equations or build up your own equation using a library of mathematical symbols.
- Symbols – This option is used to insert the special symbols that are not present on the keyboard.

Page Layout Tab

Page Layout Tab in Microsoft Word is used for document page settings like the appearance of the page, page orientation, page size, page margins, page effect, border, etc. The ribbon or toolbar of the Page Layout Tab in Microsoft Word is divided into multiple sections with the name Themes, Page Setup, Page Background, Paragraph, and Arrange. A detailed explanation of the Page Layout Tab in Microsoft Word is given below.

Themes

- Themes – With this option user can change the overall design of the entire document including colors, fonts, and effects.

Page Setup

- Margins – This option is used to select and change the pre-set margins size for the entire document. Users can also customize the margin as per need.
- Orientation – With this option user can switch the page between landscape and portrait mode. Landscape means the page is oriented horizontally. Portrait means the page is oriented vertically.
- Size – This option is used to choose the paper size for the current document for printing or give a custom size.
- Columns – This option is used to divide the document information into two or more columns.
- Breaks – This option is used to add page breaks, section or column breaks, and wrap the text matter to the document.
- Line Numbers – With this option user can insert the line numbers to each line.
- Hyphenation – This option is used to break the word with a – to the next line if no more room to fit the word end of the line. In books or magazines, it creates much more uniform text in order to have more spacing between words.

Page Background

- Watermark – A watermark is a translucent image or picture behind the primary text in a document. This is a very beneficial option that indicates that the document is confidential, and urgent and needs to be treated carefully. Users can add a custom watermark to any text or picture.
- Page Color – This option is used to choose and apply the colors for the background of the document page.
- Page Border – This option is used to add a border around the page. There are many different borders available in this feature.

Paragraph

- Indent Left – This option is used to move the line or paragraph right side from the left margin by giving a value.
- Indent Right- This option is used to move the line or paragraph left side from the right margin by giving a value.
- Spacing Before – Using this option we can add space above the selected paragraph.
- Spacing After – Using this option we can add space below the selected paragraph.

Arrange

- Position – This option is used to set the object or image with text. There are multiple wrapping options as left, right, center, middle position, etc. It automatically wraps the text around the object.
- Bring to Front –Using this option we can bring the selected object forward one level to the other object or all objects.
- Send to Back – Using this option we can send the selected object behind one level to the other object or all objects.
- Text wrapping – This option is used to set the object with text by multiple text wrapping options.

- Align – This option is used to align the multiple selected objects to the page or margin with the top, bottom, right, left, center, or middle.
- Group – This option is used to group multiple objects together so that they can be treated like a single object.
- Rotate – This option is used to rotate or flip the selected object.

Reference Tab

The Reference Tab in Microsoft Word is used to create the table of contents, images caption, cross–references, footnotes, citations & bibliography, etc. The ribbon of the Reference Tab in Microsoft Word is divided into multiple sections with the name Table of Contents, Footnotes, Citations & Bibliography, Captions, Index, and Table of Authorities.

Mailing Tab

Mailing Tab in Microsoft Word is used for creating envelopes, labels, and Mail Merge. Mail Merge is a feature for sending a letter to multiple recipients at a time with separate details like address and greetings lines.

The Mailing Tab in Microsoft Word is divided into multiple sections with the name Create, Start Mail Merge, Write & Insert Fields, Preview Results and Finish.

Review Tab

Review Tab in Microsoft Word is used for checking spelling & grammatical mistakes in the document, writing comments, tracking changes if the document is modified, comparing the document with another, and document protection, etc. The ribbon of the Review Tab in Microsoft Word is divided into multiple sections with the name Proofing, Comment, Tracking, Changes, Compare and Protect.

Proofing

- Spelling & Grammar – This option is used to check the spelling & grammatical mistakes of the text written in the document.
- Thesaurus – This option searches the suggested words with multiple similar meanings.
- Word Count – This option is used to check the number of words, lines, and paragraphs of the selected text or document.

Language

- Translate – This option translates the selected text into a different language. To use this feature, the languages feature must be installed on the computer.
- Set Language – This option is used to set the different language to check the spelling & grammar of the selected text.

Comments

- New Comment – This option is used to add a comment (info about the text) for a section or selected text.
- Delete – This option is used to delete the selected comment or delete all the comments in the documents.
- Previous – This option is used to navigate the previous comment.
- Next – This option is used to navigate the next comment.

Tracking

- Track Changes – This option tracks all the changes made in the document, including insertion, deletion, and formatting changes.
- Final Showing Markup – Final show markup allows the document with all proposed changes included and shows the original document before any changes were made.
- Show Markup – This option allows the user to choose what kind of markup to show in the document and is also used to show or hide comments.

- Reviewing Pane – This option is used to open a separate pane window to show the reviewing.

Changes

- Accept – This option is used to accept the changes one by one or all at once in the document.
- Reject – This option is used to reject the changes one by one or all at once in the documents.
- Previous – This option is used to navigate the previous revision in the document so that we can accept or reject changes.
- Next – This option is used to navigate the next revision in the document so that we can accept or reject changes.

Compare

- Compare – This option is used to compare or combine the multiple versions of a document.

Protect

- Protect Document – This option is used to restrict permissions to the document, that is how people can access the document. Such as passwords.

View Tab

View Tab in Microsoft Word is used for changing the layouts of the pages, showing/hiding elements such as – ruler or grid lines, adjusting pages with the zoom option, working with multiple windows, and recording a macro, etc. The ribbon of the View Tab in Microsoft Word is divided into multiple sections with the name Document Views, Show/Hide, Zoom, Window, and Macros.

Document Views

- Print Layout – This option is used to show the document as it will appear on the printed page. This view is specially used for printing mode.

- Read Mode– This layout hides the contents of the application and adjusts pages with full screen to read.
- Web Layout – This Layout looks like a web page. It has a single page for much matter. Certain elements such as header & footer, and pages number will not be visible in this mode.
- Outline – This Layout looks like an outline mode and shows the outlining tools for generating heading levels.
- Draft – In this view, the document looks like a draft to quickly edit the text. Certain elements such as header & footer will not be visible in this mode.

Show/Hide

- Ruler – This option is used to show or hide the ruler.
- Gridlines – This option is used to show or hide the gridlines. Gridlines are used to align the object in the document.

Zoom

- Zoom – This option is used to set the zoom level according to the pre-set size or custom size.
- 100% Zoom – This option returns 100% zoom level.
- One Page – This option fits the current page in the window.
- Two Pages – This option fits the two pages in the window.
- Page Width – This option fits the page according to the matching width of the window.

Window

- New Window – This option is used to open a new window containing a view of the current document.
- Arrange all – This option is used to arrange all the opened window side by side on the screen.
- Split – This option is used to split the current window into two parts so that the user can view a different section of the document at the same time.

- View Side by Side – This option is used to view the two-document side by side so that user can compare their contents.
- Synchronous Scrolling – This option synchronizes the scrolling of two documents, so that they scroll together.
- Reset Window Position – This option resets the window position of the document being compared side by side so that they share the screen equally.
- Switch Window – This option switches the currently opened window.

Macros

- Macros – This option is used to record a macro, view the list of recorded macros, run the recorded macro, and create or delete a macro. Macro is a program in MS Word that is used to record work including mouse actions and keystrokes.

To create a new blank document:

Word files are called documents. Whenever you start a new project in Word, you'll need to create a new document, which can either be blank or from a template.

Steps to create a new blank document.

- Select the File tab to access Backstage view.
- Select New, then click Blank document.
- A new blank document will appear.

Saving a Document

When you create a new document in Word, you'll need to know how to **save** it so you can access and edit it later.

Save and Save As

Word offers two ways to save a file: Save and Save As. These options work in similar ways, with a few important differences.

- **Save**: When you create or edit a document, you'll use the Save command to save your changes. You'll use this command most of the time. When you save a file, you'll only need to choose a file name and location the first time. After that, you can click the Save command to save it with the same name and location.
- **Save As**: You'll use this command to create a copy of a document while keeping the original. When you use Save As, you'll need to choose a different name and/or location for the copied version.

To save a document using the options provided by MS Word in its File menu, go through the following steps:

Step 1: Click on the File Menu.

Step 2: Go to the Save or Save As button provided.

Step 3: Select the location where you want the file to be saved.

Step 4: Provide a name to the file or use the default one.

Step 5: Click on the Save button.

Saving a File with New Name:

To save an existing file with a new name, **Save As** will be used. The save button will just overwrite the existing file with the same name.

Step 1: Click on the File Menu.

Step 2: Go to the Save As button provided.

Step 3: Select the location where you want the file to be saved.

Step 4: Provide a name to the file.

Step 5: Click on the Save button.

Opening a Document

We use MS-Word to create different types of documents. So, a situation may come where we need to open a document i.e., already existing, and we need to modify that document. So, for this follow the steps below:

Step 1: First Open MS-Word on your PC.

Step 2: Now, click on the “File” tab and the file tab will open.

Step 3: Now, on the left-hand side of the page or screen there are many options. Select the “Open” command to display an open dialog box.

Step 4: The open dialog box displays the list of names of files of the given type in a particular drive or folder. Word documents have the file extension docx.

Step 5: Select the required file from the list displayed in the open dialog box.

Step 6: Click the Open button in the open dialog box.

How to Password Protect a Microsoft Word Document?

Microsoft Word is a software tool that allows us to create document files like articles, office letters, project files, and many more in a very simplest and easy manner

Adding Password on Document

Protecting your Word file helps to prevent other users from accidentally adding, changing, moving, deleting, or modifying data or information in your document. MS Word can lock your document file and then protect it with a password so that other users can’t open and access it. By doing this you can save your data from tampering.

Steps to Protect the Document

Step 1: On the navigation menu bar click on the File option.

Step 2: Next click on the Info option from the left panel.

Step 3: Then click on the down arrow icon in Protect Document option.

Step 4: Then select Encrypt with password option.

Step 5: A Encrypt Document dialog box will open where you can set your password.

Step 6: Create a strong password for your document.

Step 7: Then click on OK.

Step 8: Again, a confirm password dialog box will open to confirm your password.

Step 9: Re-enter your password.

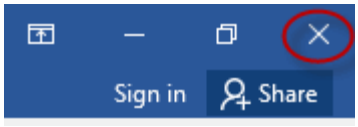
Step 10: Next click on the OK option.

Finally, your document is protected with a password. Whenever anyone tries to open it, it will first ask for a valid password.

Quitting Word

There are two ways to close a Microsoft Word document.

Method 1: Click the Gray "X" in the upper-right-hand corner.



Method 2: Select Close option from the File menu.

Unit 2

Editing Document

Entering Text

A blank document titled Document1 is automatically opened ready, to begin with when MS Word is first started.

Text can be entered in the document window by typing it in using the keyboard. The insertion point or the cursor is where the text is placed in the document when it is typed. The insertion point moves over to accommodate the new text being typed in.

Microsoft word uses the Word Wrap feature in which text automatically wraps to the next line when the user reaches the right margin of the page while typing. Within a paragraph, the text automatically wraps to the next line.

Word offers many features, which make editing of documents very simple and easy. To edit existing text, simply move the I-beam pointer to the text to be edited and click. The insertion point moves to the new position where the user can type new text, format existing text or even delete it.

The <Backspace> and <Delete> keys can be used to delete text. <Backspace> deletes one character to the left of the insertion point, while the <Delete> key erases one character to the right of the insertion point. Press <Ctrl>+<Backspace> and <Ctrl>+<Delete> to delete whole words at a time.

Editing Text

- Making changes in a document like inserting or deleting text, copying or moving text, replacing text, etc. is called editing text.
- Editing a word file means making changes in the text contained in a file.
- To make changes in the block of text, it is necessary to select it. You can select a word, line, paragraph, or entire document text.

Selecting text

- In word, a lot can be done with the text after it is typed. A user can change the appearance of the text, move it around, copy it align it, resize it or delete it if required.
- Whenever you want to edit or format text, you first need to select it.
- Word offers a number of ways in which text can be selected using both the mouse and the keyboard.
- The simplest way to select an area of text is to drag the mouse pointer over the text while holding down the left mouse button.

There are some shortcuts for selecting individual words, lines, sentences, paragraphs, and entire documents.

To Select	Action
A word	Double-click anywhere on the word
Adjacent words	Double click on the first word and drag to select the adjacent words holding down the mouse button
A sentence	Hold <Ctrl> and click anywhere in the sentence
A paragraph	Triple-click anywhere in the paragraph or Move the pointer to the selection bar, point to the paragraph and double-click.
Single line	Move pointer into the left margin or the selection bar the blank strip running down the left edge of the document window where the pointer changes to a right-pointing arrow. Point to the desired line and click.
Multiple lines	Move the pointer top the selection bar. Point the right-pointing arrow to the first desired line and drag to select additional lines holding down the mouse button.
Entire document	Choose Edit → Select All from the menu bar, or hold <Ctrl> and click in the selection bar.
Multiple words, lines, or sentences	Move the I-beam mouse pointer onto the first word, hold down the mouse button, drag to the last word and release the button.
Multiple words, lines or sentence using Shift-select	Click on the first word, move to the last word (with mouse button releases), hold <Shift> and click. Everything between the two clicks is selected.
Graphics and other objects	Click anywhere within the graphic or object. The selected object is surrounded by a border usually with handles.

Table 2.1 Selecting Text using the mouse

Text can also be selected using the keyboard shortcuts shown in Table 2.2.

To increase a selection	Press
One character to the right of the insertion point	<Shift> + <→>
One character to the left of the insertion point	<Shift> + <←>
To the start of a word	<Ctrl> + <Shift> + <→>
To the end of a word	<Ctrl> + <Shift> + <←>
To the start of a line	<Shift> + <Home>

Down one line	<Shift> + <↓>
Up one line	<Shift> + <↑>
To the end of a paragraph	<Ctrl> + <Shift> + <↓>
To the start of a paragraph	<Ctrl> + <Shift> + <↑>
Down one screen	<Shift> + <Page Down>
Up one screen	<Shift> + < Page Up>
To the document's end	<Ctrl> + <Shift> +<End>
To the start of the document	<Ctrl> + <Shift> +<Home>
To a vertical block of text	<Ctrl> + <Shift> + F8 and arrow keys
The Entire document	<Ctrl> + <A>

Table 2.2 Selecting with the keyboard

Moving and Copying text

- Moving and copying text is one of the many features provided by MS Word.

Moving Text

- When a text is moved, it means that the text is deleted or vanished from the current location and pasted to the new location.
- Text can be moved using Cut and Paste, in which text is cut from its current location and pasted in a new location.

To Move text using cut and paste:

1. Select the text to be moved.
2. Now, click on the home tab
3. Now, go to the clipboard section in the home tab
4. Now, in the clipboard section select the cut option or press <Ctrl> + <X> from the keyboard
5. Now, put the cursor where you want the text to be placed
6. Now, select the paste option or press <Ctrl> + <V>from the clipboard section. So, the selected text is successfully moved.

Copying Text

- Copying text creates a copy of the original at the new location leaving the original too in place.
- When a portion of text is copied, the word moves a copy of it to the clipboard leaving the original too in place. Text can be copied using Copy and Paste.

To copy use the Copy and Paste command:

1. Select the text to be copied
2. Now, click on the home tab
3. Now, go to the clipboard section in the home tab
4. Now, in the clipboard section select the copy option or press <Ctrl> + <C> from the keyboard
5. Now, put the cursor where you want the text to be placed
6. Now, select the paste option or press <Ctrl> + <V>from the clipboard section. So, the selected text is successfully Copied.

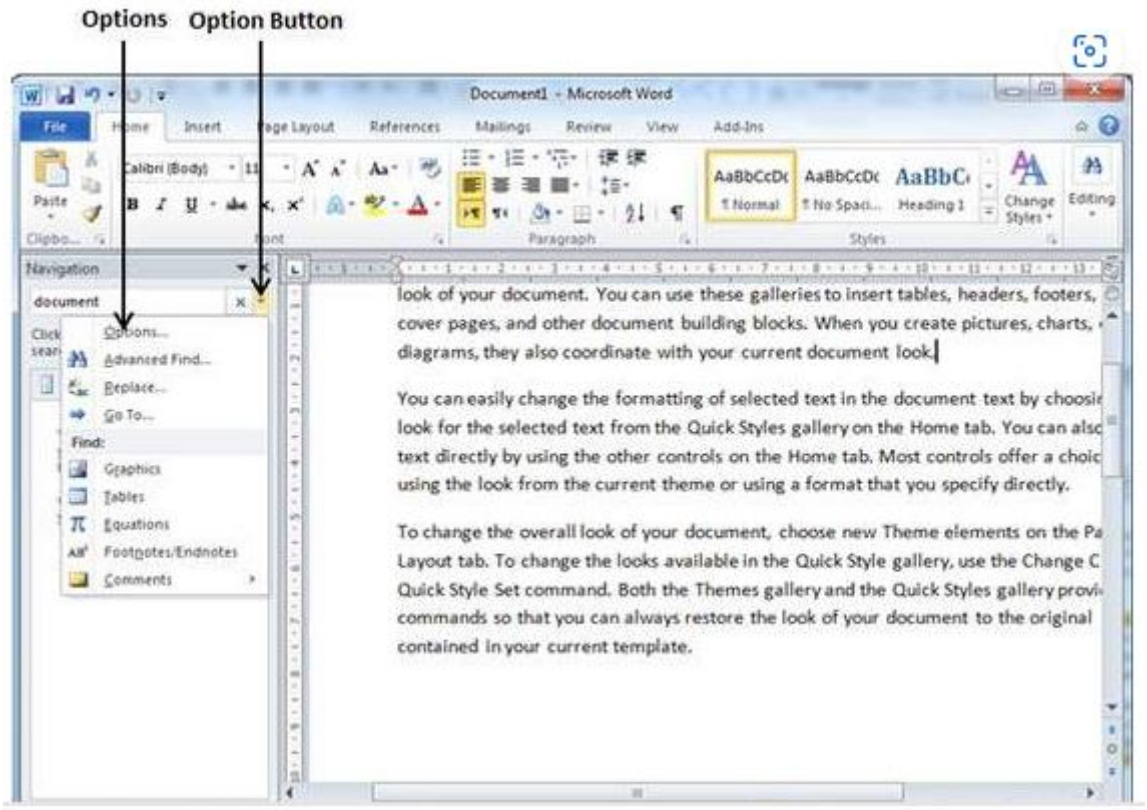
Finding and Replacing text

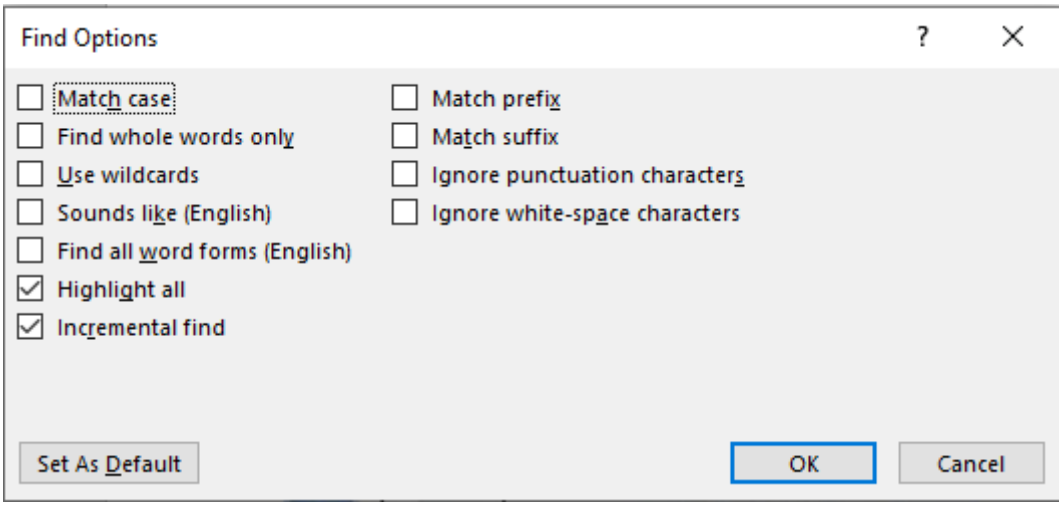
- When you are working with a longer document it can be difficult and time-consuming to locate a specific word or phrase, one of the fastest ways to make repetitive changes to a document is through Find and Replace.
- In MS word, Find can be used to search and locate a specific text string that is required for review or editing and replace helps to automatically substitute new text for the existing string.

Using Find

The Find command enables you to locate specific text in your document.

1. Click the Find option in the Editing group on the Home tab or press Ctrl + F to launch the Navigation pane.
2. Enter a word that you want to search in the Search box, as soon as you finish typing, Word searches for the text you entered and displays the results in the navigation pane and highlights the word in the document.
3. You can click the clear button (X) to clear the search and results and perform another search.
4. You can use further options while searching for a word. Click the options button to display the options menu and then click the Options button this will display a list of options. You can select options like Match case to perform a case-sensitive search.
5. Finally, if you are done with the Search operation, you can click the close button (X) to close the Navigation Pane.





Using Replace:

To replace text

1. To find and replace a particular word or phrase for another, navigate to the Editing group in the ‘Home’ tab and click the ‘Replace’ command. Alternatively, you can also press the keyboard shortcut Ctrl + H.
2. This will open the Find and Replace dialog box. In the Find and Replace dialog box, fill out the two fields:
3. Type the text you want to search for in the ‘Find what:’ field.
4. Type the text you want to replace within the ‘Replace with:’ field.

5. Once you enter the necessary texts, you can either click the ‘Replace’ button to replace the text one by one or ‘Replace All’ to replace all the instances of the word at once.
6. Click the ‘Find Next’ to find the first instance of the text highlighted in gray and click ‘Find Next’ again to jump to the next instance. Review each instance and click ‘Replace’ to exchange the currently highlighted text.

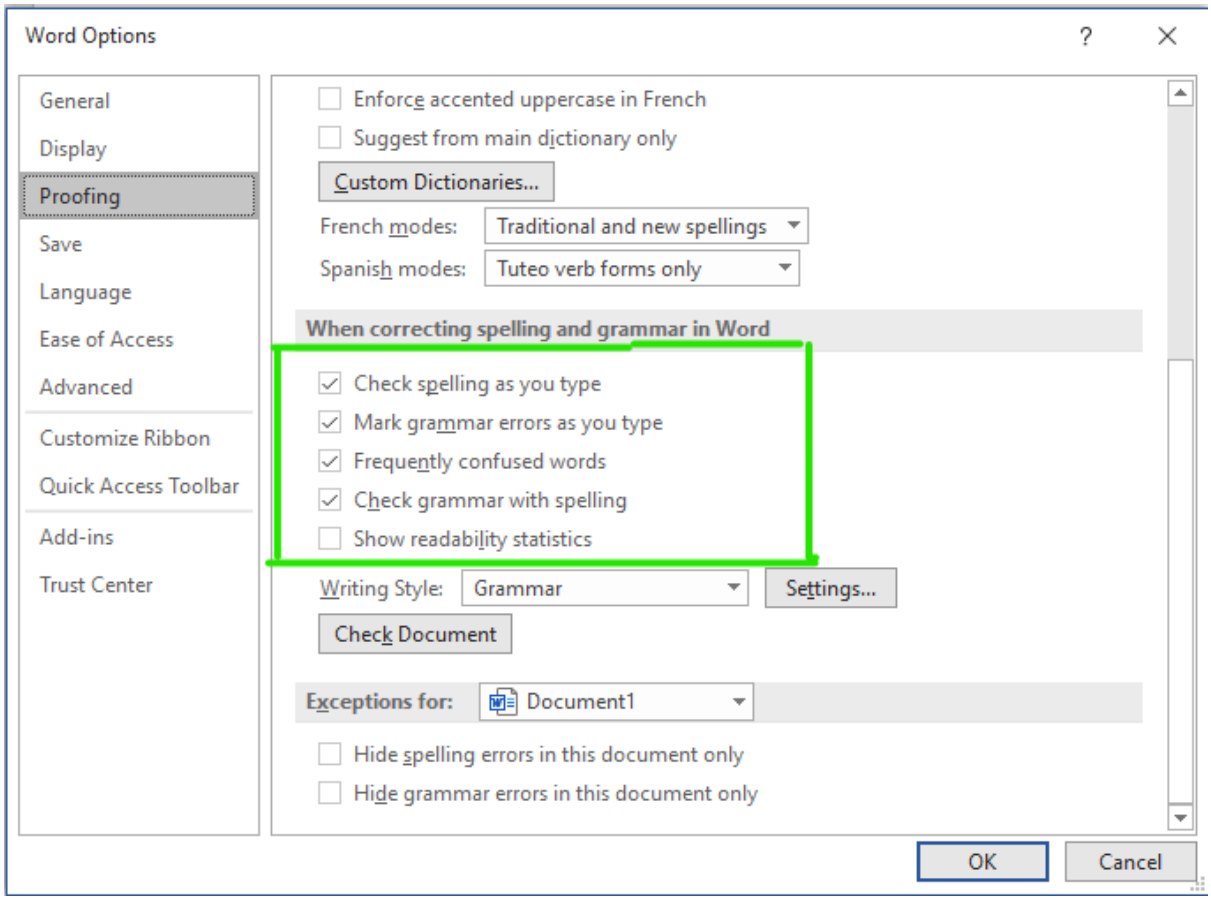
Working with Tools:

Spell Check and Grammar Check

- Microsoft Word has a special feature called spell check and grammar check that allows you to check spelling and grammatical mistakes that you made in the document.
- Basically, spell check is a software tool that identifies the misspelled words present in the document. It also allows you to search for a particular word in the document that you know you’ve misspelled in the whole document.
- In Microsoft Word documents, Word’s spell check function is set to automatically check your spelling while you type.
- Errors in your document will have color-coded underlines reflecting your choices, like red for spelling errors, green for grammar errors, and blue for contextual spelling errors.
- Microsoft Word provides several ways to check spelling and grammar:

Steps to enable Spell Checker in MS Word

Step 1: On the navigation menu bar click on the File Tab
Step 2: Next click on the options button.
Step 3: A word options dialog box will appear on the screen.
Step 4: Now select the Proofing option from the left menu.
Step 5: Check all boxes as shown in the figure:



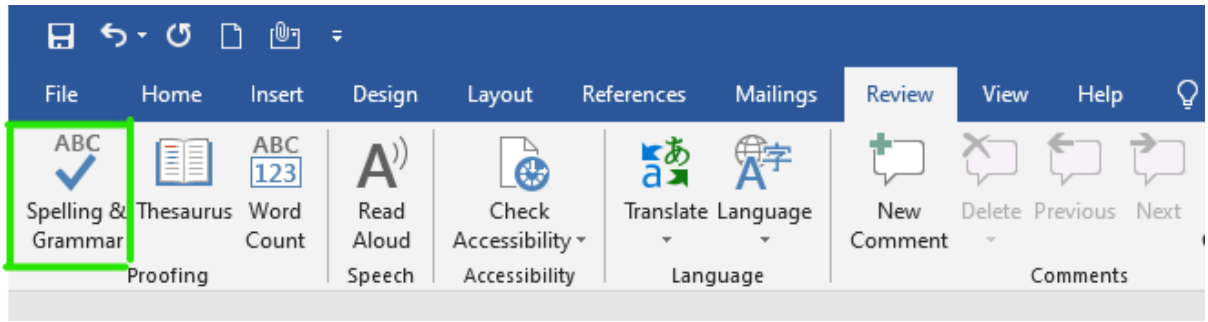
Step 6: Next click on the OK button.

The user can choose to check spelling and grammar all at once and then confirm each correction.

To Check Spelling and Grammar at once:

Step1: From the Proofing group on the Review tab, select Spelling & Grammar

Step 2: Next click on the spelling & grammar option as shown in the figure:



Step 3: If there is a spelling mistake in the document then it will appear in the red underlined text.

Step 4: If there is a grammar mistake in the document then it will appear in the

Step 4: Select the correct spelling from the suggestion menu.

Step 5: Next click on the YES option.

Step 6: Click on the OK option.

Shortcut Method to Check Spellings

Step 1: Right-click on the red-underlined word.
Step 2: Now choose the correct spelling.
Ignore: This will skip the word without changing it.
Ignore All: This will skip the word without changing it. And it will also skip all other instances of the word in the document.
Add: This adds the word to the dictionary so it will never come up as an error.

Shortcut Method to Check Grammar

Step 1: Right-click on the blue-underlined word.
Step 2: Now choose the correct Grammar.

Ignore: This will skip the word or phrase without changing it.

AutoCorrect and How to Create AutoCorrect entry:

The AutoCorrect feature in word can be used to automatically detect and correct typos, misspelled words, grammatical errors, and incorrect capitalization, AutoCorrect can also be used to quickly insert text, and symbols.

The dialog box lists certain common errors, which are corrected automatically if the corresponding check box is chosen. This feature offers to Correct Two Initial Capitals, capitalize the first letter of a sentence, Capitalize the names of days, and Correct accidental use of the CAPS LOCK key. A user can leave these options on or turn them off, based on personal preference.

AutoCorrect can use a list of built-in corrections, called AutoCorrect entries, to detect and correct misspelled words, grammatical errors, and common symbols.

A user can easily add his own AutoCorrect entries or remove unwanted ones.

Steps to create AutoCorrect Entry

Choosing File Menu ⟶ Options ⟶ Proofing ⟶ AutoCorrect Options

This will open the AutoCorrect dialog box.

The Replace and With text boxes under the option ‘Replace text as you type’, allows a user to build his own list of common misspellings.

Simply type in the with its commonly misspelled word in the replace box and type in the correct spelling in the With box.

For Example, we can enter ‘adress in the Replace box and correct the word ’address’ in the With box. When MS Word encounters one of these words, it automatically fixes them by replacing the wrong spelling with the given correct one (Wherever ‘adress’ is typed, it is replaced with ’address’, the correct spelling). Some words such as “adn” (misspelled and) and “teh” (misspelled the) are already present in the list.

Although AutoCorrect is designed to prevent typing and spelling errors, it is also valuable as a shortcut tool.

Words that are typed regularly can be entered into the AutoCorrect list to save time and keystrokes- long company names, for example, or legal or medical terminology.

If the name of the company is long, such as Hindustan Lever Limited, and it is to be typed many times in a document, the user can enter a suitable and recognizable code such as HLL for it in the AutoCorrect list. So, whenever HLL is typed, Word automatically expands it to Hindustan Lever Limited saving the user’s time and effort.

Auto Format

A software feature commonly found in word processor programs such as Microsoft Word that automatically changes the formatting or appearance of text. For example, if a user types "1st," AutoFormat would change it to "1st."
How to enable and disable the Microsoft Word AutoFormat feature

1. Open Microsoft Word.
2. Click the File tab.
3. Click the Options option in the left navigation pane.
4. In the Word Options window, click the Proofing option in the left navigation pane.
5. Click the AutoCorrect Options button.
6. In the AutoCorrect window, click the AutoFormat tab.

In this section, users can enable or disable each of the AutoFormat features they do not want to use.

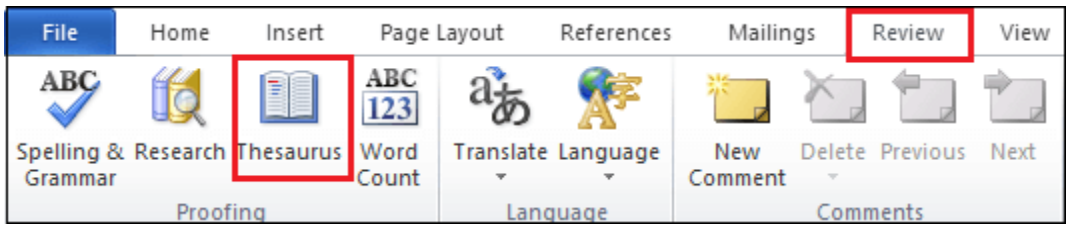
Thesaurus

The thesaurus helps the user to find out more descriptive, imaginative, or precise words to make the document text more appropriate. It can be used to find synonyms, antonyms, or related words of any selected word.

Thesaurus using the Review tab.
There are the following steps to use Thesaurus using the Review tab -
Step 1: Open the new Word document or an existing Word document.

Step 2: Type the word in the document that you want to look up (Thesaurus)

Step 3: Go to the Review tab on the Ribbon and click on the Thesaurus option in the Proofing group or you can simply press the shortcut key Shift + F7.



Step 4: The Thesaurus dialog box screen appears at the right corner of the Word document which does the following:

- Select the word.
- Click on the drop-down icon associated with the selected word.
- A small drop-down menu appears, click on the Insert.

It is also possible to find a common synonym for a word by selecting it, right-clicking the word, and pointing to synonyms on the shortcut menu that appears. Either choose from the synonyms given or choose Thesaurus to open the Thesaurus dialog box.

Questions

Q.1 Short Note: 2Marks Each

1. Find And Replace
2. Spell Check
3. Grammar Check
4. Auto Correct
5. Creating AutoCorrect entry
6. Auto Format
7. Thesaurus

Q2. How to Edit Document? Explain in detail. (10 Mark)

Unit 3

Formatting Document

What is Formatting?

- The way a document looks can be as important as its contents.
- Formatting consists of changing the appearance of the document so that it becomes more visually attractive, and the information can be presented in the most effective manner to have a greater impact on its reader.

"Formatting means changing the appearance of a document to make it attractive".

- The user can format individual characters, lines, paragraphs, pages, or the entire document.

There are three types of formatting.

1. Character Formatting
2. Paragraph Formatting
3. Page Formatting

3.1 Character Formatting

- Character Formatting is formatting that can be applied to one or more characters. Character formatting is very easy and can have a great impact on the overall appearance of a document.

"Character formatting plays a significant role in document formatting by determining the appearance of individual characters in the document".

Character formatting using the Font dialog box

- The font dialog box offers to use maximum control over font formatting by providing all the character-formatting options together in one place.

To set character formatting using the Font dialog box:

1. Select the character on which character formatting is to be applied.
2. Click the Home Tab. In the font group, click the dialog box launcher button. The button is found in the lower-right corner of the font group.

- Use the <Ctrl>+<D> keyboard shortcut to quickly open the font dialog box

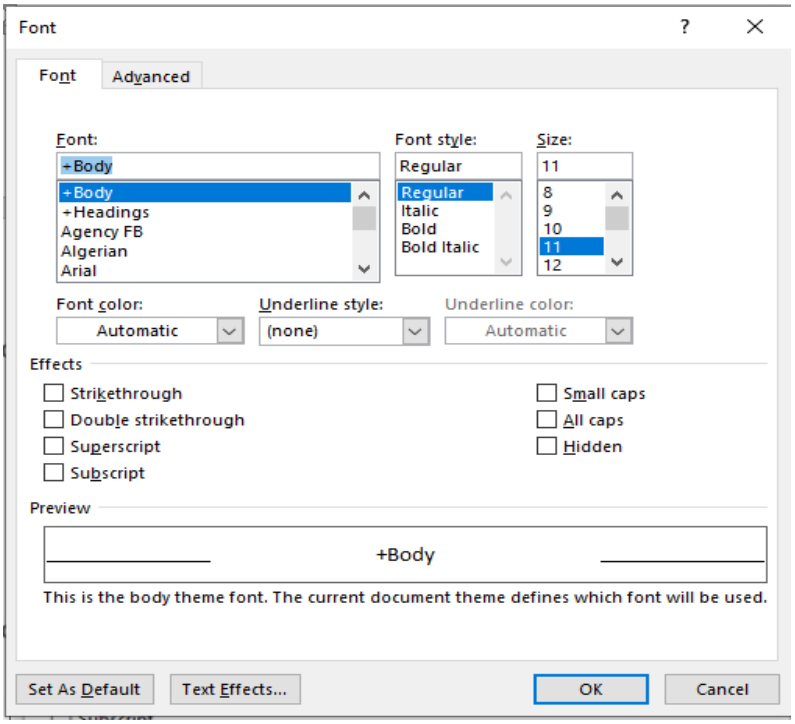


Fig. Font page of the Font Dialog Box

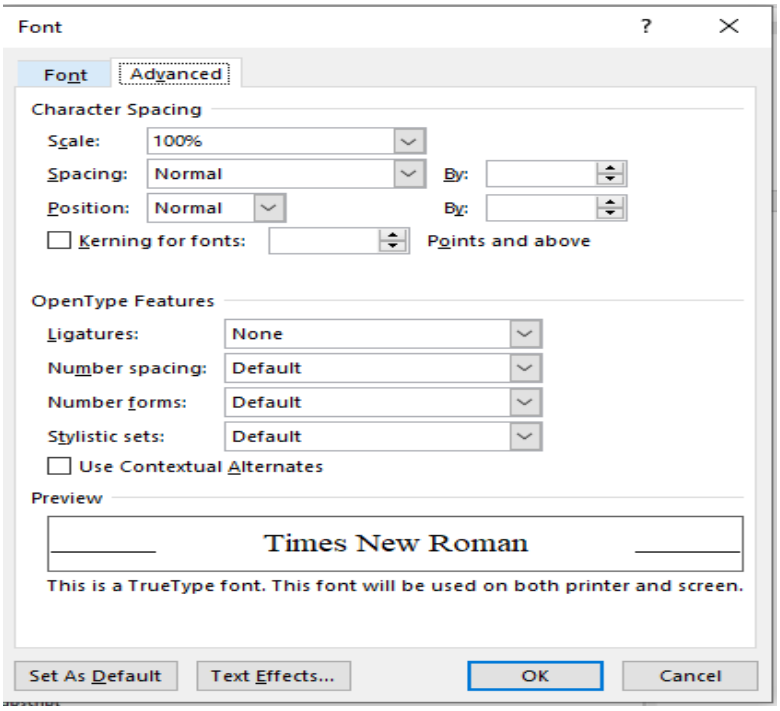


Fig. Advanced page of the Font Dialog Box

3. Font Dialog box has a two-page tab: Font and Advanced
4. Set options desired by selecting the Font, Font style, Size, and Font Color in their respective lists. Word provides several options for the Underline style that can

be chosen. The underline color option becomes available when an underline is chosen to be applied.

The font is the typeface used to display characters. This includes all fonts properly installed in the system.

Font Style Consists of several styles for formatting (Bold, Italic, Regular)

Size is the size of the characters, expressed in points where a point is 1/72 inch.

Font color can be used to apply a selected color to the characters.

5. A number of special text effects can be applied to the text by turning on the desired options in the Effect area of the font dialog box.
6. The different effects available are shown below:

Special Effects	Example
Strikethrough	~~Computer Application~~
Double Strikethrough	~~Computer Application~~
Superscript	Computer $^{1st\ year}$
Subscript	Computer $_{1st\ year}$
Small Caps	COMPUTER APPLICATION
All Caps	COMPUTER APPLICATION
Shadow	Computer Application
Outline	Computer Application

Hidden – The hidden text does not print and can only be seen when Show/Hide is enabled.

7. The preview window in the dialog box displays what the selected text looks like with the specified options applied.
8. The Font dialog box Advanced tab hosts options for changing the size and position of text on a line.
9. Character Spacing can be used to adjust the distance between characters. Text can be expanded or condensed using this option.
10. The set as default button in the Font dialog box is used to change the font that word uses for a new document. If you prefer to use a specific font face all your documents, choose the font in the font dialog box. Click the ok button afterward, all documents start with the font options you selected.

11. Some formatting options can also be applied with the use of keyboard shortcuts: Select text and press<Ctrl>+<B> to make it bold, press<Ctrl>+<I> to make it italicized, and <Ctrl>+<U> to all an underline to it.

The following table shows the lists of some more keyboard shortcuts for applying character formatting.

Shortcut	Effect
<Ctrl> + <Shift> + <.>	Increase font size (in steps)
<Ctrl> + <Shift> + <,>	Decrease font size (in steps)
<Ctrl> + <]>	Increase font size by 1 point
<Ctrl> + <[>	Decrease font size by 1 point
<Shift> + <F3>	Change case (cycle)
<Ctrl> + <Shift> + <A>	All capitals
<Ctrl> + <Shift> + <K>	Small capitals
<Ctrl> + <B>	Bold
<Ctrl> + <U>	Underline
<Ctrl> + <Shift> + <W>	Underline (single words)
<Ctrl> + <Shift> + <D>	Double-underline
<Ctrl> + <Shift> + <H>	Hidden text
<Ctrl> + <I>	Italic
<Ctrl> + <+>	Subscript
<Ctrl> + <Shift> + <=> (i.e. <Ctrl> + <+>)	Superscript
<Ctrl> + <Shift> + <Z>	Remove formatting
<Ctrl> + <Shift> + <Q>	Change to Symbol font

Table Character Formatting Using Keyboard Shortcut

3.2 Paragraph Formatting

- Paragraph Formatting is a change in the format of the text that affects an entire paragraph.
- You can change the look of a paragraph by changing its indentation, alignment, and line spacing, in addition to the space before and after it. You can also put borders around it and shade its background. Collectively, the settings you use to vary the look of a paragraph are called paragraph formatting.

To open the Paragraph dialog box

● On the Home tab or the Layout tab, in the Paragraph group, click the

Paragraph dialog box launcher.

The paragraph dialog box has a two-page tab: 1. Indents and Spacing 2. Line and Page Break.

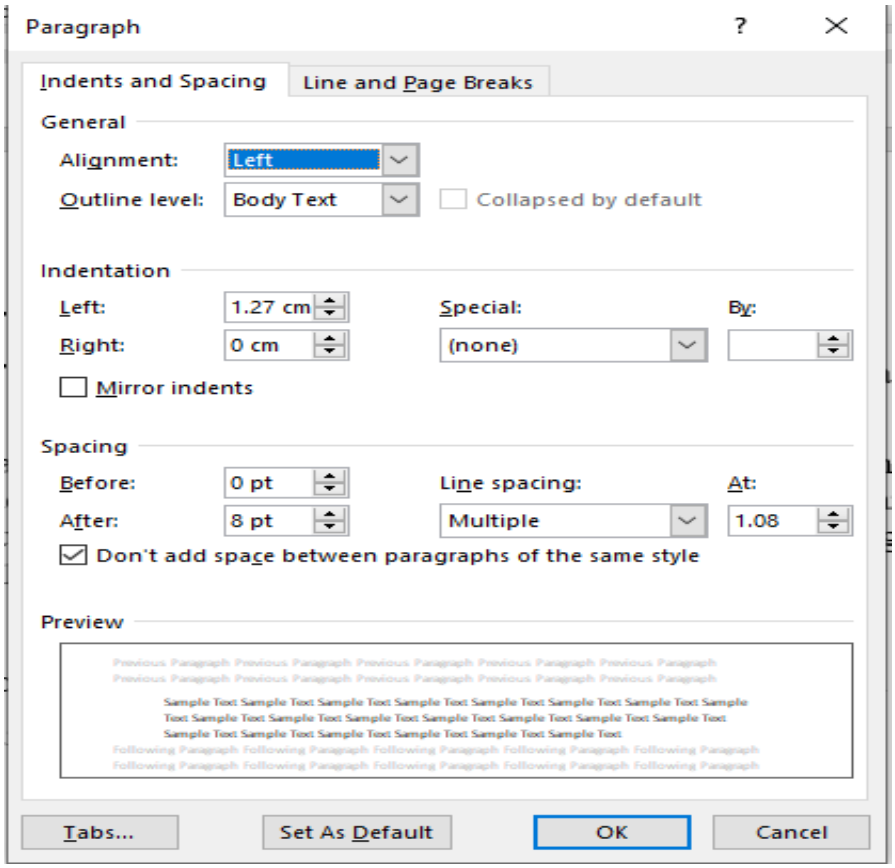


Fig. Indents and Spacing page of the Paragraph Dialog Box

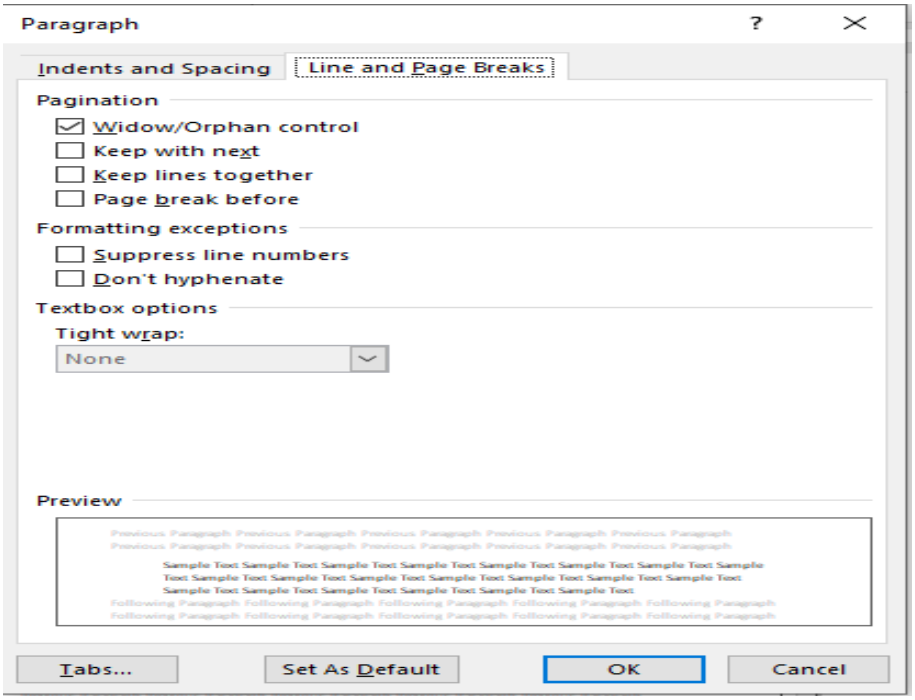


Fig. Line and Page Breaks of the Paragraph Dialog Box

General

Alignment:

The alignment settings control the horizontal position of the paragraph text between the page margins.

There are four alignment options:

- **Align Left** This is the default paragraph alignment. It sets the left end of each line of the paragraph at the left page margin.
- **Align Right** This sets the right end of each line of the paragraph at the right page margin.
- **Center** This centers each line of the paragraph between the left and right page margins.
- **Justify** This alignment adjusts the spacing between words so that the left end of each line of the paragraph is at the left page margin and the right end of each line of the paragraph (other than the last line) is at the right margin.

To set paragraph alignment

1. Position the cursor anywhere in the paragraph or select all the paragraphs you want to adjust.
2. Open the Paragraph dialog box. On the Indents and Spacing tab, in the General area, click Left, centered, Right, or Justified in the Alignment list.

Indentation

In Word, you don't define the width of paragraphs and the length of pages by defining the area occupied by the text; instead, you define the size of the white space—the left, right, top, and bottom margins—around the text.

Although the left and right margins are set for a whole document or for a section of a document, you can vary the position of the paragraphs between the margins by indenting the left or right edge of the paragraph.

A paragraph indent is a space from the page margin to the text. You can change the left indent by clicking the buttons on the home tab, or you can set the indents directly on the ruler.

Left Indent

A Left indent sets a paragraph off from the rest of the text by moving all lines in from the left margin.

Right Indent

A right indent moves text in from the right margin.

First Line Indent

A First Line Indent can be used to indent the first line in a paragraph when it is moved to the right. It works the same way as pressing <Tab> on the keyboard.

Hanging Indent

A hanging indent is when the first line of the paragraph is flush at the margin, and all succeeding lines of the paragraph are indented. Hanging indents are frequently used for bibliographic entries, and resume glossary terms.

To create a hanging indent or first-line indent

1. Position the cursor anywhere in the paragraph, or select all the paragraphs you want to adjust.
2. Open the Paragraph dialog box. On the Indents and Spacing tab, in the
3. Indents area, click First line or Hanging in the Special box.
4. In the By box, set the amount of the indent, and then click OK.

Paragraphs have two types of vertical spacing:

Paragraph spacing:

The space between paragraphs is defined by setting the space before and after each paragraph.

Adjusting the Space before and after paragraphs

The space area in the paragraph dialog box has two text boxes Before and After which allow the user to define the amount of white space to be placed before and after paragraphs. The space Spacing can be specified in points (pt), inches (in), centimeters (cm), or lines (li),

This space is usually measured in points.

To adjust the spacing between paragraphs

1. Select all the paragraphs you want to adjust.
2. Open the Paragraph dialog box. On the Indents and Spacing tab, in the Spacing area, make the adjustments you want to the paragraph spacing, and adjust the Spacing Before and Spacing After settings.
3. Then click OK.

Line spacing:

The space between the lines of the paragraph is defined by setting the height of the lines either in relation to the height of the text (Single, Double, or a specific number of lines) or by specifying a minimum or exact point measurement.

The default line spacing for documents created in Word 2016 is 1.08 lines. Changing the line spacing changes, the appearance and readability of the text in the paragraph, and, of course, also change the amount of space it occupies on the page.

Adjusting the space between Lines

Microsoft Word uses single-line spacing by default. The line spacing selected affects all lines of text in the selected paragraph or in the paragraph that contains the insertion point.

Word provides the following six options:

Single, 1.5 lines, Double, At Least, Exactly, Multiple

To adjust the spacing between the lines of paragraphs

1. Position the cursor anywhere in the paragraph, or select all the paragraphs you want to adjust.

2. Open the Paragraph dialog box. On the Indents and Spacing tab, in the Spacing area, make the adjustments you want to the paragraph spacing, and then click OK.

Tabs

A tab character defines the space between two document elements. For example, you can separate numbers from list items, or columns of text, by using tabs. You can then set tab stops that define the location and alignment of the tabbed text.

You can align text in different ways by using tabs

You can align lines of text in different locations across the page by using tab stops. The easiest way to set tab stops is directly on the horizontal ruler. By default, Word sets left-aligned tab stops every half inch (1.27 centimeters). (The default tab stops aren't shown on the ruler.) To set a custom tab stop, start by clicking the Tab button (located at the intersection of the vertical and horizontal rulers) until the type of tab stops you want appears.

To open the Tabs dialog box

1. Select any portion of one or more paragraphs that you want to manage the tab stops for.
2. Open the Paragraph dialog box.
3. In the lower-left corner of the Indents and Spacing tab, click the Tabs button

You have the following tab options:

Left Tab: This is the default tab stop. Aligns the left end of the text with the tab stop

Center Tab: Align the center of the text with the tab stop.

Right Tab: Aligns the right end of the text with the tab stop

Decimal Tab: Aligns the decimal point in the text (usually a numeric value) with the tab stop

Bar Tab: Draw a vertical line at the position of the tab stop

Line and Page Break

When a document containing more than one page is created, word paginates the document, that is, Word inserts page breaks at appropriate places.

There are six options in word to control the way text flows from page to page in the document:

Window/Orphan Control:

Keep Lines Together:

Keep With Next:

Page Break Before:

Suppress:

Don’t Hyphenate:

Preview:

The preview window in the dialog box displays what the selected text looks like with the specified options applied.

3.3 Adding Borders and Shading:

In MS Word borders are a set of lines, designs, and Shapes. Text boxes and Colourful graphics are added in word to make documents more attractive and beautiful.

In MS Word we can apply the Borders and Shading to the written text, Paragraphs, etc. to make them good.

Adding Border:

The border as the name suggests provides an outer and inner lining on the selected text or paragraph.

Borders are lines that can be added to the top, bottom, left, and right of paragraphs.

Borders can be added to documents in one of three ways: By using the formatting toolbar, using the Borders and Shading Dialog Box, Border option in the paragraph group under the home tab.

Using the Border option in the paragraph group under the home tab

1. Select the text/ paragraph to apply the border.
2. Now, go to the Border option in the paragraph group under the home menu
3. Select the border of your choice.
    - Bottom Border: Border at the bottom of the paragraph
    - Top Border: Border at the top of the paragraph
    - Left Border: Border on the left side of the paragraph.
    - Right Border: Border on the right side of the paragraph
    - No Border: Removes all the border
    - All Borders: Borders on all sides of the paragraph
    - Outside Borders: Borders outside the paragraph
    - Inside Borders: Borders inside the paragraph
    - Inside Horizontal Border: Horizontal border on both inside the paragraph

- Inside Vertical Border: Vertical borders on both sides inside the paragraph

Adding Shades:

Adding shades is like highlighting the text/ paragraph.

In MS Word there are multiple colors available in the shading tab, which you can apply to your text or paragraph like red, blue, etc.

Steps to apply to shading

1. Select the text to apply the shading.
2. Now, go to the shading option in the paragraph group under the home tab.
3. Select the shading color of your choice.

Adding Border and Shading using Border and Shading Dialog Box

The Border and Shading dialog box offer some additional features and ways of using borders. **To add borders and shading using the dialog box:**

To add a border to a paragraph, click anywhere in the paragraph. To add a border only to specific text or object, select the text or object.

1. Select the text you need to apply borders or shading
2. Click the Design tab
3. Go to Page background group
4. Click the Page Borders, you may see the Borders and Shading window popping up

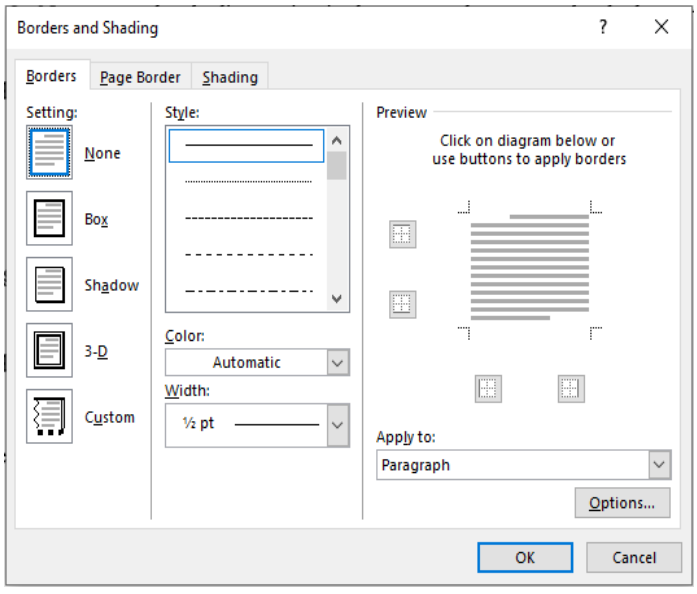


Fig. Border and Shading Dialog Box

The Border and Shading dialog box consist of three pages.

Border, Page Border, and Shading

- In the setting area, you can choose from among five types of borders:
    - None: No border, the default setting, click this setting to remove a border that already exists.
    - Box: Place a line along each of the four sides of your page.
    - Shadow: Borders all four sides but places a thicker border on the bottom and right sides to create a shadow effect.
    - 3-D: Make it appear that your page rises from its background
    - Custom: Enables you to customize your own borders, one side at a time.
- After you select a type of page border, you can control the line style, color, and width of the line you use.
- To choose a dotted, double, or triple line, select the line you want from the type scroll box.
- To specify a color for your page border, click on the color drop-down box and choose the color from the palette that appears.
- To specify a thickness for your border lines, click on the width drop-down box and select any of the options there, from ½ point to 6 points.
- As you make changes, you can view their effects in the preview box.
- Specifying which pages are bordered.
    - Whole document-Borders the entire document
    - This section: Borders all pages in your insertion points current section.
    - This section first page only
    - This section all except first page.

3.4 Page Formatting

Page formatting involves changing the appearance of entire pages. In formatting a page, the user can determine the margins between the text and the edge of the page, the orientation of the page, the size of the paper, etc.

3.4.1 Page Setup

The page setup command offers several options for formatting pages in a document before the document is sent for printing.

To use the Page Setup dialog box to specifically set page margins, obey these steps:

1. Click the Layout tab.

2. Click the dialog box launcher in the lower-right corner of the Page Setup group. Click to open the Margins Page

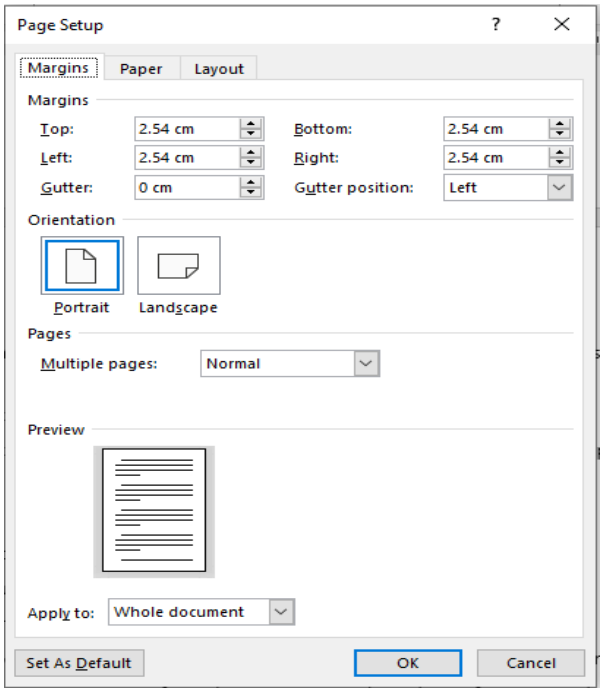


Fig. Margins page of the Page Setup dialog box

3. Use the Top, Bottom, Left, and Right spin box controls to set the amount of white space on the four edges of the document.
4. The Gutter margin should be specified for adding additional space to documents that will be bound. If the binding material takes up half an inch, a 0.5” gutter space should be specified. Specify whether the gutter space is to be added on the Top of each page or to the Left in the Gutter position area.
5. In the Orientation area, choose either Portrait or Landscape orientation. The default page orientation in the word is Portrait.
    - When the page orientation is a portrait, the page is taller than it is wide.
    - Landscape orientation, on the other hand, creates a page that is wider than it is tall.
6. Click the Mirror margins check box to activate mirror margins, if desired.
7. The mirror margin feature helps to format for back-to-back printing.
8. Indicate whether 2 pages are to be printed per sheet in the check box for the same.
9. The Apply to the drop-down list has two options: whole document or This Point Forward. The default for the apply to control is the whole document.
10. Click ok to apply the settings and return to the document.

Click the Layout tab.

Click the dialog box launcher in the lower-right corner of the Page Setup group. Click to select Paper tab:

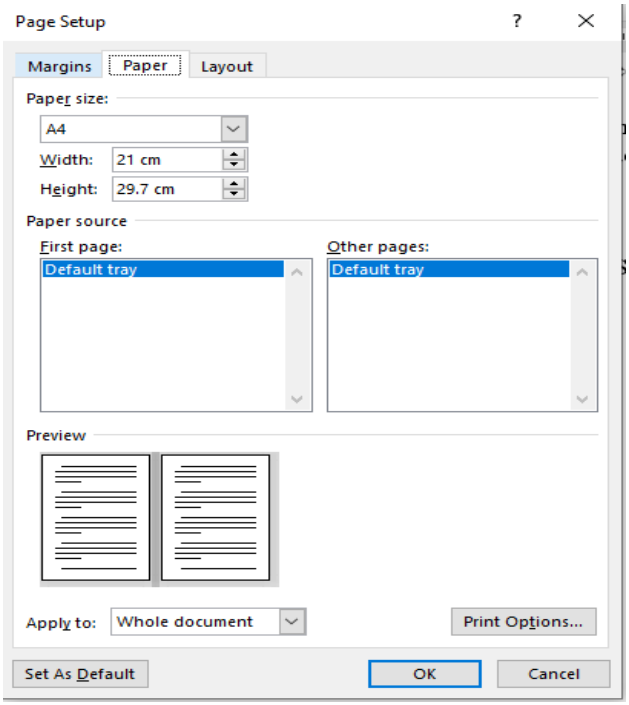


Fig. The Paper tab in the Page Setup dialog box

11. To change the paper size, Click the paper size drop-down list and select an option.
12. The width and height of the paper size chosen to appear in the corresponding controls.
13. The Paper Source is used to specify the source from where the first page and other pages of the document will be taken from for printing.

Click the Layout tab.

Click the dialog box launcher in the lower-right corner of the Page Setup group. Click on the Layout tab:

14. Sometimes it may be required to align text vertically. The vertical alignment drop-down box offers three options:

- Top: the default setting where the text lines up with the top margin.
- Center: text on the page is centered between the top and bottom margins
- Justify: the text is spread out so that each line is the same distance apart from the top line at the top margin and the bottom line at the bottom margin.
- Bottom: text lines up with the bottom margins.

15. Click OK to apply the setting and close the dialog box.

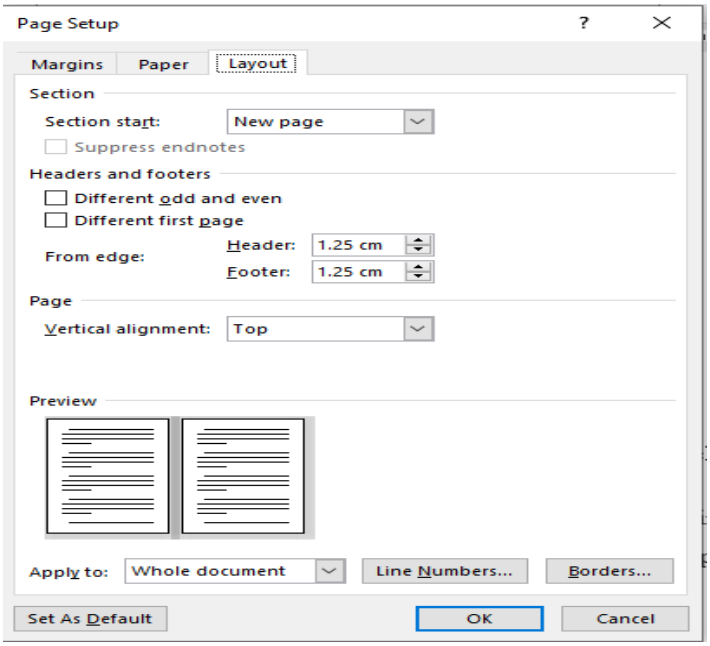


Fig. Layout Page of the Page Setup dialog box

3.5 Adding Page number

- Page numbers can be used to automatically number each page in your document. They come in a wide range of number formats and can be customized to suit your needs.
- Word can automatically label each page with a page number and place it in a header, footer, or side margin.

To insert page number:

- On the Insert tab, click the Page Number command.
- Open the Top of Page, Bottom of Page, or Page Margins menu, depending on where you want the page number to be positioned, then select the desired style of header.
- Page numbering will appear.

3.6 Headers & Footers

- The header is a section of the document that appears in the top margin, while the footer is a section of the document that appears in the bottom margin.

- Headers and footers generally contain additional information such as page numbers, dates, an author's name, and footnotes, which can help keep longer documents organized and make them easier to read.
- Text entered in the header or footer will appear on each page of the document.

To insert a header and footer in Microsoft Word, follow the below given basic steps:

Step 1: Open the new or an existing Word document in which you want to insert the header and footer.

Step 2: Go to the Insert tab at the top of the Ribbon.

Step 3: Click on either header or footer drop-down menu in the Header & Footer section.

Step 4: A Header or Footer drop-down menu will display on the screen with a list of built-in Header or Footer options. Select your desired option from the Built-in list.

Unit 4

Introduction to MS-Excel

Introduction to Excel Working Environment:

MS-Excel:

- Microsoft Excel is an office use application designed by Microsoft. It comes with Office Suite with several other Microsoft applications, such as Word, PowerPoint, Access, Outlook, and OneNote, etc.
- Microsoft Excel is an electronic spreadsheet. You can use it to organize your data into rows and columns. You can also use it to perform mathematical calculations quickly.
- Microsoft Excel is one of the most suitable spreadsheet programs that help us to store and represent the data in tabular form, manage and manipulate data, create optically logical charts, and more. Excel provides you the worksheet to create a new document in it.
- You can save the Excel file with .xls extension.
- Microsoft Excel is a full-featured spreadsheet application that can be used to create worksheets, charts, lists. It is a very user-friendly package, which is the most popular and most widely used, especially among windows user. Entering data onto a spreadsheet is quick and easy. Once data has been entered in a worksheet, Excel can instantly perform any type of calculation on it.
- Microsoft Excel basically comprises of three components:
    - Spreadsheet component, which is used to display and analyze text and numbers in rows and columns.
    - Database component, which is used to manipulate lists of information
    - Chart component, used to produced charts which help to present data in a graphical manner.
- Each component provides a different way of looking at and interacting with the data and is based on a common structure of rows and columns.

Starting Excel

how to start an excel application in simple steps. Assuming you have Microsoft Office installed in your PC, start the excel application following the below mentioned steps in your PC.

Step 1 − Click on the Start button.

Step 2 − Click on All Programs option from the menu.

Step 3 − Search for Microsoft Office from the sub menu and click it.

Step 4 − Search for Microsoft Excel from the submenu and click it.

This will launch the Microsoft Excel application.

Getting to know the excel screen.

- When Excel is launched, two windows appear, nested one within the other.
- The larger window, which covers the entire screen, is called the Excel Application Window.
- The smaller window is called the document window opens with a new a new Excel Workbook named Book1. Excel's interface combines common windows screen elements with buttons, commands, and controls that are specific to Excel. The Excel Screen or the application window is shown in below fig. 4.1.

The main components or elements of the Excel Screen are described below:

1. Quick Access Toolbar

   This toolbar is located in the upper left corner of the screen.

   Its objective is to show the most frequently used Excel commands.

   We can customize this toolbar based on our preferred commands.

2. Title Bar

   The title bar of the spreadsheet is at the top of the window.

   It displays the active document's name.

3. Control Buttons

- Control buttons are the symbols that are present in the upper-right side of the window, enabling us to change the labels, minimize, maximize, share, and close the sheet.

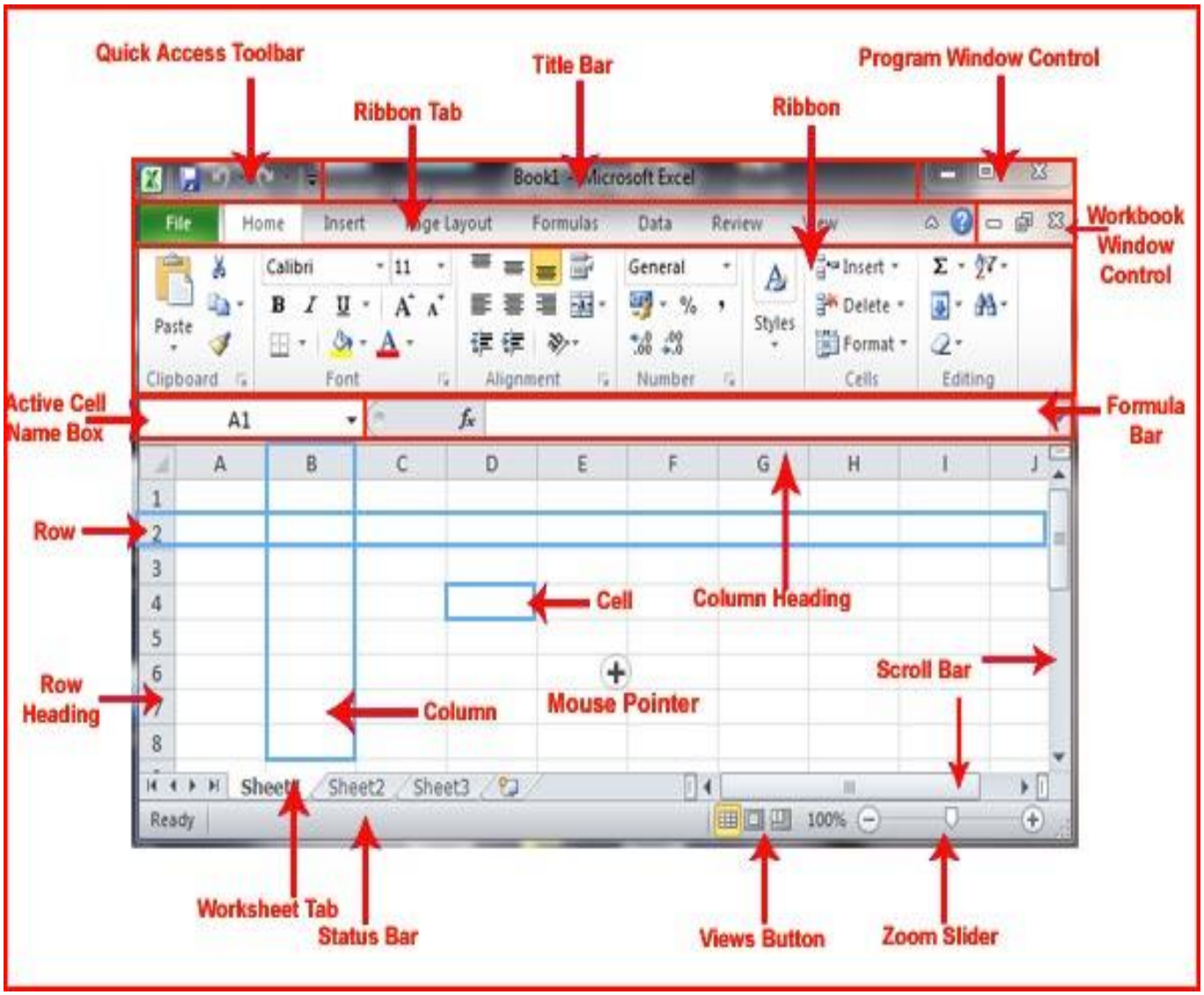


Fig. 4.1: The Excel Screen

4. Ribbon/Toolbar

Each menu bar contains several different elements. On the selection of the menu, a sequence of command options/icons will show on a ribbon. For example, if we select the "Home" tab, we will see cut, copy, paste, bold, italic, underline,

and more commands. In the same way; we can click on the "Insert" tab, we will see tables, illustrations, additional, recommended graphics, graphics maps, among others. On the other hand, if we select the "Formulas" option. Insert functions, auto sum recently used, finances, logic, text, time, date, etc.

Ribbon/Toolbar is a set of commands organized into three sections.

- Tabs
  They are the Ribbon's top part, and they include groups of related commands. Ribbon tabs include Home, Insert, Page Layout, Formula, Data.
- Groups
  They organize related commands; the name of each group is displayed below the Ribbon. For example, a set of commands related to fonts or a group of commands related to alignment, etc.
- Commands
  They appear within each group, as previously stated.

5. Dialog Box Launcher

Dialog box launcher is a very little down arrow that is present in the lower-right corner of a command group on the Ribbon. By clicking on this arrow, we can explore more options related to the concerned group.

6. Name box

Name Box is located to the left of the formula bar.
The name box displays the active cell address.

7. Formula Bar

The formula bar is generally found below the ribbon menu. It displays the content of the active cell.

The Formula bar allows you to enter or edit data and formulas in the active cell.

8. Scrollbars

The scroll bars are used to shift the window's contents to view different parts of the worksheet.

The vertical scrollbar enables the movement up or down.

The horizontal scrollbar helps move right or left in a window.

There are arrows at both ends of the scroll bars that can be clicked to move the window up, down, left, right.

9. Columns Bar

Columns are a vertically ordered series of boxes across the full sheet. This column bar is located below the formula bar.

The letters of the alphabet are used to label the columns. Begin with the letter **A** to **Z,** and then after **Z**, it will continue as **AA, AB,** and so on. The number of columns that can be used is limited to **16,384.**

10. Rows Bar

The row bar is the left part of the sheet where a sequence of numbers is expressed. Begin with number one (1), and further rows will be added as we move the pointer down. There are a total of 1,048,576 rows available.

11. Cells

The intersection of row and a column is called a cell.

Each cell has its own unique location called a cell address

A cell address is identified by its coordinates(A1)

The cell in which you are working is called the active cell.

12. Status Bar

Status bar displays the current status of the active cell in the worksheet. A cell can be in either of the four states

(a) Ready mode which indicates that the worksheet is ready to accept user input

(b) Edit mode indicates that cell is editing mode, if it is not activated the you can activate editing mode by double-clicking on a cell

(c) A cell enters into Enter mode when a user types data into a cell. (d) Point mode triggers when a formula is being entered using a cell reference by mouse pointing or the arrow keys on the keyboard.

By right-clicking on the status bar, we can configure the status bar. Any command from the specified list can be added or removed.

13. View Buttons

View buttons are a set of three buttons arranged at the left of the Zoom control, close the screen's right-bottom corner. We can see three different kinds of sheet views in Excel using this method.

- Normal View: - Normal view displays the Excel page in normal view.
- Page Layout View: - The Page Layout view shows the precise layout of an Excel page it will be printed.
- Page Break View: - This displays page break preview before printing.

14. Zoom Control

The zoom control is present at the lower-right side of the window. It enables us to ZOOM-IN or ZOOM-OUT a specific area of the spreadsheet. It is represented by magnifying icons with the symbols of maximizing (+) or minimizing (-).

15. Sheet Tabs

Each Excel document has one or more sheets combined together in a workbook. A sheet tab at the bottom of the workbook represents each of these sheets. The movement from one sheet to another within the workbook can be done using the sheet tabs. Excel workbook contain 3 worksheets by default.

Moving Around in a Worksheet:

The Worksheet

Excel allows a user to move or copy a worksheet in the same or to another workbook. You can use the Move or Copy Sheet command to move or copy entire worksheets (also known as sheets), to other locations in the same or a different workbook.
You can use the Cut and Copy commands to move or copy a portion of the data to other worksheets or workbooks.

Move a worksheet within a workbook

1. Select the worksheet tab, and drag it to where you want it.
2. Copy a worksheet in the same workbook
3. Press CTRL and drag the worksheet tab to the tab location you want.

OR

1. Right click on the worksheet tab and select Move or Copy.
2. Select the Create a copy checkbox.
3. Under Before sheet, select where you want to place the copy.
4. Select OK.

Moving Row and Column

Moving a column in excel means shifting the data of one column to another. It is similar to interchanging or swapping the data of two or more columns. When a column or row is moved, Excel copies the formulas, formatting, hidden cells, outputs, and comments to the new location.

Steps to Move a Row in Excel

1. Firstly, select the row that you want to move up or down.
2. Then, move your cursor to the edge of the row and you will see the icon with four-sided arrows.
3. Now, while holding the Shift key and left button of the mouse together drag the selected row to the desired location.
4. When you will find the right point for the row, then release the shift key and mouse click.

Steps to Move a Column

To quickly move a column in Excel, use the below steps.

1. Select the column first that you want to move in the worksheet.
2. Next, hover over the cursor to the border of the selected column and now the four-sided arrow appears there.
3. After that, press and hold the Shift key and the mouse left button at once.
4. Now, drag the cursor to the new position where you would like to relocate the column.

Cells

Excels offers two ways to move the contents of one cell to another: using Drag and Drop and Cut and Paste commands.

To move with drag and drop

1. Select the cells or range of cells that you want to move or copy.
2. Point to the border of the selection.
3. When the pointer becomes a move pointer, drag the cell or range of cells to another location.

To move with Cut and Paste

1. Select a cell or a cell range.
2. Select Home > Cut or press Ctrl + X.
3. Select a cell where you want to move the data.
4. Select Home > Paste or press Ctrl + V.

What is Formula and Function

Introduction to Formula

Excels stores not only numbers and text but also formulas. For example, a formula that adda the contents of cell B1 to the contents of B5 is given as:

=B1+B5

Or a formula that sums the contents of B1 to B5 is:

=B1+B2+B3+B4+B5

Other mathematical operators can also be used in formulas:

=B1*B2+(B3/B4)

Formulas can be either simple as the ones shown above or complex involving a number of operands and operators.

Excel uses standard computer operator symbols for mathematical and logical operators as shown in below:

Operators	Use	Example
+	Addition	=B1+B2
-	Subtraction	=B1-B2
*	Multiplication	=B1*B2
/	Division	=B1/B2
^	Exponential	B1^B2
%	Percentage	= 50%
=	Equal to	
<>	Not Equal to	
>	Greater than	
<	Less Than	

When calculating the results of expressions with a variety of operators, Excel follows the following order of precedence for mathematical operators:

1. Negation
2. Expressions in parentheses
3. Percentage
4. Exponentials
5. Multiplication or Division
6. Addition or Subtraction

When the operators have the same precedence, the expression is calculated from left to right.

A Formula must begin with an equal sign (=); otherwise, Excel treats it bas text. The operands in Excel formulas may be constants or references. When working with

formulas, it is to be remembered that the formula is stored in the cell, but the answer is displayed.

Function

A Function is a predefined formula for making a specific kind of calculation.

A Function always begin with an equal sign, followed immediately by the name of the function, a left parenthesis, a combination of cell references, and mathematical operators; the function ends with a right parenthesis.

Formula can contain any combination of values, references, operators and functions.

Formulas are not case sensitive

When the cell is active, the function itself is displayed in the formula bar.

Following are some basic statistical function

Average Function

Returns the average (arithmetic mean) of the arguments. For example, if the range A1:A20 contains numbers, the formula =AVERAGE(A1:A20) returns the average of those numbers.

Syntax

=AVERAGE(number1, [number2], ...)

SUM

The SUM function adds values. You can add individual values, cell references or ranges or a mix of all three.

The SUM function is used when you need to calculate totals for a range of cells or a group of selected cells on a worksheet.

Syntax

=SUM(number1,[number2],...)

MAX & MIN

The MAX and MIN functions help in finding the maximum number and the minimum number in a range of values.

Syntax

=MIN(number1, [number2], …)

Count Function

We use the count function when we need to count the number of cells containing a number.

The COUNT function is used to determine how many cells in a range contain a numeric entry. The COUNT function will not work for counting text or other non-numeric entries.

Syntax

=COUNT(value1, [value2], …)

Using Functions

In the previous section, the SUM function was used (by clicking the AutoSum button) to total numbers. Excel includes hundreds of functions that can be used to calculate results used in statistics, finance, engineering, math, and other fields. **Functions** are structured programs that calculate a specific result: a total, an average, the amount of a monthly loan payment, or the geometric mean of a group of numbers. Each function has a specific order or **syntax** that must be used for the function to work properly. Functions are formulas, so all functions begin with the = sign. After the = is the **function name**, followed by one or more **arguments** enclosed in parentheses. The function name determines what the function does and the arguments determine what values or cell references the function should use in its calculation. Arguments are enclosed in parentheses and if there is more than one, separated by commas. The argument component of a function can consist of numbers, text, cell references, formulas, another function, error values or logical values. Excel's functions are grouped into 10 categories, as indicated in Table 2.2

Category	Examples
Financial	Calculates interest rates, loan payments, depreciation amounts
Date & Time	Returns the current hour, day of week or year, time, or date
Math & Trig	Calculates absolute values, cosines, logarithms
Statistical	Common functions used for totals, averages, and high and low numbers in a range; advanced functions for t-tests, Chi tests, deviation
Lookup & Reference	Searches for and returns values from a range; creates hyperlinks to network or internet documents
Database	Calculates values in an Excel database table
Text	Converts text to upper or lower case, trims characters from the right or left end of a text string, concatenates text strings
Logical	Evaluates an expression and returns a value of TRUE or FALSE, used to trigger other actions or formatting
Information	Returns information from Excel or Windows about the current status of a cell, object, or the environment
Engineering	Included with Office 97, but must be installed separately from the Analysis Tool pack

Table 2.2 Excel Functions

Entering Formula

- When you type an equal sign into an empty cell, Excel assumes that you are entering a formula because a formula always begins with an equal sign.
- Entering and editing the formulas in excel is a straightforward process.

Entering Formulas

A formula can be entered into any cell by simply typing it in after activating the desired cell. As the formula is typed, it appears in two places: the active cell and the formula bar, which becomes active. The entry should be completed by either pressing the <Enter> key or by clicking the Enter button ✓ on the formula bar.

There is another way in which a formula can be entered. It is called the point-and-click method. Here, the cell is first made active and the equal (=) sign is typed in. Any constant value or operator also should be typed in wherever required. To enter a cell reference, point and click on the cell that is to be referenced. For example, to enter the formula =B1*B4+B2 in the cell B6 (Figs. 2.12), first type in the equal sign, then instead of typing in the cell reference B1, simply click on that cell and its reference is automatically entered into the formula. The * sign and the + sign are to be typed in, while for the cell references B4 and B2, simply click on those cells and there references automatically appear in the formula in B6.

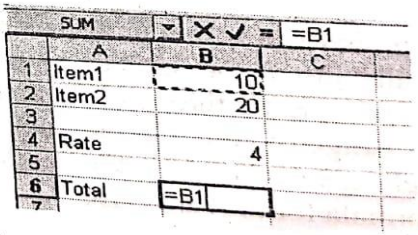


Fig 2.12 Clicking a cell to enter its reference in the formula

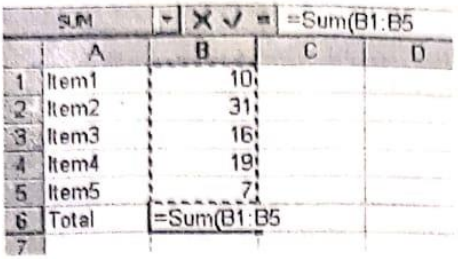


Fig 2.13 Adding the reference of a range of cells to a formula by clicking

A range of cells can be added to a formula by dragging over those cells. From figure 2.13 it can be seen that the range B1:B5 has been added to the formula in B6 by clicking on B1 and dragging over B2, B3, B4 to B5.

If the user makes a syntax error in a formula, Excel points it out. If the error is one of the common errors programmed into Excel's Formula **AutoCorrect** feature, Excel offers to correct the formula. Otherwise, the user is required to troubleshoot the formula and correct it.

Using the Formula Palette

Whatever method of entering formulas is used, the **Formula Palette** can be used to view the progress of the formula as it is being created. To activate the Formula Palette, the **Edit formula button** = on the formula bar should be clicked instead of pressing the equal (=) key on the keyboard. The formula palette (Fig 2.14) shows the results of the formula as the formula is entered.



Fig 2.14 The Formula Palette

Click OK (or press <Enter>) to finish the formula entry, or choose Cancel if the results are not as desired. The Formula Palette is not very helpful with simple formulas, but when complex formulas or functions are used, it can give additional assistance or enable us to see the intermediate results of a formula to help keep track of the process so that it moves in the right direction.

Steps for entering the formulas in excel:

1. Select a cell in which you want to enter the formula.
2. Type "=" sign to indicate that you are entering a formula.
3. Next, type the operands and operators.
4. Press Enter to confirm the formula.

We can also enter the formula using cell reference. The steps are as follows:

1. Select a cell, let C1 where we want to enter the formula.
2. Type “=” sign in order to indicate it is a formula.
3. Then select the cell reference and mention the operation to be executed and press enter.

EDITING FORMULA:

- If we want to makes changes or edit the formula. There are 3 ways of editing the formula which is as follows:

1. Double-clicking: Select the cell which contains the formula to be edited double-click on it, the formula will be displayed then we can edit the formula as per the requirement.
2. F2 key: The next method to edit formula is by selecting the cell which contains the formula to be edited and press F2 key, then the formula will be displayed and we can edit.
3. Formula bar: We can also edit the formula using the formula bar. Click on the formula bar and edit the formula then press Enter.

Use of Function in Formula

AutoSum is the only function included on the Standard toolbar. All the functions (including the SUM function can be accessed using the Formula Palette.

Entering Functions

To enter a Function:

1. Activate the cell where the results are to be displayed.
2. Click the Edit Formula button = in the formula bar to open the Formula Palette.

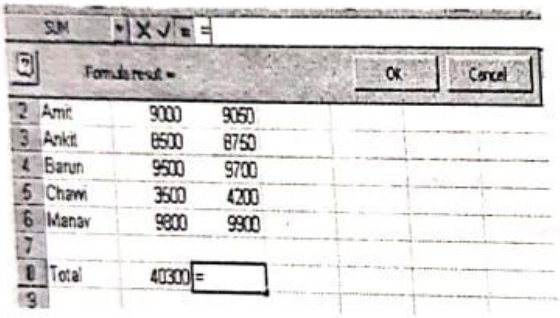


Fig2.48 Formula Palette and Function box

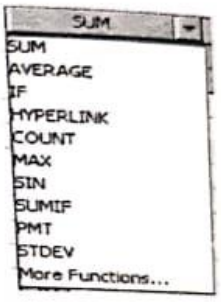


Fig 2.49 Drop-down menu of the function box

The Name Box changes to a Function box, displaying the name of the last function that was used (SUM) as shown in fig 2.48.

3. Click the Function box drop-down to open the list of recently used functions (Fig 2.49).

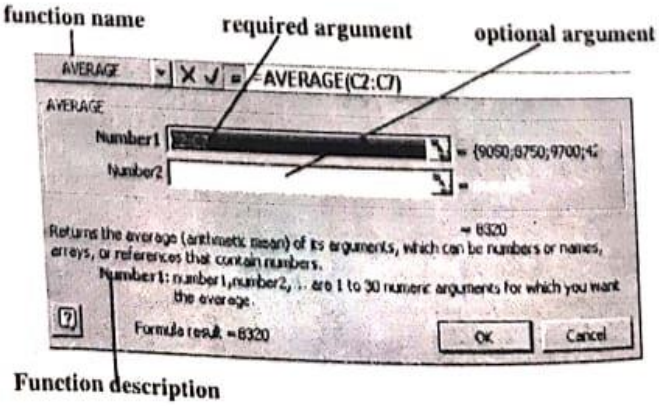


Fig 2.50 Function Palette (AVERAGE function)

4. Select the desired function, if it is in the list, and Excel moves the function to the formula bar and the Formula Palette. The Formula Palette includes a description of the function and one or more text boxes for the function's arguments, as shown in fig. 2.50.

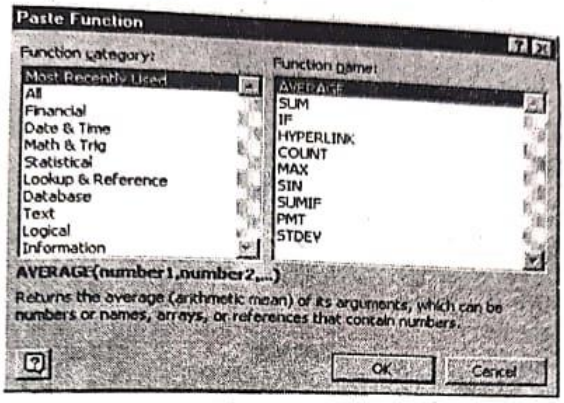


Fig 2.51 The Paste Function dialog box

For common functions that use a single range of cells as argument, Excel "guesses" which numbers to sum or average and places the range in the argument text box. Required arguments are bold, like Number 1 in Fig 2.50. The text boxes must be filled in to successfully use the function.

5. If the Formula Palette covers the cells that are to be selected, click the Collapse Dialog button to shrink the Formula Palette.

6. Confirm that the correct cells are selected or use the mouse button to select the correct cells and then expand the Palette with the Expand Dialog button.

7. Select all the required arguments and click OK to finish the entry and close the Formula Palette. As with any formula, the results of the function are displayed in the active cell; the function itself is displayed in the formula bar when the cell is active.

8. In step 4, if the function desired is not listed in the Function box, choose More Functions at the bottom of the list to open the Paste Function dialog box, shown in fig. 2.51. If a rarely used function needs to be used, the Paste Function dialog box can be opened directly by clicking the Paste Function button on the Standard toolbar.

9. Select a function category from the left pane of the dialog box, then scroll the right pane and select a function. If more information is required, clicking the Office Assistant button opens the Assistant, which offers help on the selected function.

10. Click Ok to choose the selected function and return to the Formula Palette.

5

CHARTS

What is a chart?

Charts are graphical representation of numeric data. Charts make it easier for users to compare and understand numbers and also make data analysis much easy and interesting. In Excel, a chart can be embedded in a worksheet or have a chart sheet of its own.

Charts should be included with worksheets whenever worksheet results need to be emphasized. Charts can often communicate information like trends and comparative results better than numbers alone. Charts are visually appealing and make it easy for users to see comparisons, patterns, and trends in data. For instance, rather than having to analyze several columns of worksheet numbers, a user can see at a glance whether sales are falling or rising over quarterly periods, or how the actual sales compare to the projected sales.

Charts are constructed with **data points** – individual numbers in a worksheet, and **data series** – groups of related data points within a column or row. A data series normally corresponds to the values in a linked range of cells in a single column or row of a single worksheet. When plotted on a chart, each data series is assigned its own color or pattern. Each number within a data series is a data point.

	A	B	C	D	E	F	G
1	Sahara Ticket Sales						
2	First Quarter report						
4	Destination	Jan	Feb	Mar	Total	Average	Percent of Total
5							
6	Kolkata	96	104	100	300	100	23%
7	Mumbai	115	135	125	375	125	28%
8	Chennai	87	91	83	261	87	20%
9	New Delhi	124	140	132	396	132	30%
10							
11	Total for month	422	470	440	1332		
12	Average for month	105.5	117.5	110			
13	Minimum for month	87	98	101			
14	Maximum for month	124	140	145			

Fig 5.1 Sahara Ticket Sales

Scanned with OKEN Scanner

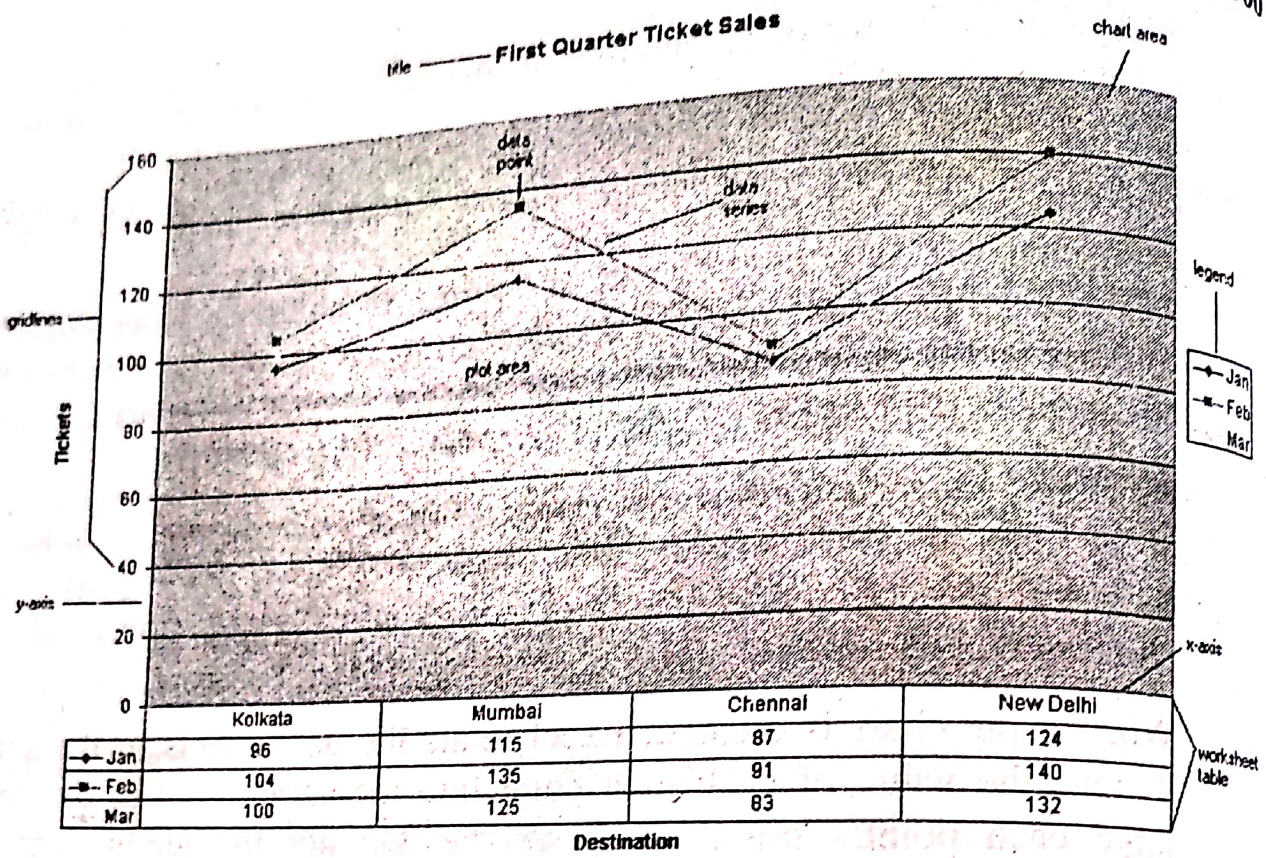


Fig 5.3 Line Chart for Tickets Worksheet

An **area chart** is a line chart with the area below the line filled. Fig 5.4 shows an area chart for the same set of data as the line chart in Fig 5.3.

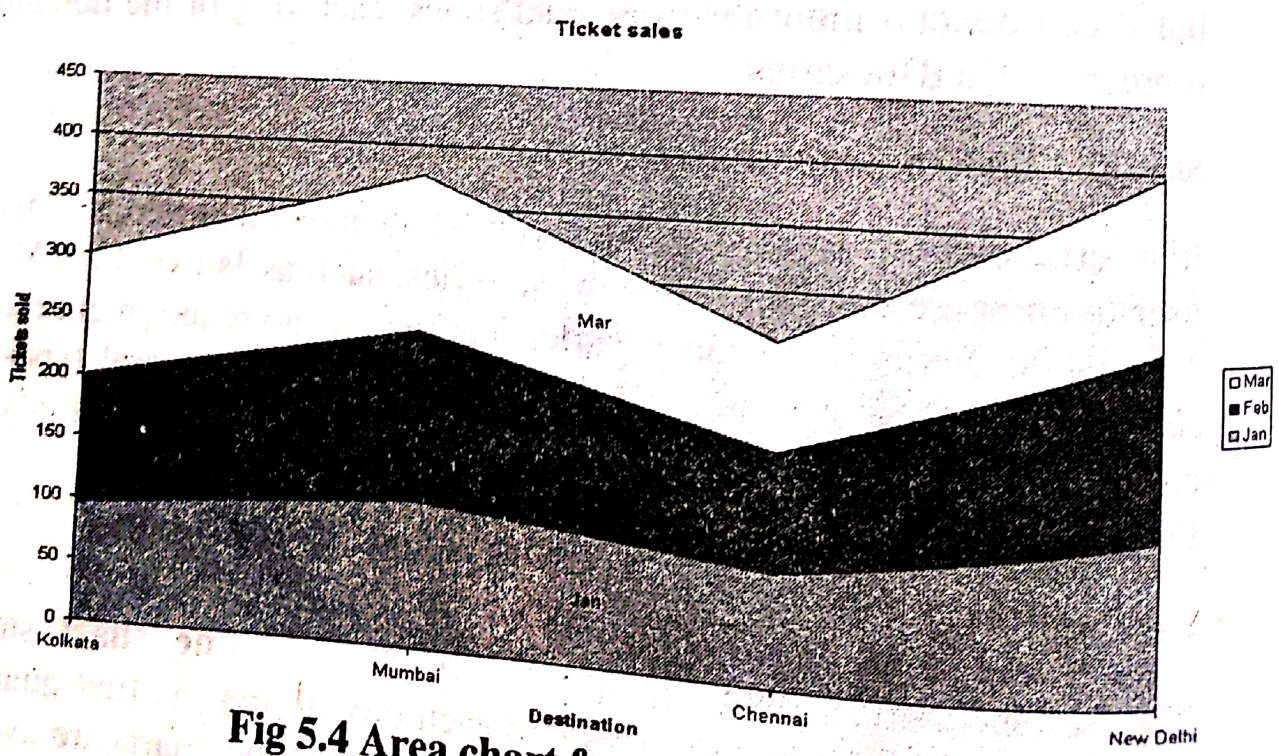


Fig 5.4 Area chart for Tickets Worksheet

Scanned with OKEN Scanner

An area chart emphasizes the magnitude of change over time. By displaying the sum of the plotted values, an area chart also shows the relationship of parts to a whole. Line charts and area charts are typically used to show one or more variables (sales, income, price) changing over time.

Column and Bar Charts

The Tickets Worksheet information is presented as a bar chart in Fig 5.5. A **bar chart** illustrates comparisons among individual items. Categories are organized vertically, values horizontally, to focus on comparing values and to place less emphasis on time. The bars give added substance to the chart. In the line chart (Fig. 5.3), the trends go up or down in each line and the gaps between the lines can be seen. The bar chart makes all ticket sales seem more substantial. It also makes the difference between destinations even more clear – like largest number of tickets is sold for New Delhi.

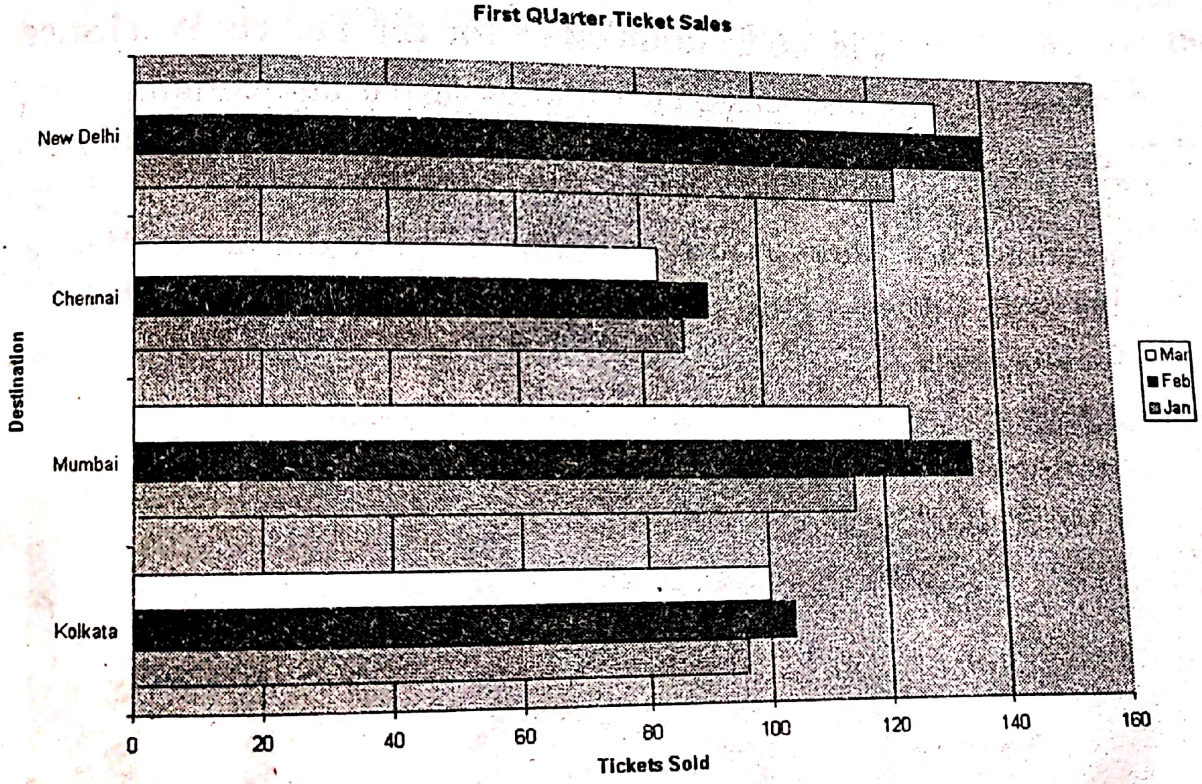


Fig 5.5 Bar Chart for Tickets Worksheet

Line and area charts share a common layout. The horizontal line is called the **X-axis**, and the vertical line is the **Y-axis**. In a bar chart, however, the axes are turned 90 degrees so the X-axis is on the left side.

Column charts are the same as bar charts, but with the X-axis at the bottom as shown in Fig 5.5. A column chart shows data changes over a period of time

Scanned with OKEN Scanner

or illustrates comparisons among items. Categories are organized horizontally, values vertically, to emphasize variation over time.

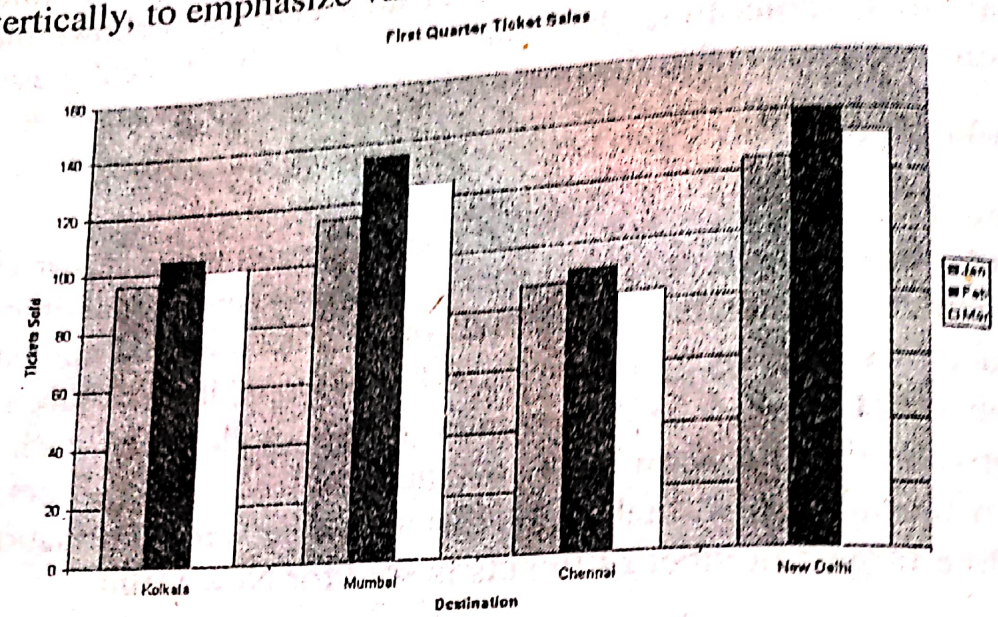


Fig 5.6 Column chart for the Tickets Worksheet

There are three-dimensional varieties of bar and column charts also that add depth to the regular chart. The cone, cylinder, and pyramid data markers can lend a dramatic effect to 3-D column and bar charts. Cylinders, cones, and pyramids are variations of a column chart. Excel also offers another style of bar and column chart - the **stacked chart**. Fig 5.7 shows a stacked 3-D column chart using the Tickets Worksheet.

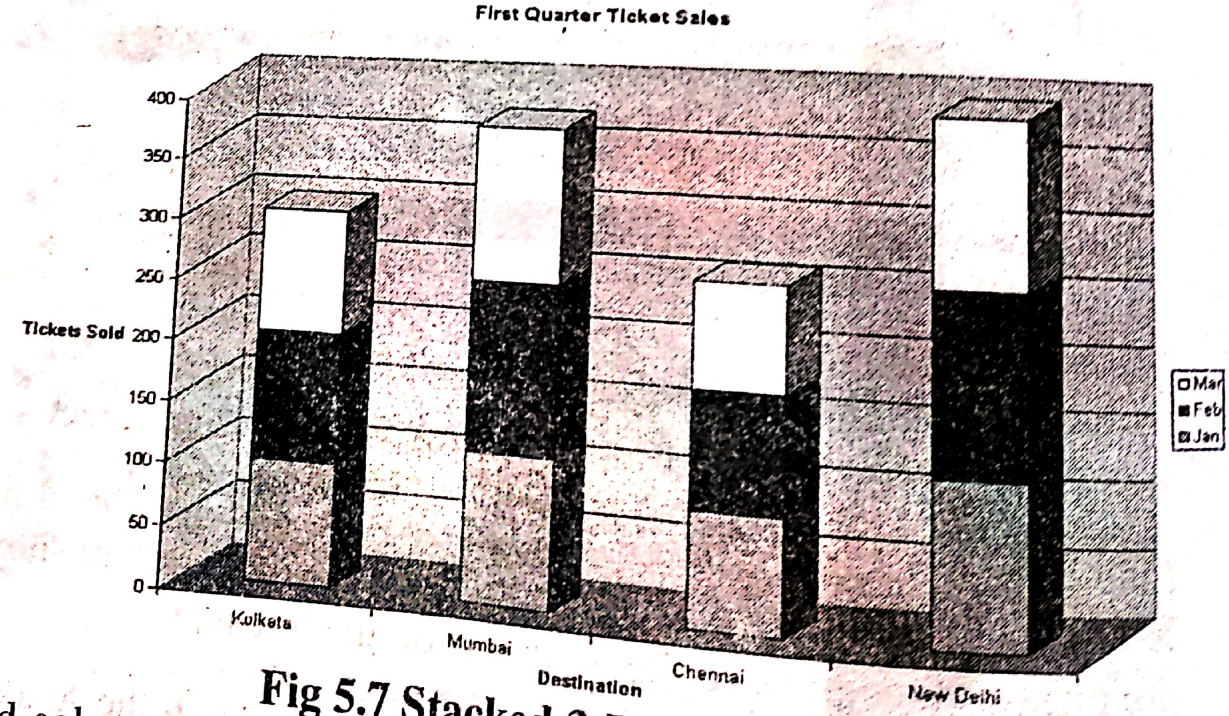


Fig 5.7 Stacked 3-D column chart

Stacked column charts show the relationship of individual items to the whole. The 3-D perspective column chart compares data points along two axes. In a

stacked chart, parallel data points in each data series are stacked on top or to the right of each other. Stacking adds another dimension to the chart, since it allows the user to compare sales between as well as within time periods – like providing a column chart and a pie chart for each time period

The 3-D charts have three axes. In a 3-D column chart, the X-axis is on the bottom. The vertical axis is the **Z-axis**; the Y-axis goes from front to back, providing the 'third-dimensional' depth in the chart.

Some other types of charts are the XY (scatter), Bubble, Stock, Radar and Surface charts which are used for specific types of data.

5.3 Creating a chart

The easiest way to create a chart is by using the Chart Wizard. The Chart Wizard guides the user step-by-step through the creation of a chart. It uses illustrated dialog boxes to prompt for information. The Wizard shows how the chart would look like in each step. At any point it is possible to go back and make changes to selections. The chart appears when the user makes all the required selections. A variety of chart formatting commands and buttons can be used to change the appearance of the chart. To create a chart using the Chart Wizard, the steps are:

1. Select the data to be included in the chart (Fig 5.8).

	A	B	C	D	E	F	G
	Sahara Ticket Sales						
	First Quarter report						
		Jan	Feb	Mar	Total	Average	Percent of Total
	Kolkata	96	104	100	300	100	23%
	Mumbai	115	135	125	375	125	28%
	Chennai	87	91	83	261	87	20%
	New Delhi	124	140	132	396	132	30%
	Total for month	422	470	440	1332		

Fig 5.8 Selection of data

2. Choose Insert → Chart from the menu bar or click the Chart Wizard button on the Standard toolbar.
3. In the Chart Wizard – Step 1 of 4 - Chart Type dialog box (Fig 5.9) that opens, click to select one of the chart types in the Chart type scrolling list.

Scanned with OKEN Scanner

Then click to select one of the chart sub-types on the right side of the dialog box. Click Next.

If the desired type of chart is not listed, check out the chart types on the Custom Types tab. For more information about a chart type, select the chart type, then click the Chart Wizard's Office Assistant button and choose Help With This feature. The Assistant will offer to provide a sample of the selected chart.

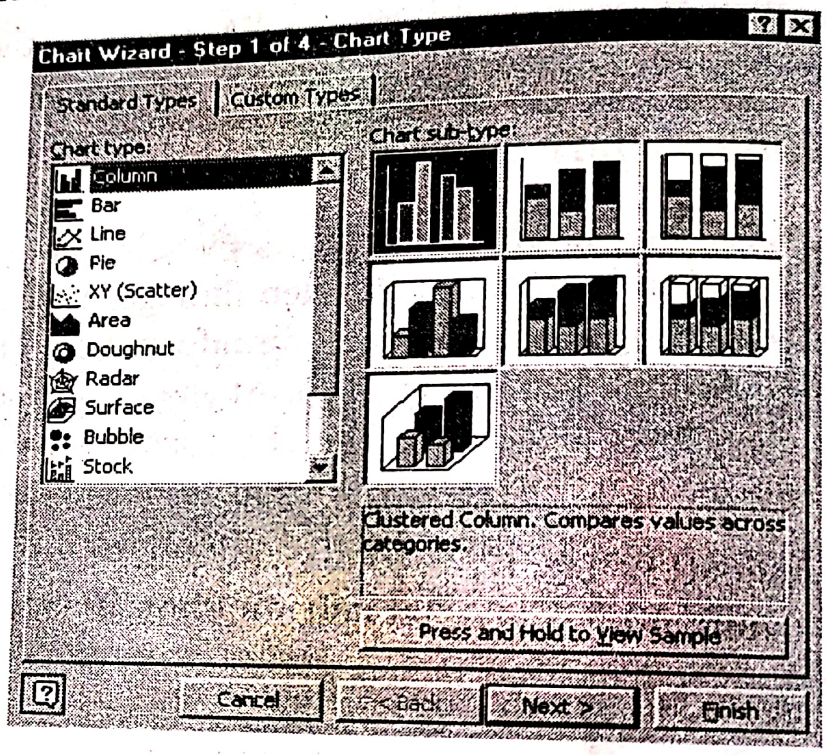


Fig 5.9 Chart Wizard – Step 1 – Chart Type dialog box

To see a rough sample of the type and subtype of the selected chart using the current data, use the mouse pointer to press the Press and Hold to View Sample button in the Chart Wizard.

4. In the second step, in the Chart Source Data dialog box (Fig 5.10), check the contents of the Data range box to ensure that it correctly indicates the data to be charted. If it does not, use the Collapse dialog button and reselect the proper range before continuing. Click the Expand dialog button to display the entire dialog box again. Choose Rows or Columns in the Series in option group.

Scanned with OKEN Scanner

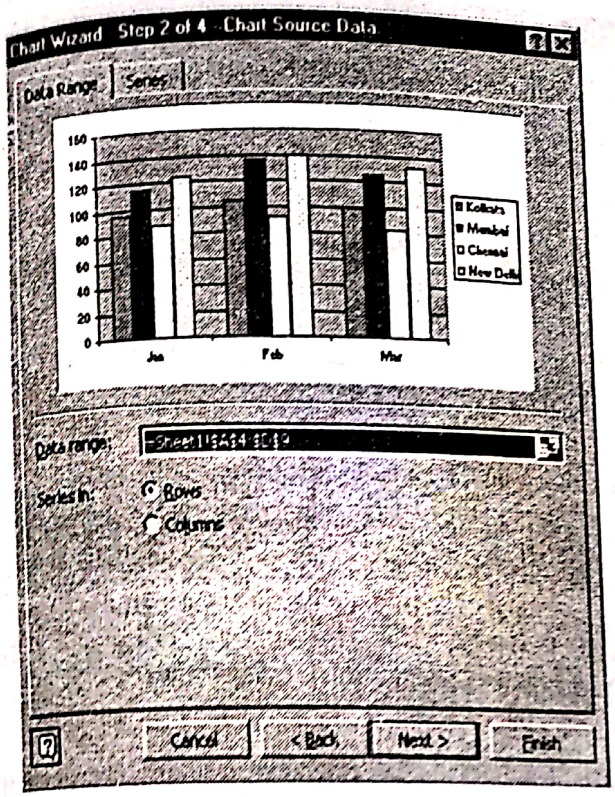


Fig 5.10 Data Range page of the Chart Wizard - Step 2 of 4 - Chart Source Data dialog box

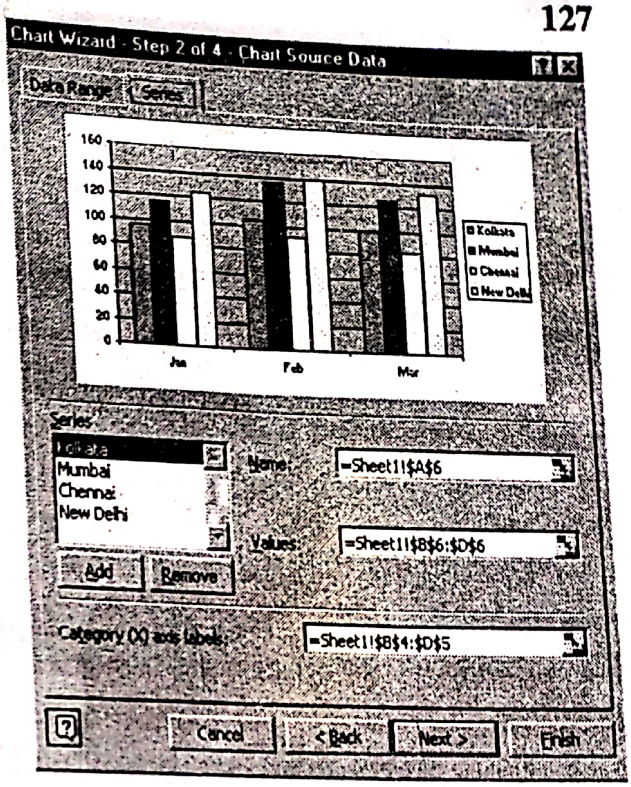


Fig 5.11 Series page of the Chart Wizard – Step 2 of 4 – Chart Source Data dialog box

The preview displayed in the top portion of the dialog box changes to reflect the specified range and series arrangement. A different series option can also be selected to change the way data is charted. Click Next.

5. In the third step, in the Chart Wizard – Step 3 of 4 – Chart Options (Fig 5.12), use the tabs to change options for various aspects of the chart:

   - Titles page enables a user to enter titles for the chart and axes.
   - Axes page enables a user to display or hide axes.
   - Gridlines page enables a user to display gridlines and display or hide the third dimension of a 3-D chart.
   - Legend enables a user to display and place a legend.
   - Data Labels enables a user to display text or values as data labels.
   - Data Table enables a user to show the selected range from the worksheet as part of the chart.

Scanned with OKEN Scanner

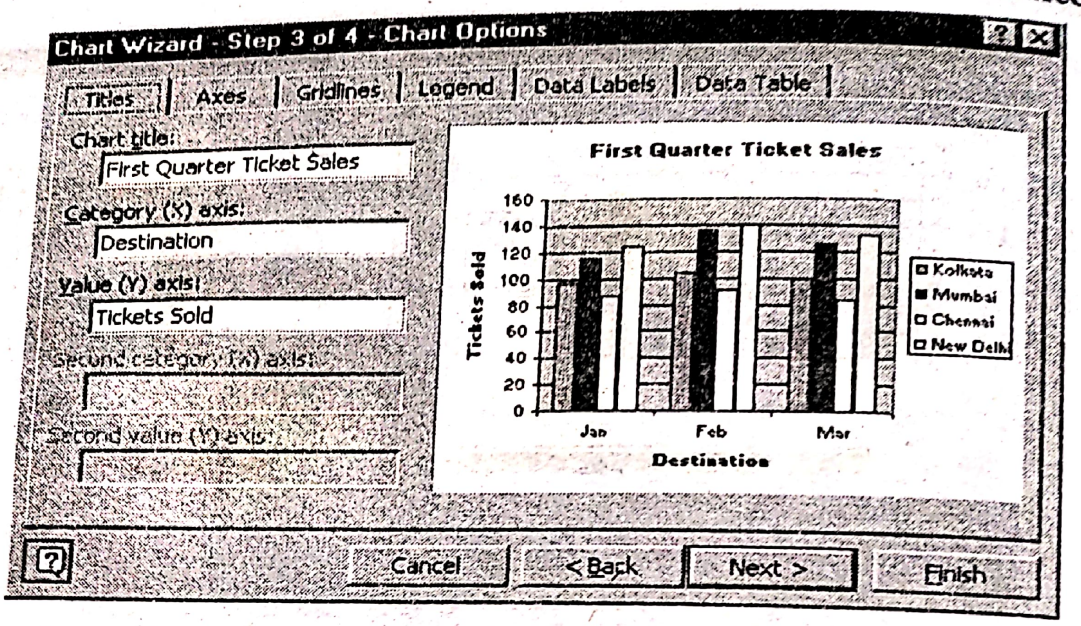


Fig 5.12 Titles tab of the Chart Wizard – Step 3 of 4 – Chart Options dialog box

Every chart needs a title. The title provides information that is not already included in the graphical portion of the chart. The chart's picture, legend, and title taken together should answer any questions about the timing, location, or contents of the chart. The Axes and Gridlines options are only available for charts that have axes or gridlines. Pie charts, for example, have neither axes nor gridlines.

As the options are changed, the chart preview reflects the changes. After setting the appropriate options, click Next to continue.

6. In the fourth step, in the Chart Wizard – Chart Location dialog box (Fig 5.13), a choice of location of the chart is made – the chart can either be placed on the current worksheet (default setting), or on a new, blank sheet in the same workbook. If the chart is placed on its own sheet, it will print full-size, single-page chart whenever it is printed. If it is added to the current worksheet as an object, it will print as part of the worksheet, but it can also be printed separately. Wherever the chart is chosen to be located, it can be easily moved later.

Scanned with OKEN Scanner

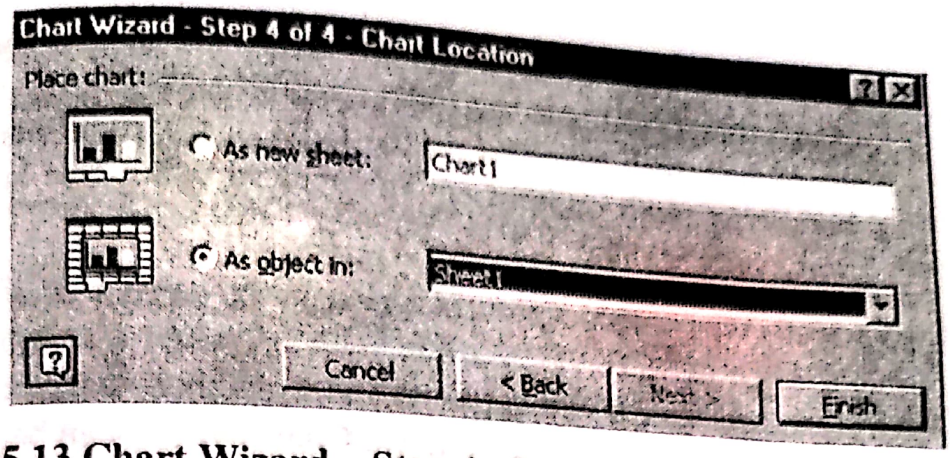


Fig 5.13 Chart Wizard – Step 4 of 4 – Chart Location dialog box

Enter a new sheet name in the Place chart As new sheet text box, or leave the default setting for Place chart As object in to have the chart placed as an object in the current worksheet. Click the Finish button to create and place the chart.

Excel creates and inserts the chart with the settings and in the location that was specified.

A user can always move a chart object to the current worksheet, or a chart can be made an object in another worksheet. To do this, select the chart or chart object, right click, and choose Location from the shortcut menu to open the Chart Location dialog box.

4 Moving, Sizing and Printing Chart Objects

If the chart is placed as an object in the current worksheet, it may have to be moved or resized – especially if the chart is too small or if it covers part of the worksheet data. A chart, if required, can be moved easily. When the chart placed in a worksheet is selected, square handles appear on the corners and sides. To move and size a Chart:

1. If the chart is not selected, click once on the chart to select it. (To deselect the chart and return to the worksheet, click once on part of the worksheet that is not covered by the chart.)
2. To move the selected chart, point to the chart. The pointer changes to a four-headed arrow. Hold the mouse button down and drag the outline of the chart to its new location. Release the mouse button to drop the outline at the new location to have the chart move to that location.

Scanned with OKEN Scanner

3. To change the size of the chart, move the mouse pointer to one of the chart's handles. Press the mouse button and drag the handle to stretch or shrink the chart. Handles on the sides of the chart change the size in one direction (width or height). To increase width and height in proportion, use a corner handle.

Even if the chart is placed as an object in the current worksheet, it can still be printed separately. If the chart is selected when the print order is given, it prints by itself on a full page. If the worksheet is selected, the worksheet prints, including the chart object. So, Chart Objects can be printed as follows:

1. To print a worksheet, including a chart object, activate any worksheet cell before printing.
2. To print a chart object as a full-page chart, select the chart before printing.

The chart is printed accordingly when the print order is given.

5.5 Modifying and Formatting Charts

A chart is a collection of graphic objects or chart elements each of which can be selected, then modified or formatted so that the chart has the desired form and appearance. To access the objects, first select the chart and then click on the object to be selected. Selection handles or selection box (or both) appear around the selected object (data point, data series, title, etc.). When an object is selected, it can be formatted or modified or even deleted. To identify a chart element, position the mouse pointer on it. Excel displays the element's name (and values, if there are) for the element in a yellow Chart Tip box (Fig 5.14).

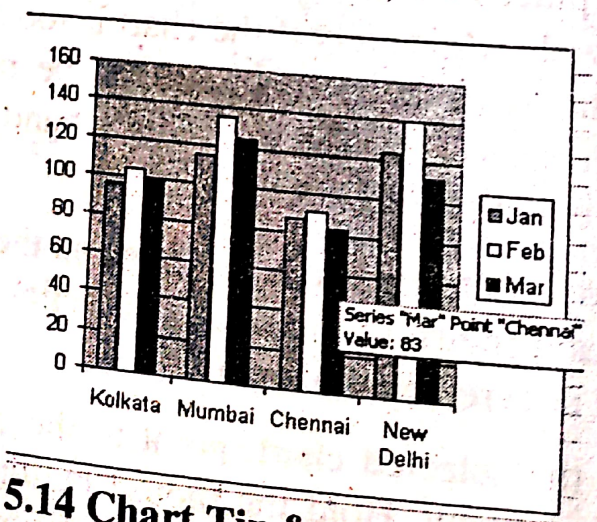


Fig 5.14 Chart Tip for a chart element

Scanned with OKEN Scanner

Once a chart has been created, it is very easy to make simple modifications in it. Excel also provides for formatting individual chart objects using the Chart toolbar or the Format dialog box for each item.

CHANGING THE CHART TYPE

The Chart Type dialog box that is used in the first step of the Chart Wizard can be opened directly also. Click once anywhere on the chart, the Chart menu appears on the menu bar. Choose Chart → Chart Type from the menu bar to open the Chart Type dialog box, which lists the available standard types and subtypes of charts. Another way to open the Chart Type dialog box is to right-click anywhere in the chart and choose Chart Type from the shortcut menu that appears. The basic chart type of the current chart can be changed by selecting any other type from the list. A custom chart type can also be chosen if required.

ADDING, MODIFYING OR DELETING DATA SERIES

Excel's charting tools allow a quick and easy modification of charts. When a chart is created based on worksheet data, the worksheet and chart are linked. The worksheet is the data source and the specified entries in its cells are represented as points plotted on the chart. If the content of any of those cells changes, the chart too changes accordingly. The links work both ways. With some chart types, a data point in the chart can be dragged to change the data in the source worksheet. This makes a good planning tool for businesses interested in maintaining trends.

It is possible to create a simple series chart and then add another data series to it. The data series included in a chart can also be changed at any time. To ad or change the data series included in a chart, four different methods can be used:

- Range Finder Handles can be used to modify a series in an embedded chart.
- The Source Data dialog box can be used to add, modify, or remove a series.
- A new series can be pasted in using the Copy & Paste commands.
- Drag & Drop editing can be used to drag in a series.

Scanned with OKEN Scanner

Modifying a series in an embedded chart using Range Finder Handles

When a chart is selected, Excel's Range Finder feature places a color-coded box around the ranges that were used to provide the source data. Fig 5.15 shows how Range Finder frame appears around the entire data source range when the entire chart is selected.

To see or edit he links between a chart and a worksheet, activate the chart and select one of the data series in it. Range Finder frames are placed around the range of cells in the worksheet that list that data series, and the formula linking the chart with the data range is displayed in the formula bar as can be seen in Fig 5.16. This formula is a SERIES function that specifies the sheet name and absolute cell references for the range making up that series.

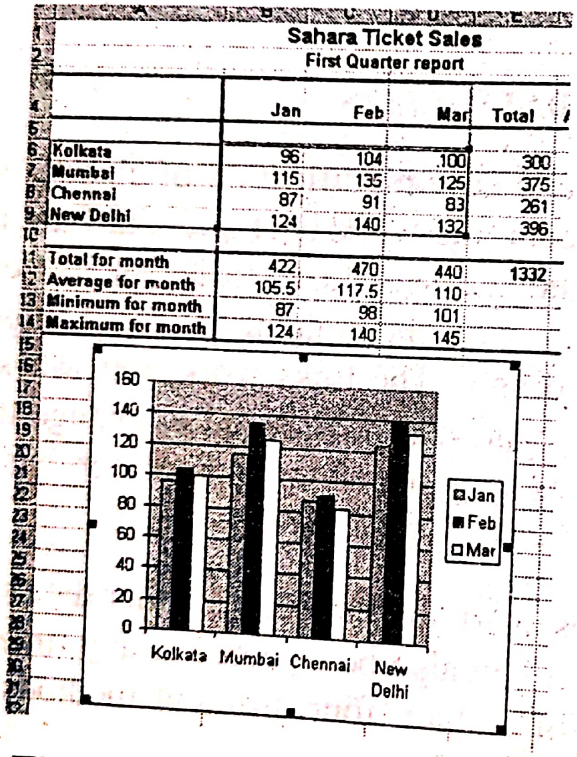


Fig 5.15 Range Finder frames around the entire data source range in the worksheet

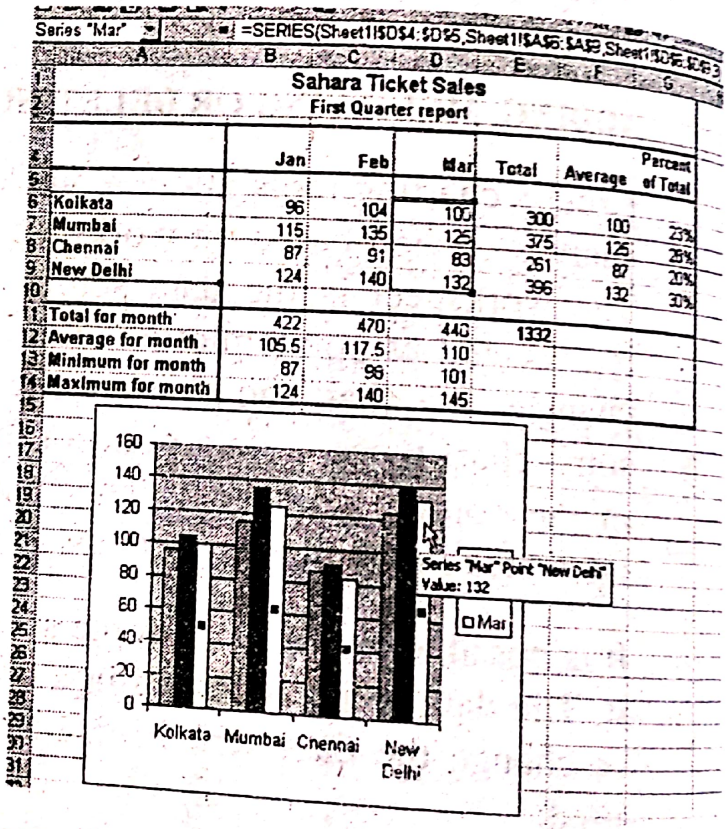


Fig 5.16 Range Finder frames around a data series

To modify the data series in a chart, using Range Finder handles:

1. Click and activate the chart. Range Finder frames with handles appear around each data series (Fig 5.15).
2. Position the mouse pointer on the handle for the series to be changed. The mouse pointer changes to a thick cross-hairs pointer.

Scanned with OKEN Scanner

3. Press the mouse pointer down and drag the handle to stretch or shrink the Range Finder frame to include or exclude one or more data series.
4. When the mouse pointer is released, the data source range changes and so does the information plotted in the chart.

This method can only be used with embedded charts.

To modify the data series using the Source Data dialog box:

1. Activate the chart by clicking the chart sheet or the chart itself if it is an embedded one.
2. Choose Chart → Source Data to open the Source Data dialog box.
3. Click the Data Range page tab (Fig 5.17), which indicates the data range used to plot the chart and whether the Series is in Rows or Columns. The user can change the range of data plotted in the chart. A new range can also be selected. The Series in option can also be changed if desired.
4. The series used and the ranges where the Name and Values are listed are displayed in the Series page of the dialog box (Fig 5.18). A data series can be added by clicking the Add button and the ranges for its Name and Values can be selected from the worksheet.

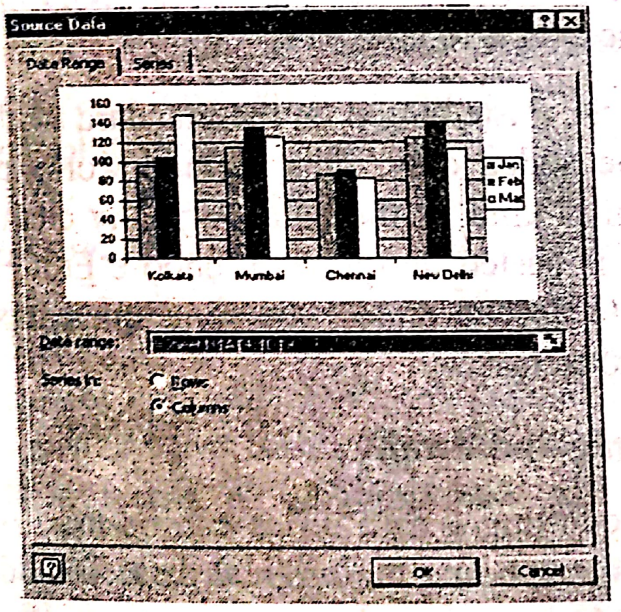


Fig 5.17 Data Range tab of the Source Data dialog box

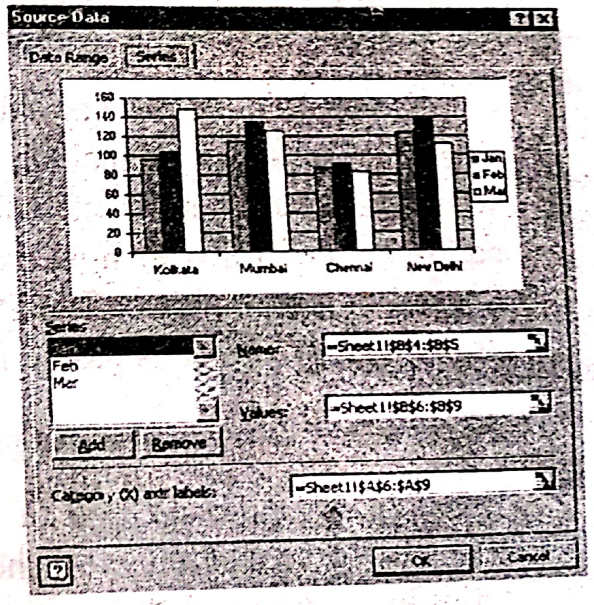


Fig 5.18 Series tab of the Source Data dialog box

To modify a data series, select the name of the series to be modified in the Series list and reselect the ranges entered in the Name and Values boxes to have a different data series. To remove a data series, select the name of the

Scanned with OKEN Scanner

series to be removed from the Series list and simply click the Remove button. Changes made in the data series used in the chart do not affect the actual data in the source worksheet in anyway; they only modify the data range or series that the chart uses.

The sample chart in the dialog box reflects whatever changes are made in the data series.

5. When the data ranges in the chart are as desired, click OK.

The Source Data dialog box looks and works just like the second step of the Chart Wizard dialog box. The Chart Wizard can also be used to make the changes in the data series as described above.

To add a data series using the Copy and Paste commands:

1. In the worksheet, select the data to be added to the chart. The column and row headings should be included in the selection if they are to be included as labels.

2. Choose Edit → Copy, or click the Copy button on the Standard toolbar. A marquee appears around the selected cells.

3. Activate the chart to which the data is to be added.

4. Choose Edit → Paste or click the Paste button on the Standard toolbar. The chart changes to include the added data. This method works properly when the added data is of a similar kind as the data originally charted. For additional control over how data is pasted into a chart, choose Edit → Paste Special in step 4 above. The Paste Special dialog box sometimes appears on its own when a range is pasted into a chart.

To modify a data series using drag-and-drop:

The drag and drop method of modifying a data series can be used only for charts embedded in the worksheet containing the original data. To add a data series using drag-and-drop:

1. In the worksheet, select the data to be added to the chart. The column and row headings should be included in the selection if they are to be included as labels.

Scanned with OKEN Scanner

2. Position the mouse pointer on the border of the selection. The mouse pointer turns into an arrow.
3. Press the mouse button down and drag the selection on to the chart. The mouse pointer gets a little plus sign next to it and the chart border changes.
4. Release the mouse button.

The chart changes to include the additional data. This method works properly when the added data is of a similar kind as the data originally charted.



To remove a data series from a chart:

1. Click to select the data series to be removed. Selection handles appear at each data point in the series.
2. Choose Edit → Clear → Series or press <Delete>.

The series disappears from the chart. If the chart included a legend, it is revised to exclude the deleted data.

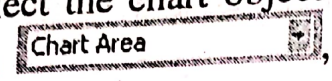
FORMATTING CHARTS

The **chart area** (see Fig 5.3) is a rectangular area within the chart window bounded by the chart border. Changing the size of the chart window changes the size of the chart area. All objects in a chart must be within the chart area. The **plot area** is bounded by the axes and contains the columns, lines, wedges, or other objects used to represent the data points. Objects within the plot area have fixed locations and cannot be moved or individually sized. For example, the X-axis labels must be located near the X-axis. However, it is possible to resize all the objects in the plot area by increasing or decreasing the plot area itself. Objects outside the plot area and axes can be sized or moved to other locations in the chart area. The title and legend can be placed above, below, or in the plot area.

Any element or object in the chart can be selected, then formatted or deleted, with the exception of individual data points. Data points can be formatted, but only data series can be added or deleted. To select a data point, first select the data series, and then click once on the data point.
Formatting of chart elements can be done either using the chart toolbar or the Format dialog box specific to that element.

Scanned with OKEN Scanner

Using the Chart Toolbar

Choose View → Toolbars → Chart to open the Chart Toolbar (Fig 5.19). Common formatting options are available on the Chart Toolbar, as indicated in Table 5.1. Select the chart object to be formatted from the Chart objects drop-down menu 

, then use the toolbar buttons to format the object or the entire chart.

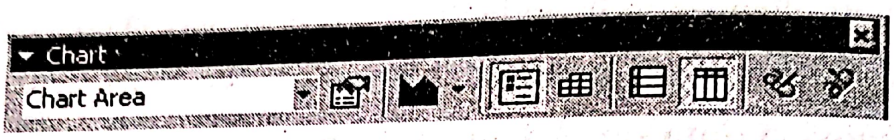


Fig 5.19 Chart Toolbar

Button	Button Name	Function
	Format Object	Opens the Format dialog box for the selected object
	Chart Type	The drop-down opens a menu of chart types; clicking the button applies the type indicated on the button face.
	Legend	Displays or hides the legend
	Data Table	Displays or hides the data table
	By Rows	Uses the selected worksheet rows as a data series
	By Columns	Uses the selected worksheet columns as a data series
	Angle Text Downward	Angles selected text downward
	Angle Text Upward	Angles selected text upward

Table 5.1 Chart Toolbar buttons

Using the Format dialog box:

Double-click on any object to open the formatting dialog box for the object. For example, double clicking any column in a data series opens the Format Data Series dialog box, shown in Fig 5.20.

Scanned with OKEN Scanner

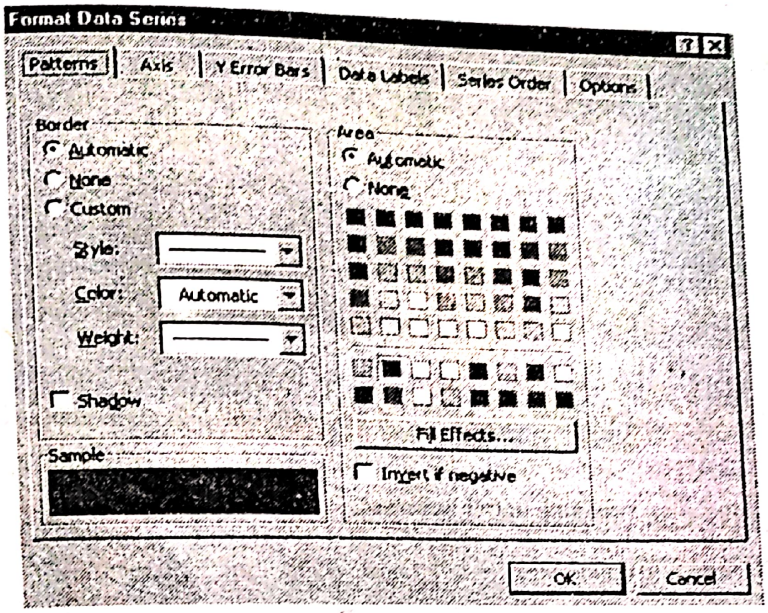


Fig 5.20 The Format Data Series dialog box

There are five or six pages in this dialog box. Each contains a group of settings for the selected data series. The dialog box provides options for setting:

- **Patterns:** used to set the color and pattern used for the series.
- **Axis:** if the chart has more than one series, this option allows the user to add a second vertical axis at the right end of the plot area scaled to this data series
- **Y Error Bars:** adds a graphic display of the standard error; used to approximate sampling error when the data in a chart is a statistical sample being applied to a larger population.
- **Data Labels:** adds a descriptive label or the numeric value for each data point in the series
- **Series Order:** allows a user to reorder the series in a chart; this is especially helpful with 3-D charts, where the selected range is charted in reverse order
- **Options:** settings for the bar or column overlap, gap, and color variation.

The dialog box always includes options suitable to the chart type of the selected chart. Depending on the chart type, the dialog box may include a tab for Shape and not include Axis or Y Error Bars tabs.

Scanned with OKEN Scanner

For more information on a specific control within the Format Data Series dialog box, right-click the control and click on the What's This? button that pops up.

The Format dialog box for other chart objects can also appear when that chart object - a data point, the plot area, chart area, etc.- is double-clicked. A user can also select any object or series and right click and select the Format command for that object from the shortcut menu. The Format dialog box that appears can be used to apply a variety of formatting to chart elements. Some of the formatting options available such as patterns, axes, data labels have already been described. There are some more formatting options that are available with specific objects:

- **Font**: enables the user to change the appearance of an object's font characters.
- **Number**: enables the user to change number formatting of values.
- **Alignment**: enables the user to change the alignment and orientation if text.
- **Scale**: used to change the values that appear on an axis.
- **Shape**: used to change the shape of data points.
- **Placement**: used to set the position of the chart legend.

To Format Chart Objects:

1. Select the chart.
2. Double-click on the object to be formatted or select the object, right click, and choose Format from the shortcut menu to open the appropriate dialog box.
3. Change formatting options as desired.
4. Click OK to apply the changes and close the dialog box.

The selected object is formatted as specified.

CHART OPTIONS

Activate the chart and choose Chart → Chart Options from the menu bar to open the Chart Options dialog box used in step 3 when creating a chart using the Chart Wizard. Alternatively, right-click anywhere on the chart (except on

Scanned with OKEN Scanner

specific elements) or select the entire chart and right-click and choose Chart Options from the shortcut menu that appears to open the Chart Options dialog box (Fig 5.21).

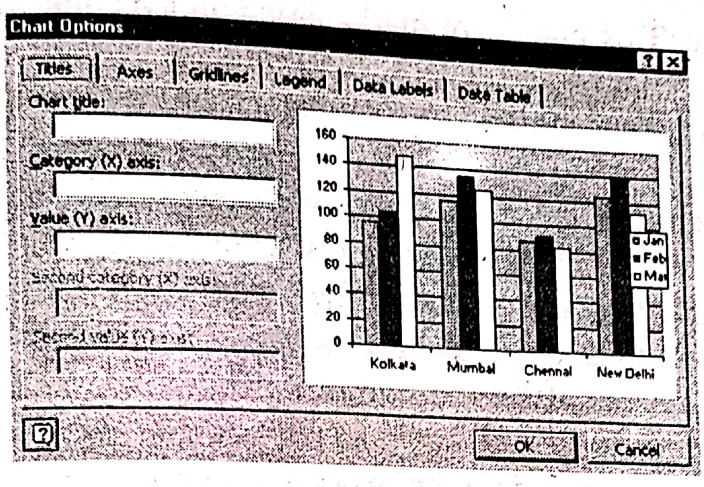


Fig 5.21 Chart Options dialog box

Setting of chart options refers to the inclusion and basic formatting of chart objects such as, titles, axes, gridlines, legend, data labels, and data tables. The Chart Options dialog box contains tabs for the following six pages:

- **Titles:** Titles are textual labels that appear in specific locations on the chart. This page has text boxes for entering the Chart Title, Category (X) axis, Series (Y) axis and Value (Z) axis, etc. The text boxes that appear depend on the type of chart selected.

- **Axes:** These are the bounding lines of a chart. 2-D charts have two axes: X and Y. 3-D charts have three axes: X, Y and Z. Pie and Doughnut Charts have no axis at all. The axes page of the Chart Options dialog box has check boxes for the axes to be displayed – Category (X) axis, Series (Y) axis and Value (Z) axis. Value (Z) axis appears only for 3-D charts. If the Category (X) axis option is turned on, three formatting options are enabled – Automatic, Category and Time-Scale. Automatic instructs Excel to check the formatting of the category data to determine whether it should use time-scale or category formatting. Category instructs Excel to use the category data, while Time-Scale instructs Excel to create a time scale for the X-axis.

- **Gridlines:** These are lines indicating major and minor scale points along a chart's walls. They can make it easier to follow chart points to corresponding values on a chart axis, though too many of them can also

Scanned with OKEN Scanner

clutter the chart walls making the data difficult to read. Pie and Doughnut charts do not have gridlines. The Major and Minor Gridlines options for each axis correspond to major and minor tick marks units for the axis scale. Tick marks are marks along an axis that correspond to scale values. For certain Chart Types, the option 2-D walls and gridlines is available which can be turned on to apply a 2-D appearance to a 3-D chart.

- **Legend:** A legend is a box with color-coded labels identifying a chart's data series. The user can show or hide a legend and set its position within the chart area: Bottom, Corner, Top, Right, or Left. A legend can also be moved to a new position within the chart by dragging. The legend's position can affect the size of a chart's plot area. Excel creates the legend based on cells selected as part of the data source. So a user should preferably include column and/or row headings in the range selected for creating the chart.

- **Data Labels:** Data Labels provide additional information about specific data points. The Chart Options dialog box enables a user to set data labels for all chart points. One of the available Data Label options can be selected from among – None, Show value, Show Percent, Show Label, Show Label and percent, and Show bubble sizes. The options available in the dialog box vary based on the type of chart that is selected. To show the legend color key beside a data label, turn on the Legend key next to label check box. For a pie or doughnut chart, to display a line from the data point to the data label, turn on the Show leader lines check box.

- **Data Table:** A data table is the data plotted on the chart, in tabular format. Data Tables are more useful on chart sheets than on embedded charts. They are not available for pie and doughnut charts. The Show data table box can be used to show or hide the data table. To show the legend color key in the data table, turn on the Show legend keys check box.

To use the Chart Options dialog box:

1. Activate the chart.
2. Choose Chart → Chart Options from the menu bar or right-click and select Chart Options from the shortcut menu to display the Chart Options dialog box.

Scanned with OKEN Scanner

3. Click the tab for the type of option to be set.
4. Set the option as desired.
5. Repeat steps 3 and 4 for each type of option to be set.
6. Click OK to apply the specified settings.

OTHER FORMATTING OPTIONS

Exploding Pies

In order to emphasize specific data points in a pie chart, it is possible to explode the pie chart by moving one or two pieces of the pie farther from its center (Fig 5.22).

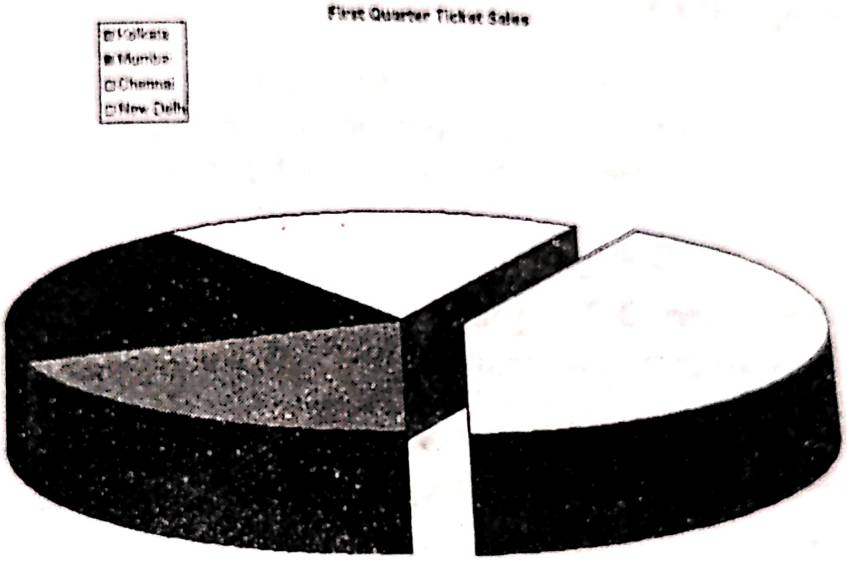


Fig 5.22 Exploded Pie Chart

To 'explode' a pie chart:

1. Select the chart
2. Click on the pie to select the data series.
3. Click to select the slice to be exploded
4. Drag the slice away from the center of the pie.
5. Release the mouse button. The piece moves away from the pie.

Scanned with OKEN Scanner

Unit 5.

Introduction to PowerPoint

5.1 PowerPoint Basics

What is PowerPoint?

- PowerPoint is a software program developed by Microsoft to produce effective presentations. It is a part of Microsoft Office suite.
- PowerPoint is a powerful, easy-to-use presentation graphics software program that allows you to create professional-looking electronic slide shows.
- PowerPoint can be used to create professional presentations for a wide variety of audiences and for a wide variety of purposes.
- PowerPoint offers the tools to create a basic presentation that includes text, graphics, charts, and even digital movies, etc.
- The presentation can be displayed as an electronic slide show on any computer.
- By default, documents saved in PowerPoint 2010 are saved with the .pptx extension whereas, the file extension of the prior PowerPoint versions is .ppt.
- "A combination of various slides depicting a graphical and visual interpretation of data, to present information in a more creative and interactive manner is called a PowerPoint presentation or PPT."

Starting PowerPoint

The Steps for starting PowerPoint are:

1. Click on the start button
2. Then choose "All Programs"
3. Next step is to select "MS Office"
4. Under MS Office, click on the "MS PowerPoint"

Understanding the PowerPoint Screen

The PowerPoint program Screen is very similar to that of the other Microsoft Office applications.

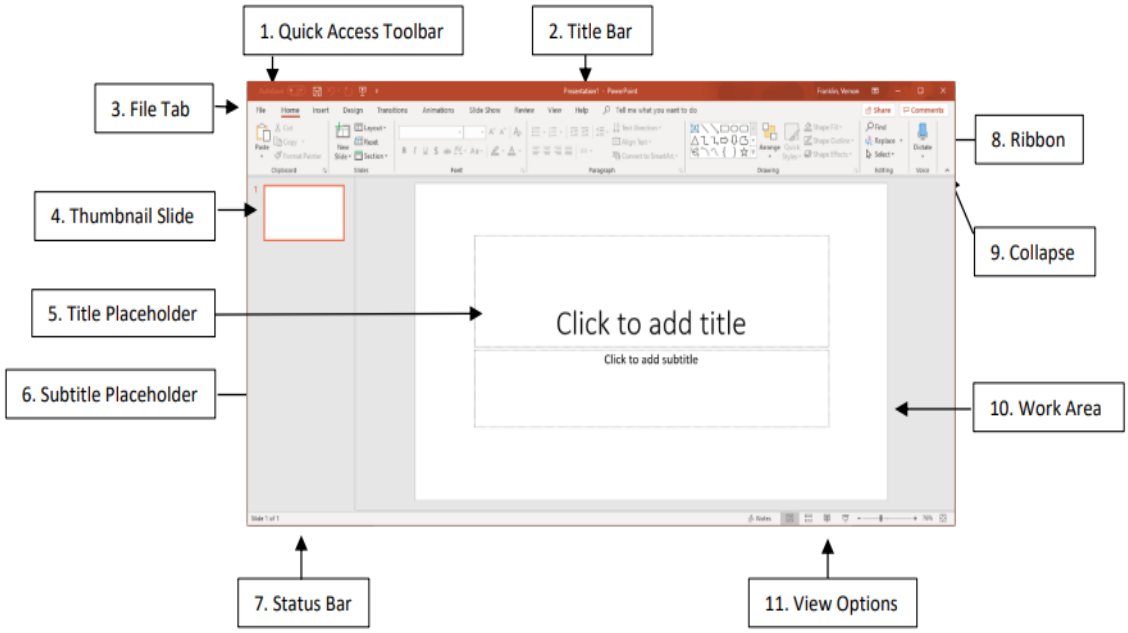


Fig. The PowerPoint Program Screen (Normal View)

	Term	Description
1	**Quick Access Toolbar**	Displays quick access to commonly used commands.
2	**Title Bar**	Displays the name of the open file.
3	**File Tab**	The File tab has replaced the Office 2007 button. It helps you to manage the Microsoft application and provides access to options such as Open, New, Save As, Print, etc.
4	**Thumbnail Slide**	Displays a snapshot of each slide.
5	**Title Placeholder**	Section where text is entered.
6	**Subtitle Placeholder**	Section where text and/or graphics are entered.
7	**Status Bar**	Displays information about the slide presentation, such as page numbers.
8	**Ribbon**	Displays groups of related commands within tabs. Each tab provides buttons for commands.
9	**Collapse**	Collapses the ribbon so only the tab names show.
10	**Work Area**	Each slide has an area where text and graphics are entered for a presentation. There are various slide layouts to work from.
11	**View Option**	Displays several View modes for slides.

5.2 Creating a new Presentation

Every PowerPoint presentation consist of a series of slides: text or objects displayed on a graphic background. A Presentation is created by adding text and objects to slides. It is required to go through the following series of steps for every presentation that is created in PowerPoint:

1. Create the presentation, entering and editing text, and rearranging slides.
2. Apply a presentation design. Modify the design if necessary.
3. Format individual slides if desired.
4. Add objects to the presentation.
5. Apply and modify transitions, animation effects, and links for electronic presentations.
6. Create audience materials and speakers notes.
7. Rehearse and add slide timings.
8. Present the presentation.

In PowerPoint, a presentation can be created in several different ways. The user can choose to create a presentation using a Blank Presentation or Using a Design Template.

Creating a Black presentation:

To create a blank presentation:

1. Select the File tab to go to Backstage view.
2. Select New on the left side of the window, then click Blank Presentation.
3. A new presentation will appear.

A blank presentation with one slide will be created. Also, further slides can be added to the presentation and can change the layout of an existing slide as well.

Add a slide

1. In the thumbnails on the left pane, select the slide you want your new slide to follow.
2. In the Home tab, in the Slides section, select New Slide.
3. In the Slides section, select Layout, and then select the layout you want from the menu.

Using a Design Template

A template is a predesigned presentation you can use to create a new slide show quickly. Templates often include custom formatting and designs, so they can save you a lot of time and effort when starting a new project.

To create a new presentation from a template:

1. Click the File tab to access Backstage view.

2. Select New. You can click a suggested search to find templates or use the search bar to find something more specific. In our example, we'll search for Business presentations.
3. Select a template to review it.
4. A preview of the template will appear, along with additional information on how the template can be used.
5. Click Create to use the selected template.

Saving a Presentation

Procedure for saving a presentation is same as saving a MS-Word document and MS-Excel workbook. For saving your presentation, do the following:

1. Click on the File tab and then, click Save As option from the menu that appear.
   Or
   Press Ctrl+S from keyboard.
2. Save As dialog box will appear, in the File name box, enter a new name to save the file.
3. Then, click Save button.

To open an existing presentation:

Select the File tab to go to Backstage view, then click Open.

Click Browse. Alternatively, you can choose OneDrive to open files stored on your OneDrive.

The Open dialog box will appear. Locate and select your presentation, then click Open.

5.3 Presentation Views in PowerPoint

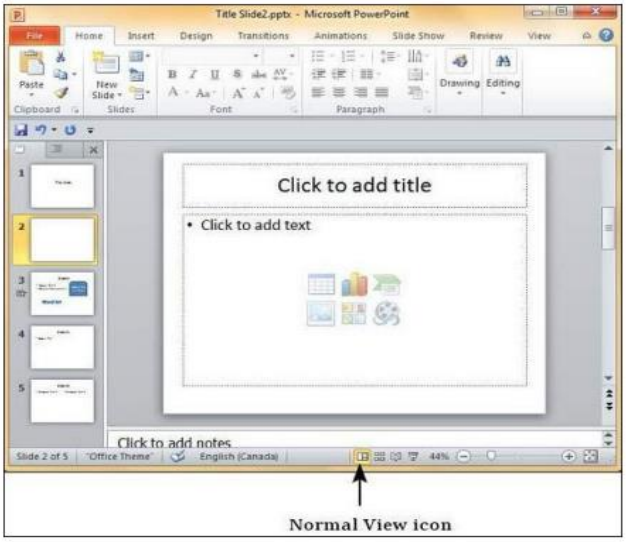
PowerPoint supports multiple views to allow users to gain the maximum from the features available in the program. Each view supports a different set of functions and is designed accordingly.

PowerPoint views can be accessed from two locations.

- Views can be accessed quickly from the bottom bar just to the left of the zoom settings.
- Views can also be accessed from the Presentation Views section in the View ribbon
- Working With Graphics in PowerPoint

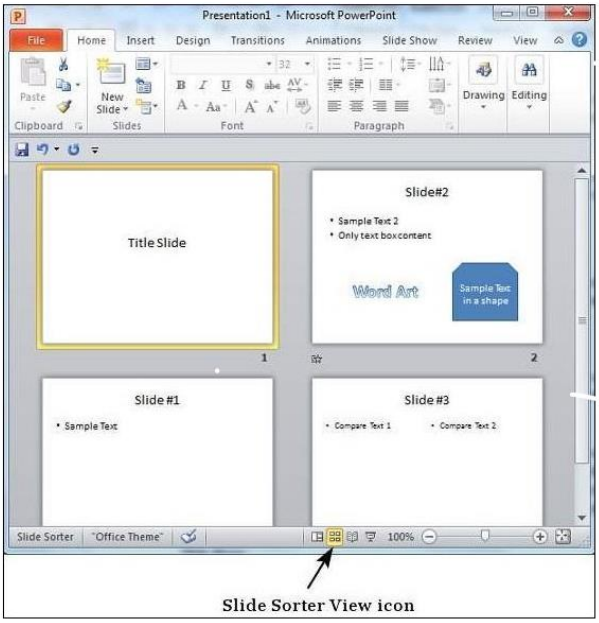
Normal View

This is the default view in PowerPoint and this is primarily used to create and edit slides. You can create/ delete/ edit/ rearrange slides, add/ remove/ modify content and manipulate sections from this view



Slide Sorter View

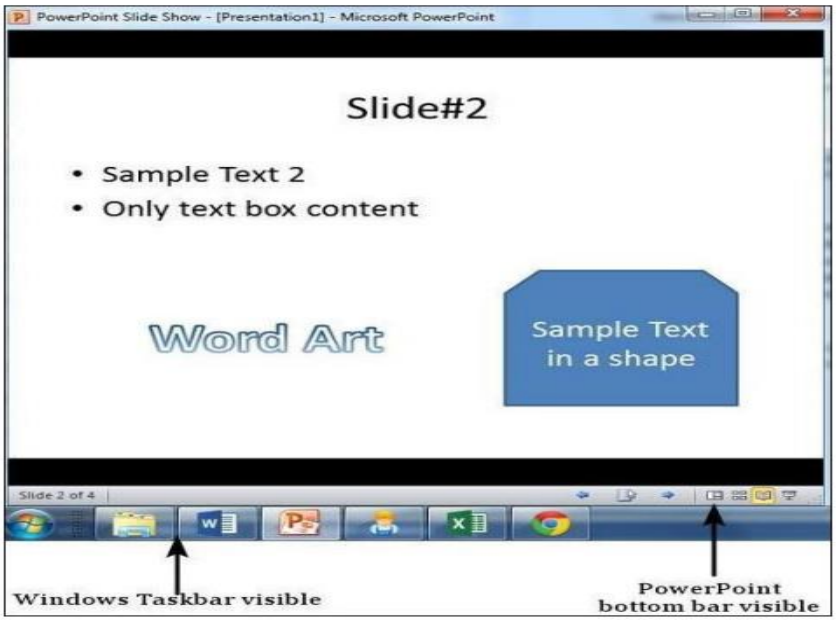
This view is primarily used to sort slides and rearrange them. This view is also ideal to add or remove sections as it presents the slides in a more compact manner making it easier to rearrange them.



Reading View

This view is new to PowerPoint 2010 and it was created mainly to review the slideshow without losing access to rest of the Windows applications. Typically, when you run the slideshow, the presentation takes up the entire screen so other applications cannot be accessed from the taskbar.

In the reading view the taskbar is still available while viewing the slideshow which is convenient. You cannot make any modifications when on this view.



5.4 Working with graphics

Adding Graphics in PowerPoint

Slide shows with pictures, graphics, and visuals are much more compelling and effective at conveying messages than slide shows that contain only boring text.

PowerPoint includes a variety of drawing and object formatting options.

Inserting Shapes

A set of ready-made Shapes is available in PowerPoint and can be used in slides.

You can add shapes, such as boxes, circles, and arrows, to your documents, email messages, slide shows, and spreadsheets.

Steps To add a shape:

1. On the Insert tab, click Shapes.

2. Click the shape you want, click anywhere in the workspace, and then drag to place the shape.

After you add one or more shapes, you can add text, bullets, and numbering to them, and you can change their fill, outline, and other effects on the Format tab.

Understanding WordArt

WordArt is used to insert decorative text in presentations.

To insert Word Art image in a slide:

1. Click Insert > WordArt, and pick the WordArt style you want.
2. In the WordArt gallery, the letter A represents the different designs that are applied to all text you type.

Understanding SmartArt

SmartArt allows you to communicate information with **graphics** instead of just using text. There are a variety of styles to choose from, which you can use to illustrate many different types of ideas.

A SmartArt graphic is a visual representation of your information and ideas.

To insert a SmartArt graphic:

1. Select the slide where you want the SmartArt graphic to appear.
2. From the Insert tab, select the SmartArt command in the Illustrations group.
3. A dialog box will appear. Select a category on the left, choose the desired SmartArt graphic, and click OK.
4. The SmartArt graphic will appear on the current slide.

Change the colors of an entire SmartArt graphic

You can apply color variations that are derived from the theme colors to the shapes in your SmartArt graphic.

1. Click your SmartArt graphic.
2. Under SmartArt Tools, on the Design tab, in the SmartArt Styles group, click Change Colors.
3. If you don't see the SmartArt Tools or Design tabs, make sure that you've selected a SmartArt graphic. You may have to double-click the SmartArt graphic to open the Design tab.
4. Click the color variation that you want.

Convert text into SmartArt

1. Select your text.
2. Select Home > Convert to SmartArt.
3. Select the SmartArt you want.

Insert pictures in SmartArt

Select a SmartArt option with pictures, and then insert your pictures.

1. Select your text.
2. Select Home > Convert to SmartArt.
3. Select More SmartArt Graphics and then select Picture.
4. Select the SmartArt you want and select OK.
5. To insert a picture, select the picture icon, go to where your picture is located, and then select it.
6. Select Insert.

5.5 Show Effect

- Presenting Slide Show Once your slide show is completed, you will need to learn how to present it to an audience. PowerPoint offers several tools and features that can help make your presentation smooth, engaging, and professional.
- To present your slide show, you'll need to know how to start it. PowerPoint allows you to start your slide show from the first slide or from any slide within the slide show. Once your slide show has started, you'll need to know how to advance through the slides. This Slide Show tab provides options for presenting the Slide Show to the audience. Narration for the show can be recorded. Decisions can be made as to where to start the presentation. In addition, custom shows can be created.



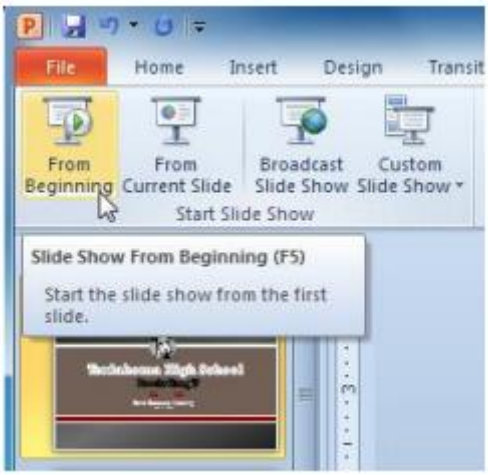
Groups/Buttons	Description
Start Slide Show Group	
From Beginning	Click this button to start the slide show at the first slide. This will start the show at the beginning even though the insertion point is not in the first slide.
From Current Slide	When the insertion point is in a different slide than the first one, clicking this button will start the show on the selected slide.
Present Online	This option is used to present a PowerPoint presentation on the Internet using the default presentation service. The viewers can watch the presentation through a Web browser.
Custom Slide Show	A custom slide show displays only selected slides. This option makes it possible to have several slide shows within the same presentation.
Set Up Group	
Set Up Slide Show	This button is used to set up advanced options for a presentation such as viewing the show on a kiosk.
Hide Slide	To hide a slide within the presentation, click this button. This means that the slide will not be shown during the presentation. However, it will still display in Normal view. This is a toggle button that when clicked the first time will hide the slide. When it is clicked the second time, it will display the slide
Rehearse Timings	This option is used to rehearse the timing for a presentation. These timings can be used when a presentation is a self-running show, such as when displaying it on a kiosk.

Record Slide Show	To record a narration to go with the slides in the presentation, click this button. The Record Narration dialog box will display where the options for recording can be set.
Play Narrations	This option allows narrations and laser gestures to be played back during a slide show.
Use Timings	Click the check box for this option if the timings that were set up will be used during the presentation.
Show Media Controls	Select this option to have media controls for audio and video show when the mouse pointer is moved over an object during a slide show presentation.
Monitors Group	
Show Presentation On	Click the arrow for this option to choose whether to let PowerPoint chose the monitor to show the presentation on or to allow the presenter to select the monitor. This option is only available when more than one monitor is connected to the computer.
Use Presenter View	This option is used when a computer is connected to more than one monitor. One monitor can be used to show the slide show while the other screen can be used to show speaker notes for the presenter. If the computer only has one monitor Alt+F5 can be used to display the slide show in Presenter View.

To start a slide show:

1. Select the Slide Show tab.

2. Click the From Beginning command in the Start Slide Show group to start the slide show with the first slide.



- You can also start the slide show from the slide you prefer by selecting the slide and clicking on From Current Slide from the Start Slide Show group. This option is convenient if you only want to view or present certain slides.
- Another option for starting the slide show is to select Slide Show view at the bottom of the window.



5.6 Animation Effects

- Animations in MS PowerPoint are the most interactive way to present the data to someone.
- In the simplest sense, animation is something that makes slide objects move on a slide. And slide objects can be anything on a slide. These include text, pictures, charts, SmartArt graphics, shapes, even movie clips.
- Animation can help make a PowerPoint presentation more dynamic, and help make information more memorable.
- Add animation and effects to text, pictures, shapes, and objects in your PowerPoint presentation.
- Animations are one of the most important features of MS PowerPoint from which we can make creative templates out of it.

Preview

- Animation effects that you have applied will show when you play the slide show. You can preview the animations quickly for the current slide without clicking the option of the slideshow.

Steps to preview:

1. Navigate to the slide you want to preview.
2. From the Animations tab, click the Preview command. The animations for the current slide will play.

Type of Animation

There are so many in-built animation effects provided by MS PowerPoint, and they are categorized into four types:

Entrance: It has animations saying that how an object should enter. For example, with the fly animation, the object will "fly" onto the slide from any of the 4 directions.

Emphasis: This type of animation usually occurs by the mouse clicks i.e when the object is already on the slide. For example, we can make the object wave after the entrance animation has finished.

Exit: These have a set of animations to control the exit animations. For example, in the Fade animation, the object will fade away.

Motion Paths: They are similar to emphasis animation effects, but these animations are a little advanced as the object will move in a specified path. These paths can be edited also.

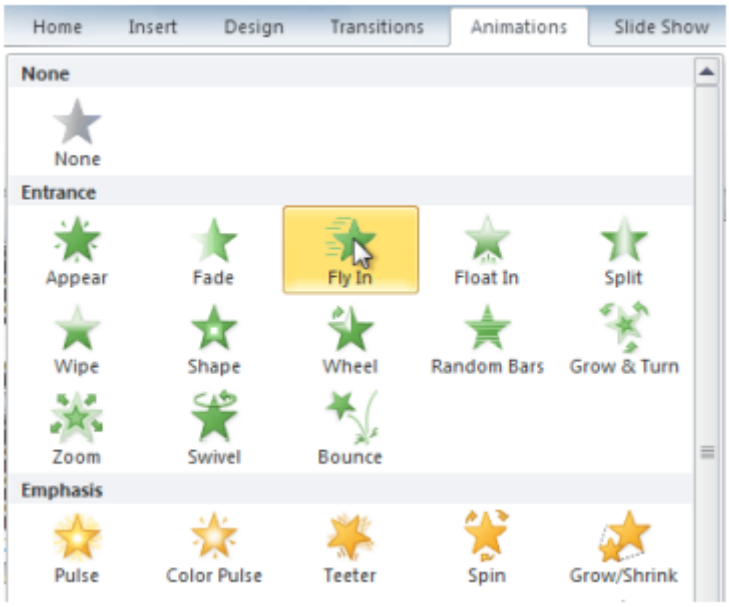
Steps to apply the animation:

To apply animation effects to an object:

Step 1: Select the object you want to animate.

Step 2: Go to the animation tab and click on the dropdown to view all types of animations.

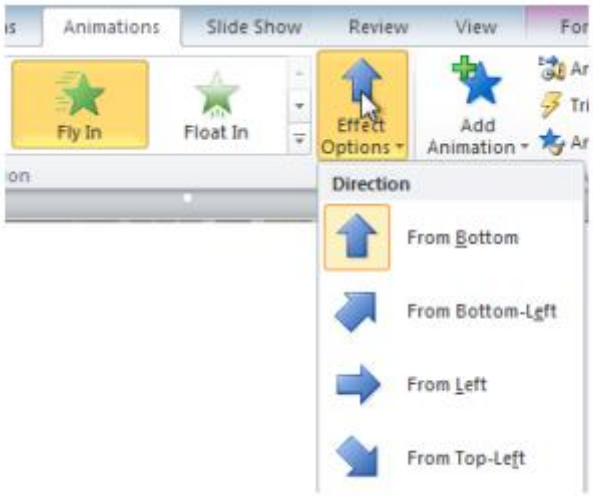
Step 3: A drop-down menu of animation effects will appear. Select the desired effect.



Step 4: The effect will apply to the object. In the Slide pane, a star symbol appears next to the slide, by which we can confirm that animation is applied.

Effect Options

Some effects will have options you can change. For example, in the effect of the random bars, we can control that we need vertical or horizontal bars. We can use these options by accessing the Effect Options command in the Animation group. It totally depends on the animation which is selected.



To add multiple animations to an object:

If you want to add new animation then we can reselect in the animation group but to add new animation without changing the existing one we use this option. And sometimes to make the content interactive we need to add 2 or more effects. To apply these changes, click on Add Animation command, this option will allow adding new animations without changing the old ones.

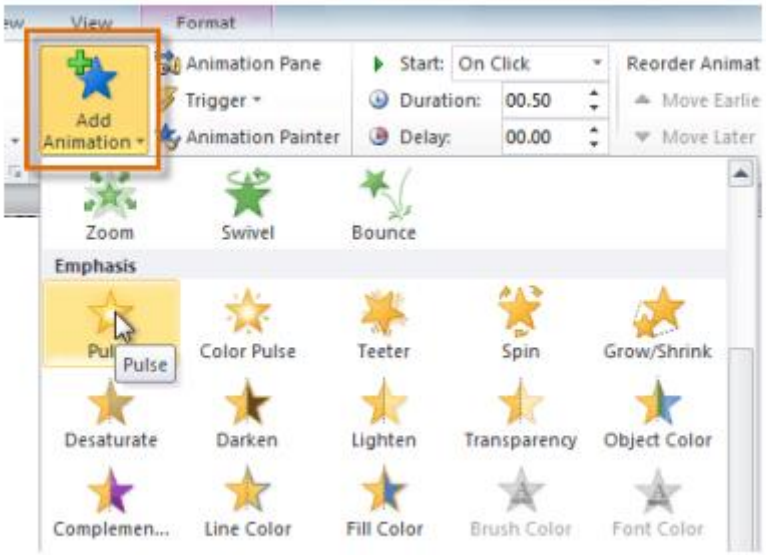
Steps to apply multiple animations to a single object:

Step 1: Select an object.

Step 2: Click the Animations tab.

Step 3: In the Advanced Animation group, we need to click the Add Animation command to view the available animations.

Step 4: Select the desired animation effect.


View
Format
Add Animation
Animation Pane
Trigger
Animation Painter
Start: On Click
Duration: 00.50
Delay: 00.00
Reorder Animat
Move Earlie
Move Later
Zoom
Swivel
Bounce
Emphasis
Pulse
Color Pulse
Teeter
Spin
Grow/Shrink
Desaturate
Darken
Lighten
Transparency
Object Color
Complemen...
Line Color
Fill Color
Brush Color
Font Color

XI. Indian Economy

End of Semester Examination 40 Marks
Continuous Assessment 35 Marks
Total 75 Marks

Learning Objectives: - The objective of the course is to equip the students with the ability to analyse, interpret and apply the basic concepts and theories of Indian economy in managerial decision making.
Utility: Student can acquire basic-level knowledge of Indian economy for business.
Prerequisite: Basic knowledge of Indian economy.

Unit. No.	Particulars	No. of Lectures
1	**INDIAN ECONOMY-A PROFILE** A) Nature of Indian economy B) Role of Agriculture, Industry and Service sector in development of Indian economy: their problems and growth. C) Basic understanding of tax system in India —Direct tax and Indirect tax	11
2	**SELECTED ASPECTS OF INDIAN ECONOMY** A) Population —Its size, rate of growth, measures to control the growth of population. B) Poverty- Absolute and relative poverty and main programs for poverty alleviation C) Unemployment- Types, causes and incidence of unemployment. D) Infrastructure- Energy, Transportation, communication, Health and education.	11
3	**MONEY AND BANKING** A) Money —Meaning and function B) Commercial banks- Role and functions. C) Reserve bank of India- Role and functions, monitory policy	11
4	**INFLATION AND DEFLATION** A) Concepts and features of inflation, demand pull and cost push inflation; measures to control inflation. B) Deflation and measures to control it.	11
5	**NATIONAL INCOME** Concept, significance, methods of measurement of National income, growth of national income and per capita income in various plans.	10
	Total	54

Reference books :

1. Indian Economy - Misra and Puri - Himalaya Publication
2. Indian Economy - Datt and Sundaram -S. Chand
3. Indian Economy - Francis Cherunilam - Himalaya Publication

Unit I

INDIAN ECONOMY-A PROFILE

1.1 Nature of Indian Economy

Over a period of about 6 decades the NNP has increased by more than 18 times. On an average, the NNP has increased at a rate of a little less than 5 per cent per annum. During the last three decades, NNP rose at a rate of 6.6 per cent per annum as against 3.5 per cent per annum during the first three decades of planning. Thus, we see that India is growing. In fact, today India is the second fastest growing country in the world after China. Its fundamentals are strong and because of this, even the global recession did not hurt India much. Following are some features of Indian Economy

(i) **Rise in Per Capita Income:** Per capita income in India was ` 6,122 in 1950-51 (at constant prices). It rose ` 35,917 in 2010-11. Thus, over a period of about six decades the per capita income has increased by nearly 5 times. If we consider the period 1950-51 to 1990-91, the rate of increase in per capita NNP was 1.6 per cent per annum. Since 1990-91, the per capita income shows an uptrend. It has increased at a rate of about 5.5 per cent per annum . Considering the fact that the economy nearly stagnated during colonial rule and many years after, this is a good achievement.

(ii) **Significant changes in occupational distribution of population**: By occupational structure of a country we mean the distribution of work force in different occupations of the country. All occupations are broadly divided into three groups.

   (i) Primary Sector: Primary sector includes agriculture and other activities related with agriculture such as animal husbandry, forestry, poultry farming etc.

   (ii) Secondary sector: This includes all types of manufacturing activities including construction etc.

   (iii) Tertiary sector: This sector includes trade, transport, communication, banking and other such services.

In general, it has been found that as an economy grows, there is a shift of labour force from primary sector to secondary and tertiary sectors. The proportion of working population in agriculture and allied activities falls and the proportion of working population in secondary sector and tertiary sector rises. This happens basically because of two reasons. Firstly, as economic development takes place income increases but demand for agricultural goods does not increase proportionately. On the other hand, rise in incomes brings about a large increase in demand for goods and services produced by secondary and tertiary sectors. Secondly, as an economy develops, better techniques of production become available to the agricultural sector which improve productivity of land and labour in this sector. The result is that there is a less need for labour in agriculture. On the other hand, although productivity also improves in the industrial sector, the increase in demand for industrial goods is far greater than the rise in productivity in

this sector. This necessitates engagement of more labour in this sector, hence the shift takes place.

Occupational structure in India: The following table shows the occupational structure in India.

Table 1: Occupational Distribution of Working Population in India

Occupation	1951	1991	2001	2009-10
Primary sector	72.1	66.9	56.7	49.3
Secondary sector	10.6	12.7	18.2	21.9
Tertiary sector	17.3	20.4	25.2	28.8
Total	100.0	100.0	100.0	100.0

Primary Sector includes, agriculture, forestry, logging and fishing. Secondary sector includes, mining and quarrying, manufacturing, electricity, gas and water supply and construction. Tertiary sector includes, trade, hotels and restaurants, transport, storage and communication, financial, insurance, real estate, business, community, social and personal services.

During 1951, Primary sector offered work to about 72 percent of the working population, secondary sector to 10.6 percent and tertiary sector to 17.3 percent of the working population. In 2001 there was some change in the occupational distribution. The primary, secondary and tertiary sectors respectively occupied 56.7 percent, 18.2 percent and 25.2 percent of the working population.

In 2009-10, around 49.3 per cent of the working population was engaged in primary sector, 21.9 per cent in secondary sector and 28.8 per cent in tertiary sector.

Thus, over a period of six decades there has been shift of work force from primary to secondary and tertiary sectors signifying development in the economy.

(iv) **Important changes in sectoral distribution of domestic product:** An important indicator which shows that India is growing is decline in the share of agricultural sector in the overall gross domestic product. The following table shows how over a period of about six decades, structural changes have taken place in India - the share of agricultural and allied activities has fallen and shares of industrial sector and (services) sectors have improved in the GDP.

Table 2 : Composition of GDP

	1950-51	1990-91	2000-01	2011-12
Agriculture	53.1	29.6	22.3	13.9
Industry	16.6	27.7	27.3	27.0
Service	30.3	42.7	50.4	59.0

Economy Survey 2011-12

(v) **Growing capital base of the economy**: Another characteristic which hints that the economy is growing is the development of strong industrial base in the country. At the time of

Independence, we had very few basic and capital goods industries. But after Independence, especially in the Second Plan a high priority was given to establishing basic industries. As a result, a large number of industries have been established during the planning period. These include, iron and steel, heavy chemicals, nitrogenous fertilizers, heavy engineering, machine tools, locomotives, heavy chemicals, heavy electrical equipment, petroleum products and many more.

(vi) **Improvements in social overhead capital**: Social overhead capital mainly includes transport facilities, irrigation facilities, energy, education system, health and medical facilities. When there is an expansion in these facilities, we say that the economy is growing. In India, since Independence, these facilities have improved a lot as can be seen from the following points.

- Indian railways covers about 63,000 kilometres. Indian railways is Asia's largest and world's second largest rail network under a single management.
- Diesel and electrical locomotives have replaced steam engines.
- The Indian road network has become second largest network in the world aggregating 3.34 million kilometers.
- Although the country is still facing energy crisis, there has been an impressive increase in the installed capacity. In 2011-12, the installed electricity generating capacity was about more than 2,00,000 MW (Mega Watt) against 2,300 MW in 1950-51.
- Similarly, irrigation facilities have increased raising the land under irrigation from 22.6 million hectares in 1950-51 to 87.2 million-hectares in 2007-08.
- In the field of education, during the planning period, the number of primary educational institutions has nearly quadrupled; the numbers of middle/senior basic schools and higher secondary educational institutions have increased by around 23 times. There are now more than 31,000 colleges for general education and 7000 colleges for professional education and more than 600 universities. The literacy rate has increased from 18.33 per cent in 1951 to 74 per cent in 2011 (Economic Survey 2011-12).
- In the field of medicine and health also, some development has taken place. The number of doctors has increased by more than 12 times increasing from 61,800 in 1951 to more than 8.15 lakh in 2010. The bed-population ratio is now 1.03 per 1,000 population increasing from .32 per 1,000 in 1950-51.

(vii) **Development in the banking and financial sector**: Since Independence, important developments have taken place in the banking and financial sector. Initially banks were under private ownership. But after Independence, the process of nationalization was started. In 1949 Reserve Bank of India was nationalised and later in 1969 and 1980 many big banks were nationalised. As a result, banks which earlier catered to very small population have now reached at every nook and corner. Agricultural sector, small scale industries and other sectors have been getting bank's funds on a priority basis and at concessional rates of interest.

Thus, we can say, that although India is economically not so strong economy, but it is on the road of development. If its present pace of development continues, in the near future it will become an economic force to reckon with.

1.2 Role of Agriculture in India

Agriculture is a very important sector of the Indian economy. It plays a major role in the overall development of the country as it contributes 12.3 of GDP and engages around 50 per cent of the population of the country. Its role is discussed in the following points:

(i) Providing employment: Agriculture provides employment to a large number of people in India. At the time of Independence around 72 per cent of the population was engaged in agriculture and allied activities. As economy developed, its other sectors (industry and services) also developed and the percentage of people working in agriculture sector came down to around 58 per cent in 2001-02. Even today, around 50 percent of working population is engaged in agriculture. It must however, be noted that in absolute terms there has been a big increase in the number of people engaged in agricultural activities. Besides, a large number of people earn their living by working in occupations dependent on agriculture like storage, procuring, trade and transport, marketing and export of agricultural products.

(ii) Share in national income: Agriculture contributes a large share in the country's gross domestic product. Its share in total GDP in 1950-51 was around 55 per cent which has come down to 12.3 per cent in 2010-11 (constant prices). This reduced share indicates that the economy and its non-agricultural sectors are growing, nevertheless share of 12.3 per cent is an indicator of the fact that in India, the growth of agriculture is still a critical factor in the overall performance of the Indian economy.

(iii) Supporting industries: Agriculture has a big role in the development of industries specially the agro-based industries such as textiles, sugar, tea, paper. There are several other industries like handloom, weaving and other cottage industries which also depend upon inputs from agriculture. The prosperity of these agro-based industries is directly dependent upon the availability of inputs from the agricultural sector. The prosperity of industries depends on agricultural prosperity from another angle also. The demand for industrial products depends upon the income of the farmers which in turn depends upon agricultural production.

(iv) Share in foreign trade: At the time of Independence and a number of years thereafter our export basket mainly consisted of three agro-production-cotton textiles, jute and tea. These three accounted for more than 50 percent of our export earnings. If we consider other agricultural commodities like cashew, kernels, tobacco, coffee, sugar, vegetable oil etc., the total share of agriculture in total exports was about 70 per cent. As economy developed, the share of agricultural exports in total exports fell down. In 2010-11 agricultural exports formed about 10 per cent of the national exports. In recent years, special schemes (like Special Agricultural Product Scheme) have been started to promote exports of fruits, vegetables, flowers, dairy products and forest products.

As far as the agro-imports are concerned they constituted just 3 per cent of national imports in 2010-11. This is very meager, considering the fact that nearly one-fourth of our total import expenditure at the time of Independence and many years thereafter was on agro-imports (food grains, pulses, edible oil etc.). India, over a period of time has become self-sufficient in the production of agro-products and need to import them only when there are severe shortages resulting from unfriendly weather conditions like droughts and floods.

(v) Supplier of food and fodder: Agriculture meets almost the entire food-needs of the people. In India, people spend a very large proportion of their incomes on food and food products. Thus, the cost of living of people also gets affected by agricultural prosperity. If food is costly; the cost of living of the people also gets affected to a great deal. Agriculture also provides fodder to sustain livestock comprising of cattle, buffaloes, sheep and poultry etc.

(vi) Savings of capital: Agriculture has a low capital output ratio; in other words it requires lesser capital per unit of output produced compared with industries. A capital poor economy like India can make

efforts to develop this sector which along with increase in production could increase employment opportunity in the rural areas and could help in solving problems of urban congestion and pollution in the cities.

(vii) Contributions to Government's revenue: The government revenues also depend a great deal on agricultural prosperity. The direct contribution of agricultural taxes to the revenues of the centre and the states is not significant but indirectly agriculture has a considerable influence on the revenues of the central and state governments. Particularly, when due to agricultural droughts, the revenue suffers a set-back, government expenditure on relief, etc., goes up a great deal leading to heavy deficit in government budgets.

(viii) Solving problems of urban congestion and brain drain: Migration from rural areas to urban areas and metropolitan cities has created a dual problem: on the one hand, it has deprived rural areas of skilled and educated persons and, on the other hand, it has created the problem of urban congestion. If agriculture is on the road to prosperity and is in a position to absorb fruitfully the growing talent in rural areas, the dual problem of urban congestion and rural brain drain will be solved.

1.3 Growth of agriculture during planning period:

In the following points we will learn how agriculture sector has developed in India over the years:

I Increase in production and productivity: The following table shows how agricultural production has progressed over the years.

Table 3 : Agriculture Production **Million Tonnes**

	Commodity	1950-51	2001-02	2010-11
1.	Foodgrains (m.t)	51.0	212.9	244.8
2.	Pulses (m.t.)	8.4	13.4	18.2
3.	Sugarcane (m.t.)	69.0	297.2	342.4
4.	Oilseeds (m.t.)	5.1	20.7	32.5
5.	Cotton (m. bales)	2.1	10.0	33.0
6.	Jute and Mesta (m. bales)	3.5	11.7	10.6

We can see from the table that over the last 6 decades, agriculture production has increased by more than thrice. In 1950-51, food grains production was 51 million tonnes which increased to about 245 million tonnes in 2010-11.

Significant breakthrough in the production of food grains (often termed as Green Revolution) had been made possible due to the adoption of the new agricultural strategy since 1966. This strategy stressed upon the use of high-yielding varieties of seeds, proper irrigation facilities, extensive use of fertilizers, pesticides and insecticides often termed as High Yielding Varieties Programmes (HYVP). Due to adoption of this Programme, the production and productivity of food grains especially of wheat have increased sharply. The food grains production increased from 81 million tonnes in the Third Plan (i.e. before HYVP) to 245 million tonnes in 2010-11.

The country has not witnessed any significant technological breakthrough in agriculture since then. The yield levels in wheat and rice (which were the major crops of Green Revolution) have plateaued. Since the scope of bringing more land under cultivation is very less, the food safety net for all on a continuous basis

requires increased production and productivity of food crops. Thus, there is an urgent need for technological breakthrough in the form of a Second Green Revolution.

Similarly, we find the production of pulses increased from 8.4 million tonnes to 18.2 million tonnes, of sugarcane from 69 million tonnes to 342 million tonnes in 2010-11, of oilseeds from 5.1 million tonnes to 32.5 million tonnes in 2010-11, of cotton from 2.1 million bales to 33 million bales, and of Jute and Mesta from 3.5 million bales to 10.6 million bales over the period 1950-51 to 2010-11.

HYVP was restricted to five crops - wheat, rice, bajra, jawar and maize. But among these, wheat made wide strides with production increased by about 7 times from 11 million tonnes (annual average) in the third plan to 86 million tonnes in 2010-11. The productivity of wheat during the same period has increased from 827 kilograms per hectare in 1965-67 to 2938 kilograms per hectare in 2010-11. On account of this, it is often said that the green revolution is largely wheat revolution.

Since pulses, oilseeds, fruits and vegetables are essential for nutritional security of people; there is a need for improving their production and productivity. Hence, technological breakthrough in these products is also necessary.

The long-term annual growth of food grains output has been around 2 per cent during 196061 to 2010-11. According to Economic Survey 2011-12, the compound growth rate of food grains production has deteriorated from 2.85 per cent during 1980-81 to 1989-90 to 2.32 during 2000-01 to 2011-12. The per capita availability of foodgrains has improved from about 395 gm in 1951 to 439 gm in 2010.

Agricultural productivity is measured in terms of yield per hectare of land. The productivity of food grains has improved from 2.74 per cent in 1980-1990 to 2.91 per cent during 2000-12. It is to be noted that in two major crops – wheat and rice, the yield levels have plateaued. It is important and necessary that productivity of land improves as there is hardly any scope for bringing more land under cultivation.

II **Diversified agriculture:** Indian agriculture has become diversified as will be clear from the following facts:

- ⌘ The share of non-crop sectors (fishery, forestry and animal husbandry) in total agricultural output has increased.
- ⌘ Area under commercial crop like sugar, cotton, oilseeds, etc. has increased.
- ⌘ Within food grains, area under superior cereals (rice and wheat) has increased and area under the inferior cereals has declined.

III **Modern agriculture:** Some qualitative changes have taken place in agricultural sector especially in India since 1966 when Green Revolution was started. These are:

- ⌘ The use of high-yielding varieties of seeds, chemical fertilizers, pesticides, threshing machine is rising.
- ⌘ Farmers are increasingly resorting to intensive cultivation, multiple cropping, scientific water management in some states.
- ⌘ There have been noticeable changes in the attitudes of farmers. They are ready to accept new and scientific techniques of production.
- ⌘ Agricultural capacity has improved a lot. This has been made possible due to use of modern techniques such as irrigation facilities, high-yielding varieties of seeds, tractors and other modern machines.
- ⌘ A number of institutions have come up for marketing agricultural products for providing agricultural credit, for purchasing and distributing of agricultural inputs and storage etc. They have facilitated growth of agriculture.

IV **Improved agrarian system:** At the time of Independence, there were three types of land tenure systems prevailing in the country - the zamindari system, the ryotwari system and the mahalwari

system. In all these systems, the land was cultivated by tenants and they paid rent for the use of land. Only the system of collecting rent or land revenue was different in these land tenure systems. Whatever the system of collecting land revenue, the tenant or the actual tiller of the land was exploited by the land owners. More than 25 per cent of the produce was taken away by these intermediaries (zamindar etc.) in the form of rent. These intermediaries did not work at all on the land but took away whatever surplus above the minimum subsistence the cultivators produced. The cultivators did not show any interest nor did they have any surplus left for modernisation of agriculture. Hence, there was virtual stagnation in the agricultural sector.

In order to stop the exploitation of the actual tillers of the soil and to pass on the ownership of land to them land reforms were introduced after Independence. Three measures were contemplated to achieve these objectives:

(i) Abolition of Intermediaries

(ii) Tenancy reforms

(iii) Reorganisation of agriculture

Legislations were passed in all states to abolish zamindari system and bring tenants into direct contact with the state.

Under the tenancy reforms, three measures were taken (a) Regulation of rent (b) Security of tenure and (c) Conferment of ownership rights on tenants.

Before Independence, the rent changed by the zamindars from the tenants was very high. It ranged between 30 to 75 per cent of the produce. So after Independence, legislations were passed to fix rents between 25-50 percent for different states. Security of tenure has also been provided by these states by passing legislations which disallow ejectments of the tenants except in accordance with the provisions of the law. Many States have also conferred ownership rights on the tenants. It has been estimated that as result of laws conferring ownership rights on tenants in various states, approximately 12.42 million tenants have acquired ownership rights over 6.32 million hectare of land. Ceilings were also imposed on agricultural holdings. That means limits were imposed on the amount of land which a family could hold. Accordingly, a family could hold 18 acres of wet land or 54 acres of unirrigated land. It has been estimated that 2.98 million hectares of land had been declared surplus of which 2.18 million hectares has been distributed to 5.58 million beneficiaries.

In order to solve the problem of fragmentation of holdings, the land was reorganized. Accordingly it was decided to consolidate holdings by giving to the farmer one consolidated holding equal to the total of the land in different scattered plots under his possession. Cooperative farming was also started but it did not succeed much.

Thus, we see that since Independence, we have a much improved agrarian system which has resulted due to the land reforms undertaken by the government.

V **Other developments :** Apart from the above, the following developments have also taken place:
⌘ Farmers have been getting material inputs at subsidised rates. ⌘ They are getting credit at low rates of interest.

- ⌘ Government is helping them in procuring their products at predetermined rates and marketing them.
- ⌘ Minimum wage levels have been fixed for agricultural labourers.
- ⌘ Special programmes such as Swarnjayanti Gram Swarojgar Yojana and Mahatma Gandhi National Rural Employment Scheme (MGNREGS) etc., are in place in rural areas to provide employment to the rural people.

- ⌘ The National Food Security Mission (NFSM) was launched in 2007-08 in identified districts of different states. The aim is to have self sufficiency in different food crops like rice, wheat and pulses. In order to give special attention to pulses production, an Accelerated Pulses Production Programme (A3P) has been launched as a part of NFSM Pulses.
- ⌘ The Rashtriya Krishi Vikas Yojana (RKVY) (now converged with MGNREGS) is being implemented for integrated development of food crops. The scheme focuses on agricultural mechanisation, improvement of soil health and productivity, development of rain fed farming systems, improvement of agricultural marketing and pest management.
- ⌘ In order to give boost to oilseeds, pulses, oil palm and maize, an Integrated Scheme of Oilseeds, Pulses, Oil Palm and Maize was started in Eleventh Plan.

1.4 Problems of Agriculture Sector in India

The agricultural sector in India faces the following problems:

(1) **Slow and uneven growth:**

(a) The growth of agricultural sector is not sufficient to meet the rising demands of fast growing population. While the population is growing at a rate of around 2 per cent per annum, food grain production has increased at an annual rate of around 2 per cent. (1960-2011) Growth in the agricultural sector has quite often fallen short of the Plan targets. The Ninth and Tenth Plans witnessed agricultural sectoral growth rate of 2.44 per cent and 2.3 per cent respectively compared to 4.72 per cent growth rate during Eighth Plan. During Eleventh Plan, agricultural growth is estimated at a little over 3 per cent against the target of 4 per cent. The poor performance of agricultural sector resulting mainly from deficient and uneven rainfall in the recent years has led to creating inflationary pressure in some primary products and reduction in the potential growth of other sectors by dampening growth. This rate is just sufficient to maintain the existing standard of consumption of people. If we desire better standards of consumption and nutrition, then agriculture will have to grow at a higher rate.

The Approach Paper to Twelfth Plan emphasizes the need to enhance our efforts to ensure 4 per cent average growth of agriculture in the 12^{th} plan.

(b) Certain crops (like wheat) are growing at a higher rate than other crops (like maize, jawar etc.).

(c) Low yield per unit area across almost all crops has become a regular feature of Indian agriculture. Though India is one of the largest producers of most of the agricultural crops but ranks very low in terms of yield. For example, it has the largest area under rice and wheat in the world and is the second largest producer of these crops, but in terms of productivity its world rank is 52^{nd} in rice and 38^{th} in wheat. There is a need for a renewed focus on improving productivity.

(d) There are regional imbalances in the spread of growth. The growth has remained confined to certain areas like Punjab, Haryana and Western Uttar Pradesh.

(e) Till very recently, the attention and resources were devoted to the development of agricultural crops and animal husbandry, fisheries and forestry were not given much attention.

(2) **Not so modern agriculture:**

(a) The HYVP was initiated on a small area of 1.89 million hectares in 1966-67 and even in 2003-04 only 80 million hectare of land was covered by this program which is just 44 per cent of the gross cropped area. Naturally, the benefits of the new technology have remained confined to this area only.

(b) In many areas and in a number of crops old methods of ploughing, sowing and harvesting etc. are still used. As a result, productivity in such areas and crops is very low.

(c) About 60 per cent net sown area is rain fed and there are no appropriate dry-farming techniques.

(d) Only 40 per cent of the gross cropped area has irrigation facilities.The irrigation sector requires a renewed thrust both in terms of investment as also modern management.

(3) **Flaws in Land reforms:**

(a) The legislation measures have not been completed in all the states.

(b) There are snags in legislation like definitions of 'personal cultivation' and 'tenants' were inadequate, substantial area were given to zamindars for their personal cultivation, landlords often forced their tenants to surrender the lands voluntarily, ceiling laws were inadequate and zamindars indulged in large scale transfer of land to their family members in order to escape these laws.

(4) **Problems relating to finance:**

Since agriculture is an unorganized profession dependent mainly on rains, banks and other financial institutions are reluctant to provide finance to this sector. In fact, till a very long period after Independence the main source of agricultural credit was the moneylender as organized institutions insisted on collateral securities. In 1951, moneylenders accounted for as much as 71.6 percent of rural credit. Moneylenders used to charge exorbitant rates of interest ranging from 18 to 50 percent. They often manipulated accounts and cheated the poor uneducated farmers. Therefore, after Independence steps were taken to free farmers from the clutches of money lenders, the most important being the expansion of institutional credit to agriculture. Fourteen banks were nationalised in 1969 and six banks were nationalised in 1980 with an important objective of providing credit to the rural and other priority sectors. In 1975, the government established Regional Rural Banks (RRBs) to specifically meet the requirements of the farmers and villages. This was followed by the setting of an apex bank called National Bank for Agriculture and Rural Development (NABARD) in 1982. Cooperative credit societies were also established to finance rural projects at lower rates of interest. As a result of all these efforts the share of moneylenders has reduced to about 27 per cent now and that of institutional credit has increased.

Of late a number of steps have been taken to enhance credit support to farmers. These include:

- Introduction of the “Farm Credit Package” in 2004. As a result of this package the flow of credit to the farm sector has consistenly exceeded the target.
- There has been a massive increase in the institutional sources of credit to agricultural sector in India.
- Farmers are given crop loans at concessional rate of interest and on easy terms.
- Kisan Credit Card scheme was started in 1998 to provide adequate and timely support for the banking system to the farmers for their cultivation needs. Nearly 11 crore credit cards have been issued till date.
- NABARD provides refinance to State Cooperative and Agriculture and Rural development banks, State Cooperative Banks, Regional Rural Banks, commercial banks and other financial institutions so that they can meet the credit requirements of rural people.
- From time to time, the government announces agricultural debt waiver and debt relief schemes so as to mitigate the distress amongst farmers.

Although much improvement has taken place in agricultural finance, the following problems have emerged:

- Agricultural loans are concentrated in certain region and states. For example, nearly half of the agricultural bank credit is concentrated in Southern States.
- The proportion of overdue to demand has been increasing. Nearly 40 per cent of the amount financed does not come back to the society.

- ⌘ The major beneficiaries of the agricultural credit have been the large and medium farmers.
- ⌘ There is a lack of experienced and skilled staff in these institutions.

(5) **Problems relating to warehousing and marketing:**

(a) The storage facilities with the individual farmers are normally very primitive types in the form of dug-holes and pits. As a result 10 - 15 percent of agriculture produce gets spoiled or eaten by rats. Government agencies like Food Corporation of India provide storage facilities but these are inadequate.

(b) There is a lack of organization among farmers so they do not get a fair price from the purchasers who are generally well-organized.

(c) There are a number of agents between the producers (farmers) and the consumers (buyers). They charge a heavy amount as their fees or as commission. As a result, the farmers do not get a fair share in the total product price charged.

(d) Because of heavy indebtedness, the farmers are many times forced to sell their produces at low prices and sometimes due to lack of proper transport facilities in the nearest market at not so great prices.

(e) A great number of farmers live just for subsistence. Their marketable surplus is very low or almost nil.

(f) Several malpractices exist in unorganized agricultural markets such as under weighing, levying of a number of unauthorised fees and taxes etc.

(g) The farmers are many a times not well informed about the prevailing market conditions including prices presvailing in the markets.

(h) Grading and standardisation are at a very low level. So often inferior quality gets mixed up with superior one, killing the motivation of farmers to produce superior quality products.

(i) In order to meet the needs of poor people in the country, the government runs a network of ration shops and fair price shops which provide food grains and other essential commodities at very low prices to consumers. Despite the massive coverage of these shops, the total requirements of food grains of all vulnerable sectors are not met.

There is a need to develop marketing infrastructure, storage, warehousing, cold chains and spot markets that are driven by modern technology. In this direction some steps have been recently taken. These include:

- ⌘ To bring about reforms in agricultural marketing, Agricultural Produce Market Committee (APMC) Act is being amended by various states.
- ⌘ To transfer agricultural technologies and information to the farming sector a number of initiatives have been taken by the Department of Agriculture & Cooperation. These include, setting up of Agri Clinics and Agri Business Centres (ACABC), setting up of Kisan Call Centres and developing of Kisan Knowledge Management System (KKMS) etc.
- ⌘ The National Policy for Farmers, 2007 is being adopted by the government. Major policy provisions include provisions for assets reforms, water use efficiency, use of technology, inputs oand services, good quality seeds, disease free planting material, credit insurance etc.
- ⌘ Modified National Agricultural Insurance Scheme and Weather Based Crop Insurance Schemes are being run for insuring the farmers against crop loss and loss arising out of bad weather . The agriculture Insurance Company of India Ltd. is the implementing agency for these.
- ⌘ Food Corporation of India (FCI) is the nodal agency for procurement, distribution and storage of food grains. It is taking many steps for improving food management in the country.

1.5 Role of Industry in India

In any economy, industries have an important role to play. In fact, it has been noticed that countries which are industrially well developed (example USA) have higher per capita income than those countries where industries are not well developed (example: India, Pakistan). The only exception to this could be the petroleum exporting countries (like UAE) which have a higher per capita income due to abundance of petroleum products and virtual monopoly in export of petroleum products.

In India, industrial sector plays the following roles:

i) **Modernising agriculture:** It modernises agriculture and improves productivity in it. It provides agriculture with the latest tools and equipments which enhance efficiency in this sector.

ii) **Providing employment:** Indian economy being labour surplus economy needs sectors which absorb ever increasing labour-force. Industries can play an important role here. It is the establishment of industries alone that can generate employment opportunities on an accelerated rate. In 2009-10, industries engage only about 22 per cent of the working population of India and there is a need to industrialise country and that too quickly.

iii) **Share in the GDP:** Over the years, the value-added by industrial sector in the GDP has improved from 16.6 percent in 1950-51 to 27 per cent in 2011-12 (constant prices).

iv) **Contribution to exports:** Indian industries contribute tremendously to the export earnings of India. In fact, manufactured goods alone contribute more than two third of the export earnings of India.

v) **Raising incomes of the people:** Industries generally help in raising the incomes of the people of a country. This is possible because industries are not dependent on vagaries of nature. By putting in more efforts, capital and improved technology industrial output and production can be raised. In fact, in this sector, the benefits of large scale production can be reaped. Higher industrial output results in higher income per head. In fact, in the industrially developed countries, the GNP per capita is very high as compared to the GNP per capita in industrially developing countries. For example, in USA GNP per capita was $ 48,450 and in India it was just $1,410 in 2011.

vi) **Enhancing further the economic growth:** As industrialisation grows, the role of capital goods vis-à-vis consumer goods gains strength. This helps in enhancing further the economic growth. It helps an economy to attain self-sustaining growth.

vii) **Meeting high-income demands:** Beyond certain limits, the demand of the people for agricultural products falls and for industrial products rises. Industries help in meeting these ever-increasing demands.

viii) **Strengthening the economy:** Industries help strengthen the economy in a number of ways: (a) The growth of industries producing capital goods i.e. machines, equipments etc. lets a country to produce a number of goods in large quantities and at low cost. This gives industrial character to the economy and strengthens its infrastructure (b) It makes possible the production of economic infrastructure goods like railways, dams etc. which in any case are non-importable. (c) Agriculture gets improved farm-implements, chemical fertilizers and transport and storage facilities due to industries. (d) Dependence on foreign sources for defense materials is a risky matter. Industrialisation helps a country to become selfreliant in defense materials.

1.6 Growth in Indian Industry

All the industries of a country can be grouped in two major ways (i) on the basis of the size of industries and (ii) on the basis of end-use.

(i) On the basis of size of industries: On the basis of size of the industries, they can be divided into large industries, medium industries and small industries. Large industries which largely form the basis of the country's index of industrial production include the following industries: (a) mining and quarrying (often referred as mining); (b) manufacturing; and (c) electricity, gas and water supply (often referred as electricity).

(ii) On the basis of end-use: On the basis of end-use of output, industries are divided into:

(a) Basic goods industries (like minerals, fertilizers, cement, iron and steel, non-ferrous basic metals, electricity etc.)

(b) Capital goods industries (like machinery, machine tools, rail-road equipments etc.)

(c) Intermediate goods (like chemicals, rubber, plastic, coal and petroleum products)

(d) Consumer goods - consumer durables and non durables (like man-made fibers, beverages, watches, cosmetics, perfumes etc.).

The growth rate of industrial output over the planning period has been nearly satisfactory. The industrial production has grown at an annual average rate of 6.2 per cent per annum over the planning period. Industrial growth in the country has in terms of long run trend, remained aligned with the growth of GDP. The long-term average annual growth rate of industries comprising mining, manufacturing, electricity and construction, during the post reform period between 1991-92 and 2011-12, averaged 7 per cent as against GDP growth rate of 6.9 per cent.

2 Pattern of Industrial Development since Independence: Now in the following points we will make a comprehensive review of the pattern of industrial development during the planning era since 1951. The industrial development pattern on the eve of Independence was characterized by the following elements:

(a) Lop-sided pattern of development dominated by too large and too small industrial units, with very few medium size units. There was a high concentration of employment in small industries and household industries (31%) and in large industries (43%).

(b) Capital employed per worker in industry was very low because of low priority given to industry, low level of domestic demand and low per capita income.

(c) Consumer goods industries were well-established to the utter neglect of capital goods industries like steel, machinery, heavy electrical and chemicals. The country had to largely depend on imports for capital goods.

The progress of industrialisation during the last six decades has been striking. There has been a remarkable development of capital goods industries, substantial diversification and broad-basing of manufactured products, phenomenal development of small scale industries. An impressive base has been created in sophisticated and high technology industrial sectors like electronics, machine tools, telecommunication equipment and the like.

(1) Industrial growth experienced ups and downs during the period 1951 to 2010-11. There was steady growth of about 8 per cent during the first three plan period viz., 1951-65. Thereafter, a significant decline was experienced for 15 years 1965-80 when the annual rate of industry fell down to 4.1 per cent per annum. In the period 1965-80 not only there was a deceleration in the industrial growth but the industrial structure also witnessed the phenomenon of retrogression. In other word, the growth of elite-oriented consumption goods such as man-made fibers, beverages, perfumes, cosmetics, watches,

clocks, etc. was considerably higher than the goods which satisfied the needs of mass of the people such as coal, cotton, railways, etc. The situation improved during 1980-91 when the annual rate of industrial growth became 7.8 per cent. In the year 1991 major economic reforms were carried out in the country. Wide scale reduction in the scope of industrial licensing, simplification of rules and procedures, limiting the domain of public sector, increasing

the limits of foreign investments in India, liberalization of trade and rationalization of taxes etc. were some of the measures taken. However, it has been noticed that the performance of industrial sector post reform period has not been very good. The average annual growth rate of industrial production fell to 5.7 per cent during 1990-91 to 19992000. The next two years saw deterioration in the industrial growth because of slow down in domestic demand for industrial goods, high prices of oil and infrastructural bottlenecks. The Tenth Plan (2002-2007) aimed at achieving a growth rate of 10 per cent in the industrial sector. But the overall industrial growth at an average of around 8.7 per cent per annum during this plan has remained well short of the target.

The Eleventh Plan aimed at 10 per cent per annum growth in the industrial sector.

Performance of the industrial sector during the Eleventh Plan was not very good. The industry grew at an annual average rate of about 7.4 per cent as against the target of 10 per cent during this Plan. A number of factors were responsible for this. The important ones are the global financial shock, persistent rise in crude oil prices, decline in foreign direct investment, shrinkage in demand for exports and decline in domestic demand.

(2) The structure of industry has shifted in favour of basic and capital goods and intermediate goods sector during the period of planning since 1951. The programme of industrialisation was started on a massive scale in the Second Plan (1956-61). Based on the Mahalanobis model, this Plan emphasized on building basic and capital goods industries so that a strong base for development in future could be made. Three Steel Plants were set up in the public sector at Bhilai, Rourkela and Durgapur. Public Sector made advances in machine building, machine tools, railway locomotives, heavy electrical, ship building, fertilizers etc. In subsequent plans and for almost four decades (1951-90), the strategy to favour basic and capital goods and to give public sector the responsibility to develop these industries was followed. As a result, we now have a strong industrial base in the country. **Table 4: Annual Average Growth Rate of Industrial Production**

Use Based Classification	1980-81 to 1991-92	Eighth Plan (1992-97)	Ninth Plan (1997-2002)	Tenth Plan (2002-2007)	Eleventh Plan (Mean growth rate from April 2006 to Dec. 2011)
Basic Goods	7.4	6.8	4.1	6.6	6.1
Capital Goods	9.4	8.9	4.7	14.4	18.0
Intermediate Goods	4.9	8.5	5.8	6.2	5.6
Consumer Goods	6.0	6.6	5.5	9.6	9.8

Sources: Economic Surveys Handbook of Statistics on Indian Economy, RBI

(3) There has been a remarkable growth of consumer goods industries especially those manufacturing elite-oriented consumer goods such as man made fibers, finer varieties of textiles, beverages, cigarettes, motor cars, motor cycles and scooters, refrigerators, TVs, air-conditioners, electrical goods like fans, watches and clocks, cosmetics and so on. In first four decades after Independence, the stress

was on the establishment of basic and capital goods. Since 1991, important changes have occurred in the industrial structure. Intermediate and consumer goods have got more importance than basic and capital goods. As a result, the output of consumer durables goods has expanded at rapid pace especially since 1991.

(4) Industrial sector has become broad-based and modernised. The role of traditional industries like textiles has reduced and role of non-traditional industries like engineering goods, chemical goods and electrical goods has improved tremendously. Manufacturing capabilities over a period of time have strengthened. We now manufacture a wide array of goods like food products, leather products, chemicals products, rubber and plastic products, metal products, machinery, transport and equipments, paper products, wood products and so on.

Manufacturing sector which was targeted to grow at 12 per cent could grow at only 7.7 during the Eleventh Plan. The manufacturing sector's performance even in the preceding plans was not very encouraging. The share of manufacturing sector in India's GDP is only 15 per cent, compared with 34 per cent in China. The slow pace of growth of manufacturing sector is a cause of concern. Unless the manufacturing sector becomes engine of growth, it will be difficult to generate more jobs and make growth inclusive. Concerned with this, the government released a National Manufacturing Policy (NMP) in November, 2011. The Objectives of the Policy are:

⌘ Increase manufacturing sector growth to 12-14 per cent over the medium term. ⌘ Enable manufacturing to contribute at least 25 per cent of GDP by 2022.

⌘ Create appropriate skill sets among the rural and poor urban for their easy absorption in manufacturing.

⌘ Increase value addition and technological depth in manufacturing. ⌘ Enhance global competitiveness of Indian manufacturing.

The Policy has identified sectors that will create large employment (e.g. textiles, leather, gems and jewellery, handlooms, etc.), sectors that will deepen technological capabilities in manufacturing (e.g. IT hardware and electronics), sectors that will provide strategic security (e.g. defence equipment), and sectors in which India has competitive advantage (automotive and pharmaceuticals and medical equipment). The NMP also emphasized the role of Medium and Small enterprises in manufacturing and in their contribution to the generation of employment.

(5) India is a mixed economy where public and private sectors co exist. After Independence, sustained efforts were made to strengthen the public sector in India. Since the country was in initial stages of development and massive investments were required which private enterprises could not afford, the public sector was given dominant position. Besides strong industrial and infrastructural base, removal of regional disparities, check over concentration of income and wealth, export promotion and import substitution were other benefits which emanated from existence of public sector in India. Public sector domain ranged from core industries like iron and steel, heavy engineering, heavy electrical, petroleum and natural gas, chemicals, fertilizers to non-infrastructure or non core areas like bread. Since Independence there has been a massive increase in the size and diversification of public sector. The number of central public sector units was just 5 with a total capital of about

Rs. 30 crore. In March 2011, the number of central public sector enterprises (CPSEs) increased to 248 with a cumulative investment of about ` 6,66,800 crore. Several public sector giants like Oil and Natural Gas Corporation, Indian Oil Corporation, Steel Authority of India, Bharat Heavy Electricals, Hindustan Aeronautics Limited, Bharat Electronics Limited, Cement Corporation, Coal India, etc dominate the Indian industrial scene. Out of 248 CPSEs, 220 were in operation. Out of 220, 158 earned profits and 62 earned losses in 2010-11.

Public sector faced certain criticism in the form of low profitability, high under utilization of capacity, low efficiency, over staffing, political interference, etc.

With the introduction of economic reforms in 1991, a number of steps were taken to dilute the role of public sector in India. Public sector was exposed to competition from domestic as well as foreign companies. Only 8 industries group where security and strategic concerns dominate were reserved for public sector. Rest of the sectors were opened to the private sector. At present only 3 industries are reserved for the public sector. These are atomic energy, minerals specified in the schedule to the atomic energy 1953 and rail transport.

(6) So far as the private sector is concerned, the dominance of large and monopoly business houses has increased several times. There were hardly two large business houses - Tata and Birla in 1951 but now not only the number of big business houses has increased enormously to about 80 (which include Reliance Group, Bajaj, Thapars, Mafatlals, Kirloskars, Goenka, Chhabria, Shriram, Walchand, Singhania and so on) their assets have also increased enormously during the last 50 years.

(7) A remarkable expansion and sophistication took place in infrastructural facilities since 1951, in such respects as power generation, development of energy sources, railway transport, telecommunication, roads and road transport and the like, which are basic prerequisites for industrial development. Large scale railway electrification and dieselisation, extensive discovery of petroleum and gas reserves and their extraction, nationalisation of coal mining and its development, petroleum refineries, pipelines, storage and distribution arrangements, hydro, thermal and atomic power generation, together with manufacture of heavy electrical equipment, electricity grids, electronic telephone exchange and microwave long distance telephone facilities, cellular mobile telephone services, electronic mail services and so on were taken up; significant strides made in these and other infrastructural facilities have catalyzed industrial development in several ways. Industrial finance was supported heavily by financial institutions (LIC, IDBI, ICICI for example) and commercial banks. Port facilities for imports and exports have been substantially expanded.

(8) The country could be proud of achieving remarkable progress in the science and technology front. Several Research laboratories were set up under the leadership of Council of Scientific and Industrial Research. R & D facilities were installed in public and private sector units. Technological know-how was extensively imported through foreign technical collaboration arrangements. Science, Engineering, Management and other professional educational institutions have been established on a large scale. An elite cadre of scientific, technical and professional manpower has been built; India ranks high in the world in respect of technological talent and manpower and in development of information and communication technology, space research, nuclear technology, electronics and so on.

(9) One of the notable features of the planning era since 1951 has been the mammoth growth of small-scale industrial units. Small-scale industrial units are those who operate with a modest investment in fixed capital, relatively small-scale work force and which produce a relatively small volume of output of goods/services. They differ from large-scale industries with respect to size of capital, employment, production and management, flow of input and outputs and so on.

The present Act - Micro, Small and Medium Enterprises Development (MSMED) Act, 2006 has broadly classified the enterprises in those engaged in (i) manufacturing and (ii) providing of services. Both categories have been further classified in Micro, small and medium based on their investment. In manufacturing sector units with investment upto 25 lakh are called micro enterprises, units with investment between ` 25 lakh and ` 5 crore are called small enterprises and units with investment between ` 5 crore and ` 10 crore are called medium enterprises. In the service sector, units with investment upto ` 10 lakh are called micro units and units between 10 lakh and ` 2 crore are called small emterprises and units with investment between ` 2 crore and ` 5 crore are called medium enterprises.

Since Independence, there has been an all-round development of small-scale and cottage industries in India. Their performance and contribution to the growth of the industrial economy of India has been quite remarkable. This will be clear from the following points: (a) The growth rate of small-scale sector at a rate of more than 10% per annum in terms of production has been far faster than that of large scale sector since 1973. It is estimated that MSMEs contribute about 45% of the gross value of output in the manufacturing sector.

(b) The number of registered and unregistered small-scale units which stood at 16,000 units in 1950 increased to 7 million in 1991-92 and to around 30 million MSMEs in 2009-10.

(c) The MSME sector employed nearly 70 million persons in 2009-10 compared to 7.5 million and about 17 million persons in 1981-82 and 1991-92 respectively.

(d) It is estimated that this sector contributes over 40 per cent of the total exports.

(e) Small-scale industrial units produce a very wide range of producer goods and consumer goods items needed by the economy. They include both simple and sophisticated engineering products, electrical, electronics, chemicals, plastics, steel, cement, textiles, paper, matches, ready made garments and so on.

(f) Ancillary units contribute greatly and cater to the requirements of medium and large industrial units for materials, components, consumables and so on.

(g) The traditional village and cottage industries which are generally clubbed with modern small-scale industries provide means of living to artisans, sustain viability of countless number of villages and towns, enrich the quality of life in society by providing fine handicrafts and pieces of art and project the heritage of India.

(h) A large number of MSMEs are engaged in the manufacture of consumer goods of mass consumption, thereby making them available in plenty which serves as a noninflationary force.

(i) The encouragement of MSMEs with their favourable employment-capital ratio has become a stabilizing force in the Indian economy. According to All India Census of Small Scale Industries, 2001-02, employment generated by the small scale sector per ` 1 lakh investment was 1.39 as against only 0.20 in respect of the large manufacturing units. In other words, an investment of ` 10 lakh generated employment for 14 persons in small scale sector and for 2 persons in large scale sector.

As far as the investment-output ratio is concerned, the small scale industries sector is almost at par with the large scale industries. It is estimated that they need slightly higher amount of investment (` 48,000) for producing the same output (worth ` 1 Lakh) than their larger counterparts (` 43,000).

1.7 Problems of Industry in India

(1) Failure to achieve targets: Except a few years, when targets of overall growth in the industrial sector were achieved, in the entire period of planning, achievements have been below targets. The average industrial growth rate during 1951 to 2011-12 has been around 6.2 per cent relative to the target of 8 per cent per annum. The following table shows how far our targets on the industrial front have been met:

Table 5 : Growth Rate of Industrial Production

(Per cent per annum)

Five year Plan/Annual Plan	Target	Actual
I (1951-56)	7.0	7.3
II (1956-61)	10.5	6.6
III (1961-66)	11.0	9.0
Annual Plans (1966-69)	-	2.0
IV (1969-74)	12.0	4.7
V (1974-79)	8.0	5.9
Annual Plan (1979-80)	-	1.4
VI (1980-85)	8.0	5.9
VII (1985-90)	8.7	8.5
Annual Plan (1990-92)	-	8.0
VIII (1992-97)	7.4	7.3
IX (1997-02)	8.2	5.2
X (2002-07)	10.0	8.7
XI (2007-12)	10.0	7.4

(2) Under-utilization of capacity: A large number of industries experience endemic under-utilization of production capacity. The average under-utilisation is in the region of 40% to 50%.

The factors responsible for under-utilization of capacity are said to be: (a) indiscriminate grabbing and creation of capacities by private enterprise (b) demand short-falls (c) overoptimistic demand projections (d) supply bottle-necks, (e) labour problems and (f) deliberate under-utilisation to create shortages and thereby to corner more profits.

(3) Absence of world class infrastructure : The most critical barrier to growth of the industrial sector is the absence of world class infrastructure. The short supply of transport facilities, frequent power failures and poor condition of roads have hampered the growth of industry in general.

(4) Increasing capital-output ratio: Another very disturbing future of industrial development of India is the ever-rising average and incremental capital output ratio (ICOR). The latter which was 2.95 during the first plan increased to 3.9 during the Seventh Plan and further to around 4 during Eighth, Ninth, Tenth and Eleventh Plans. The increasing trend of capital output ratios could be explained in terms of increasing capital costs of new industrial units, highly capital intensive nature of basic and heavy units, under-utilisation of capacity, unremunerative administered prices in respect of basic goods and services and so on.

(5) **High cost industrial economy:** The costs and prices of manufactured goods and services in India are generally much higher than international costs and prices. The high cost economy is attributed to import substitution, government protection to indigenous industries, monopolistic tendencies in several industrial areas, high wage rates, increasing capital intensity in industrial units, outdated technology low productivity of labour, uneconomic size of industrial units, lack of cost consciousness among industrial magnates and managers and so on.

(6) **Inadequate employment generation:** One of the most serious deficiencies of industrial development over the decades since Independence has been its inadequate employment generation in relation to investment made. The process of industrialisation has failed to make a marked dent on the unemployment problem in India. Factory employment absorbed only 2% of the labour force in 1980. There was only a marginal increase in the rate in 1980s. Employment generation through industrialisation has also been decreasing over the decades. According to the Annual Survey of industries, there has been a decline in the absolute number of persons engaged in the industrial sector between 1995-96 and 2004-05. Even labour intensive manufacturing sub-sectors like leather, food products, jute and leather products failed to generate adequate employment in these years. However, as per the Annual Survey of Industries, 2009-10, the situation has improved now as the absolute number of persons employed in factories has been increasing every year since 2004-05.

(7) **Poor performance of public sector:** Though public sector has grown by leaps and bounds over the planning period backed by massive public investment, its performance on production and profit fronts has been generally disappointing. Though profit may not always be the appropriate criterion for evaluating the performance of public sector industrial units, its relevance cannot be ignored altogether. Commercial and economic efficiency is to a large extent reflected in profit. A loss making undertaking becomes weakened in course of time. It loses dynamism and survival capability. A large number of public sector units are 'loss leaders' in the industrial sphere while the rate of profitability of others is low. The net loss of the loss making enterprises (62 in number) stood at more than about `22,000 crore in 2010-11 compared with `14,600 crore in 2008-09.

(8) **Sectoral imbalances:** In India, industrial development on an over-all basis suffered several set- backs because of inadequate support from agriculture and infrastructure. Even within the industrial sector the input-output relations between individual industries like steel and machine building, petro-chemicals and fertilizers, are such that they have to be developed in harmony. But in real practice, several sectoral imbalances plague the industrial economy of India.

(9) **Regional imbalances:** Industrial development continues to be lopsided, region-wise. Large scale industries are concentrated in a very few states like Tamil Nadu, Maharashtra, Andhra Pradesh and Gujarat. These four States account for 50% of total factories and 50% of productive capital. It is true that a large number of new industrial growth centers have emerged since Independence in several States like U.P., Punjab, Karnataka, M.P., and Rajasthan. But these States continue to be industrially backward. The industrial units established in these States have somehow failed to generate further industrialisation.

(10) **Industrial sickness:** Industrial sickness has become a serious problem affecting small, medium and large units. It is a major area of concern due to its implications for the entire economy and health of the industrial sector in particular. In March 2009, there were 1.07 lakh sick units out of which more than 96 percent were small units. Industrial sickness has been spreading over the years. The causes of sickness are identified as financial mismanagement, demand recession, labour unrest, working capital shortage, cost escalations, shortage of raw materials, uneconomic size, out-dated machinery and equipment and so on.

1.8 Role of Service Sector in Indian Economy

such as information technology that is dependent on the skilled labour. Growth in tourism and tourism-related services such as hotels holds a large potential for employment generation.

IT enabled services, such as Business Process Outsourcing (BPO) have been growing rapidly (60-70 per cent) in the recent past and will continue to grow. Outsourcing has changed the image of India. Western companies are seeing India as their top destination for outsourcing work.

Factors underlying the Services Sector Growth

The Services sector has grown at a fast rate in India due to the following reasons:

- ⌘ It has been noticed that income elasticity of demand for services is greater than one. Hence, the final demand for services grows faster than the demand for goods and commodities as income rises.
- ⌘ Technical and structural changes in the economy have made it more efficient to out source certain services that were once produced with in the industry.
- ⌘ With the advent of the information technology revolution, it has become possible to deliver services over long distances at a reasonable cost, thus trade in services has increased world wide. India has been particular beneficiary of this trend.
- ⌘ Economic reforms initiated since 1991 also impacted on the performance of the services sector. Increased demand for manufacturing industry provided synergies to the services sector. Also, liberalisation of financial sector provided an environment for faster growth of the financial services. Moreover, reforms in certain segments of infrastructure services also contributed to the growth of services.

2.2.2 Problems of service sector in India: Although service sector is doing remarkably well, it is facing lots of problems, important ones are:

(1) Achieving rapid growth of the economy and its counterparts requires a very high quality of infrastructure. Unfortunately, our infrastructure is inadequate not only in the rural areas but also in the urban areas. For example, power shortage and traffic congestions are very common in Bangalore, the silicon city of India. These affect the quality of services provided.

(2) Though service sector has been the fastest growing sector in the last dacade and has contributed about 59 per cent of the GDP, it provides less than 30 per cent of the total employment. Services that have witnessed a very high growth rate e.g. business and communication services have a low share in GDP or employment.

(3) Although economic reforms have been undertaken in all the sectors but they are inadequate. In financial sector many controls and bottlenecks are still there. These need to be removed if financial sector of India has to achieve the international standard.

(4) India has great potential in the tourism sector. But there is a need to create proper set up for attracting tourists. Foreign tourists often get harassed and cheated in the hands of babus and officialdom, touts and conmen.

(5) Etiquettes and good behaviour are the hallmark of the service sector. Indian service providers whether they are in banks, in hotels and restaurants, in hospitals or in public administration, they need to be trained thoroughly in public dealing, etiquettes, hospitality and manners.

(6) The airports, railways etc. in India are not clean and well organized. They need to be revamped and reorganized.

(7) Our consular division also is not proper. It takes many days to issue visas. This hampers the growth of tourism sector. For this, we need to have a system which entails singlewindow clearance.

(8) Service trade also faces a number of problems. These include lack of set up like export promotion councils (other than for computer software), various visible and invisible barriers to service trade for example, visa restrictions, sector specific restrictions and preferential market access.

(9) Service sector cannot grow in isolation. It needs strong backing of other sectors, primary and secondary. In India, such backing needs to be strengthened. In other words, other sectors especially industries need to grow up if they have to provide market to the service sector.

(10) Indian service providers (like BPOs and IT service providers) are facing stiff competition from other countries. They need to improve their quality and reduce their costs. There are certain political problems also like a backlash from European and North Americans countries especially in the case of BPOs.

1.10 Tax System in India

Tax is the most important source of revenue of the Government. A tax is a compulsory contribution from a person to the expenses incurred by the State in common interest of all without reference to specific benefits conferred on any individual. Taxes are generally classified into direct taxes and indirect taxes. Taxes which are not shifted i.e., the incidence of which falls on persons who pay them to the Government are direct taxes. Examples of direct taxes are income tax and wealth tax. Where the burden is shifted through a change in price, the taxes are indirect. Examples of indirect taxes are sales tax, custom duty, excise duty etc.

4.1 MERITS AND DEMERITS OF DIRECT AND INDIRECT TAXES

Merits of Direct Taxes :

(i) They are imposed according to the ability of the person to pay. Therefore these taxes are considered progressive.

(ii) The revenue is income elastic; because of the progressive character revenue will increase faster than the increase in income.

(iii) These taxes create better civic consciousness because the person paying knows clearlyhow much he has paid. This incidentally fulfils the objective of certainty.

(iv) They best serve the purpose of transference of income from the rich to the poor, throughprovision of amenities to the poor or even direct monetary help like old age pensions.

Demerits of Direct Taxes

(i) The ability to pay is difficult to determine; only a rough idea can be formed.

(ii) Because of undeclared sources of income or evasion, the actual payment may not be strictly according to the ability to pay. It is also sometimes said that direct taxes are taxes on the honesty of the person.

(iii) Such taxes necessitate proper maintenance of accounts which some of the tax payers may not be able to do.

The assessment procedure is also cumbersome requiring expert assistance of tax advisers.The direct tax system is often very complicated.

Merits of Indirect Taxes

(i) The most important merit is convenience in assessment and a relative difficulty in evasion. This is because they are assessed at flat rates and realised at appropriate point such as at the factory site in the case of an excise duty on production or at point of entry in the case of imports.

(ii) Since the tax is included in the price, the consumer may not even realise that he is paying a tax. The amount of tax on each item is often so small as really not to hurt the tax payer.

(iii) Even these taxes may not be really regressive if they are levied on ad valorem basis or on the basis of value. The rates may also be differential-higher for luxury articles and lower for necessaries; the latter are sometimes fully exempt.

(iv) Such taxes are difficult to evade. Unless the producers resort to manipulation of accountsor smuggling, it is difficult to evade the excise duty. In case of customs duties, articles are taxed the moment they enter the country. However, it is difficult to make a similar claim in respect of sales tax.

(v) Indirect taxes on drinks, narcotics and tobacco, serve a social purpose by discouraging their consumption.

Demerits of Indirect Taxes:

(i) These taxes are often criticised for their regressive character. Taxes on necessaries of life will certainly mean taxing the poor and that will mean taxing the rich and the poor alike.

(ii) Also it is contended that these taxes do not create social consciousness because they are often not felt by tax payers.

(iii) Government is not certain about the proceeds of these taxes.

(iv) The burden of indirect taxes can be shifted forward or backward. In most of the cases, theconsumers have to bear the ultimate burden of indirect taxes.

(v) These taxes can also be evaded by such methods as smuggling, falsification of accounts etc.

TAX STRUCTURE IN INDIA

Direct Taxes in India : Under this mainly income tax, wealth tax and gift tax are included.

Income Tax : Income tax is a tax on the income of an individual or an entity. Income Tax in India was introduced in India in 1860 but was discontinued in 1873. It was reintroduced in 1886 and since then it has stayed. Since its reintroduction a number of changes have been made in its structure, rates, exemptions and other dimensions of this system. Important types of Income tax are Personal income tax and Corporate income tax. Though the State Governments have power to levy a tax on agricultural income, in practice this tax has not developed as a major source of revenue for the State Governments.

Personal income tax is levied on the income of individuals, Hindu Undivided Families, unregistered firms and other association of people. For taxation purposes incomes from all sources are added. Certain rebates, deductions, expenditure etc. on account of Life insurance,

medical insurance, savings in Public Provident Fund and certain notified instruments are allowed. Whole income is divided into different slabs and it is taxed on the basis of slab into which it falls.

Like all other countries India has a progressive income tax. That means, as income increases, the rate of tax also increases. There was a time when the income tax rate , inclusive of surcharge was as high as 97.75 per cent for the highest income slab. Since 1974-75, it has been brought down(in stages) to 30 per cent in 1997-98. At present also, the marginal rate of income tax (i.e., tax for the highest slab) is 30 per cent. Thus the degree of the progressivity of the income tax schedule has been considerably reduced.

Corporate Tax is levied on the incomes of registered companies and corporations. The rationale for the corporation tax is that a joint stock company has a separate entity and thus should be taxed separately.Untill 1960-61, corporations were taxed in a partial sense. A corporation was required to pay income tax on behalf of its shareholders on dividends paid to them, and each shareholder got a credit to this effect. Since 1960-61, corporations are being treated as independent entities and shareholders are not given any credit. Though the corporates are being taxed at a flat rate, there are provisions for various kinds of rebates and exemptions. Tax rates are different for Indian companies and foreign companies. Certain types of companies (e.g. export houses) are given tax exemptions and tax holidays.

Taxes on Wealth and Capital : Taxes which are levied on wealth and capital are mainly estate duty, annual tax on wealth and gift tax. Estate duty was first introduced in India in 1953. It was levied on the total property passing to the heirs on the death of a person. From the point of view of proceeds, the estate duty was a minor source of revenue and it was abolished in 1985. An annual tax on wealth was introduced in 1957. It was levied on the wealth such as land, bonds, shares etc. of the people. Certain types of properties such as agricultural land and funds in Provident Account were exempt. Like estate duty this is also a minor source of revenue. With effect from 1.4.1993, wealth tax has been abolished on all assets except certain specified assets such as residential houses, farm houses, urban land, jewellery, bullion, motor car etc. A gift tax was first introduced in 1958 and was leviable on all donations to recognised charitable institutions, gifts to women dependents and gifts to wife. Gift tax was abolished in 1998.

Income Tax on gifts (received without adequate consideration) was partially reintroduced in April 2005. Gifts received from any person or persons, if the aggregate value exceeds ` 50000, have been made taxable under the head "Income from other sources". Certain exemptions have, however, been given. With effect from 1.10.09 even movable and immovable property given as gift would attract tax if its value exceeds ` 50000.

In order to overcome the complications relating to all direct taxes (income tax, dividend distribution tax and wealth tax) and to put in place an efficient, effective and equitable direct tax system, the Direct Tax Code (DTC) Bill, 2010 was introduced in Parliament. The DTC is likely to be implemented in coming years.

4.2.1 ***Indirect Taxes:*** The main indirect taxes levied in India are custom duties, excise duties, sales tax *and service tax*.

Custom Duties: Custom duties are levied on exports and imports. From the point of view of revenue, the importance of export duty is limited. Import duties are generally levied on the basis of ad valorem which means they are determined as a percentage of the price of the commodity. On some commodities, specific import duties i.e, per unit taxes on imports are levied. In pre-tax reform period, India had become a country with one of the highest levels of custom tariffs in the world. As a part of rationalisation measures carried out since 1991, the custom duty structure has been pruned.

The peak rate of custom duty on non-agricultural products, which was 20 per cent in 2003-04, was successively reduced in subsequent budgets to reach a level of 10 per cent in 2007-08. In the subsequent budgets, no major change has been made in the overall rate structure of custom duties.

In the first half of 2008-09 fiscal, prices of certain commodities like crude oil, steel and food etc. shot up world wide. This affected their domestic prices and demand. Consequently, as an anti-inflationary measure, custom duties on many commodities were reduced.

Subsequently, in the aftermath of the impact of global financial meltdown since September 2008, there was a sharp decline in international commodity prices and some of the duty cuts were, therefore, reversed.

But since the growth momentum of the economy was slowing down in the second half of the 2008-09, fiscal stimuli in the form of certain custom duties cut and exceptions were also given. These stimuli were continued in 2009-10, 2010-11 and 2011-12 as well.

Excise Duties : An excise duty is levied on production and has absolutely no connection with its actual sale. It is a tax on manufacturing, which is paid by a manufacturer, who passes its incidence on to the customers. Excise duties are mostly levied by the central Government in India and are known as Central Excise duties. There are certain goods, however, on which only states levy excise duty in India. These are alcoholic liquid for human consumption, opium, Indian hemp, narcotic drugs and narcotics. It is levied in a number of forms. Over the years, number of rate categories has been reduced and number of exemptions has also been brought down.

Taxation on inputs, such as raw materials, components and other intermediaries has a number of limitations. In production process, raw material passes through various processes till a final product emerges. Thus, output of the first manufacturer becomes input for second manufacturer and so on. When the inputs are used in the manufacture of product 'A', the cost of the final product increases not only on account of the cost of the inputs, but also on account of the duty paid on such inputs. As the duty on the final product is on ad valorem basis and the final cost of product 'A' includes the cost of inputs, inclusive of the duty paid and duty charged on product 'A'. This means doubly taxing raw materials. In other words, the tax burden goes on increasing as raw material and final product passes from one stage to other because, each subsequent purchaser has to pay tax again and again on the material which has already suffered tax. This is called cascading effect or double taxation. This very often distorts the production structure, and does not allow correct assessment of the tax incidence. In order to remove these defects, the government introduced Modified Value Added Tax (MODVAT) in 1986-87. Value added is the difference between a firm's revenues and its payments to other firms. It is the value difference between sales and purchased items.

However, MODVAT suffered from many shortcomings. The most important being, the existence of a number of rates on output and inputs leading to disputes relating to classification of both the output and inputs.

In order to combat the problem, the Budget 2000-01 introduced the Central Value-Added Tax [CENVAT]. Under the Cenvat Scheme, a manufacturer of final product or provider of taxable service, shall be allowed to take credit of duty of excise as well as of service tax paid on any input received in the factory or any input service received by manufacturer of final product.

Sales Tax : Sales Tax in India is a form of indirect tax that is imposed by the government on the sale or purchase of a particular commodity within the country. Sales tax is an indirect form of tax, wherein it is the responsibility of the seller of the commodity to collect and recover the tax from the purchaser. Sales Tax in India is imposed under both, Central Government (Central Sales Tax) and State Government (Sales Tax) Legislation. Generally, each state follows its own sales tax act and levies tax at various rates. In India, many commodities are not covered by sales tax. Sales tax is more in the case of luxury items and less or almost nil in the case of necessities. Under sales tax, the registered trading concerns are required to pay the sales tax to the government. These registered concerns shift the burden of sales tax to the customers. Sales tax regime suffers from many problems, main being, cascading effect,lack of transparency, narrow base, different procedures followed by different states and so on.

In India, sales tax was in two forms – state sales tax and central sales tax. State sale tax (i.e. tax on transactions within a state) is being replaced by Value Added Tax in all states. Central sales tax is inter-state sales tax. This tax is non-rebatable tax and is incongruent with the system of VAT. Therefore, it is being phased out in stages. At present central sales tax is 2 per cent.

VAT : Value Added Tax (VAT) is a multistage sales tax with credit for taxes paid on business purchases. One of the major benefits of VAT over sales tax is that the VAT avoids double taxation.

One of the important components of tax reforms initiated since liberalization relate to introducing state-level value added tax (VAT). The VAT is a multi-point destination based system of taxation, with tax being levied on value addition at each stage of transaction in the production/distributional chain. If for example,

inputs worth ` 1,00,000/- are purchased and sales are worth ` 2,00,000/- in a month, and input tax rate and output tax rate are 4% and 10% respectively, then input tax credit and calculation of VAT will be as shown below:

(a) Input purchased : ` 1,00,000.

(b) Output sold : ` 2,00,000.

(c) Input tax paid : ` 4,000.

(d) Output tax payable : ` 20,000.

(e) VAT payable after set-off input tax credit : ` 16,000.
[(d) - (c)]

The following are the benefits of VAT:

⌘ A set off will be given for input tax as well as tax paid on previous purchases. ⌘ Other taxes such as turnover tax, surcharge etc. will be abolished.

⌘ Overall tax burden will be rationalised.

⌘ Price will in general fall. ⌘ There will be higher revenue growth.

⌘ The zero rating of exports would increase the competitiveness of Indian exports.

⌘ There is a provision of self-assessment. ⌘ There will be more transparency.

VAT was introduced in 1999 and was implemented in April, 2005 in some states.

At present all states/union territories have successfully implemented VAT.

Service Tax : *Service tax is a form of indirect tax imposed on services. Introduced in the year 199495, service tax network has expanded to cover almost all services except a negative list of services.*

The indirect tax regime in India is being replaced by a comprehensive dual Goods and Services Tax (GST). GST would replace most indirect taxes currently in place i.e. octroi, Central Sales Tax, State-level sales tax, entry tax, stamp duty, telecom license fees, turnover tax, tax on consumption or sale of electricity, taxes on transportation of goods and services, etc., thus avoiding multiple layers of taxation that currently exist in India. The GST is likely to be implemented in very near future in India.

4.2.2 ***Features of Tax Structure in India :*** Following are the main features of tax structure of India :-

(i) Tax revenues (on account of the centre, state and union territories) form about 16 per cent *(2010-11)* of the total national income of India. This was only 6.7 per cent in 1950-51 and 11 per cent in 1960-61.

(ii) Over the last 59 years tax revenue collected both by the Central and State governments has increased many folds from ` 460 crore in 1951-52 to more than ` *11,60,000* crore in *2010-11*.

(iii) The ratio of direct to indirect taxes which was 40:60 in 1950-51 declined to 20:80 in 199091. Thus there has been an increasing reliance on indirect taxes which is not good since they fuel inflationary trends in the country. But since 1990-91, in wake of rationalisation, the proportion of direct taxes has been on rise, whereas that of indirect taxes on the decline. The share of direct taxes in the gross tax revenue (Centre and States combined) was *41* per cent in 2009-10 while that of indirect taxes declined to *59%.*

(iv) The population of the economy is more than *120* crore. But very small fraction (less than 3 per cent) of the population pays income tax in India. Thus Indian tax structure relies on a very narrow population base.

(v) The total tax revenue is highly insufficient to meet the expenditure requirements of the economy. Over time there has been an increasing reliance to internal and external debts.

(vi) The structure of taxes in India has under gone changes. In 1970-71, excise duties were contributing more than half (about 56 per cent) of the central tax revenue. The relative share of personal income tax, corporation tax and custom duties was 5 per cent, 15 per cent, and 21 per cent respectively. In 1990-91, custom duties became the most important source of tax revenue and contributed nearly half of the total tax revenue. In the post economic reforms period, there has been a change in the scenario. In 2010-11, the most important contributor to tax revenue is corporation tax (37 per cent) followed by excise duties (19 per cent), personal income tax (18 per cent) and custom duties (16 per cent). Similarly, land revenues were important source of state revenue. Then sale tax became more important.

(vii) In India, the direct taxes are progressive, indirect taxes are differential in nature. In otherwords, direct tax rates increase with increase in income and indirect tax rates are higher for luxury items and lower for necessities.

(viii)The agriculture income is mostly exempt from the income-tax.

4.2.3 *Evaluation of the Indian tax system*

Although direct taxes are progressive there is a differential rate schedule for indirect taxes. With the increase in national income, the tax yields in general and from direct taxes in particular have not increased at a rate high enough to show a high degree of income elasticity. Direct taxes (centre and states combined) were 2.1 per cent of the GDP in 1950-51. It has increased to around 7 per cent in *2010-11*.

Indian Tax system largely depends on urban incomes and leaves out almost completely agricultural incomes from the purview of direct taxes. India's tax system has a much reduced scope of manoeuvrability in the field of personal taxation. Thus, while national income rises, with about *one-eighth* of it originating in the agricultural sector, the tax system is not able to tap fully the rising income. The indirect tax system too is characterised by inelasticity.

Both the coverage and the rate schedule have been modified from time to time so that the tax system plays a truly functional role for economic growth, stability and social justice.

It is to be noted that while the service sector accounts for about 59 per cent of GDP, service tax (2010-11) contributes just 9 per cent towards tax revenues and about 1 per cent towards GDP.

In respect of canon of convenience, several, measures have been taken such as self assessment, advance payment, deduction of tax at source, assessment on the basis of returns submitted, etc. However, changes in tax laws in quick succession disturb long-term business decisionmaking. Indirect taxes, although considered to be regressive, are quite convenient from the collection point of view.

Simplification of tax system has also been attempted. Income tax returns have been simplified and made handy. The Booth Lingam Committee and Chelliah Committee recommended simplification and rationalisation of tax system in India. The proposed Direct Taxes Code and Goods and Services Tax (GST) also aim at simplification of tax laws.

The cost of tax collection (all taxes - direct and indirect) has increased over the years. It has increased from ` 543 cores in 1990-91 *(Central Government)* to more than *6,500* crore in *2010-11*.

However, it is also noticed that the cost of tax collection for the Income Tax Department is one of the lowest in the world at the rate of less than 60 paise for every Rs. 100 collected as a direct tax.

Evasion and tax avoidance are reported to be very high. It has been estimated that black money is generated at the rate of 50 per cent of the country's' GDP. Because of this, the black money accumulation is of considerable magnitude. It is growing every year. The unaccounted funds are invested into business through

diverse means and add further to the existing funds of black money. A part of it is squandered and wasted lavishly on social functions and on anti- social activities. Besides, the Indian tax system is also accused of (i) discouraging employment (ii) distorting prices (iii) and adversely affecting savings.

SUMMARY

⌘ A tax is a compulsory contribution from a person to the expenses incurred by the state in common interest of all without reference to specific benefit conferred on any individual. ⌘ Taxes are generally classified into direct taxes and indirect taxes.

⌘ Direct taxes means taxes which are not shifted. The incidence and the impact are on the same person. For e.g. income tax. ⌘ Indirect tax is that where the burden can be shifted to others. The tax payer is not a tax bearer. For e.g. sales tax, custom duty.

⌘ Direct tax is progressive but indirect taxes are generally regressive in nature.

⌘ Direct taxes contributed 7% of the GDP in 2010-11. ⌘ **Direct Tax**

- ④ Income tax is a tax on the income of an individual or an entity
- ④ Corporate tax is levied on the incomes of registered companies and corporations
- ④ Taxes which are levied on wealth and capital are mainly estate duty, wealth tax and gift tax.
- ④ Wealth Tax in India is levied on certain specified assets. ⌘ **Indirect Tax**
- ④ Custom duties are levied on exports and imports.
- ④ An excise duty is levied on production and has absolutely no connection with its actual sale.
- ④ Sales tax is a tax on sale of commodities.
- ④ Value added tax is a multistage sales tax with credit for taxes paid on business purchases. At present all states /union territories have successfully implemented VAT. ④ Service tax is imposed on services.

⌘ The tax revenue on account of centre, state and union territories form about 16 per cent (2010-11) of the total national income of India ⌘ Due to tax evasion and avoidance the black money is generated at the rate of 50% of the country's GDP.

UNIT II
SELECTED ASPECTS OF INDIAN ECONOMY

Population refers to the total number of people residing in a place. Thus, population of India means the total number of people living in India. There was a time when growth in population was considered desirable. There are still certain countries (example, Australia), which give incentives to people to have large families and hence have big population of the country. For them, more number of persons is desirable as

It provides work force to produce.
It provides market for the products produced. It may promote innovative ideas.
It may promote division of labor and specialization.

However, there are countries (example, India) for whom more number of persons is not desirable as

There may not be adequate jobs to absorb all additional people.
They put pressure on means of subsistence.
They put pressure on social overheads (hospitals, schools, roads etc.)
They may result in increased consumption and reduced savings and hence slow down capital formation.
They may increase dependency.

Actually, whether a big and growing population is an asset or a liability for the economy depends upon economic condition of country.

2.1 Demographic trends in India

Size of Population: The size of population is determined in terms of number of persons. Considering the present boundary of India (i.e., except Pakistan and Bangladesh), the population in 1901 was 23.84 crores and after one hundred years, in 2001 it was 102.27 crores. Thus, over a period of 100 years, our population has more than quadrupled. In 2011 the population was more than 121 crores.

Rate of Growth: It shows growth rate of population per decade and per annum. During 1901-11, the population grew by 5.74 per cent over the decade or 0.56 per cent per annum. The next decade saw a fall in the growth rate. In fact, there was a decrease in the population and growth rate became negative. Since 1921, population has again started increasing. In fact, year 1921 is known as 'Year of Great Divide' for India's population. The

slow or negative growth during 1901-21 was due to rapid and frequent occurrence of epidemics like cholera, plague, influenza and famines. Since Independence, the rate has not only been positive but crossed the 2 per cent mark. Between 1961-1991, the growth rate has remained above 2 per cent per annum. Only in 1991-2001 decade, the growth rate has come down to 1.97 per cent per annum. According to the provisional figures of Census 2011, the growth rate of population during 2001-2011 decade was 1.64 per cent per annum.

Birth Rate and Death Rate: Birth rate refers to number of birth per thousand of population. Similarly, death rate refers to number of deaths per thousand of population. We see from the table the death rate has declined significantly from 27.4 in 1951 to 8.4 in 2001 and 7.1 in 2010 and birth rate, although has declined but the decline is not so remarkable. Birth rate was 39.9 in 1951, it fell to 25.4 in 2001 and to 21.8 in 2010.

Density of population: Density of population refers to the number of persons per square kilometer

Density of the population before Independence was less than 100. But after Independence, it has increased rapidly from 117 in 1951 to 267 in 1991 to 325 in 2001 and further to 382 in 2011. Thus the pressure of population on land has been rising. The density of population is not the same for all the States; while Kerala, West Bengal, Bihar and U.P. have density higher than the average density, Andhra Pradesh, Himachal Pradesh, Gujrat, Madhya Pradesh, Maharashtra, Karnataka, Orissa, Rajasthan, Sikkim etc. have density lower than the nation's average. This difference could be due to the differences in natural resources endowment, level of development etc. Bihar is the most densely populated state in the country with 1102 persons living per sq. km. followed by West Bengal with 880

Life-expectancy at birth : Life expectancy refers to the mean expectation of life at birth. If death rate is high and/or death occurs at an early age, life expectancy will be low and, it will be high if death rate is low and/or death occurs at an advanced age.

Life expectancy has improved over the years. During 1901-11, life expectancy was just 23 years. It remained below 30 years till the decade 1921-30 and remained below 40 years till the period 1941-50. However, it improved to 50 in 1981 and to 59 in 1991 and further to 62.5 in 2001 and 63.5 years in 2011. Considerable fall in the death rate is responsible for improvement in the life expectancy at birth. This could be one of the reasons for favourable sex ratio for males.

Literacy ratio : Literacy ratio refers to number of literates as a percentage of total population

In 1951, only 27% of males and 9% of females were literate. Thus on an average only 18% of the people of the country were literate. During 1951-2011, there has been considerable improvement in literacy. This is clear from the figures of 2011. In 2011, 82 per cent of males and 65 of females were literate giving an overall literacy rate of 74. Compared with other developed countries, this rate is very low. Illiteracy among 26 per cent of the population is bound to affect the progress of family planning programme. It has been found that literate persons are more responsive to family planning programme than illiterate ones.

2.2 Causes of Rapid Growth of Population

Population generally increases because of

(i) high birth rate.

(ii) relatively lower death rate.

(iii) migration.

growth of population has affected economic growth in India

(i) **Growth of national income :** National income rose by nearly 18 times during 1950-51 to 2011-12, but on account of increase in population by more than 2 times, the per capita income rose by less than five times only. The average annual growth during this period was about 4.9 per cent but could effect an increase of only 3 per cent in per capita income.

(ii) **Food supply :** The total production of food grains increased from 51 million tons in 1951 to about 260 million tons in 2011-12. During the same period population increased from 361 million to more than 1210 million. Consequently, the per capita domestic availability of foodgrains increased from 395 grams to 463 grams signifying a very small increase in per capita availability. As compared to increasing demand for food, per capita availability of food grains is insufficient. If the present trend of increase in population continues demand for food grains might outgrow population unless the increase in population is contained by vigorous family planning programmes. Also, increase in production of food grains has to be brought about more and more through increasing productivity because per capita availability of cultivable area is coming down gradually. In 1951 it stood at 0.33 hectare per capita which came down to less than 0.17 hectare per capita in recent years. A falling land-man ratio has to be compensated by increase in productivity per acre.

(iii) **Unproductive consumers:** With a rapid increase in population, the ratio of children and old persons in total population has a tendency to increase which leads to higher burden of unproductive consumers on the total population. In

India, around 63 per cent of the population is in the age group 15-64 and 37 per cent of the population is under 15 or above 64. That means about 37 percent of the population is dependent on 63 per cent of the population or we say that the dependency rate is more than 57 per cent. An increase in the ratio of unproductive consumers places additional burden on the resources of the family as well as the public utility services like education, health etc. Economists have calculated the burden of dependency in terms of food, education and health. Such dependency load is an important contributing factor to the vicious circle of poverty and under-development in developing countries.

(iv) **Problem of unemployment :** With fast growing population, the labour force increases rapidly and there is a pressure for creating jobs for the growing labour force. In absence of insufficient number of jobs, the number of unemployed people increases.

In India, in successive plans job opportunities have been created but they have fallen short of the required numbers and as a result the backlog of unemployed people has become very large. It is estimated that unemployed and underemployed persons constitute nearly 10 per cent of the labour force in India. Moreover, it is projected that total labour force would increase by 24.5 million in the Twelfth Plan. Jobs would have to be created for these people plus the people in the backlog.

(v) **Capital formation :** It is said that a part of capital formation investment normally goes in maintaining the existing standard of living for the additional population. For example, with a population growth of 2 per cent an investment of around 8 per cent (with a capital output ratio of 4 : 1) of national income would just be able to maintain the existing standard for the additional population. Therefore, for any improvement in the standard of living, the capital investment has to be very large. Thus, for 5 per cent rise in per capita income the resources needed for the economic development will be 20 per cent. Thus to bring about an increase of 5 per cent in per capita income, an investment of 28 per cent (8 per cent for demographic investment and 20 per cent for economic investment) will be needed.

(vi). Ecological degradation: A rapid growth population in India, as in many other countries, has somewhat upset the ecological balance. There is a gradual shrinkage of area covered by forests as also open land. Denudation of forest means serious soil erosion and floods with their adverse consequences on food production.

2.3 Government Measures for solving the Population Problem

National Population Policy, 2000

With a view to encourage two-child norm and stabilizing population by 2046 A.D. the Government adopted the National Population Policy (NPP-2000). The following are the main features of the NPP :

- Address the unmet needs for basic reproductive and child health services, supplies and infrastructure.
- Make school education up to age 14 free and compulsory, and reduce dropouts at primary and secondary school levels to below 20 per cent for both boys and girls.
- Reduced infant mortality rate to below 30 per 1000 live births.
- Reduce maternal mortality ratio to below 100 per 100,000 live births.
- Achieve universal immunisation of children against all vaccine preventable diseases

Maternal Mortality Ratio and Infant Mortality Rate

Maternal Mortality Ratio (MMR) refers to the number of maternal deaths in a given period per 1,00,000 live births. Infant Mortality Rate (IMR) refers to number of babies dying before the age of one, per 1000 live births. As per 2010 data, Infant mortality rate in India is highest for Madhya Pradesh (62) and lowest for Kerala (13).

3.4 Poverty

Poverty is not having sufficient material possessions or income for a person's needs. Poverty may include social, economic, and political elements.

Poverty can be classified into absolute and relative.

Absolute poverty:

- When a household does not have sufficient income to sustain even a basic acceptable standard of living meet basic needs.
- Absolute poverty is the complete lack of the means necessary to meet basic personal needs, such as food, clothing and shelter.
- Absolute poverty thresholds will vary between developed and developing countries.
- The extreme poverty measure now used by the World Bank is the percentage of the population living on less than $1.90 a day (PPP).

Relative poverty

- A level of household income that is considerably lower than the median level of income within a country.
- relative poverty occurs when a person cannot meet a minimum level of living standards, compared to others in the same time and place
- The official UK relative poverty line is household disposable income (adjusted for household size) of less than 60% of median income.

Poverty Estimates: The National Sample Survey Office (NSSO) undertakes surveys from time to time in order to analyze the expenditure pattern of people in rural and urban areas. It uses two types of recall periods – uniform recall period (URP) and mixed recall period (MRP). While the URP uses 30-day recall/ reference period for all items of consumption, MRP uses 365 day recall/reference period for five infrequently purchased non-food items namely, clothing, footwear, durable goods, education and institutional medical expenses.

Causes of Poverty

Economic Causes : Various causes of poverty can be classified under economic, political and social heads. Economic backwardness or inactivity is often the characteristic of the countryside of a developing country like India where majority of the population lives. Agriculture is the main occupation of the rural poor and contributes 14 per cent of the GDP. Yet the income it provides to agricultural workers is substantially below average and almost at the existence level. There are a number of factors which are responsible for low income in the agricultural sector such as small size of land holdings, inadequate irrigation facilities, lack of enough financial resources needed for investment for ensuring development and raising productivity. Thus, productivity in small farms is generally low resulting in very low levels of returns. The condition of landless agricultural laborer is worse. The economic conditions of persons engaged in non-agricultural activities in the rural sector are equally dismal.

Political and Social Causes : Political vested interests are also equally responsible for widespread poverty in the economy. But whereas these interests can be countered by following the right type of policies, social factors responsible for promoting poverty are more severe and are interwoven in the web of society itself. Inhibitions and handicaps arising from caste and religion are hard to overcome and require considerable effort by way of propaganda and education through mass media, reorientation of education system and so on.

Other Causes : Apart from these, other factors such as family size and family composition, poor levels of education and skills, lack of motivation and will to get out of the rut of poverty and misery, the feudalistic system of bonded labour in some parts of the country and so on, are also responsible for depressed standards of living among people.

3.5 Government Programmes for Poverty Alleviation

(1) **The Mahatma Gandhi National Rural Employment Guarantee Scheme (MGNREGS):** The MGNREG Act was notified in 2006 in selected districts and was later extended throughout the country in 2008. It aims at enhancing livelihood security of households in rural areas of the country by providing at least 100 days of guaranteed wage employment in a financial year to every household whose adult members volunteer to do unskilled manual work. During 2012-13, more than 4.39 crore households were provided employment under the scheme.

(2) **Swaran Jayanti Gram Swarozgar Yojana (SGSY) :** This was introduced in April, 1999 as a result of restructuring and combining the Integrated Rural Development Programme (IRDP) and allied programmes and Million Wells Scheme (MWS). It is the only self-employment programme for the rural poor. It aims at bringing the self employed above the poverty line by providing them income generating assets. Upto February, 2012, more than 168 lakh swarojgaries have been assisted. The SGSY had been restructured as the National Rural Livelihoods Mission (NRLM) and now has been renamed as Aajeevika. The NRLM aims at reducing poverty by enabling poor households to access gainful self-employment and skilled wage employment opportunities.

(3) **The Swarna Jayanti Shahari Rozgar Yojana (SJSRY) :** The SJSRY which came into operation from December'97, sub-summing the earlier urban poverty alleviation programmes viz., Nehru Rozgar Yojana (NRY), Urban Basic Services Programmes (UBSP) and Prime Minister's Integrated Urban Poverty Eradication Programme (PMIUPEP). The scheme which was revamped in 2009, aims to provide gainful employment to the urban unemployed or underemployed poor by encouraging the setting up of self-employment ventures or provision of wage employment. A total of more than 4,00,000 beneficiaries have been assisted in 2012-13.

2.6 Unemployment

Roughly every sixth person in the world is an Indian and every third poor person in the world is also an Indian. The current condition show about the reality of the problems of unemployment and poverty which require immediate solution. It has been observed that poverty increases due to increase in the number of unemployed persons.

Every year government try to remove poverty and unemployment by implementing many schemes but due to lack of implementation and lack of excellence in the execution of plans it has not been giving expected results. The increasing unemployment over the years is generally recognized as the basic weakness in the approach of the Government.

Generally a person who is not gainfully employed in any productive activity is called unemployed. Unemployment is a complex phenomenon and takes many forms. The important forms are:

Labour force : Labour force or in other words, the economically active population refers to the population which supplies or seeks to supply labour for production and, therefore, includes both 'employed' and 'unemployed' persons and the labour-force participation rate (LFPR) is defined as the number of persons in the labour force per 1000 persons.

Work-force : Work force is a part of labour force and refers to the population which is employed. Thus work force participation rate (WPR) is defined as the number of persons/ person-days employed per 1000 person/person days.

Unemployed rate : Unemployment rate is defined as the number of persons unemployed per thousand persons in the labour-force.

Measurement of Unemployment: There are three main measures of employment and unemployment.

1. **Usual Status (US) :** This measure estimates the number of persons who may be said to be chronically unemployed. This measure generally gives the lowest estimate of unemployment especially for a poor economy because only a few can afford to remain without work over a long period.
2. **Current Weekly Status (CWS) :** This estimate reduces the reference period i.e. the period for which data is collected to one week. According to this

estimate a person is said to be employed for the week even if he is employed only for a day during that week.

3. **Current Daily Status (CDS) :** The reference period here is a day. It counts every half day's activity status of the respondent over the week. For working out the rate of unemployed person-days the aggregated count of unemployed days during the reference weeks constitutes the numerator and the aggregated estimate of the total number of labour force days constitutes the denominator.

2.7 Types of Unemployment

i) **Voluntary unemployment :** In every society, there are some people who are unwilling to work at the usual wage rate and there are some people who get a continuous flow of income from their property or other sources and need not work. All such people are voluntarily unemployed. Voluntary employment may be a national waste of human energy, but it is not a serious economic problem.

ii) **Frictional Unemployment :** Frictional unemployment is a temporary phenomenon. It may result when some workers are temporarily out of work while changing jobs. It may also result when the work is suspended due to strikes or lockouts. To some extent, frictional unemployment is also caused by imperfect mobility of labour. We may also say that frictional unemployment is due to difficulties in getting workers and vacancies together.

iii) **Casual unemployment :** In industries, such as construction, catering or agriculture, where workers are employed on a day to day basis, there are chances of casual unemployment occurring due to short- term contracts, which are terminable any time.

iv) **Seasonal unemployment :** There are some industries and occupations such as agriculture, the catering trade in holiday resorts, some agro-based activities like sugar mills and rice mills, in which production activities are seasonal in nature. So they offer employment for
only a certain period of time in a year. People engaged in such type of work or activities may remain unemployed during the off-season. We call it seasonal unemployment.

v) **Structural Unemployment :** Due to structural changes in the economy, structural unemployment may result. It is caused by a decline in demand for production in a particular industry, and consequent disinvestment and reduction in its manpower requirement. In fact, structural unemployment is a natural concomitant of economic progress and innovation in a complex industrial economy of modern times.

vi) **Technological unemployment :** Due to the introduction of new machinery, improvement in methods of production, labour-saving devices, etc., some workers tend to be replaced by machines. Their unemployment is termed as technological unemployment.

vii) **Cyclical unemployment :** Capitalist biased, advanced countries are subject to trade cycles. Trade cycles - especially recessionary and depressionary phases cause cyclical unemployment in these countries. During the contraction phase of a trade cycle in an economy, aggregate demand falls and this leads to disinvestment, decline in production and unemployment. The solution for cyclical unemployment lies in measures for increasing total expenditure in the economy, thereby pushing up the level of effective demand. Easy money policy and fiscal measures such as deficit financing may help. Since cyclical phase is temporary, cyclical unemployment remains only a short- term phenomenon.

viii) **Chronic unemployment :** When unemployment tends to be a long- term feature of a country it is called chronic unemployment. Underdeveloped countries suffer from chronic unemployment on account of the vicious circle of poverty, lack of developed resources and their under utilisation, high population growth, backward, even primitive state of technology, low capital formation, etc.

ix) **Disguised unemployment :** So far, the types of unemployment which we have discussed above are all related to open unemployment. Apart from open unemployment we have disguised unemployment. Disguised unemployment commonly refers to a situation of employment with surplus manpower in which some workers have zero marginal productivity so that their removal will not affect the volume of total output. Disguised unemployment in the strict sense, implies underemployment of labour. To illustrate, suppose a family farm is properly organized and four persons are working on it. If, however, two more workers are employed on it and there is no change in output, we may say that these two workers are disguisedly

unemployed. This kind of unemployment is a common feature of under developed economies especially of their rural sector. In short, overcrowding in an occupation leads to disguised unemployment. It is a common phenomenon in an over populated country.

The various causes responsible for widespread unemployment in India are as follows :

1. **Growth without adequate employment opportunities :** As economy grows usually employment also grows. But in India, most of the time, the economic growth has been inadequate and adequate number of jobs could not be created. In fact, for almost three decade 1950-80, GDP growth rate was as low as 3.5 per cent per annum. Such a low rate of growth did not push many jobs in the market. Since 1980s however, growth has accelerated to around 5-6 per cent but job creating capacity of the economy has not improved much.

2. **Growing Population :** Population has increased at a very fast pace since Independence but jobs have failed to keep pace with the population.

3. **Inappropriate technology :** India is a labour surplus and capital scarce economy. Under such circumstances, labour-intensive industries should have been given preference. But not only in industry but also in agriculture producers are increasingly substituting capital for labour. This has hindered the growth of job opportunities.

Inappropriate education system : The education provided in India has not much practical utility. The students receiving such education, even very high one, fail to get appropriate jobs

If we look into the data on unemployment in India we find that the unemployment rates have first decreased then risen and then fallen again. This is clear from the following Table:

Table 9 : Unemployment Rate

Current Daily Status Basis

1983	9.22
1993-94	6.06
1999-00	7.31
2004-05	8.28
2007-08	8.10
2009-10	6.60

Recent Approach of government to solve the problem of unemployment.

- The Eleventh Plan targeted generation of additional employment opportunities in services and manufacturing, in particular, labour intensive manufacturing sectors such as food processing, leather products, footwear and textiles and in services sectors such as tourism and construction and village and small scale enterprises.
- It underlined the need to have a proper policy on developing skills. Therefore, a National Policy on Skill Development was formulated in 2009 which envisions empowering all individuals through improved skills, knowledge, and nationally and internationally recognized qualification to gain access to decent employment and ensure India's competitiveness in the global market

  Educated youth are expected to join the labour force in increasing numbers during the 12^{th} Plan. The 12^{th} Plan therefore lays great stress on improving job opportunities in manufacturing sector, agro-processing, supply chains, rural infrastructure and services. It lays high stress on skill building as it is critical for achieving faster, sustainable and inclusive growth on the one hand and for providing decent employment opportunities to the growing young population on the other.
- It has identified priority sectors for employment generation and skill development. These include, Textiles and Garments, Leather and Footwear, Gems and Jewellery, Food Processing Industries, Handlooms and Handicrafts, Machine Tools, IT Hardware and Electronics, Telecommunications equipment, Aerospace Shipping, Defense Equipment, etc.

2.8 Infrastructure

Energy

Energy is an important input for most of the production processes and consumption activities. It plays a crucial role in the economic development. Economic growth and demand for energy are positively co-related. A study shows a 3 per cent rise in industrial production in the world is accompanied by a 2 per cent increase in energy consumption. A similar relation is also observed in India. India is both a major energy producer and consumer. India currently ranks as the world's fifth largest energy producer, accounting for about 4 per cent of the world's total energy production. It is also the world's fourth largest energy consumer after U.S.A., China and Russia accounting for about 6 per cent of the world's total energy consumption.

There is a distinction between primary energy resources and final energy resources. When coal is consumed for generating electricity and electricity is consumed by industry, we call coal as primary energy resource and electricity as the final one. Coal, petroleum products and natural gas are both primary resources and final resources as they are consumed directly as well as indirectly, while electricity is, by and large, the only final energy resource Major users of electricity are industry (37 per cent), domestic (25 per cent), agriculture (21 per cent) and commercial establishment (10 per cent). We shall concentrate on electricity as it is the main mover of manufacturing industry.

Electricity: Electricity or power is the most important source of commercial energy. Over the planning era, the production and consumption of electricity has increased tremendously. Our total installed capacity of generating power was around 2300 Mega Watt (MW) in 1950-51. It increased to 117800 MW in 2000-01 and further to about more than 2,36,000 MW in 2011-12. Thus, over a period of 61 years, there has been more than 100 times increase in the installed capacity. We are roughly adding 4000-5000 MW every year.

Sources of Electricity: There are 5 major sources of electricity (1) water (2) coal (3) oil (4) gas and (5) radio active elements like uranium, thorium and plutonium. Electricity generated from water is known as hydro-electricity. Electricity generated from coal, oil and gas is called thermal electricity and electricity generated from radio-active elements is called atomic energy. Of the present capacity, 66 per cent is in the thermal (coal, gas and oil), 17 per cent in the hydel 2 per cent is in the nuclear and 15 per cent is in the other renewable (wind, bio-gas, waste etc.) sectors. In terms of generation of power, thermal plus non conventional energy source is contributing 72.5 per cent, hydel around 12.5 per cent and nuclear 3 per cent and others are contributing 12 per cent.

(2) Oil prices and inflationary pressure : Since 1973, oil prices have been rising in the international market. During 1973-2013, the Organisation of Petroleum Exporting Countries (OPEC) has increased the prices many folds. This has contributed to the inflationary pressure in India. Mineral oil is presently the major source of energy for transport, industry and agriculture. It is also used as household fuel. Therefore, rising oil prices has led to rising general prices in India.

(3) Growing oil imports bill : Since 1973, India's oil imports bill has increased substantially. In 1973-74 India's oil import bill was ` 1100 crore. It increased to ` 10,816 crore in 1990-91 and to a record level of more than ` 6,00,000 crore in 2011-12. Petroleum, oil and lubricants (POL) constitute around 35 per cent of our import bill. The oil import bill is also responsible to a great extent for the existing large balance of trade gap.

(4) Transmission and distribution losses : One of the major problems faced by the power companies are transmission and distribution (T&D) losses. The T&D losses are very high in many of the SEB systems. These losses include substantial amount of theft of power. National average of this loss is more than 20 per cent while in many states it is more.

(5) Sick SEBs : Many SEBs have become financially sick. A large portion of these losses is accounted for by almost free supply of power to agriculture. Besides, operational inefficiencies, high cost structure, lower power tariffs and large overdues have made SEBs sick.

(6) Operational inefficiency : Most of the thermal power plants are operating inefficiently. Plant load factor (PLF) measures the operational efficiency of a thermal plant. If total generating capacity is say 600 billion kilo-watt hours (kwh) and we are producing only 300 billion kwh, plant load factor here is 50. Plant load factor varies across the regions. It is lowest in Eastern region (62 per cent in 2012-13) and highest in Southern region (81 per cent in 2012-13). If we consider SEBs, central sector and private sector, we find that PLF was 65 per cent in SEBs, 79 per cent in central sector and 79 per cent in the private sector in 2012-13.

(7) Inadequate electrification : Till date, many of villages are not electrified. In many villages, there are a very few houses which are lighted. This is sad considering the fact that more than 60 years have passed since we got Independence and still our rural areas are without electricity.

Transportation

Today, along with energy, transport is the basic infrastructural requirement for industrialization. Transport provides a useful link between production centers, distribution areas and ultimate consumers. Important means of transport are railways, roads, water transport and air transport.

Railways : Indian Railways, is the fourth largest railway network in the world. It has been contributing to the industrial and economic landscape for over 165 years. There

are two main segments of railways - freight and passenger. The freight segment accounts for about 70 per cent of revenues and passengers thirty per cent of revenues. The total route length of railways is about 64,600 kilometers. Out of which about 21 thousand kilometers is electrified. During 2011-12 it carried more than 8200 million of passengers and about 970 million tonnes of freight traffic. Railways however, faces the following problems:

(1) The existing technology of both electric and diesel locomotive is very old. The entire system
is in urgent need of modernization.

(2) The railway network is smaller and inadequate vis-à-vis the requirements of the economy.

(3) The railways is facing the problem of financial crunch. The conventional methods of increasing the net revenue, like raising of tariffs and expenditure control are inadequate for generating the levels of investment required.

(4) Because of social responsibilities, railways is forced to operate a number of unremunerative
lines and suffer heavy losses. Often, essential goods like foodgrains, fruits and vegetables
have to be carried at a loss.

(5) Railways also suffers from over crowding and poor passenger services.

(6) There is need to develop safety measures on an extensive scale.

Road

Road: The Indian road network is one of the largest networks in the world. At the beginning of the first plan, India had about 4,00,000 kms of total road length out of which 1,57,000 kms was surfaced road. Today, India has a network of 4.69 million kilometer

Problems of road transport : Following problems are faced in the case of road transport:

(1) The road length is inadequate considering the size of the country.

(2) A number of areas, particularly interior areas and hilly tracts remain to be linked with roads.

(3) Large tracts of rural roads are mud roads which cannot be used for plying heavy traffic.

(4) A number of urban roads are also poorly maintained. This is due to constraints of financial
resources, organizational inadequacies, procedural delays, shortage of essential materials etc.

(5) Most of the State Road Transport Corporations are running on heavy losses. This is because

of rising cost of operations, inefficiency in operations and corruption

Water Transport

Water transport: Water transport can be divided into inland water transport and shipping. Shipping can again be divided into coastal shipping and overseas shipping. India has about 14500 km of navigable waterways which comprise rivers, canals, backwaters creeks etc. About 50 million tonnes of cargo is being annually moved by inland water transport. Over the years, the importance of this mode of transport has declined considerably due to expansion of rail and road transport and navgational inadequacies. The government approved the Inland Water Transport Policy which includes a number of incentives to encourage private sector participation in inland water transport.

Problems faced by Indian ports : The main problems are low productivity and poor competitiveness. Major factors contributing to these are:

- Operational constraints such as frequent breakdown of cargo handling equipment due to obsolescence.
- Inadequate dredging and container handling facilities.
- Inefficient and non optimal deployment of port equipment.
- Lack of proper coordination in the entire chain.

Communication

Communication means transmission of information. For the development of industries, commerce and trade in the country, communication is very necessary. The important means of communications are the postal services, telephone services, tele printers, radio and television etc. Telephone, tele-fax and e-mail have been gradually evolving and telex and telegraph are getting out of fashion.

Postal services

India's postal system dates back to 1837 and today our postal network is the largest network in the world. Today, we have about 1.55 lakh post offices and out of which around 90 per cent are in the rural areas. On an average, one post office serves 7814 persons and 21.23 sq. km area. Postal services suffer from many weaknesses such as inadequate number of post offices, use of outdated techniques, delays in reaching of posted material etc. A number of steps have been taken for resolving these problems. Such as speed post, business

post, express parcel post, media post, speed post passport etc. services have been introduced. With a view to improve the speed and volume of transactions, a range of e-enabled services such as electronic money order (eMO), e-payment and Instant Money Order (IMO) have been started. To provide better services, mechanization and computerization of postal operations is being progressively introduced. Automatic mail processing centers (AMPC) have been set up at various places for faster processing of mails. E-post services were started in 2001 in some states. Under e-bill post, customers are able to pay multiple utility bills at post office counters. They are now being upgraded for multiple messaging to make them useful for corporate houses. This is a major initiative to provide user-friendly services to its vast customer base. 'Logistics Posts', 'Retail Post Services' are other new services which are now being provided. Post offices are also providing a number of financial products such as saving bank

Telecommunications

Communications all over the world has progressed rapidly and the most important factor accounting for increased communication has been the development of telecommunications which include (i) the telephone service, and (ii) the telex service. At the time of Independence, India had a total of only 321 telephone exchanges with about 8200 working connections. There were only 338 long distance public call offices and 3324 telegraph offices. The growth of telecommunications has gained momentum after Independences and by October 2012 India had more than 935 million connections (wireline and wireless). As on December 2011, more than 5.8 lakh villages were connected using a village public telephone (VPT). Thus, 98 per cent of villages in India have been covered by the VPTs. India's telephone network is the second largest in the world (after China) with a tele density
(number of phones per 100 persons) of 76.75 per cent. While tele density in rural areas is 40.6

per cent, the urban tele density shot up to 159 per cent in October, 2012.

Health

The general health standard in India is quite low. This is quite inevitable as nearly 30% of the population lives below the poverty line. These people do not have nutritional diet, adequate medical care and hygienic conditions. As a result, the overall health conditions are poor in India. Prevalence of child under weight and child malnutrition is widespread across states. It is not that nothing has been done on the health front but they are far from satisfactory. The following table shows trends in health care in India since Independence. Since diseases like, T.B., malaria, gastrointestinal infections are related with unhygienic sanitation, efforts have been intensified in providing hygienic conditions and opening new hospitals and strengthening existing hospitals especially in rural areas. In order to bridge the gap between existing health infrastructure and to

provide affordable and equitable health care, a large number of schemes have been started in India.
Important among them are:
_ The National Rural Health Mission was started in 2005 to provide accessible, affordable, and quality health services to rural areas. Seeing its success in the rural areas, the government aims to extend it to towns also by converting it in National Health Mission.
_ Accredited Social Health Activists (ASHAs) have been selected and trained in health care for various villages.
_ Janani Suraksha Yojana was started to bring down maternal mortality rate in India and under the scheme the number of beneficiaries has increased from 90 lakh in 2008-09 to nearly 109 lakh in 2011-12.
_ Pradhan Mantri Swasthiya Yojana was launched with the objectives of correcting regional imbalances in the availability of reliable health care services in the country.
_ Janani Shishu Karyakaram (JSSK) is a new initiative launched in 2011 for mother and child care. Under the programme, free entitlements are given to pregnant women and sick new borns for cashless delivery, drugs and consumables, diet etc.
_ National Vector borne Disease control Programme is being implemented for prevention and control of vector borne diseases such as malaria, filarsis, kala-azar, dengue, etc.
_ Apart from these, there are Rogi Kalyan Samitis, Village Health and Sanitation Committees, Mobile Medical Units, AYUSH services, Integrated Child Development Services, Rajiv Gandhi Scheme for Empowerment of Adolescent Girls - sabla. National Programme for health care

Education

Education plays an important role in the overall development of a human being and a society. It is the single most important instrument for social and economic transformation. Therefore, stress on imparting education has been given in our Constitution which says education should be free for children below 14 years of age. Under the various plans, education facilities have been expanded at all levels in India and as a result, not only the literacy rate has risen but the percentage of children availing school education has also increased. India, now, has the one of the largest education systems in the world. Eighty-four per cent of rural habitation in India now have a primary school located within a distance of 1 kilometre. The National Policy on Education (NPE) was made in 1986 and further modified in 1992. It emphasized on: universal access and enrolment; universal retention of children up to 14 years of age and a substantial improvement in the quality of education.
NPE had set a goal of expenditure on education at 6 per cent of the GDP. As against this target, the actual expenditure of central and state governments was over 4 per cent of

GDP in 2011-12. Right of Children to Free and Compulsory Education Act (RTE Act) 2009, has

Problems of India's Education system

The education system in India suffers from the following problems:

1. Unplanned expansion of higher education deterioration of academic standards, low level of research etc.
2. Inadequate number of institutions which can impart education through correspondence or in the evening.
3. Low standard of education, deterioration of academic standards, outdated curriculum, low level of research, etc.
4. Large number of unemployed educated people.
5. Large-scale migration of educated people to the developed western countries.
6. Lack of infrastructure in many rural schools – absence of rooms, blackboard, teachers, water etc.
7. Neglect of primary education.
8. Regional disparities.
9. Large number of dropouts especially among females

Unit 3
Money and Banking

3.1 Money

Meaning

It is very difficult to define money in exact sense. This is because, there are various categories of assets which possess the attributes of money. Many things such as clay, cowry shells, tortoise shells, cattle, slaves, rice, wool, salt, porcelain, stone, gold, iron, brass, silver, paper and leather etc. have been used as money. Traditionally, money has been defined on the basis of its general acceptability and its functional aspects.

3.1.1 Functions of Money

As a unit of account: Money is a common measure or common denominator of value. The value in exchange of all goods and services can be expressed in terms of money. We can say that it is the general language we use to estimate prices and compare them.

It would be possible to use any good as a unit of account. Let us consider it as mobile phones. That means the prices of tables, chairs, books and groceries would all be estimated in terms of the number of mobile phones required to buy them. In theory it is possible, practically it is not possible. Which ever the commodity that can be used as medium of exchange but not as accurately as money is. Money is the only way to exchange the goods with accurate estimate of the value. Following are the functions of money.

As a medium of exchange : The fundamental role of money in an economic system is to serve as a medium of exchange or as a means of payment. In the barter system goods are exchanged for other goods. This system prevailed throughout the world in the olden times. This system suffered from many shortcomings. For exchange of goods, persons desiring to exchange goods must specifically want those goods what others offered in exchange. Money has removed this difficulty. Now a person A can sell his goods (say clothes) to another person B for money and then can use to buy goods (say Mobile phone) he wants from others who have these goods.

As standard of deferred payments: Money is a unit in terms of which debts and future transactions can be settled. Thus loans are made and future contracts are settled in terms of money.

As store of value : Money being a permanent place of purchasing power holds command over goods and services all the times-present and future. Money is a convenient means of keeping any income which is surplus to immediate spending needs and it can be exchanged for the required goods and services at any time. Thus it acts as a store of value.

Directs economic trends: Money directs idle resources into productive channels and there by affects output, employment, consumption and consequently economic welfare of the community at large. When it is converted into investment in various sector it contribute for economic growth.

As encouragement to division of labor: In a money economy, different people tend to specialize in the different goods and through the marketing process, these goods are bought and sold for the satisfaction of multiple wants. In this way, occupational specialization and division of labor are encouraged by the use of money.

Smoothens transformation of savings into investments: In a modern economy, savings and investments are done by two different sets of people - households and firms. Households save and firms invest. Households can lend their savings to firms. The mobilization of savings can be done through the working of various financial institutions such as banks. Money so borrowed by the investors when used for buying raw materials, labor, factory plant etc. becomes investment. Saved money thus can be channelized into any productive investment.

3.2 Commercial Banks

A modern industrial society cannot be run by self-financing of entrepreneurs. Some institutional assistance is necessary to mobilize the savings of the community and to make them available to the entrepreneurs

From a broader social point of view, banks act as a bridge between the users of capital and those who save but cannot use the funds themselves. The idle resources of the community are thus activated and brought to productive use.

Besides, the banking system has capacity to add to the total supply of money by means of credit creation. The bank is a dealer in credit - its own and other people's. It is because of the ability to manipulate credit that banks are used extensively as a tool of monetary policy

3.2.1 Role of Commercial Banks

Banks play a very useful and dynamic role in the economic life of every modern state. Their economic importance may be viewed in the followed points:

1 A developing economy needs a high rate of capital formation to accelerate the speed of economic development. But the economic development depends upon the rate of savings. Banks offer facilities for keeping savings and thus encourage the habits of thrift in the society.

2 Not only do the banks encourage savings but they also mobilize savings done by several households and make them available for production and investment to the entrepreneurs in various sectors of the economy. Without banks these savings would have remained idle and would not have been utilized for productive and investment purposes.

3 Allocation of funds or economic surplus among different sectors, users or producers so as to make maximum social return and thus to ensure optimum utilization of savings is another important function performed by the banks. However, it may be mentioned, that commercial banks do not always work and allocate resources in the way that maximizes production or social welfare. For example, before nationalization in 1969, the commercial banks in India neglected socially highly desirable sectors such as agriculture, small scale industries and weaker sections of the society. Therefore, it was thought necessary to nationalize them so that they should allocate resources in socially desirable directions.

(4) By encouraging savings and mobilizing them from public, banks help to increase the aggregate rate of investment in the economy. Banks not only mobilize saved funds from the public, but they also themselves create deposits or credit which serve as money. The new deposits are created by the banks when they lend money to the investors or other users. These deposits are created by the banks in excess of the cash reserves they obtain through deposits from the public. These days, the bank deposits, especially demand deposits are as much good money as the currency issued by the government or the central bank. This creation of credit, if it is used for productive purposes greatly enlarges production and investment and thus promotes economic growth.

3.2.2 Functions of Commercial Bank

The functions of a bank can be summarized as follows:

(a) Receipt of deposits: A bank receives deposits from individuals, firms, and other institutions. Deposits constitute the main resources of a bank. Such deposits may be of different types. Deposits which are withdrawable on demand are called demand or current deposits, others are called time deposits. Savings deposits are those from which withdrawals are not restricted as regards the amount and the period. Deposits withdrawable after the expiry of an agreed period are known as fixed deposits. Interest paid by banks is different for each kind of deposit - highest for fixed deposits and lowest or even nil for current deposits.

(b) Lending of money: Banks lend money mainly for industrial and commercial purposes. This lending may take the form of cash credits, overdrafts, loans and advances, or discounting of bills of exchange. Interest charged by banks on such lending varies according to the amount and period involved, social priority-nature of security offered, the standing of the borrower, etc.

(c) Agency services: A bank provides various services to consumers, such as : (i) collection of bills, promissory notes and cheques; (ii) collection of dividends, interests, premiums, etc.; (iii) purchase and sale of shares and securities; (iv) acting as trustee or executor when so nominated; and (v) making regular payments such as insurance premiums.

(d) General services : A modern bank performs many services of general nature to the public, e.g. (i) issue of letters of credit, travellers cheques, bank drafts, circular notes; etc. (ii) safe keeping of valuables in safe deposit vaults; (iii) supplying trade information and statistics; conducting economic surveys; and (iv) preparation of feasibility studies, project reports, etc. Banks in some foreign countries also underwrite issue of shares and make loans for long-term purposes.

With development in technology, new methods of banking have been evolved. Now people have the benefit of banking anytime and anywhere. Important tools of modern banking are Automatic Telling Machine (ATM), Real Time Gross Settlement (RTGS) and the National Electronic Funds Transfer (NEFT).

An automated or automatic teller machine (ATM) also known as an automated banking machine (ABM) is a computerized telecommunications device that enables the clients of a financial institution to perform financial transactions without the need for a cashier, Nowadays, transactions which are bulk and repetitive in nature are routed through electronic clearing service (ECS). India has two main electronic funds settlement systems for one to one transactions: the Real Time Gross Settlement (RTGS) and the National Electronic Funds Transfer (NEFT) systems.

Real Time Gross Settlement (RTGS): RTGS system is a funds transfer mechanism where transfer of money takes place from one bank to another on a 'real time' and on 'gross' basis. This is the fastest possible money transfer system through the banking channel. Settlement in 'real time' means payment transaction is not subjected to any waiting period. The transactions are settled as soon as they are processed. In India, the Reserve Bank of India (India's Central Bank) maintains this payment network. Core Banking enabled banks and branches are assigned an Indian Financial System Code (IFSC) for RTGS and NEFT purposes. This is an eleven digit alphanumeric code and unique to each branch of bank. The first four letters indicate the identity of the bank and remaining seven numerals indicate a single branch. This code is provided on the cheque books, which are required for transactions along with recipient's account number.

National Electronic Fund Transfer (NEFT): The National Electronic Fund Transfer (NEFT) system is a nation-wide system that facilitates individuals, firms and corporates to electronically transfer funds from any bank branch to any individual, firm or corporate having an account with any other bank branch in the country. NEFT requires Indian financial system code (IFSC) to perform a transaction.

3.3 RBI(Reserve Bank of India)

A Central Bank is one which constitutes the apex of the monetary and banking structure of a country and which performs, in the national economic interest, the following functions :

1. The regulation of currency in accordance with the requirements of business and the general public.

2. The performance of general banking and agency services for the State.
3. The custody of cash reserve of the commercial banks.
4. The custody and management of the nation's reserves of international currency.
5. The granting of accommodation, in the form of rediscounting or collateral advances to commercial banks, bill brokers and dealers.
6. The clearance arrangements among banks; and
7. The control of credit in accordance with the needs of business with a view to carrying out broad monetary policy adopted by the State.

The above is quite comprehensive but, in addition, central banks perform additional functions to meet the specific requirements of the country. Broadly speaking, a central bank has three objectives, namely monetary stability, including stability of domestic price levels, maintenance of the international value of the nation's currency and issue of currency.

3.3.1 Role of RBI

The Reserve Bank of India (RBI) is the Central Bank of India and occupies a pivotal position in the Indian economy. Its role is summarized in the following points:

- The RBI is the apex monetary institution of the highest authority in India. Consequently, it plays an important role in strengthening, developing and diversifying the country's economic and financial structure.
- It is responsible for the maintenance of economic stability and assisting the growth of the economy.
- It is India's eminent public financial institution given the responsibility for controlling the country's monetary policy.
- It acts as an advisor to the government in its economic and financial policies, and it also represents the country in the international economic forums
- It also acts as a friend, philosopher and guide to commercial banks. In fact, it is responsible for the development of an adequate and sound banking system in the country and for the growth of organized money and capital markets.

- India being a developing country, the RBI has to keep inflationary trends under control and to see that main priority sectors like agriculture, exports and small scale industry get credit at cheap rates.

- It has also to protect the market for government securities and channelize credit in desired directions.

3.3.2 Functions of RBI

The Reserve Bank of India being the Central Bank of India performs all the central banking functions. These are:

(i) Issue of currency: The RBI is the sole authority for the issue of currency in India other than one rupee coins and notes and subsidiary coins, the magnitude of which is relatively small.

(ii) Banker to the government: As a banker to the government, the RBI performs the following functions:

(a) It transacts all the general banking business of the Central and State Governments. It accepts money on account of these governments and makes payment on their behalf and carries out other banking operations such as their exchange and remittances.

(b) It manages public debt and is responsible for issue of new loans. For ensuring the successes of the loan operations it actively operates in the gilt-edged market and advises the government on the quantum, timing and terms of new loans.

(c) It also sells Treasury Bills on behalf of the Central Government in order to wipe away excess liquidity in the economy.

(d) The RBI also makes advances to the Central and State Governments which are repayable within 90 days from the date of advance.

(e) The RBI also acts as an adviser to the government not only on policies concerning banking and financial matters but also on a wider range of economic issues including those in the field of planning and resource mobilization. It has a special responsibility in respect of financial policies and measures concerning new loans, agricultural finance and legislation affecting banking and credit and international finance.

(iii) Banker's Bank : The RBI has been assigned with extensive power to control and supervise commercial banking system under the Reserve Bank of India Act, 1934 and the Banking Regulation Act, 1949. All the scheduled banks are required to maintain a certain minimum of cash reserve ratio with the RBI against their demand and time liabilities. This provision enables the RBI to control the credit position of the country.

The RBI provides financial assistance to scheduled banks and state cooperative banks in the form of discounting of eligible bills and loans and advances against approved securities.

The RBI also conducts inspection of the commercial banks and calls for returns and other necessary information from banks.

(iv) Custodian of Foreign Exchange Reserves : The RBI is required to maintain the external value of the rupee. For this purpose it functions as the custodian of nation's foreign exchange reserves. It has to ensure that normal short-term fluctuations in trade do not affect the exchange rate. When foreign exchange reserves are inadequate for meeting balance of payments problem, it borrows from the IMF.

The RBI has the authority to enter into exchange transactions on its own account and on account of government. It also administers exchange control of the country and enforces the provisions of Foreign Exchange Management Act.

(v) Controller of Credit : Credit plays an important role in the settlement of business transactions and affects the purchasing power of people. The social and economic consequences of changes in the purchasing power are serious, therefore, it is necessary to control credit. Controlling credit operations of banks is generally considered to be the principal function of a central bank. The RBI, like any other Central Bank, possesses power to use almost all qualitative and quantitative methods of credit controls. (For details discussion on instruments of credit controls please refer to the topic Indian Monetary Policy).

(vi) Promotional Functions : Apart from the traditional functions of a Central Bank, the RBI also performs a variety of developmental and promotional functions. It is responsible for promoting banking habits among people and mobilizing savings from every corner of the country. It has also taken up the responsibility of extending the banking system territorially and functionally. Initially, it had also taken up the responsibility for the provision of finance for agriculture, trade and small industries. But now these functions have been handed over to NABARD, EXIM Bank and SIDBI respectively. The Reserve Bank is responsible for over all credit and monetary policy of the economy.

(vii) Collection and publication of Data : It has also been entrusted with the task of collection and compilation of statistical information relating to banking and other financial sectors of the economy.

3.4 Monetary Policy

Monetary Policy is usually defined as the Central Bank's policy related to the control of the availability, cost and use of money and credit with the help of monetary measures in order to achieve specific goals. In the Indian context, monetary policy includes those decisions of the government and the Reserve Bank of India which directly influence the volume and composition of money supply, the size and distribution of credit, the level and structure of interest rates, and the effects of these monetary variables upon related factors such as savings and investment and determination of output, income and price.

The broad concerns of monetary policy in India have been -

(a) To regulate monetary growth so as to maintain a reasonable degree of price stability and

(b) To ensure adequate expansion in credit to assist economic growth;

(c) To encourage the flow of credit into certain desired channels including priority and the hitherto neglected sectors; and

(d) To introduce measures for strengthening the banking system and creating institutions for filling credit gaps.

Monetary policy is implemented by the RBI through the instruments of credit control. Generally two types of instruments are used to control credit.

These are (i) quantitative or general measures and (ii) qualitative or selective measures. The quantitative measures are directed towards influencing the total volume of credit in the banking system without special regard for the use to which it is put. Selective or qualitative instruments of credit control, on the other hand, are directed towards the particular use of credit and not its total volume.

3.4.1 **Quantitative or General Measures**

Quantitative weapons have a general effect on credit regulation. They are used for changing the total volume of credit in the economy. Quantitative measures consist of (a) Bank Rate Policy (b) Open Market Operations and c) variable reserve requirements.

Bank Rate Policy: It is the traditional method/weapon of credit control used by a Central Bank. When the Central Bank(RBI) wishes to control credit and inflation in the economy, it increases the Bank Rate. Increased Bank Rate increases the cost of borrowings of the commercial banks who in turn charge a higher rate of interest from their borrowers. This means the price of credit will increase. This will affect the profits of the business community who will feel discouraged to borrow. As a result, the demand for credit will go down. Decreased demand for credit will slow down investment activities which in turn will affect production and employment. Consequently, income in general will be down, people's purchasing power will decrease and aggregate demand will fall and prices will fall down. This in turn will lead to a cumulative downward movement in the economy.

On the other hand, if the Central Bank wishes to boost production and investment activities in the economy, it will decrease the Bank Rate. Decreasing the Bank Rate will have a reverse effect. As regards Bank Rate in India, it was 10 percent in 1981, 12 percent in 1991, 11 per cent in 1997 which was reduced (in stages) to 6 per cent in 2003. However, it was increased to 10.25 per cent (in stages) in 2013 in order to control inflationary trends in India.

(b) Open market operations: Open market operations involve direct sales and purchases of securities and bills in the market by the Central Bank on its own initiative to control the volume of credit. When the Central Bank sells securities in the open market, other things being equal, the cash reserves of the commercial banks decrease to the extent that they purchase these securities. In effect, the credit-creating base of commercial banks is reduced and hence credit contracts. On the other hand, open market purchases of securities by the Central Bank lead to an expansion of credit made possible by strengthening the cash reserves of the banks. Thus, on account of open market operations, the quantity of money in circulation changes. This tends to bring about changes in money rates. An increase in the supply of money through open market operations causes a down ward movement in the interest rates, while a decrease of money supply raises interest rates. Change in the rate of interest in turn tends to bring about the desired adjustments in the domestic level of prices, costs, production and trade.

(c) Variable reserve requirements: The Central Bank also uses the method of variable reserve requirements to control credit. There are two types of reserves which the commercial banks are generally required to maintain (i) Cash Reserve Ratio(ii) Statutory Liquidity Ratio(SLR). Cash reserve ratio refers to that portion of total deposits which a commercial bank has to keep with the Central Bank in the form of cash reserves. Statutory liquidity ratio refers to that portion of total deposits which a commercial bank has to keep with itself in the form of liquid assets viz - cash, gold or approved government securities. By changing these ratios, the Central Bank controls credit in the economy. If it wants to discourage credit in the economy, it increases these ratios and if it wants to encourage credit in the economy, it decreases these ratios. Raising of the reserve rates will reduce the surplus cash reserves of the banks which can be offered for credit. This will tend to contract credit in the system. Reverse will be effects of reduction in the reserve ratio requirements reflected in the expansion of the bank credit. At present, (July 2013) cash reserve ratio is 4 per cent and statutory liquidity ratio is 23 per cent for entire net demand and time liabilities of the scheduled commercial banks.

(**d) Repo Rate and Reverse Rate:** In addition to these, there are tools of Repo and Reverse Repo Rates. Repo rate is the rate at which our banks borrow rupees from RBI. Whenever the banks have any shortage of funds they can borrow it from RBI. RBI lends money to bankers against approved securities for meeting their day to day requirements or to fill short term gap. A reduction in the repo rate will help banks to get money at a cheaper rate. When the repo rate increases borrowing from RBI becomes more expensive. At present, Repo rate is 7.25 per cent. [July, 2013]

Reverse Repo rate is the rate at which Reserve Bank of India (RBI) borrows money from banks. An increase in Reverse repo rate can cause the banks to transfer more funds to

RBI due to this attractive interest rates. It can cause the money to be drawn out of the banking system. At present Reverse Repo rate is 6.25 per cent. [July, 2013]

3.4.2 Qualitative or Selective Measures : Qualitative or selective measures are generally meant to regulate credit for specific purposes. The Central Bank generally uses the following forms of credit control -

(a) Securing loan regulation by fixation of margin requirements : The Central Bank is empowered to fix the margin and thereby fix the maximum amount which the purchaser of securities may borrow against those securities. Raising of margin controls the borrowing capacity of the security holder. This is a very effective selective control device to control credit in the speculative sphere without, at the same time, limiting the availability of credit in other productive fields. This device is also useful to check inflation in certain sensitive spots of the economy without influencing the other sectors.

(b) Consumer credit regulation : The regulation of consumer credit consists of laying down rules regarding down payments and maximum maturities of installment credit for the purchase of specified durable consumer goods. Raising the required down payment limits and shortening of maximum period tend to reduce the demand for such loans and thereby check consumer credit.

(c) Issue of directives : The Central Bank also uses directives to various commercial banks. These directives are usually in the form of oral or written statements, appeals, or warnings, particularly to curb individual credit structure and to restrain the aggregate volume of loans.

(d) Rationing of credit : Rationing of credit is a selective method adopted by the Central Bank for controlling and regulating the purpose for which credit is granted or allocated by commercial banks.

(e) Moral suasion : Moral suasion implies persuasion and request made by the Central Bank to the commercial banks to co-operate with the general monetary policy of the former. The Central Bank may also persuade or request commercial banks not to apply for further accommodation from it or not to finance speculative or non-essential activities. Moral suasion is a psychological means of controlling credit; it is a purely informal and milder form of selective credit control.

(f) Direct Action: The Central Bank may take direct action against the erring commercial banks. It may refuse to rediscount their papers, and give excess credit, or it may charge a penal rate of interest over and above the Bank Rate, for the credit demanded

Unit 4

Inflation and Deflation

4.1 Inflation

Inflation refers to a persistent upward movement in the general price level. It results in a decline of the purchasing power. According to most economists inflation does not occur until price increase averages less than 5% per year for a sustained period. Inflation can broadly be of the following types:

(i) Demand-pull inflation : In a market there is interaction between the flow of money and flow of goods and services. When more money used to purchase relatively less quantity of goods and services the excess of demand relative to supply increases the prices of goods and services. Such inflation, as a result of increased money expenditure, is called demand-pull inflation. In other words, when demand for goods and services is more than their supply, their prices rise. Such price rise is called demand pull inflation.

(ii) Cost-push inflation : Cost push inflation refers to a situation where prices persistently rise because of increase in cost required to produce goods. Cost push inflation results when factors of production especially wage earners try to increase their share of the total product by raising their prices. A rise in factor prices leads to a rise in the total cost of production and consequently a rise in the price level. This may result in an inflationary spiral. Inflation once set in motion due to phenomenon of cost push in one industry or sector spreads throughout the economy. For example, due to rise in wages in the steel industry, prices of steel may rise and this will raise the prices of vehicles, machines, etc., using steel as input. The rise in prices of vehicles may in turn raise the cost of transport and manufactured goods. The cost of agriculture may also rise due to high prices of tractors. Ultimately, food and raw material prices will also go up leading to higher cost of living. Higher cost of living will further push wage rates. Cost push inflation is much more difficult to control than demand pull inflation. This is because cost push inflation is not susceptible to direct control. Often the demand pull inflation precedes the cost push inflation. When the former sets in, there is an increasing demand for factors of production; the prices of these also rise, leading to rise in general prices.

(iii) Stagflation : The combined phenomenon of demand-pull and cost-push inflation is found in many countries, both the developed and the developing. One of these situations is in the form of stagflation under which economic stagnation, in the form of a low rate of growth, combines with the rise in general price level. In the developing countries, this happens when aggregate demand increases at a fast rate due to high

public expenditure and expansion of credit money, aggregate cost increases due to higher wages resulting from pressure from labour organisations thus combining cost-push effect with the demand pull inflation. Such inflationary situations when unchecked by appropriate monetary and fiscal measures, may lead to galloping or hyper-inflation leading to price increase of even 40% to 100% every year.

Stagflation in India has been interpreted to mean that the economy is growing slowly or stagnating (i.e. GNP is either increasing slowly or remaining constant or even declining) and at the same time experiencing a high rate of inflation. In India during 1991, partly as a result of large budget deficits resulting in rapid expansion in money supply and partly due to supply shocks delivered by Gulf war in 1990-91 and sharp increase in the procurement prices of foodgrains the high rate of inflation emerged in the economy. Along with high inflation rate, rate of industrial and economic growth, was very low. Thus, during the period 1991-94 high inflation occurred in India while the economy was stagnating. Therefore, it is correct to say that India was experiencing stagflation during the period.

4.1.2 Price Trends in India

In India, the variation in prices are measured in terms of Wholesale Price Index (WPI) . They can also be measured in terms of Consumer Price Index (CPI).

Wholesale Price Index (WPI) : The Wholesale Price Index (WPI) is the price of a representative basket of wholesale goods. The WPI focuses on the price of goods in the wholesale market. The Indian WPI which was earlier updated weekly is now updated on a monthly basis. This basket comprises 676 items which carry different weights. WPI measures headline inflation i.e. it includes the entire set of commodities. This is different from core inflation in which the commodities which have volatile prices (like food and fuel)are not considered. WPI, however, does not include services and non-tradable commodities. Currently, in India WPI series with base 2004-05 is being used to assess price changes.

Consumer Price Index (CPI) : A consumer price index (CPI) measures changes in the price level of consumer goods and services purchased by households. It reflects the cost of living for a homogeneous group of consumers. There are 4 CPI indices in India. These are CPI for industrial workers, CPI for agricultural labour, CPI for rural labour and CPI for urban non-manual employees.

4.1.3 Causes of Inflation in India

A general price rise can take place either as a result of rise in aggregate demand or a failure of aggregate supply or both. Increase in public expenditure, deficit financing, and rapid growth of population can be mentioned as demand factors and erratic agricultural growth, agriculture price policy, inadequate rise in industrial production and upward revision of administered prices etc. can be mentioned as supply factors which have led to inflationary price rise in India.

(i) **Increase in public expenditure :** Public expenditure has risen from 18.6 per cent of GDP in 1961 to around 28 per cent in 2012-13 (current prices). With a rise in national income and also rapid growth of population an increase in public expenditure is unavoidable. But the spectacular rise in the public expenditure is not justifiable. Approximately 40 per cent of the government expenditure in India is on non-developmental activities. No doubt, defence and maintenance of law and order are essential for the stability of the society. At the same time, it must not be forgotten that due to their unproductive nature, expenditure on these activities results in inflationary price rise. The government expenditure on non-developmental activities, by putting purchasing power into the hand of its employees, creates demand for goods and services, but it does nothing whereby their supply could increase. Under these circumstances the general price level shows an inevitable tendency to rise.

(ii) **Deficit financing :** Deficit financing means financing of budget deficits (shortages) by borrowing from the banks or printing of more currency. The Government of India has frequently resorted to deficit financing in order to meet its developmental expenditure. A small dose of deficit financing is helpful in tiding over the gap between public revenue and public expenditure and making available funds for the growth of the economy but a large dose and that too in a period of relatively slow growth turns out to be inflationary. This happens because by financing the deficit the government puts purchasing power in the hands of people but it does nothing for creating real resources for the economy at least in near future. In India from plan to plan, the recourse to deficit financing has been increasing which has led to inflationary trends in the country.

(iii) **Erratic agricultural growth :** The Indian agriculture largely depends on monsoons and thus crop failures due to drought have been regular feature of agriculture in this country. In the years of scarcity of food grains not only price of food articles increases but the general price level also rises.

(iv) **Agricultural price policy of the Government :** The government has been pursuing a policy of price support to the agriculturists. For this, it announces the price at which it would be buying agricultural products. This ensures certain minimum price to the farmers. This policy benefited farmers in India but this has been a major contributory factor to the inflationary price rise in the country.

(v) **Inadequate rise in industrial production :** Performance of the industrial sector, particularly in the period 1965 to 1985, has been rather disappointing. Over the 20 years period, industrial production increased at a modest rate of 4.7% per annum. The performance of essential consumer goods sector which includes industries like oil, food manufacturing, textiles, weaving, apparel and footwear was particularly

disappointing. Moreover, in the ten years period from 1991-92 (except for 1995-96), the industrial sector registered slow growth of around 6 per cent per annum. In the wake of a large expansion of the money supply creating big demand for these goods, inadequate increase in their production pushed up their prices.

(vi) Upward revision of administered prices : There are a number of important commodities for which price level is administered by the government. Many of these commodities are produced in the public sector. The government keeps on raising prices from time to time in order to cover the losses in the public sector which often arise due to inefficiency and unimaginative planning. This policy results in cost push inflation.

(vii) Other factors : Besides the above factors, the following have also contributed to inflationary trends in price in India: Failure of the government to fully bring within the ambit of taxation the increasing income of the people, large scale tax evasion and avoidance, increasing reliance on indirect taxes, black marketing and hoarding of essential commodities, unused capacity in industries, high capital-output ratio, shortage of essential raw materials, low surplus from public sector undertakings, infrastructural bottleneck and rising prices of imports

4.1.4 Measures to control Inflation

Since inflation is a phenomenon where money income is rising faster than the real goods and services, the measures to check inflation should either be of a check on the increase in money incomes or making available more of real goods and services. The various measures can be studied under three main heads – monetary and fiscal measures, control over investment and other measures. In India also, in order to check inflation, various monetary, fiscal and administrative measures are taken in the years of high inflation (e.g. 2006-07, 2008-09). These have helped in containing inflation to some extent. These measures are discussed below:

(i) Monetary measures : Monetary measures are applied to check the supply of currency and credit. These measures consist of quantitative measures (open market operations, statutory reserve requirements and Bank Rate) and qualitative measures (margin requirements, moral suasion etc.). When the Reserve Bank of India wants to control inflation it uses any one or more of the above measures. Thus, it may sell government securities in the open market. By issuing (i.e. by selling) government securities, the government takes away liquidity (i.e. cash etc.) from the people. This lowers the balances with the banks; which in turn will reduce their capacity to create credit or lend money for investment purposes. This will reduce liquidity in the economy and bring the prices under control. Similarly, by raising Bank Rate or statutory reserve rates the RBI controls liquidity and credit and ultimately prices. The extent to which these will be effective will depend on the intensity of the investment demand. The raising of the statutory reserve ratio is very widely used measure in India but the effect of this measure also depends

on the banking habits of the people. However in those countries where banking is not fully spread and all money is not quickly banked, the effect will be much smaller.

There are several selective measures of credit, such as variable interest rates, variable margin requirement, ceiling on certain types of loans, minimum and maximum rates of interest etc. There is need for selective control of credit when the rise in prices is confined to some commodities only e.g. necessities of life.

(ii) Fiscal measures : These are the measures taken by the government with regard to taxation, expenditure and public borrowings. Taxes determine the size of the disposable income in the hands of the public. In the case of inflation, a proper tax policy will be to avoid tax cuts, or to introduce some increase in the existing rates so as to reduce the purchasing power in the hands of the people and thus reduce the pressure of demand on prices. The fiscal tools have been extensively used as tools to control inflation in India. The progressive income tax system, control over public expenditure, introduction of new types of taxes, improving profits of public sector units, etc. are all meant to control inflation in the country.

(iii) Control over investment : Controlling investments is also considered necessary because, due to the multiplier effect, the initial investment leads to large increase in income and expenditure and the demand for both the consumer and capital goods goes up speedily. Therefore, it is necessary that the resources of the community should be employed for investment which does not have the effect of increasing inflation.

(iv) Other measures : These measures can be divided broadly into short term and long term measures. Short term measures can be in regard to public distribution of scarce essential commodities through fair price shops. There may also be control over movement of commodities from one state to another. In India whenever shortage of basic goods has been felt, the government has resorted to imports so that inflation may not get triggered. It has also resorted to rationing of essential goods in times of shortages. The long term measures will require accelerating economic growth especially of the wage goods which have a direct bearing on the general price and the cost of living. Some restrictions on present consumption may help in improving saving and investment which may be necessary for accelerating the rate of economic growth in the long run.

4.2 Deflation

Deflation is a state in which the prices are falling and thus the purchasing power of money is increasing. Deflation is just the opposite of inflation.

Deflation is a serious economic issue that can exacerbate a crisis and turn a recession into a full-blown depression. When prices fall and are expected to drop in the future, businesses and individuals choose to hold on to money rather than spend or invest. This leads to a drop in demand, which in turn forces businesses to cut production and sell off inventories at even lower prices.

4.2.1 Measures to control Deflation

Monetary Measures

Central banks buy treasury securities in the open market and, in return, issue newly created money to the seller. This increases the money supply and encourages people to spend those money.

Central banks can lower the target interest rate on the short-term funds that are lent to and among the financial sector. If this rate is high, it will cost the financial sector more to borrow the funds needed to meet day-to-day operations and obligations. Short-term interest rates also influence longer-term rates, so if the target rate is raised, long-term money, such as mortgage loans, also becomes more expensive. Lowering rates makes it cheaper to borrow money and encourages new investment using borrowed money. It also encourages individuals to buy a home by reducing monthly costs.

Fiscal Measures

If governments cut taxes, more income will stay in the pockets of businesses and their employees, who will feel a wealth effect and spend money that was previously earmarked for taxes. One risk of lowering taxes during a recessionary period is that overall tax revenues will drop, which may force the government to curtail spending and even cease operations of basic services. There has been conflicting evidence as to whether or not general and specific tax cuts actually stimulate the real economy. (For more, see: Do Tax Cuts Stimulate The Economy*?*)

Unit V

National Income

National income is the money value of all the final goods and services produced by a country during a period of one year. National income consists of a collection of different types of goods and services of different types. Since these goods are measured in different physical units it is not possible to add them together. Thus, we cannot state national income is so many millions of metres of cloth, so many million litres of milk, etc. Therefore, there is no way except to reduce them to a common measure. This common measure is money. The value of all goods and services produced is measured in money.

5.1 Basic Concepts of National Income

(1) Gross Domestic Product (GDP) : Gross domestic product is the money value of all final goods and services produced in the domestic territory of a country during an accounting year. Domestic territory is defined to include the following:

(i) Territory lying within the political boundaries, including territorial waters of the country.

(ii) Ships and aircrafts operated by the residents of the country between two or more countries.

(iii) Fishing vessels, oil and natural gas rigs, and floating platforms operated by the residents of the country in the international waters or engaged in extraction in areas in which the country has exclusive rights of exploitation.

(iv) Groups, missions and military establishments of the country located abroad.

(2) GDP at Constant Prices and at Current Prices: GDP can be estimated at current prices and at constant prices. If the domestic product is estimated on the basis of the normal prices it is called gross domestic product at current prices. Thus, when we say that GDP of India at current prices in 2009-10 is ` 65,50,271 crores, we are measuring GDP on the basis of the prices prevailing in 2009-10. On the other hand, if GDP is measured on the basis of some fixed prices, that is prices prevailing at a point of time or in some base year it is known as GDP at constant prices or real gross domestic product. Thus when we say that GDP in 2009-10 is ` 44,93,743 crores at 2004-05 prices, we are measuring GDP on the basis of the prices prevailing in 2004-05.

(3) GDP at Factor Cost and GDP at Market Price : The contribution of each producing unit to the current flow of goods and services is known as the net value added. GDP at factor cost is estimated equal to the cost involved in their production (factor cost). GDP at market price includes indirect taxes and excludes the subsidies given by the government. Therefore, in order to arrive at GDP at factor income we must subtract indirect taxes from and add subsidies to GDP at market price.

(4) **Net Domestic Product :** While calculating GDP no provision is made for depreciation allowance. It is a matter of common knowledge that capital goods like machines, equipment, tools, buildings, tractors etc., get depreciated during the process of production. After some time these capital goods need replacement. A part of capital is therefore, set aside in the form of depreciation allowance. When depreciation allowance is subtracted from gross domestic product we get net domestic product.

In brief

NDP = GDP - depreciation

(5) Gross National Product (GNP) : It has already been seen that whatever is produced within the domestic territory of a country in a year is its gross domestic product. It, however, includes, the contribution made by non-resident producers by way of wages, rent, interest and profits. The non-residents work in the domestic territory of some other country and earn factor incomes. For example, Indian residents go abroad to work. Indian banks are functioning abroad. Indians own property in foreign countries. The income of all these people is the factor income earned from abroad. In other words, it is factor income earned from abroad by the residents of India by rendering factor services abroad. Similarly, factor services are rendered by non- residents within the domestic territory of India. Net factor income from abroad is the difference between the income received from abroad for rendering factor services and the income paid for the factor services rendered by non-residents in the domestic territory of a country.

Gross national product is defined as the sum of the gross domestic product and net factor incomes from abroad. Thus in order to estimate the gross national product of India we have to add net factor income from abroad i.e., income earned by Indian residents abroad minus income earned by non-residents in India to form the gross domestic product of India.

In brief GNP = GDP + NFIA (where NFIA is the net factor income from abroad).

(6) Net National Product (NNP) : It can be derived by subtracting depreciation allowance from GNP.

NNP = NDP + NFIA

5.2 Methods of Measuring National Income

(i) Value Added Method (alternatively known as Product Method);

(ii) Income Method; and

(iii) Expenditure Method.

(i) Value Added Method or Product Method or Output Method

This method involved in measuring money value of all the goods and services produced or measures the contribution of each producing enterprise in the domestic(national) territory(area) of the country. This method involves the following steps:

(a) Identifying the producing enterprise and classifying them into industrial sectors according to their activities.

(b) Estimating net value added by each producing enterprise as well as each industrial sector and adding up the net value added by all the sectors.

All the producing enterprises are broadly classified into three main sectors namely:

(1) Primary sector which includes agriculture and allied activities;

(2) Secondary sector which includes manufacturing units and

(3) Tertiary sector which include services like banking, insurance, transport and communications, trade and professions.

These sectors are further divided into sub-sectors and each sub-sector is further divided into commodity group or service-group.

For calculating the net product of the industrial sector we need to know about gross output of the sector, the raw materials and intermediate goods and services used by the sector and the amount of depreciation. For an individual unit, we subtract from the value of its gross output, the value of the raw material and intermediate goods and services used by it and, from this, we subtract the amount of depreciation to get net product or value added by each unit. Adding value-added by all the units in one sub-sector, we get value-added by the sub-sector. Again adding value-added or net products of all the sub-sectors of a sector we get value-added or net product of that sector. For the economy as a whole, we add net products contributed by each sector to get Net Domestic Product. If the information regarding the final output and intermediate goods is available in terms of market prices we can easily convert it in terms of factor costs by subtracting (or adding as the case may be) net indirect taxes to it. If we add or subtract net income from abroad we get Net National Product at factor cost which is nothing but National Income.

Care should be taken to include the value of the following items :

(a) Own account production of fixed assets by government, enterprises and households.

(b) Production for self-consumption.

(c) Imputed rent of owner occupied houses.

Care should also be taken not to include sale of second-hand machines because they were counted as a part of production in the year in which they were produced. However, brokerage and commission earned by the dealers of second- hand goods are a part of production and hence included while calculating total value-added. There is a difference of opinion among the national income accountants regarding raw materials, intermediate goods and depreciation. For example, a question arises whether government services are final (because they add to satisfaction) or intermediate (because they are essential for economic activity). In India, we treat them as final services but in Soviet Union (now called Commonwealth of Independent States) these are treated as intermediate services. Similarly, it is very difficult to ascertain the actual amount of depreciation because a fall in the value of capital stock depends upon many factors which are difficult to measure.

Moreover, large areas of production activities are excluded for varying reasons. Their net products cannot be valued either because there is no acceptable way of valuing them (which is true in the case of services of housewives or self-services in homes or services of friends) or because of the difficulty of securing data of the subsistence producing units particularly in underdeveloped countries.

The product method, thus, gives information about the industrial origins of national income. Additionally, net income from abroad should also be included or subtracted to get a true picture of national income.

Example:

Industry A sells raw jute to industry B for Rs. 10 crore
Industry B Sells jute fibre to Industry C for Rs. 25 crore
Industry C sells canvas ropes to final consumers for Rs. 50 crore

The total output is Rs. 10 crore+ Rs. 15 crore + Rs. 25 crore = 50 crore. To avoide double counting the intermediate products are ignored,

(ii) Income Method

In this method all the incomes are considered for measuring national income. It may be from wages, salaries, bonus, commission, non-labour income includes dividends, undistributed profits of corporations before taxes, interest, rent, royalties and profites of unincorporated enterprises and of government enterprises.

However, normally, it is difficult to separate labour income from capital income because in many instances people provide both labour and capital services. Such is the case with self-employed people like lawyers, engineers, traders, proprietors etc. In economies where subsistence production and small commodity production is dominant most of the incomes of people would be of mixed type. In sectors such as agriculture, trade, transport etc. in underdeveloped countries (including India), it is difficult to differentiate between labour element and capital element of incomes of the people. In order to overcome this difficulty a new category of incomes, called mixed income is introduced which includes all those incomes which are difficult to separate. Care has to be taken to see that transfer incomes do not get included in national income. In this context it is worthwhile to note that personal income which is income of household sector should not be confused with national income. While personal income includes transfer payments, national income does not. Similarly, illegal incomes, windfall gains, death duties, gift tax and sale proceeds of second-hand goods are not included while calculating national income.

Net income from abroad need not be added separately since the incomes received by people include net foreign incomes as well. But if national income is calculated not from incomes received by the people but from data regarding incomes paid out by producers then net income from abroad would have to be added separately because incomes paid by producers would total

to domestic income. To arrive at national income, net income from abroad should be added to domestic income.

(iii) Expenditure Method

The various sectors - household sector, business sector and government sector either spend their incomes on consumer goods and services or save a part of their incomes or we can say that they spend a part of their incomes on non-consumption goods (or capital goods).

Total expenditure in an economy consists of expenditure on financial assets, on goods produced in preceding periods, on raw materials and intermediate goods and services and on final goods and services produced in the current period.

Expenditure on financial assets which are produced and owned within the country is excluded but expenditure on financial assets of foreign countries is included in national expenditure. However, only the net expenditure i.e., the difference between expenditure on foreign financial assets by residents and expenditure on the country's financial assets by non-residents or foreigners is incorporated. This difference is also called net foreign investment. Goods produced in preceding years are also excluded from national income because they have been accounted for in the national incomes of the periods when they were produced. Similarly, expenditure on raw materials and intermediate goods and services are excluded because otherwise there would be double counting of some of the items included in the national income. Government expenditure on pensions, scholarships, unemployment allowance etc. should be excluded because these are transfer payments.

Thus, only expenditure on final goods and services produced in the period for which national income is to be measured and net foreign investment are included in the expenditure method of calculating national income.

Expenditure on final goods and services is broadly classified into expenditure on consumer goods and service (also called consumption expenditure) and expenditure on capital goods (also called investment expenditure). Consumption expenditure is classified into private consumption expenditure of the household sector and government consumption expenditure; and investment expenditure is classified into private investment expenditure by business sector and investment expenditure by government. To the total domestic investment, we add net foreign investment in order to arrive at national investment. Thus, the aggregates resulting from the expenditure method measured at market prices are as follows:

Gross national expenditure = Consumption expenditure + net domestic investment + net foreign investment + replacement expenditure (i.e., expenditure on replacement investment).

Net national expenditure = Consumption expenditure + net domestic investment + net foreign investment.

Net domestic expenditure = Consumption expenditure + net domestic investment.

All the three methods mentioned above should ideally lead to the same figure of national income and therefore national income of a country should be measured by these methods separately to get a three dimensional view of the economy. This helps the government to analyse the level of production and economic welfare in the economy, to analyse stability and growth of the economy and to formulate appropriate economic policies of the government. Moreover, each method provides a check on the accuracy of the other methods. However, it is easier said than done. Because of lack of proper and reliable data it is very difficult to estimate national income by each method separately. This is especially so in underdeveloped economies.

As a matter of fact, countries like India are unable to estimate their national income wholly by one method. The contributions of different sectors to the total national income are estimated by different methods. Thus, in agricultural sector net value added is estimated by the production method, in small scale sector net value added is estimated by the income method and in construction sector net value added is estimated by the expenditure method.

Income method may be most suitable for developed economies where people properly file their income tax returns. With the growing facility in the use of the commodity flow method of estimating expenditures, an increasing proportion of the national income is being estimated by the expenditure method.

Estimation of the national income of a country is not an easy task. Appropriate and completely reliable data for accomplishing this work is not available even in developed countries. The following problems require particular mention:

(1) Presence of a large non-monetized sector

(2) Lack of appropriate and reliable data

(3) Problem of double counting

(4) Problem of transfer payments

(5) Difficulties in classification of working population

(6) Unreported illegal income

5.3 Growth of National Income

(i) The real national income of India has increased at an annual average rate of about 4.9 per cent based on 2004-05 prices during 1950-51 to 2009-10.

There are two distinctive phases of economic growth in India since Independence: 1950-51 to 1980-81 and 1980-81 to 2009-2010. During the period 1950-1951 to 1980-81, growth in GDP was 3.2 per cent and during 1980-81 to 2009-10, growth in GDP was 6.6 per cent per annum. If we consider the period between 1991-92 and 2009-10, GDP growth rate substantially increased to 7.3 per cent per annum.

Colonial past, restrictive trade policy, licensing system, inward looking foreign policies, too much stress on public sector and socialistic society, anti-market and anti-competition attitude of the State, and vagaries of nature are some of the reasons for very poor performance of the economy during 1950-51 to 1980-81

Acceleration in economic growth since 1980 was attributable to several factors. Expansionary macro economic policies, economic reforms including trade liberalization and deregulation of industries especially since 1991, reasonably well established social and legal framework, well developed higher education system, improvement in infrastructural facilities, change in attitude of national leadership, adoption of pro-market policies, well fostered entrepreneurial skills and improvement in science and technology etc. all led to improving the rate of growth of Indian economy.

Restructuring measures by domestic industry, overall reduction in domestic interest rates, improved profitability, a benign investment climate amidst strong global demand and commitment based fiscal policy have led to real GDP growth averaging 9 per cent per annum during 2004-08.

Economic growth decelerated in 2008-09 to 6.7 per cent. The global financial crisis and consequent economic recession in developed economies have been major factors in India's economic slow down. The deceleration of growth in 2008-09 was spread across almost all the sectors.

In 2009-10, a fast and strong turnaround has been noticed. The economy grew by 8.4 per cent in 2009-10 and 2010 -11. In 2011-12, it is expected to grow by 6.9 per cent. With growth in real GDP at 8.4 per cent in 2009-10 and 2010-11, the economy moved closer to the pre crisis levels. The decomposition of growth in 2010-11 indicated that it was relatively broad based across the major sub-sectors in industry and services, besides the rebound in agriculture.
Slow-down of growth in 2011-12 can be attributed to weak industrial growth, high and sticky inflation and unfavourable global economic environment.

How a country is performing can be judged in a two ways- by comparing its performance with other countries and by comparing its performance with the targets set by it.

A broad comparison with a few of the developing countries is shown in the following Table:

Table 1: Real GDP Growth –Select Countries (per cent)

Country	1960s	1970s	1980s	1990s	2000-06	2010
Brazil	5.9	8.5	3.0	1.7	3.1	7.5
China	3.0	7.4	9.8	10.0	9.5	10.4
India	4.0	2.9	5.6	5.7	7.0	9.6
Korea	8.3	8.3	7.7	6.3	5.2	6.3
Thailand	7.8	7.5	7.3	5.3	5.0	7.8

(World Development Indicators, World Bank)

As can be seen in the Table, India's rate of growth is improving. In fact, India now ranks among the top ten fastest growing countries in the world along with China, Vietnam, South Korea, Malaysia Thailand, Singapore, among others.

Plan-wise study of growth of real income in India, also indicates an encouraging fact that although the annual rate of increase in national income was pretty low during the first three decades of planning, it rose to 5.6 per cent per annum during the eighties, around 5.7 per cent per annum during the nineties and more than 7 per cent per annum during 2000-10.

Table 7 shows the targeted and the achieved rates of growth during various plans.

Table 2 Growth Rate in the Five Year Plan

(per cent per annum)

Plan	Target	Actual
I (1951-56)	2.1	4.4
II (1956-61)	4.5	4.0
III (1961-66)	5.6	2.6
IV (1969-74)	5.7	3.1
V (1974-79)	4.4	4.9
VI (1980-85)	5.2	5.4
VII (1985-90)	5.0	5.5
VIII (1992-97)	5.6	6.7
IX (1997-2002)	6.5	5.3
X (2002-2007)	8.0	7.6
XI (2007-2012)	9.0	7.7 (P)

(P) - provisional

Source: Eleventh Plan Document (vol I), Economic Survey (2011-12)

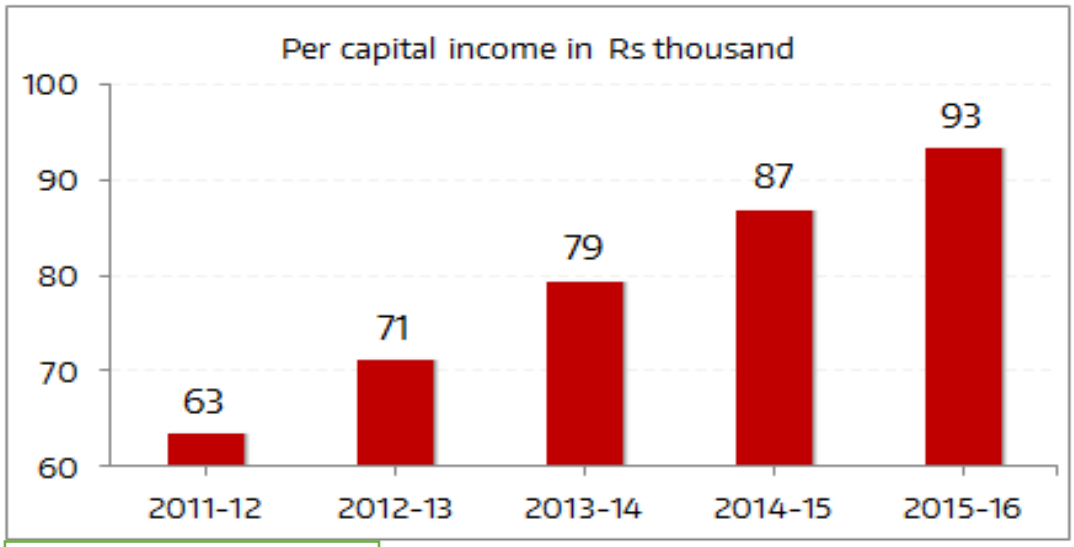
5.4 Per Capita Income

Per capita income calculated by taking national income in the aggregate and dividing it by the total population. It indicates national standard of living. It can be known as income per person in country. India's per capita net national product i.e., during the last 60 years of planning has increased at a rate of about 3 per cent per annum. This is modest performance by all means. The rate of increase in per capita net national product was not only conspicuously low but despite 60 years of economic planning, was still unsteady and erratic.

There were years when the rate of growth of per capita income was quite impressive, but there were also years in which rate of growth remained static or declined. It has been noticed that per capita growth rate in India to a large extent depends upon the agricultural growth rate. In the years of good harvests resulting from good monsoons, there has been growth in the per capita income. But when there are crop failures resulting from bad monsoons, the per capita income has also not grown much.

It is to be noted that during 1950-51 to 1989-90, growth in GDP per capita per annum was 1.6 per cent per annum. Economic reforms undertaken since 1991 have helped in pushing the growth rate of per capita income. It accelerated to 5.8 per cent per annum during 1990-91 to 2009-10.

Fig. per capita income from the year 2011-12 to 2015- 2016



Source: Mospi, Economictimes.com

Syllabus of BBA 2ndnd Year

Sr. No.	**Unit 1: Import Export Management Overview**	**Lectures Required**
1	Concept of import export management, Features of export management, Need for export management, • Process of export management, Function of export manager.	11
Sr. No.	**UNIT II: Foreign Trade Policies and Export Promotion**	**Lectures Required**
2	Objectives of foreign trade policies, Export import policy(latest) • Export promotional organizations in India, EPCS, Commodity board, STC, FIEO, Chamber of commerce, IIP, ITPO, • Facilities for unit in EOUs, EPZs.	11
Sr. No.	**Unit III: International Trading Environment**:	**Lectures Required**
3	Meaning and definition & Importance of international trade, • Factor influencing international trade • Problem in international trade	11
Sr. No.	**Unit IV: Export Product Planning**	**Lectures Required**
4	Product planning strategies, Pricing strategies, Product mix, Product branding • Product packaging, Product Labelling strategy.	10
Sr. No.	**Unit V: Import -Export Procedure and Documentation**:	**Lectures Required**
5	Stages in import procedure, Import document-letter of credit, documentary bill, bill of exchange and bill of lading. • Export procedure: Registration, pre-shipment, post shipment procedure. • Export documents: related to goods, shipment, payment, inspection and excisable goods.	11

Unit 1: import export management overview:

Concept of import export management

Export is the selling of goods or services from one country to another. Import is when one country buys goods or services from another. Exports and Imports are two critical components which maintain the economic well being of a country.

Exports and Imports establish a relation between the two countries. This is how important Exports and Imports are. The Export/Import Management course imparts knowledge in such a way that one can identify foreign markets, product development, payments, financial processes and documentation. Export/import management, operations management, international supplies are areas in which export and import management graduates work.

Export management means conducting the export activity in an orderly, efficient and profitable manner. Since the heart of each business is marketing, export management can be termed as export marketing management. Because if needs to be managed efficiently so that the export should increase and exporter should get more profit and importer should get more satisfaction. Therefore, export management activity is growth oriented and dynamic in nature.

Export marketing management and domestic marketing management are two aspects of the same coin i.e. total marketing management.

However, export marketing management is more difficult and complicated as compared to domestic marketing due to several factors such as, three faced competition, varied regulations of different countries, language, etc. If requires systematic approach for comprehensive oversea, marketing research: therefore, export management involves the study of foreign markets, requirements, of foreign buyers, potential marketing opportunities and using and them tactfully for large-scale exporting.

Export management is basically planning, organizing, coordinating and controlling all activities relating to export of goods and services to other counties. It involves various activities such as production of exportable good, collection of orders from foreign buyers and their execution, publicity in abroad, adoption of sales promotion techniques, price fixation and looking after various procedures and formalities relating to exporting of goods.

It is rightly said that export management involves functions and activities undertaken by the department/ division of a large manufacturing enterprise. The scope of export management is vast as everything concerned with exporting comes within the scope of export management. It is also argued that export management means what an export manager does.

1.2 DEFINITIONS OF THE EXPORT MANAGEMENT

The term export of management is rather difficult to define precisely as it is dynamic in scope. Secondly, standard definition of the term export management is not available as it is an applied subject. Here, the principles of management are applied to the management of export trade/ marketing activities. However, it is possible to note some simple definition of export management. Such definitions are as noted below:

1. Export management means managing export marketing activity efficiently, smoothly and in an orderly manner.
2. Export management means finding at opportunities for marketing goods & services in foreign markets and exporting such opportunists for the benefit of an exporting firm, subject to existing export rules and regulations.
3. Export management is one specific area of business management and it is concerned exclusively with exporting goods abroad. It is concerned with international marketing activities and operations.
4. Export Management means planning, organizing, coordinating and control export efforts or activities to achieve desired export objectives smoothly and with continuance.
5. According to B. S. Bathor, "Export Marketing includes the management of marketing activities for products across the national boundary or a country".

1.3 NEED FOR EXPORT MANAGEMENT

Increase in exports provides several benefits to exporter as well as the Nation, Export sector has been termed as a priority sector for Indian Economy. It is treated as an engine of economic growth and an instrument for employment generation. Government has made every effort to increase export, and raise the volume and value of exports. Better export, performance leads to do industrial development, economic development, foreign Exchange reserves, favorable balance of payment etc. All these benefits will be available only when exports are made at large scale.

In the early 1960s the slogan "Export or Perish" was coved by the then Prime Minster of India Pandit Jawaharlal Nehru. It is applicable even to present economic condition of India with a little change i.e. "Globalization or Perish". Therefore, export promotion is rightly heated as a national challenge.

Therefore, export management is receiving top priority in the country to take effective steps in the right direction for economic growth and prosperity.

We can discuss the need for export management at two different levels.

(A) At the National level.

(B) At the Business level.

Need For Export Or Export Management at international level.

Earning foreign Exchange-

Export management enables the country to earn foreign exchange. The foreign exchange can be utilized for following purposes.

Import of consumer good

Imports of Raw materials, spares and components Import of capital goods and technology Servicing of External Debts.

International Relations-

Export helps to develop international ties with importing countries due to the following reasons.

a) The international trade brings together the exporters and importers of various countries.
b) Trade talks take place between nations at international forums like WTO.
c) Also, trade agreements are singed between Governments of participating counties.

Balance of payments-

A country's external economic strength depends upon its balance of payment position. Naturally, every country would like to have a strong and favorable balance of payments position since exports bring in foreign exchange, it helps a country to solve and improve its Balance of payments position.

Reputation in the World-

Exports bring reputation and goodwill for a nation in the international markets. For instance:

a) Japan commands reputation for electronic products.
b) India has goodwill for handicrafts including germs and jeweler.
c) Germany is famous for engineering of goods.

Employment-

Exports help to generate employment in the country Export facilitates.

a) Direct Employment in the export sector
b) Indirect Employment in the supporting sectors such as banking, insurance, transport etc.

Research and Development-

In international markets, quality of products is of at most importance. Therefore, government provides assistance to exporters to undertake R&D. R&D helps to: a) Reduce costs

b) Develop new products
c) Improve quality of existing products

The fruits of R&D benefit the consumers not only in the overseas markets but also in the domestic market.

Regional Development-

Exports facilitate regional development of instance, about 1/3rd of India's exports are from small sector. The small units are located throughout India. (for instance, in India, The maximum number of small units, is located in the industrially backward state of Uttar Pradesh). Therefore, export sector contributes, towards regional development of a nation.

Optimum use of Resources-

Exports facilitate, optimum use of resources in the country, such as

a) Physical resources such as materials, machines, etc.
b) Capital resources
c) Man power

For instance, Gulf countries have enough supply of petroleum, which is exported, thereby making optimum use of resources.

Standard of Living-

Exports increase demand, which leads to higher production and distribution. Increase in production and distribution generates more employment. Increase in employment leads, to higher purchasing power with the people. Therefore, people can enjoy new and better products, which improves standard of living

Economic Growth-

Due to export, the demand increases. Increase in demand leads to higher production. Higher production increases the GDP of the country which leads to economic growth.

Spread Effect-

Due to increases in export trade in the country service sector also expand, like banking, transport, etc. Similarly other ancillary industries are established to support the export activities.

Need For Export or Export management At Business Level

1. Export Obligation

It means the firm which intends to import capital goods at concessional rates has to export the goods, under EPCG scheme. Thus, the firm can fulfill its export obligation. In India, units operating in the SEZ are expected to honor export obligation against special concession offered to them. Therefore, every business unit has to export first in order to import goods.

2. increase Production Capacity

For every business unit increase production is necessary, in order to meet domestic demand and export order. Exports are possible when surplus production is available after meeting domestic demand.

3. Organizational Efficiency

Export management enables a firm to improve its organizational efficiency. E.g. firms have to emphasize on training and development of employees. This helps to improve knowledge, attitudes, skills and social

behaviors. Therefore, the over all efficiency of the organisation improves due to training, research and other much activities which are encouraged by export management.

4. Higher Profits

Export management enables a business unit to export quality goods at higher prices and there by raise the profit margin.

5. Reputation and Goodwill

Exports bring reputation to the export firm in international market as well as in the domestic market. It is assumed that export firms. Produce quality goods which help to develop goodwill. e.g., some of the world-famous firms include Microsoft for computer and Nike for Sports.

- Sony for Electronics etc.

6. Economies of Scale

Because of increase in export there will be large scale production and distribution. This will result in-

I. Economies of large-scale production like discount in bulk purchase of material and reduce cost.

II. Economies of large-scale distribution such as freight concession on bulk shipment of goods.

7. Technological Up gradation

Continuous research and development activities lead technological development and improvement in other organizational activities which help in improvement in quality standards which is beneficial to the form and customers both.

8. Imports are liberalized

Business organizations exporting on a large scale collect huge foreign exchange which can be utilized for the import of new technology machinery and component. This also raises their competitive capacity.

9. spreading of Marketing Risk

A firm engaged in domestic as well as export marketing activities can spread its marketing risk. The loss in domestic market can be compensated by the profit, earned in export market and vice versa.

10.Government Incentives

Exporter gets various assistances and incentives for export promotion. These are Duty Drawback, Octroi exemption, Excise duty exemption, Income tax exemption, liberal finance etc. These incentives make export marketing attractive and profitable.

In present global trade, all countries developed as well as developing take special interest and initiative in making export trade at a very large scale. Thus, large scale exports are necessary for survival and growth of developed as well as developing countries. Presently U.S.A., European counties, China, Japan, India,

and many other countries, take special measure, for promoting export. This suggests the significance of exports to all counties developed, developing and even to poor counties. Therefore, it is correct to say that, "Exports are necessary not only for developed counties but also for developing countries like India".

1.4 NATURE /FEATURES OF EXPORT MANAGEMENT

The main features of export management are as follows-

1. Large scale operations

Export management involves large scale marketing and production operations of goods and services. Because of large scale business operation, the firm gets the benefit of economics of scale and increase profit margin. Import, of other counties also prefer in placing large orders. Exporters get advantage of reduce cost and quoting competitive prices in the increase market.

2. Systematic Process

It is a systematic process became the export manager under takes various marketing activities such as marketing research, product design, branding, packaging, pricing, promotion etc. All these aspects require collection of data, analysis of data, then in perpetration of data in order to take systematic export marketing decisions.

3. Three faced Competition

Foreign trade market is highly competitive in nature. The competition is three dimensional i.e.

I. Competition from Indian exporters
II. Competition from local producers of Importing country.
III. Competition from exporter of other nations

4. Trade Barriers

Export trade is subject to trade barriers tariff and nontariff barriers. The trade barriers are the restrictions on free movement of goods between countries. Normally countries impose trade barriers in order to restrict import. The export marketing manager must have a good knowledge of trade barriers imposed by importing counties.

5. Domination of MNC

Multinational Corporation has huge investment and conduct business operation all over the world. Major share of foreign trade is captured by MNCs, and TNCs, (Transnational corporations). Therefore, they dominate in export management activities of the world. Due to large scale business, they get the benefit of economies of scale.

6. Domination of Development counties

Most of the MNCs belong to industrially developed countries. Such countries like USA, Japan, Germany etc. produce and sell good quality of goods at low cost on massive scale with the help of advanced technology. In this way rich and developed countries always dominate in international business activities.

7. Foreign Exchange Regulation

Export trade is subject to foreign exchange regulations imposed by countries. These foreign exchange regulations relate to payment and collection of export proceeds. In addition, export marketing is subject to other rules and regulations relating to health and safety, environment protection, etc. All such regulation affects free movement of good among the countries.

8. Documentation formalities

Export marketing is subject to various documentation formalities. Exporters require various documents to submit them to various authorities including customs, port trust, etc. The documents include Bill of lading, Commercial consular invoice, Shipping bill, Certificate of origin etc.

- Shipping Bill
- Consular Invoice Certificate of Origin etc.

9. Marketing Mix

Export marketing requires the right marketing mix for the target market, i.e., exporting the right product at the night price, at the right place and with the right promotion, the exporter can adopt different marketing mixes fro different export markets, so as to maximize exports and earn higher retunes.

10.International Marketing Research

Knowing more about customers, dealers, and competitors is a must not only in the domestic markets but also in the export markets. Marketing research is a must in export business due to various factors, such as diversities on social, economic, and political environments of distant markets.

11.Advance Technology

Export marketing is highly competitive. An exporter should be able to sell quality articles at competitive price. Use of advanced computer – oriented technology is a must for making the goods globally competitive. World markets are dominated by developed countries due to intensive use of computer technology.

12.Globalise or perish

Foreign trade is the need of each country. Because some important goods a country has to import like technology and goods which are not available in domestic market to export to get foreign exchange, otherwise it will perish economically.

13.Subject to Regulation

Foreign exchange regulation may be imposed by importing countries. These may relate to payment and collections of export proceed. Similarly export trade is subject to other rules and regulations relating to

health and safety, environment protection etc. All such regulations affect free movement of goods among the countries.

14.Diverse customs and Traditions

The export markets differ in languages, customs and traditions. The exporter may not be able to cope up with these diversities. Therefore, he has to be selective, he should be deal in only such markets where he can easily handle or overcome such differences or diversities.

15.High Amount of Risk

Export business is profitable than domestic business. But it is riskier also. Such as cancellation of order, non-collection of documents, nonpayment, transport risk, foreign regulation risk etc. this risk can be reduced by taking various insurance cover form ECGC and insurance agents. Risk can be spread also by exporting goods to many countries, so that loss in one market is compensated by the project other market.

16.Sensitive and Flexible Character

An exporter has to identify the specific requirement or foreign buyers and design the goods accordingly, but some times because of technological development new design of good may be supplied by other exporters due to which demand for these goods may go down. Therefore, exporter has to offer continuous support and loyalty.

CHECK YOUR PROGRESS

Enlist the main features of Export management.

"Export is treated as an engine of economic growth and an instrument for employment generation". Explain. Define the term Export Management.

1.5 FUNCTIONS OF AN EXPORT MANAGER

To decide export objectives of the organization and prepare comprehensive short term and long-term plans and programmed to achieve such objectives.

To conduct marketing research so that export efforts will be concentrated on certain commodities and on foreign markets which are highly promising.

To introduce product development and to produce quality goods as per specific needs of foreign markets/buyers.

To execute long-term export promotion programmes for the products with promising overseas demand.

To fix up the prices of exportable items with proper care.

To find out new designs for packaging of export items.

To look after the advertising and publicity abroad and to maintain effective communication with prospective buyers.

To look after prompt execution of export orders so as to avoid inconvenience to foreign buyers.

To analyze the EXIM policy of the government and the current export regulations and procedures.

To look after the opening of new branches/offices aboard.

To face the challenges of international competition and changing marketing environment.

To evaluate export incentives/facilities offered by the government and to secure benefits from them.

To look after the accounting and financial aspects of export transactions.

To look after the training of staff working in the export division, to motivate them and to develop human relations.

In brief the functions of an export manager are to develop export markets for which he should plan, organize, direct and control the export marketing activities.

Therefore, the export manger should have a clear understanding of there
major export marketing functions. Planning the export marketing activities is very important in order to avoid bottleneck, which may be proved expensive in foreign market. Organizing expert marketing system requires and understanding of fundamental organizational concepts.

Unit 2: foreign trade policies and export promotion.

Objective of foreign trade policies:

Main objectives of Foreign Trade Policy as noted below:

1. To strengthen the base for export production for promoting exports.
2. To create sound and favorable situation for export promotion through diversification.
3. To facilitate technological up gradation of domestic production as to make Indian good globally competitive.
4. To reduce imports through import substitution and encouragement to indigenous production.
5. To simplify and stream line import, export procedures.
6. To enable exporters to draw long term export plans and strategies.
7. To provide necessary institutional support to export initiatives by exporters / export organizations.
8. To provide liberal import facilities to promote exports.
9. To offer different types of exports incentives, concessions and facilities so as to encourage manufactures and exporters to take more initiative in export promotion.
10. To facilities availability of necessary imported inputs for sustain industrial growth.
11. To establish the frame work for globalization of Indian's foreign trade.
12. To promote productivity, modernization and competitiveness of Indian industry and thereby to enhance its export capabilities.
13. To foster the country's Research and Development (R & D) and technological capabilities.

MEANING of EPCs

The EPCs are nonprofit organizations registered under the Companies Act or the Societies Registration Act, as the case may be. The basic objective of EPCs is to promote and develop the export of the country. Each EPC is responsible for the promotion of a particular group of products, projects and services.

ROLE / FUNCTIONS OF EPCS

The main role of EPC is to project good image of Indian products. The EPC performs its role by performing the following functions.

1. **Issue of Certificate of Origin:**

Certain counties demand certificate of origin from the exporters. In India EPCs can issue certificate of origin to the exporters certifying the origin of goods.

2. Collection of Information:

It collects valuable information on overseas imports, import regulations, about competitions, customer preferences, market demand and other developments in foreign trade

3. Supplying Information:

It provides information on latest developments in the field of export trade. It may relate to various aspects of foreign trade. Such information is vital to the exporters to take export marketing decisions.

4. Organizing Seminars:

It organizes seminars, workshops discussions, meeting, and conferences on various aspects of foreign trade. Exporters are invited to take part in such seminars and to know the latest developments in foreign trade.

5. Trade Fairs & Exhibitions:

It may also assist the concerned authorities in organizing trade fairs and exhibitions in India and abroad. It may also arrange buyer seller meets, so as to promote Indian exports.

6. Recommendations to Government:

It provides recommendations to the government authorities to solve export problems and suggest measures for export growth. It may advise the government in framing people export import policies. It may recommend certain modification in the existing government policies and programmes.

7. Inviting Trade Delegations:

It may invite trade delegations from abroad, both at private level and at governmental level. Such trade delegations are very important to promote export trade of India. Foreign delegation visits India and sign contracts with Indian exporters.

8. Sending Delegations Abroad:

It may undertake the responsibility of sending trade delegations abroad comprising Indian businessmen, Indian delegations, may visit abroad and enter into contracts with overseas buyers. Thus, foreign trade of India can be expanded.

9. consultancy Services:

It may offer professional advice to exporters in areas such as technology up gradation, quality and design improvement, innovation etc. Such advice goes a long way to improve product and organizational efficiency of the exporter.

10.Exploration of Overseas Markets:

It may assist the exporter in exploration of overseas markets and identify items having export potential. It may also assist the exporter to open offices or branches abroad. It may guide the exporter in setting up of joint ventures abroad.

11.Developing Export Consciousness:

This organization makes all the possible efforts to develop export consciousness in our country. This is because there is great need for a country like ours so as to earn foreign exchange.

COMMODITY BOARDS

MEANING

CBs have been set up by GOI in order to organize, develop and promote production and exports of certain traditional commodities. At present there are about nine commodity boards namely:

1. Tea Board
2. Coffee Board
3. Rubber Board
4. Spices Board
5. Coir Board
6. Tobacco Board
7. Coconut Development Board
8. Central Silk Board
9. National Horticulture board.

ROLE /FUNCTION OF COMMODITY BOARD

The main role of Commodity Board (CB) is to projects good image of Indian products. The CB's performing the following functions:

1. Issue of certificate or origin

Certain countries demand certificate of origin to the exporters certifying the origin of goods.

2. Collection of information

It collects valuable information on overseas imports, import violations, about competitors' customers' preferences, market demand and other developments in foreign trade.

3. **supplying information**

It provides information on latest development in the field of export trade. it may relate to various aspects of foreign trade. Such information is vital to the exporters to take export marketing decisions.

4. **Organizing Seminars**

It organizes seminars, workshops discussions, meeting, and conferences on various aspect of foreign trade. Exporters are invited to take use of such participations to know the latest development in foreign trade.

5. **Trade Fairs & Exhibitions**

It may also assist the concerned authorities in organizing trade fairs and exhibitions in India and abroad. It may also assist the exporters to take part in such trade faire and exhibitions. It may also arrange buyer seller meets so as to promote India exports.

6. **Recommendation to Government**

It provides recommendation to the government authorities to solve export problems and suggest measures for export growth. It may advise the government in framing proper export import policies. It may recommend certain modifications in the existing government policies and programmes.

7. **Inviting Trade Delegations**

It may invite trade delegation from abroad, both at private level and at government level such trade delegations are very important to promote export of India. Foreign delegations visit India and sign contracts with Indian exporters.

8. **Sending Delegations Abroad**

It may undertake the responsibility of sending trade delegation abroad comprising Indian businessmen. Indian delegation may visit abroad and enter into contracts with overseas buyers. Thus, foreign trade of India can be expanded.

9. **Consultancy Services**

It may offer professional advice to exporters in areas such as technology up gradation quality and design improvement, standards and specification, product development, innovation, etc. Such advice goes a long way to improve product and organizational efficiency of the exporter.

10.**Exploration of Overseas Markets**

It may assist the exporter in exploration of overseas markets and identify items having export potential. It may also assist the exporter to open offices or branches a broad. It may guide the exporter in setting up of joint ventures abroad

11.**Developing Export Consciousness**

This organization makes all the possible efforts to develop export consciousness in our country. This is because there is a great need for exports for a country likes so as to earn foreign exchange.

12.**Other Functions**

It may allocate or distribute quota/slips in respect of certain items.

It may fix minimum floor prices or may advise the government in such fixation of floor prices.

EPC may undertake publicity through schemes like joint foreign publicity in export markets.

STATE TRADING CORPORATION (STC)

MEANING

STC was established in May, 1956 as an autonomous corporation for diversifying foreign trade and supplementing efforts of the private trade and industry in developing India's foreign trade. STC is the first public sector organization which participates in foreign trade. It is a premier international trading house set up by the central government in 1956 and is operating for 45 years.

The institution of public sector commercial corporations was created in the fifties when India started developing trade exchanges with the East European countries. India was one of the few developing countries to have established trade with the East European countries. The Rupee payment Agreement was instrumental in improving trade with these countries. In the initial years of India's planning era, this agreement had helped the country to have access to capital goods and equipment as well as key inputs like fertilizers, non-ferrous metals and petroleum for which otherwise payment would have been made in hard earned foreign exchange, India was also able to develop some of its exports sectors, especially wool and woolen products, readymade garments and leather foot wear by exploring the market of East Europe. These exports and imports were undertaken by the state trading organizations like STC.

Over the years STC has become the premier international trading organization. If offers a wide range of services and handles exports of almost everything that India produces from coffee to compressors, tobacco to textile machinery and roses to rolling shock. It imports wide range of products from raw materials to equipment. The role of STC as path-finder for the Indian industry has been quite praise worthy. As international marketing agency, its services, relating to negotiating, contracting, financing, product development, quality control, market intelligence, shipment and settlement of trade disputes has been quite significant.

In addition to earning foreign exchange for the country, it has added stability to the industrial sector. STC continuously look for the new products and new markets. STC has excelled in marketing nontraditional commodities aboard. One of the major achievements of STC relates to formation of joint marketing groups of the manufactures of engineering equipment, drugs and pharmaceuticals to bid for international business. STC had done commendable work in promoting difficult to sell products overseas. STC imports scarce commodities through the foreign exchange earned under different marketing strategy. The Soybean project of STC aims at reducing dependence on traders or processors and to have its own supply base for export of Soya bean meal.

FUNCTIONS (SERVICES) OF STC:

1. To arrange for exports where bulk handling and long-term contracting are advantageous.
2. To promote the production of nontraditional items and opening up new fields for the export of traditional items.
3. To provide development finance for the production of export-oriented goods and boost export of small-scale sector.
4. To facilitate bulk purchasing for bulk selling abroad.
5. To undertake internal trades as and when the situation warrants it and to ensure adequate and regular supplies at reasonable prices of essential commodities to meet local demand.
6. To facilitate the implementation of trade agreements and bilateral deals.
7. To organize production to meet export demands and to help production unit to overcome difficulties of raw materials and other essential requirements.
8. To act as vehicles for the implementation of Governments trade policies and trade plans.
9. To undertake price support operations to protect the interest of growers.
10. To look after buffer stocking to ease the problem of shortage of some essential items.

FUTURE ROLE OF STC

It is a fact that public sector trading organizations like the STC and MMTC have traditionally depended heavily on canalized trade. However, the list of canalized items has been drastically reduced in recent years. This suggests that in future, STC will have to take up new areas for its business activities. The STC will have to be reoriented to achieve the objective of emerging as International Trading House capable of operating in a competitive global environment of servicing an effective instrument of public policy and of providing adequate support services to the small/cottage sector. In short, STC needs to be given more purposeful role through suitable restructuring.

I. The STC of India.
II. The Handicrafts and Handloom Export Corporation of India (HHEC), a wholly owned subsidiary of STC
III. The Minerals and Metals Trading Corporation of India Ltd. (MMTC)
IV. The projects and Equipment corporation (PEC) of India Ltd. a wholly owned subsidiary of STC.
V. The Cashew Corporation of India Ltd. (CCI) a wholly owned subsidiary of STC. **VI.** Central Cottage Industries Corporation (CCIC) a subsidiary of STC.
VII. The Tea Trading Corporation of India (TTCI) a subsidiary of STC.

VIII. The Mica Trading Corporation of India (MITCO), a wholly owned subsidiary of MMTC.

IX. Spices Trading Corporation

X. The State Chemicals and Pharmaceuticals Corporation of India Ltd. (SCPC) a subsidiary of STC.

FEDERATION OF INDIAN EXPORT ORGANIZATIONS (FIEO)

It is a nonprofit institution set up jointly by the GOI, Trade and Industry. It was set up in 1965 as a common and coordinating platform for various export organization including commodity boards EPCs and the service institutions and organizations. It functions like EPCs.

The principal functions of FIEO are:

a. To co-ordinate the export promotion activities of its members.

b. To collect and forward all important commercial and market information to its members.

c. To provide common services for the benefit of exporters and their organization.

d. To conduct meetings, conferences, seminars and workshops on export trade.

e. To arrange for buyer and seller meets

f. To project by way of advertising and publicity Indian goods and services abroad.

g. To sponsor study trams abroad.

h. To keep overseas liaison with international and UN agencies like GATT, UNCTAD, ESCAP, IMF, ILO etc.

i. To establish rapport with overseas chambers of commerce Trade Associations and Govt. Agencies.

j. To send Indian trade delegations abroad and to invite foreign trade delegations.

k. It is a registering authority for export houses/TH/STH/SSTH. It issues RCME to such houses.

Assistance to Exporters

a. It issues recommendatory letters to members for getting visa for export promotion tours.

b. It issues certificate of origin to recognized member export houses/ Trading houses / STH/SSTH.

c. It takes up cases of members with public grievance committees in the officers of it. DGFT, collector of customs or any other organization, like railways, telecommunications, P & T various ministries etc.

d. It circulates / disseminates trade information.

INDIAN INSTITUTE OF PACKAGING (IIP) MUMBAI

With a view to improve the packaging and packing standards, the IIP was set in 1966 at Mumbai. The main functions of IIP are:

1. training programmed

IIP is primarily engaged in training programmes relating to packaging industry. This institute makes the trainees familiar with packaging technology, packaging materials and current trends in packaging in the world markets.

2. Improvement in quality of packaging

IIP makes constant efforts to up grade and improve the design and quality of packaging, so as to promote Indian products abroad.

3. collection of information

IP collects information on latest trends in the packaging in respect of raw materials design etc.

4. Supply of information

The information collected on packaging is provided to the exporters by IIP, so that exporters can upgrade their packaging standards

5. organizing seminars and workshops

IIP organizes seminars and workshops on packaging designs and quality. Exporters can take the advantage of such seminars and workshops.

6. Consultancy services

It provides consultancy services to the exporters in the field of packaging.

7. Develops packaging consciousness

Export packaging is vital as it not only protects and preserves the product, but also promotes the product in international markets. Therefore, IIP develops consciousness of the need for good packaging among Indian exporters

8. Publication

It publishes two quarterly magazines one devoted to the technological aspects of packaging and the other to techno economic aspects of packaging.

9. Testing facilities

It provides testing facilities in respect of packages and packaging material to the exporters.

The IIP is one of the founder members of the Asian packaging federation established for exchange of all information on packaging and for chalking out common programmes for the development of packing standards in the Asian region.

CHAMBERS OF COMMERCE

The chambers of commerce and industry are established on regional basis. They assist export promotion in a number of ways. It's membership is open to all members of trade and industry, they provide suggestions and recommendations to the government on various issues concerning trade and industry including export marketing. The government while framing import and export policies does accept their suggestions. They help the exporters in issuing the certificates of origin and taking up specific cases of exporters to the government. The chambers of commerce also provide information and it provides a forum to their members to discuss the problems arising out of policy matters. They also organize workshops seminars and short training courses in order to guide their member on export aspect. They do send trade delegations abroad and invite trade delegations from abroad.

Role /Functions of Chamber of Commerce

The main role of Chamber of Commerce is to project good image of Indian products. The Chamber of Commerce performs in role by performing the following functions.

1. Issue of certificate of origin

Certain countries demand certificate of origin from the exporters. In India, EPCs can issue certificate of origin to the exporters' certificate the origin of goods.

2. Collection of information

It collects variable information on overseas imports, import regulations about competitors, customer preferences, market demand and other developments in foreign trade.

3. Supplying information

It provides information on latest development in the field of export trade. it may relate to various aspects of foreign trade. Such information is vital to the exporters to take export marketing decisions.

4. Organising seminars

It organizes seminars, workshops, discussion, meetings and conferences on various aspects of foreign trade exporters are invited take part in such seminars and workshops. The exporters can make use of such participations to know the latest development in foreign trade.

5. Trade fairs & exhibitions

It may also assist the concerned authorities in organizing trade fairs and exhibitions in India and abroad. It may also assist the exporters to take part in such trade fairs and exhibitions. It may also arrange buyer and seller meets, so as to promote Indian exports.

6. Recommendations to government

It provides recommendations to the government authorities to solve export problems and suggest measures for export growth. It may advise the government in framing proper export- import policies. It may recommend certain modifications in the existing government policies and programmes.

7. Inviting Trade delegations

It may invite trade delegations from abroad, both of private level and at government level. Such trade delegation are very important to promote export trade of India. Foreign delegations visit India and sign contracts with Indian exporters.

8. Sending delegation abroad

It may undertake the responsibility of sending trade delegations abroad comprising Indian businessmen. Indian delegation may visit abroad and enter into contracts with overseas buyers. Thus, foreign trade of India can be expanded.

9. Consultancy services

It may offer professional advice to exporters in areas such as technology up gradation, quality and design improvement, standards and specifications, product development, innovation etc. Such advice goes along way improve product and organizational efficiency of the exporter.

10.Exploration of overseas Markets

It may assist the exporter in exploration of overseas markets and identify items having export potential. It may also assist the exporter to open offices or branches abroad. It may guide the exporter in setting up of joint ventures abroad.

11.Developing export consciousness

This organization marks all the possible efforts to develop export consciousness in our country. This is because there a great need for country like our so as to earn foreign exchange.

12.Other functions

It may allocate or distribute quota/slips in respect of certain items. It may fix minimum floor price or may advise the government in such fixation of floor prices.

EPC may undertake publicity through schemes like joint foreign publicity in export markets.

INDIA TRADE PROMOTION ORGANIZATION (ITPO)

MEANING

The ITPO came into existence on 1st Jan, 1992, with the merger of Trade Fair Authority of India (TFAI) and Trade Development Authority (TDA). ITPO has set up five regional offices in Germany, Japan, UAE and USA.

ITPO acts as a publicity wing of Govt. of India for organizing and for participating in trade fairs and exhibitions in India and abroad.

ITPO has set up a Trade Information Centre at its headquarters in New Delhi, which is considered as the best source of information on import and export trade.

FUNCTIONS AND SERVICES OF ITPO: -

1. **Organizing trade fairs and exhibitions in India and abroad**: ITPO organizes trade fair and exhibitions in India and abroad and to book stalls /space for India exporters to participate in over seas trade fair and exhibitions. ITPO acts as a publicity wing of Government of India for organizing trade fair and
   exhibition in Indian

2. **Giving publicity to trade fair and exhibition**: ITPO gives publicity to trade fair and exhibitions organized in India so as to encourage foreign buyers to visit stalls of Indian goods and place orders for the same. ITPO also participates in trade fair and exhibitions arranged in India and abroad.

3. **Booking of space in Overseas Trade Fairs:** ITPO books necessary space/stalls for Indian exporters. This enables Indian exporters to participate in overseas trade fair and exhibitions.

4. **Inviting Trade Delegations from Abroad and sending Indian Trade Delegation Abroad:**
   ITPO invites trade delegation from abroad in order to participate in trade fair and exhibitions and book orders for Indian goods. In addition, to send Indian trade delegation abroad for market survey and for signing contracts for the supply of Indian goods.

5. Providing consultancy service to Indian exporters: ITPO provides consultancy service to Indian exporters for participation and displays their products in trade fairs and exhibition held in India and abroad.

6. **Organizing seminars and workshops:** ITPO organizes seminars, workshops for giving information, guidance to exporters about fair and exhibitions arranged in India and abroad. ITPO has set up trade information Centre at its headquarters in New Delhi. It is considered as the best source of information on import and export trade.

7. Miscellaneous Functions

   a. Promotion of exports through specialized programmes.
   b. Supply of trade information to Indian exporter and foreign importers.
   c. Undertaking research activities relating to export promotion of Indian goods services abroad.
   d. Membership of international organization such as WTO, etc.

Unit 3: International Trading Environment:

DEFINITION

According to Hess and Eater

International marketing is "the performance of business activities that direct the flow of goods and services to consumer or users in more than one nation."

According to Subhash c. Jain

"The term international marketing refers to exchanges across national boundaries for satisfaction of human needs and wants."

IMPORTANCE OF INTERNATIONAL TRADE / MARKETING

The important benefits of international marketing are explained as follows:

1. Lowers prices

International marketing enables companies to specialize in those goods and services in which they are more competent. By specialization, international marketing lowers the prices of goods and services. The comparative cost benefits enjoyed by firms due to specialization are partly passed on to the consumers.

2. Increases the real income and national wellbeing international marketing ensures that each country specializes in the production of those commodities, which it is best suited to produce, export its surplus produce and import those commodities which it can obtain cheaper from abroad. In doing so, it increases the real income and national well being of all the participating countries.

3. **Furthers technological development**

By specialization, it lowers the prices of goods and services all over the world. Consequently, it stimulates their consumption and demand for more and better goods, which causes further specialization and technological developments.

4. **Provides higher standard of living**

International marketing provides better life and welfare to people in different countries. It provides goods which cannot be produced in the home country. This raises the standard of living of the people and social welfare.

5. **Reduces dangers of monopolistic exploitation**

By ensuring free competition, international marketing reduces the dangers of monopolistic exploitation of consumers by the home producers; exploitation of one country by another country is also difficult since there are numerous suppliers in international markets.

6. **Ensures optimum use of resources**

International marketing enables to make optimum utilization of resources. Countries having surplus resources or production can export to other markets.

7. **Builds cultural relations**

International marketing changes the quality of life of people. It not only exchanges goods and services among nations but also it develops closer social and cultural relations between different nations.

8. **Bridges the technological gap**

It makes possible to transfer technology and other assistance from the developed nations to the developing ones and as such, it narrows the gap between the developed and developing countries.

9. **Bring international co-operation and world peace**

International marketing brings countries closer due to trade relations. Because of interdependence of countries, cordial relations are maintained and this ensures world peace.

10. **Brings about rapid industrialization**

Most of all, international marketing brings about a rapid growth and development not only of the developing nations but also that of developed ones. While the developed nations provide aid, capital goods and technology to the developing nations, which in turn supplies raw materials and labour to the developed nations.

FACTORS INFLUENCING INTERNATIONAL TRADE / MARKETING

International trade is the exchange of commodities, products, services, capital between people and companies in different countries. It forms a significant part of many counties' Gross Domestic Product, GDP. International trade has existed for a long time, but trade has increased hugely in the past few hundred years and has a major impact on the economies of many countries.

1. Exchange Rate

The exchange rate is the price of one currency in terms of another. Exchange rates fluctuate depending on the demand for a particular currency. If there is a high demand for a country's currency then its price will tend to rise. Because currencies fluctuate in price it can often be cheaper to buy goods in one country and sell them in another. Because of this exchange rates have a major impact on international trade.

2. Competitiveness

Competitiveness is a measure of the relative ability of different countries to provide different products or services. Competitiveness takes into account the efficiency, costs of employment, level of government regulation and the ease of doing business. Competitiveness effects international trade because the more competitive countries will tend to attain a higher level of global trade.

3. Tariffs and Trade Barriers

Sometimes governments enact trade barriers to limit trade with foreign countries. Sometimes these can take the form of quotas, where only a limited amount of a certain product or service can be purchased from businesses in foreign countries. Tariffs have been a common trade barrier in history. A tariff is a tax paid on imported **4. Rate of profit –**

The rate of profit in export business is normally higher as compare to rate of profit in domestic marketing. The unit value realization of export products normally increases. Such progressive improvement in the unit value of realization is one reason which acts as a motivator for exporting.

5. **Sales and production stability** –

Export marketing may enable a firm to maintain sales and production stability. For example, in the case of seasonal products, exporting may help to achieve sales stability, because the seasons may be opposite in certain export markets. For example, woolen clothing.

6. **Inadequate domestic demand** –

The level of domestic demand may be insufficient for utilizing the available production capacity fall, i.e., at the optimum level. Here, the company enters in export marketing so that the available production capacity will be utilized fully for meeting domestic demand and demand from abroad. This is one motivational factor for export marketing.

7. **Economic growth –**

Growth is another major reason for internationalization. The growth potential of many foreign markets is a very strong attraction for foreign companies. Several developing countries, like China, and India, have been growing at a much faster rate than the developed countries. Many multinational companies are eager to establish their foothold in such countries, considering future potential.

8. **Reducing business risks –**

Geographical diversification facilitates distribution of business risks among different export markets. Even the risks in internal marketing can be reduced by undertaking export marketing. A diversified business spreads business risks among different markets.

9. **Information and media Revolution** –

There has been tremendous growth in the field of information and media. For example, internet facility has given a big boost to a global trade. Now, business firms can conduct global operations with much investment in setting up elaborate offices. Business activities in other countries can be conducted through information network

10.**Strategic vision** –

Some firms have a strategic vision to enter in export markets. The business strategy of such firms includes systematic international growth. Therefore, the stimulus for export marketing comes from desire to grow and expand, need to become more competitive.

11.**Accepting social responsibility** –

Export promotion is a collective responsibility of all social groups including business enterprises. Some large enterprises accept this social obligation and participate in export marketing. Here, social responsibility acts as a motivation for export marketing.

12. **Government policies –**

Government policies and Regulations may also encourage the companies for international marketing. Some companies export and invest in foreign countries to avail economic incentives, and benefits provided by the government. Also, some companies internationalize due to government's emphasis on import development and foreign investment. In India, certain companies export in order to fulfill their export obligation.

13.W. T. O.

Due to WTO, member nations have reduced a number of restrictions on foreign investment, and trade in goods and services. For example, the custom duties have been reduced worldwide. This has motivated business firms to enter in the global markets to a greater extent. –

Export business is normally in bulk quantity. Export orders are much larger as compared to orders in domestic marketing. Export business is undertaken in order to take the benefits of selling in bulk i.e., in large quantities. It helps to earn foreign exchange in large.

In brief, export marketing offers many benefits to exporting organization. Such benefits encourage companies to participate in export marketing. The benefits also act as motivators for export marketing.

PROBLEMS IN INTERNATIONAL TRADE / MARKETING

At present, Indian exporters face a number of problems / difficulties. The problem demotivates the business firms to enter into foreign markets. These problem / difficulties are as follows. **a) Recession in world market**

The world market, faced recession in 2008 and in the first half of 2009. The recession was triggered due to sub-prime crisis of USA in September 2007. Due to recession, the demand for several Indian items such as Gems and Jeweler, Textiles and Clothing and other items were badly hit. During recession, exporters get low orders from overseas markets, and they have to quote lower prices. Therefore, exporter gets law profits or suffers from losses. **b) Technological differences**

The developed countries are equipped with sophisticated technologies capable of transforming raw materials into finished goods on a large scale. Less developed countries, on the other hand, lack technical knowledge and latest equipment's. And therefore, they have to use their old and outdated technologies. It leads to the lopsided development in the international market. **c) Reduction in export Incentives –**

Over the years, the Govt. of India has reduced export incentives such as reduction in DBK rates, withdrawal of income tax benefits for majority of exporters, etc. The reduction in export incentives demotivates exporters to export in the overseas markets. **d) Several competitions in global marketing –**

Export marketing is highly competitive. This competition relates to price, quality, production cost and sales promotion techniques used. Indian exporters face three-faced competition while exporting. This includes competition from domestic exporters, local producers where the goods are being exported and finally from producers of competing countries at global level. Such competition is one special problem to the exporters.

e) **Problem of product standards** –

Developed countries insist on high product standards from developing countries like India. The products from developing countries like India are subject to product tests in the importing countries. At times, the importing countries do not allow imports of certain items like fruits, textiles and other items on the grounds of excessive toxic content. Therefore, Indian exporters lose markets especially in developed countries.

f) **Fluctuations in Exchange Rate –**

Every country has its own currency which is different from international currencies. The dominant international currencies are US dollar or Sterling Pound. From the point of view of Indian exporters, we are interested to realize the payment in international currency. Foreign exchange earned by the operators is converted into Indian rupees and paid to the exporters in Indian currency; this exposes the exporters to the dangers of fluctuation in foreign exchange rates. **g) Problems of Sea Pirates Attacks –**

A major risk faced by international trade is attack by pirates in the Gulf of Aden. More than half of India's merchandise trade passes through the piracy infested Gulf of Aden. New exporters and importers are facing problem, because of increased pirate attacks as they find it difficult to get insurance cover.

h) **Problem of subsidies by Developed countries** –

The developed countries like USA provide huge subsidies to their exporters. For example, in case of agriculture exporters, USA, UK and other provide huge subsidies to their exporters. Therefore, the exporters of developing countries like India find it difficult to face competition in the world markets. **i) Problem in preparing Documents –**

Export involves a large number of documents. The exporter will have to arrange export documents required in his country and also all the documents as mentioned in the documentary letter of credit. In India, there are as many as 25 documents (16 commercial and a regulatory documents) to be filled in.

j) **Government restrictions and foreign exchange regulations** –

The Government restrictions compel the exporters to follow certain rules and regulations in the form of licenses, quotas, and customs formalities. Due to such restrictions, new problems develop before the exporters. Even trade restrictions in foreign countries create problems before exporters. Indian exporters face this difficulty of government restrictions and foreign exchange regulations even when trade policy is now made substantially liberal.

k) **High risk and Uncertainties –**

Export marketing is subject to high risks and uncertainties. The risks may be both political and commercial. Political risks involve government instability, war, civil disturbances, etc. The commercial risks involve insolvency of the buyer, protracted default on the part of the buyer dispute on quality and so on.

l) **Competition from China**

India is facing stiff competition from China in the world markets, especially in the OECD markets. As a result, India's share of export of OECD markets has declined from 53% of total exports in 2000-01 to about 38% in 2007-08. Some of the Indian exporters have lost their overseas contracts due to cheap Chinese goods and supplies. This is the major problem of exporters.

Unit 4: export product planning:

Product planning strategies:

Product planning involves a number of product related decisions such as product design, product-mix, branding, pricing etc., Therefore, the exporter has to adopt various product planning strategies such as-

(a) PRODUCT DESIGN STRATEGIES

Exporters have the following options in respect of product design strategies.

(i) Product Innovation Strategy-

Under this strategy, the exporter develops a new product in response to the demand prevailing in overseas markets. Low-cost items for developing countries and high-cost items for developed countries may be developed for example readymade garments to take advantage of fashion in different countries.

(ii) Product Adaptation Strategy-

Under this strategy, both the product and the message are changed to increase the adaptability of the product. This helps to meet the specific needs of the foreign buyers. Products like office machines, health goods, and electrical goods require this strategy. It is a costly strategy. It is generally used to serve large size markets.

(iii) Product Standardization Strategy-

Under this strategy, the exporter sells the same product all over the world with "One product, One message-worldwide". For example, Coca-Cola, Sony etc. The exporters use this strategy when their product is too well known and enjoys global reputation. It is an economical strategy. This is because it does not require any modification. Moreover, it enjoys the economies of large-scale production and marketing.

(b) PRICING STRATEGIES

As far as pricing strategies are concerned, an exporter may adopt any of the following strategies:

(i) **Skimming Pricing-**

Where high price is charged when the product is launched in the market as in the case of innovative products.

(ii) **Differential Pricing-**

Where different price is charged in different markets depending upon market and competitive conditions.

(iv) Standard pricing-

Where the same price is charged in all the global markets, which is quite rare.

(c) **DISTRIBUTION STRATEGIES**

Distribution channel includes a set of marketing institution participating in the marketing activities involved in the movement of flow of goods or services from the primary producer to the ultimate consumer. In international marketing, two categories of marketing channels are involved i.e.-

(i) **Direct Exporting-**

Direct exporting is a method of exporting goods directly to the foreign buyers by the manufacturer himself

(ii) **Indirect exporting-**

Where the manufacturer exports through intermediaries such as Star Export Houses or Merchant Exporters.

(d) **PROMOTION STRATEGIES**

The exporter also needs to adopt suitable promotion strategies-

(i) **Standard Promotion Strategy**

Where the same promotion-mix is used in all the global markets and with the same promotional theme.

(ii) **Differential Promotion Strategy**

Where different promotion-mix is used depending upon market and competitive situation and with different promotion theme.

(e) **AFTER-SALE-SERVICE STRATEGY**

After-sale-service is plays very important role in the case of durables, office equipment and machinery. The exporter needs to adopt proper after-sale-service strategy. Exporters may have a tieup with after-sale-service agencies in overseas markets. They may also depute their own service staff in important **markets.**

(f) PRODUCT POSITIONING STRATEGY

The exporter should develop effective product positioning strategy. The exporter may position the brand to create a distinct image by adopting positioning by features, price, use, benefits etc. **(g)**

RODUCT LINE STRATEGY

The exporter should decide whether to concentrate on one product or deal with product line. It is a group of related products. For example, a food items products line may include milk powers, health drinks, fruits etc.

Product mix:

A marketer needs to decide on the product mix and product line. Product mix refers to the complete set of products, which are offered for sale by a firm. For instance, a product mix may consist of stationery, food product, beverages, cosmetics, detergents, etc. A product line is a good group of dated products for instance food products may include with providers confectionaries fruit juices, health drinks, breakfast cereals, and so on.

BREADH AND DEPTH OF PRODUCT MIX

A product mix has the structural dimensions of breadth and depth. Breadth refers to the number of product line in duded in the product mix for instance the company may deal in soaps, detergents toothpastes, shaving creams, and so on.

Depth refers to the number of product items (of different shape size, color, etc.) within each product line for instance, in the soup category, the product items can be of different colors, shapes, fragrances, and so on. For instance, Hindustan lever produces Liril, Lifebuoy, Lux and so on.

REASONS FOR PRODUCT MIX

Many marketers prefer to deals with on appropriate blend of product mix rather than concentrating on a single product. There are several reasons as to why companies prefer product mix. A Typical well established large firm's product mix includes new growing maturing and declining products. The main reasons as to why firms go for a product mix rather than concentrating on a single product are as follows.

- To use company resources and move effectively.
- To spread production and marketing costs over a wide product range.
- To become better knowing by middlemen and consumer To capitalize on brand loyalty of a particular brand.
- To balance out seasonal sales patterns.
- To offset the effects of the product life cycle of product on the firm.
- To meet the challenges of the competitors.
- To meet the demands of the customers.

PRODUCT BRNADING

MEANING OF BRAND

Brand is a name term, symbol and / or special design that are intended to identify the goods or services of one seller or group of sellers. A brand differentiates one seller's products from of competitors.

Branding is the management process by which a product is branded. It is a general term covering various activities such as giving ultimate name for a product, assigning a brand mark and establishing and plating it.

METHODS OF BRANDING

There are many ways of branding a product some of them are as following.

1) **Individual brand: -** A firm may decide upon a strategy of adopting. For example, Proctor & Gamble, Hindustan Lever Limited etc., have adopted this strategy.

2) **Family brand: -** When a firm has various product lines and when each product line is given a particular brand name it is has called family brand. A company may produce different line of items like food items, health care products, and personal care products and so on. Each one of product

   line is given separate brand when used widely; a family brand strategy has considerable merit. It is much simpler and less expensive to introduce new related products to a line e.g., Maggi (noodles, Ketchup), (ketchup, jam) etc.

3) **Umbrella brand / company brand:** When all the products of a company have the name of the company as a brand name such brand name is known as umbrella brand or company brand. Companies like B.P.L. Videocon, Sony adopt only one brand name for their products.

4) **Combination of brand name and company's name:** under this method of branding each product of that company has an individual brand name but it also has the name of company brand to indicate the business house producing the product e.g., Tata Sault, Tata Tea, Amul butter, Amul Cheese, Alum Pizza etc.

5) **Private or middleman's brand**: Under this arrangement manufacture introduces his products under the distributers brand name the manufactures merely produce goods as per the specifications and requirements of distributors and the middlemen undertake marketing. Product like woolen, hosiery,

Sport goods etc. normally are sold through private brands where they have more control over pricing and distribution.

6) **Multiple brand:** In order to increase the share of the market and profitability a company may use multiple brands for example, Watch are sold in two brand names one in the name of Titan itself and the other in the name of Sonata Ltd. Television are sold in the name of LG in the Urban Areas.

7) **Symbols and marks:** Bands can be used in the form of symbols and marks. Such as camel for Camel ink. Tiger for Tiger cocks and so on.

8) **Special Names:** companies may use special name such as VXL, LUX, RIN and so on.

9) **Names of founders:** The company may also introduce the products with the name of founders or inventors' examples are Dunlop, Colgate etc.

10) **Typical Numbers:** At times companies may follow certain numbers such as 555 (Cigarettes), 50 (Cake soap) and so on.

11) **Combination of Name and Numbers:** There can be a combination of names and numbers. This includes here Honda CD 100 perimeter, 118 NE, 7UP, 7 O'clock blades and so on.

PRODUCT PACKAGING

We define packaging as all the activities of designing and producing the container for a product. Packages might include up to three levels for material. Cool water bislery comes in a bottle in a card board box in a corrugated box containing six dozen boxes.

Well – designed package can build brand equity and drive sales. The package is the buyer's first encounter with the product and is capable of turning the buyer on or off packaging also affects consumers' later product experiences. Various factors have contributed to the growing use of packaging as a marketing tool:

1) Self – Service:

An increasing number of products are sold on a self – services basis. In an average supermarket, which stocks 15,000 items, the typical shopper pusses by some 300 item per minute. Given that 50% to 70 % of all purchase are made in the store, the effective package must perform many of the sales tasks attract attentions, describe the products features create consumer confidence and make a favorable overall impression.

2) Consumer affluence:

Rising consumer affluence means consumers are willing to pay a little more for the convenience. Appearance, dependability, and prestige of better packages.

3) Company and brand image:

Package contributes to instant recognition of the company or brand. In the store, package for a brand can create a visible bill board effects such as Garnier Fructis and their bright green packaging in the hair care.

4) Innovation Opportunity:

Innovative packaging can bring large benefits to consumers and profits to producers. Companies are incorporating unique materials and features such as resalable spouts and openings. Heinz's unique colorful EZ squirt ketchup and upside-down bottle have helped to revitalize the brand's sales.

From the perspective of both the firm and consumers packaging must achieve number objectives.

1. Identify the brand.
2. Convey descriptive and persuasive information.
3. Facilitate product transportation and protection.
4. Assist at – home – storage.
5. Aid product consumption.

To achieve the marketing objectives for the brand and satisfy the desires of consumers' marketers must choose the aesthetic and functional components of packaging correctly. Aesthetic considerations relate to a package's size and shape, material, color, text and graphics. Blue is cool and serene, red is active and lively, yellow is medicine and weak, postal colors are feminine and dark colors are masculine. Functionally, structural design is crucial. For example, innovations with food products over the years have resulted in packages that are resalable, tamperproof, and more convenient use (easy to hold, easy to open or severable.) The packaging elements must harmonize with each other and with pricing, advertising and other parts of the marketing program.

After the company designs its packaging. It must test it Engineering tests ensure that the package stand up under normal conditions visual tests. That the script is legible and the colors harmonious, dealer tests so that dealer finds the packages attractive and easy to handle and consumer tests, that buyers will respond favorable Eye tracking by hidden cameras can assess how much consumers notice and examine packages.

IMPORTANCE OF PACKAGING

Proper packaging not only helps to protect and preserve the product, but also it serves a number of other purposes.

1) **Protection:** Packaging helps to protect the product from damage during transit or during its use, while packing the product, the exporter needs to consider the mode of transport, the channels the amount of handling and so on.

2) **Preservation:** Packaging helps to preserve the products quality from deterioration. The exporter needs to consider climatic conditions, the transit coverage etc. Items such as spices and flavored food items need to be packed in moisture proof packs in order to preserve the quality.

3) **Promotion:** Packaging gives advertising value to the product. Attractive packages attract the attention of the shoppers and they buy such attractively designed products.

4) **Convenience to Consumer:** Packaging offers convenience to the customers in handing and using the product. Proper designed packages can be stored in the shelves or racks in the consumers' homes. Packaging also provides direction for use in respect of certain products.

5) **Benefit to retailers:** Packaging also benefits the retailers in handing and selling the products. They can display the product in the racks. Accordingly, Customers may demand those products which are

prominently displayed in the shop. In self – service stores, proper write up on the package can facilitate customer's choice.

6) **After – use Value:** Certain packaging can offer after- use value to the customers and others. For instance, plastic cans used for packing oil can be used by the customers for some other purpose. Again, manufactures can be using certain packaging such as soft- drinks bottles.

7) **Identification:** Packaging helps to identify the product. This is because product picture is shown on the package and also due to the printed matter on the package.

8) **Develops Confidence:** Properly packed products develop confidence in the minds of the people. It builds good will for the Organisation. Again, packaging may provide a reasonable degree of protection from being duplicated.

Labeling:

MEANING

The label is the printed material that appears on the package. A products label can be a strong selling tool. The labels illustration and copy should indicate clearly the content of the product, and the directions for use. The label contains the brand name and as such it assists in identifying the brand. The label should indicate the following information.

TYPES OF LABELS

There are three types of labels
(a) Brand Labels (b) Grade Labels, and (c) Informative Labels.

(a) Brand Labels: When only the brand name, appears on the product package or item. It is a brand label. This is mostly used incase of clothing, Stickers on fruits, and such other products.

(b) Grade Labels: At times the product package or item may have a grade indication by a letter number or word. Grade label indicate the quality standard of the product.

(c) Informative Labels: These labels give information in respect of the product's ingredients, use, care, performance, or expiry date, limitations, precautions, Nutrition value, price and such other details.

LABEL DESIGN – ITS OBJECTIVES

- An effectively designed label can enhance communication with large customers and build brand and company Images.

- Labels provide are a reasonable degree of protection from being duplicated.

- Labels comply with statutory requirements in case of food items such as the expiry date, or statutory warnings on cigarette packs and so on.

- It provides direction for use in respect of certain products.
- Labels facilitate brand comparisons on the part of the consumer and as such it facilitates the purchase decision.

- It creates confidence in the mind of consumers.

Unit 5: import export procedure and documentation

❖ Stages in import procedure

What is Import Procedure?

The purchase of goods from a foreign country is referred to as import trade. The Import procedure varies by country, depending on the country's import and customs policies, as well as other statutory requirements.

Import procedures are the procedures for import and export activities that include ensuring licensing and compliance prior to shipping goods, arranging for transport and warehousing after goods are unloaded, and obtaining customs clearance and paying taxes prior to the release of goods.

Steps involved in Import transactions:

1. **Trade Enquir**y: The importing company should first gather information about the countries and companies that export the given product. The importer is able to collect such data from trade directories and/or trade associations as well as organizations. After identifying the countries and firms that export the product, the importing firm contacts the export firms by using a trade enquiry to learn about their export prices and terms of export. A trade enquiry is a written request from an importing firm to an exporter for information on the price and various terms and conditions under which the latter is willing to export goods. The importer will receive a quotation from the exporter in response to this inquiry. The quotation includes information about the goods available, such as their quality and price, as well as the terms and conditions of the sale. Procurement of Import license: Certain goods can be imported freely, while others require licensing. The importer must consult the current Export-Import (EXIM) policy to determine whether the goods he or she intends to import require import licensing. If goods can only be imported with a license, the importer must obtain an import license. Every importer (and exporter) in India must register with the Directorate General Foreign Trade (DGFT) or Regional Import Export Licensing Authority and obtain an Import Export Code (IEC) number. This number is required on the majority of import documents.

The Imports and Exports (Control) Act of 1947 governs the import trade in India. Without a valid import license, a person or company cannot import goods into India.

The Indian government declares its import policy in the Import Trade Control Policy Book, also known as the Red Book. Every importer must first determine whether or not he can import the goods he desires, as well as how much of a particular class of goods he can import during the time period covered by the relevant Red Book.

3. **Obtaining foreign exchange:** The supplier in an import transaction requests payment in a foreign currency because they are based overseas. Indian currency must be converted into foreign currency in order to make a payment in another currency. The Reserve Bank of India's Exchange Control Department oversees all foreign exchange transactions in India (RBI). Every

importer is required by the current regulations to obtain the approval of foreign currency. The importer must submit an application to a bank that the RBI has authorized to issue foreign currency in order to receive such a sanction. In accordance with the Exchange Control Act's guidelines, the application must be submitted in the prescribed format and include an import license.

The exchange bank endorses and forwards the applications to the Reserve Bank of India's Exchange Control Department. The Reserve Bank of India sanctions the release of foreign exchange after scrutinizing the application on basis of the Government of India's exchange policy in effect at the time of application. The importer obtains the necessary foreign exchange from the relevant exchange bank. It should be noted that, whereas import licenses, which are issued for a specific period of time, the exchange is only released for a specific transaction. Most restrictions have been lifted as the rupee has become convertible on a current account as the economy has liberalized.

4. **Placing order**

The importer places an import order or indents with the exporter for the supply of the specified products after obtaining the import license. The import order includes details about the cost, size, grade, and quality of the goods ordered, as well as packing, shipping, ports of departure and arrival, delivery schedule, insurance, and payment method instructions. In order to avoid any ambiguity and subsequent conflict between the importer and exporter, the import order should be carefully drafted.

It contains the importer's instructions regarding the quantity and quality of goods required, the method of forwarding them, nature of packing, mode of payment and price, and so on. Indentations are typically prepared in duplicate or triplicate. Indent types include open indent, closed indent, and confirmatory indent. Because all of the necessary particulars of goods, price, and so on are not mentioned in the indent, the exporter is free to complete the formalities at his own end. On the other hand, a closed indent is one that clearly states the full particulars of the goods, price, brand, packing, shipping, insurance, and so on. A confirmatory indent is one in which an order is placed subject to the importer's agent's confirmation.

5. Obtaining a letter of credit: If a letter of credit is the preferred method of payment between the importer and the overseas supplier, the importer must obtain one from its bank and send it to the supplier. A letter of credit, as previously mentioned, is a guarantee made by the bank of the importer that it will honor payment of export bills to the bank of the exporter up to a certain amount.

. Arrangement of finance: Prior to the arrival of the goods at the port, the importer should make arrangements to pay the exporter. To avoid grossly overpaying demurrages (fines) on imported goods that are lying uncleared at the port due to lack of payments, advanced planning for financing imports is required.

7. **Advice for shipment receipt**: The international supplier sends the importer the shipment advice after loading the goods onto the ship. Information regarding the shipment of goods is included in shipment advice. The shipment advice includes information such as the invoice number, bill of lading/airways bill number and date, vessel name with date, port of export, description of goods and quantity, and date of vessel sailing.

8. **Retirement of documents:** After the goods have been shipped, the overseas supplier gathers the required paperwork in accordance with the contract and letter of credit terms and gives it to their banker for negotiation with the importer in the manner indicated in the letter of credit. A bill of exchange, commercial invoice, bill of lading or airline bill, packing list, certificate of origin, marine insurance policy, etc. are generally included in a set of documents. Retirement of import documents refers to the acceptance of a bill of exchange for the purpose of receiving delivery of the documents. When retirement is finished, the bank will give the importer the import documents.

9. **Goods Arrival:** The international supplier ships the goods in accordance with the agreement. The person in charge of the carrier, whether it be a ship or an airline, notifies the person in charge at the dock or the airport of the arrival of goods in the importing country. The import general manifest document is presented by him. The specific details of the imported goods are listed in an import general manifest document. It is a document that the unloading of cargo is based.

10. **Custom Clearance**: After they cross Indian borders, all imported goods are required to go through customs clearance. A number of formalities must be completed in the somewhat time-consuming process of customs clearance. It is suggested that importers appoint C&F agents who are familiar with such formalities and play a crucial role in getting the goods cleared through customs.

• import documents:

letter of credit

When the buyer has agreed to provide a letter of credit as part of the payment terms, the buyer will apply to its local bank in its home country and a letter of credit will be issued. The seller should send instructions to the buyer before the letter of credit is opened, advising the seller as to the terms and conditions it desires. The seller should always specify that the letter of credit must be irrevocable. The bank in the buyer's country is called the issuing bank. The buyer's bank will contact a correspondent bank near the seller in the United States, and the U.S. bank will send a notice or advice to the exporter that the letter of credit has been opened. If the letter of credit is a

confirmed letter of credit, the U.S. bank is called the confirming bank; otherwise, it is called the advising bank. The advice will specify the exact documents that the exporter must provide to the bank in order to receive payment. Since the foreign and U.S. banks are acting as agent and subagent, respectively, for the buyer, the U.S. bank will refuse to pay unless the exact documents specified in the letter of credit are provided. The banks never see the actual shipment or inspect the goods; therefore, they are extremely meticulous about not releasing payment unless the documents required have been provided. The issuing bank and advising bank each have up to seven banking days to review the documents presented before making payment. When the exporter receives the advice of the opening of a letter of credit, the exporter should review in detail the exact documents required in order to be paid under the letter of credit.

Introduction to Letters of Credit;

Letters of credit are a payment mechanism, particularly used in international trade. The Seller gets paid, not after the Buyer has inspected the goods and approved them, but when the Seller presents certain documents (typically a bill of lading evidencing shipment of the goods, an insurance policy for the goods, commercial invoice, etc.) to his bank. The bank does not verify that the documents presented are true, but only whether they "on their face" appear to be consistent with each other and comply with the terms of the credit. After examination the bank will pay the Seller (or in LC terms the beneficiary of the letter of credit).

Example:

1) Buyer and Seller sign a purchase contract that stipulates payment by letter of credit. It is good practice to agree already in the purchase contract which documents the Seller/Beneficiary has to present.

❖ Documentary bill:

Documentary Bill is a bill of Exchange drawn payable overseas and accompanied by commercial documents such as Bills of Lading, invoice and insurance papers ❖ **Bill of exchange,**

Bill of Exchange

A bill of exchange is an order drawn by a person upon bank or another person asking the latter to make payment to a third party. Bill of Exchange is an important method of payment. It is prepared by the exporter and sent to the place of importer through a commercial bank along with the documents.

Foreign Bill of Exchange may be of two types:

(i) Ordinary or Clean Foreign Bill: The is an ordinary bill of exchange and drawn in case the exporter has sufficient confidence in the credit-worthiness of the importer.

This is also called Clean Foreign Bill of Exchange.

(ii) Documentary Bill: When necessary, documents to the title of the goods are also

sent along with the foreign bill of exchange, it is called Documentary Bill. A **Documentary bill may be of two types:**

(a) **Documents Against Payment:** In this case, the documents are delivered to the importer provided he makes the full payment. The exporter gives clear instruction to the bank that the documents should only be given provided the importer makes the full payment.

(b) **Documents Against Acceptance**: The exporter instructs the bank to deliver the documents to the importer provided the importer accepts the enclosed bill of exchange. Foreign Bill of Exchange bear the stamps of both the countries. The exporter should affix sufficient stamps while sending the bill and the acceptor of the bill also should affix the stamps according to the rules his country.

• bill of leading

BILL OF LADING

- The bill of lading is a document wherein the shipping company gives its official receipts for the goods shipped in its vessel and at the same time contracts to carry them to the port of destination.
- It is also a document of title to the goods and, as such, is freely transferable by endorsement and delivery.
- A bill of lading serves three main purposes As a document of title to the goods As a receipt from the shipping company As a contract for the transportation of goods.
- Each shipping company has its own Bill of lading.
- The exporter prepares the Bill of lading in the forms obtained from the shipping company or from the agents of the shipping company. The information continued in the bill of lading includes

    1. The date and place of shipment
    2. The name of the consignor
    3. The name and destination of the vessel
    4. Quantity and destination of the goods

5. The marks and numbers
6. The invoice number and the date of export
7. The gross weight and net weight
8. The number of packages

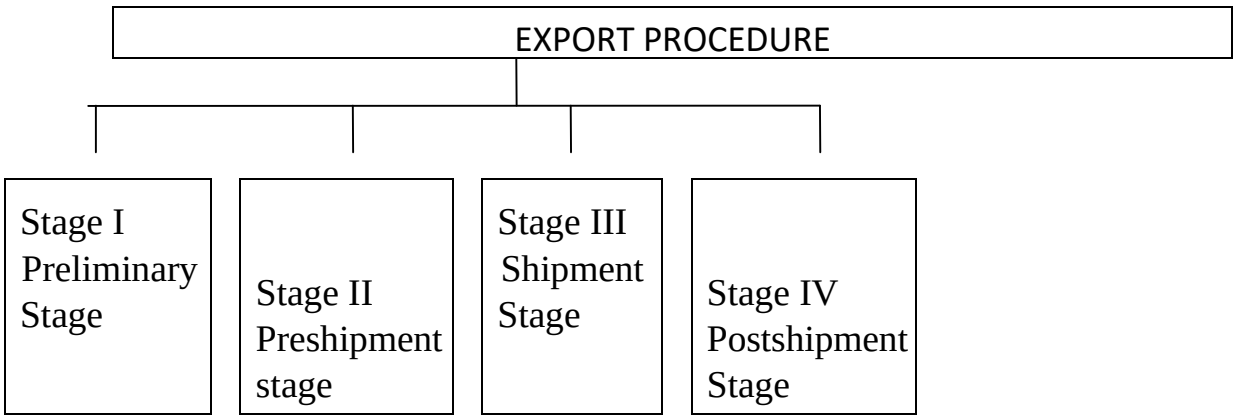
export procedure:

Export procedure refers to the execution of an order received from an overseas buyer and includes everything that the exporter is required to do right from the receipt of a confirmed order up to the realization, o/s final payment. It is not difficult to receive an export order but extremely difficult to successfully and satisfactorily execute the same. This is because exporting goods overseas involves same definite procedure and is covered by legal restrictions.

Export trade is governed by legal controls and therefore, every function of it is carried out under definite procedures. The various procedures that are followed in the export of goods facilitate execution of export in a systematic manner.

Export market is not merely an extension of domestic market. Apart from the basic principles of sound business in domestic as well as foreign market, selling abroad requires specialized knowledge regarding certain matters such as detailed market surveys, shipping, marine insurance, customs and foreign exchange formalities, etc.

Export procedure involves a number of steps. The various steps can be classified into four stages



12.2.1 PRELIMINARY STAGE

1) Organizing –

The exporter should have an organization to look after exports. Exporters may set up a completely new organization or add on export section to an existing one. At this stage the exporter may take a decision to select the right product sell abroad.

2) Registering with various Authorities –

The exporter should register his organizations with various authorities. These are as follows.

a) Income Tax Authorities to obtain permanent Account Number (PAN).

b) Jt. DGFT – to obtain Importer's exporter's code Number (IEC No.)

c) EPC – to obtain Registration – cum –membership – certificate (RCMC)

d) Other authorities, such as FIEO, sales Tax authorities, chambers of commerce etc.

3) Approaching Foreign Buyers –

The overseas agents / representatives approach foreign buyers with a quotation. The foreign buyer, if satisfied with the quotation and after clarifications, if any, will place an order with the exporter.

12.2.2 PRE-SHIPMENT STAGE

1) Confirmation of order –

When the buyer is satisfied with the terms and conditions of the seller, he will place either a formal or confirmed order along with a signed copy of the contract. The exporter should acknowledge and confirm the receipt of such order.

2) Obtaining pre-shipment Finance –

As soon as the exporter receives a confirmed order and the L/C, he should approach his bank for securing pre-shipment finance to meet his working capital requirements.

3) Obtaining Export License, if necessary's –Export control is exercised to a limited extent in India. The problem of obtaining export license arises only in the case of a few controlled items. Otherwise, export business has been declined.

4) Production and Procurement of goods –

Soon after securing the pre-shipment advance from bank, the exporter has to arrange for production and procurement of goods for shipment. A manufacturer exporter himself undertakes the entire process of production.

5) Packing and Marketing –

After procuring the goods meant for export, the exporter has to arrange for proper packing and marking of the goods. Packaging must ensure proper protection of the goods. The packing material should be selected after considering the distance to be covered, mode of transportation, types of handling of the goods at ports etc.

6) Pre-shipment Inspection -

If the export cargo is subject to qualify control and reshipment inspection, the exporter should get in touch with EIA to obtain Inspection certificate.

7) Appointment of clearing and Forwarding Agent –

It is always advisable to appoint C & F agent to look after forwarding work which includes booking of shipping space, preparing and submitting various documents to customs.

❖ SHIPMENT STAGE

1) Reservation of space in the ship –

The exporter has to contact the shipping company well in advance for booking the required space in the vessel for shipment of his consignment. He has to provide necessary information as regards date of shipment, gross and net weight of each package, particulars of the importer and that of his own, arrival and departure date of the vessel etc. when shipping company accepts the exporter's request, the company or its agent issues "shipping order".

2) Preparation and processing of shipping documents –

When goods reach the port of shipment, the exporter has to arrange for preparation of a complete set of documents to be passed on to the forwarding agent.

3) Physical Examination of goods at the port –

The C & F agent obtains the carting order from the Port Trust to Cart the goods inside the docks. He then approaches the custom Examiner, who may physically inspect the goods. The custom Examiner then given "Let Export order".

4) Loading of goods –

The duplicate copy of shipping bill which is endorsed by the custom
Examiner is handed over to the custom preventive officer, who endorses it with
'Let ship order'. The goods are then loaded on board the ship, for which Mate's Receipt is issued by the mate of the ship. The Mate's Receipt is handed over to the shipping company to obtain Bill of Lading.

❖ POST-SHIPMENT STAGE.

1) Shipmen advice to importer –

After the shipment of goods, the exporter has to send suitable intimation to the importer for his information. By this intimation, the date of shipment, the name of the vessel, date on which the goods will reach the destination should be informed to the importer. A copy of non-negotiable bill of lading is also sent for information. The importer gets the remaining documents through his bank.

2) Presentation of Documents to the Bank –

A complete set of documents is submitted by the exporter tohis bank for the purpose of negotiating the same and obtaining export proceeds in time. The bank then sends the same documents to the exporter.

3) Realization of export proceeds –

The exporter then proceeds to claim export incentives on the basis of bank certificate. The bank certificate gives description of the product, its value, the rate of conversion, the details of invoice etc. The exporter is entitled to various incentives such as IPRS, DBK and other incentives, it applicable.

4) Follow-up of Export sales –

A good exporter should always have a follow-up after sales i.e. he should provide necessary after sale service, find out buyer's opinion towards the product and so an, which will help to generate more sales in the international market.

• Shipments Procedure.

Meaning:

Exports are the goods and services produced in one country and purchased by residents of another country. It doesn't matter what the good or service is. It doesn't matter how it is sent. It can be shipped, sent by email, or carried in personal luggage on a plane. If it is produced domestically and sold to someone in a foreign country, it is an export.

Procedures:

The exporting activity involves several commercial and regulatory procedures. The export documentation involves the preparation or specified number of copies of the prescribed documents pertaining to the different procedures.

Step 1. Receipt of an Order

The exporter of goods is required to register with various authorities such as the income tax department and Reserve Bank of India (RBI). In addition to this, the exporter has to appoint agents who can collect orders from foreign customers (importer). The Indian exporter receives orders either directly from the importer or through indent houses.

Step 2. Obtaining License and Quota

After getting the order from the importer, the Indian exporter is required to secure an export license from the Government of India, for which the exporter has to apply to the Export Trade Control Authority and get a valid license. You can get a license from here too. The quota is referred to as the permitted total quantity of goods that can be exported.

Step 3. Letter of Credit

The exporter of the goods generally ask the importer for the letter of credit, or sometimes the importer himself sends the letter of credit along with the order.

Step 4. Fixing the Exchange Rate

Foreign exchange rate signifies the rate at which the home currency can be exchanged with the foreign currency i.e. the rate of the Indian rupee against the American Dollar. The foreign exchange rate fluctuates from time to time. Thus, the importer and exporter fix the exchange rate mutually.

Step 5. Foreign Exchange Formalities

An Indian exporter has to comply with certain foreign exchange formalities under exchange control regulations. As per the Foreign Exchange Regulation Act of India (FERA), every exporter of the goods is required to furnish a declaration in the form prescribed in a manner.

The declaration states:

I. The foreign exchange earned by the exporter on exports is required to be disposed of in the manner specified by RBI and within the specified period.

II. Shipping documents and negotiations are required to be done through authorized dealers in foreign exchange.

III. The payment against the goods exported will be collected through only approved methods.

Step 6. Preparation for Executing the Order

The exporter should make required arrangements for executing the order:

I. Marking and packing of the goods to be exported as per the importer's specifications.

II. Getting the inspection certificate from the Export Inspection Agency by arranging the pre-shipment inspection.

III. Obtaining insurance policy from the Export Credit Guarantee Corporation (ECGC) to get protection against the credit risks.

IV. Obtaining a marine insurance policy as required.

V. Appointing a forwarding agent (also known as custom house agent) for handling the customs and other related matters.

Step 7. Formalities by a Forwarding Agent

The formalities to be performed by the agent include –

I. For exporting the goods, the forwarding agent first obtains a permit from the customs department.

II. He must disclose all the required details of the goods to be exported such as nature, quantity, and weight to the shipping company. III. The forwarding agent has to prepare a shipping bill/order.

IV. The forwarding agent is required to make two copies of the port challans and pays the dues.

V. The master of the ship is responsible for the loading of the goods on the ship. The loading is to be done on the basis of the shipping order in the presence of customs officers.

VI. Once the goods are loaded on the ship, the master of the ship issues a receipt for the same.

Step 8. Bill of Lading

The Indian exporter of the goods approaches the shipping company and presents the receipt copy issued by the master of the ship and in return gets the Bill of Lading. Bill of lading is an official

receipt which provides the full description of the goods loaded on the ship and the name of the port of destination.

Step 9. Shipment Advise to the Importer

The Indian exporter sends shipment advice to the importer of the goods so that the importer gets informed about the dispatch of the goods. The exporter sends a
copy of the packing list, a non-negotiable copy of the Bill of Lading, and commercial invoice along with the advice note.

Step 10. Presentation of Documents to the Bank

The Indian exporter confirms that he possesses all necessary shipping documents namely; Microinsurance policy. The Consular Invoice Certificate of Origin the Commercial Invoice the Bill of LadingThen the exporter draws a bill of exchange on the basis of the commercial invoice. The bill of exchange along with these documents is called Documentary Bill of Exchange. The exporter then hands over the same to his bank.

Step 11. The Realization of Export Proceeds

In order to realize the proceedings of the export, the exporter of the goods has to undergo specific banking formalities. On submission of the bill of exchange, these formalities are initiated. Generally, the exporter receives payment in foreign exchange.

- **EXPORT DOCUMENT RELATED TO SHIPMENT.**

INTRODUCTION

The export process is made as difficult complex by the wide variety of documents that the exporter needs to complete to ensure that the order reaches its destination quickly, safety and without problems. For shipment process, certain certificates will be required.

MATE RECEIPT

A Mate Receipt is a receipt issued by the Commanding Office of the ship when the cargo is loaded on the ship and contains information about the

1. Name of the vessel
2. Berth
3. Date of shipment
4. Description of packages
5. Marks and numbers
6. Conditions of the cargo at the time of receipt onboard the ship etc.

- The mate receipt is first handed over to the Port Trust authorities so that all the port dues mat be paid by the exporter.
- After paying all the port dues, the merchant or the agent may collect the mate receipt from the Port Trust authorities.
- The bill of lading is prepared by the shipping agent after the mate has been
- obtained.

SHIPPING BILL:

- The shipping bill is the main document on the basis of which the Customs' permission for export s given.

The shipping bill contains

1. Particulars of the goods exported
2. The name of the vessel master or agent's flag
3. The port at which goods are to the discharged
4. The country of final destination, the exporter's name and address etc. It also contains details of the packages and the goods, such as number and description, marks and numbers, quantity details.

The following three forms of the shipping bill are available with the customs authorities:

1. Dutiable shipping bill for goods for which there is export duty.
2. Free shipping bill for goods for which there is no export duty.
3. Drawback shipping bill which is required for claiming the Customs drawback against exported goods.

CART TICKET

A cart ticket also known as a cart chit, vehicle and gate pass, is prepared by the exporter and includes details of

1. Export cargo in terms of the shipper's name
2. The number of packages
3. The shipping bill number
4. The port of destination
5. 5.The number of the vehicle carrying the cargo.

The driver of the vehicle cargo should possess the ticket and the vehicle is brought at port gate, it should be presented to the gate warden/inspector/keeper along with other shipping and port documents.

The gate keeper/warden/inspector, after satisfying himself that the vehicle is carrying the cargo as mentioned in the document allows it to pass the gate.

CERTIFICATE OF MEASUREMENT

- Freight is charged either on the basis of weight or measurement.
- When it is charged on the basis of weight, the weight declared by the shipper may be accepted.
- However, a certificate of measurement from the Indian Chamber of Commerce or other approved organization may be obtained by the shipper and given to the shipping company of the calculation of the necessary freight.

The certificate contains

1. The name of the vessel
2. Port and destination
3. The description of goods
4. The quantity, length, breadth, depth, etc of the packages.

BILL OF LADING

- The bill of lading is a document wherein the shipping company gives its official receipts for the goods shipped in its vessel and at the same time contracts to carry them to the port of destination.
- It is also a document of title to the goods and, as such, is freely transferable by endorsement and delivery.
- A bill of lading serves three main purposes As a document of title to the goods As a receipt from the shipping company As a contract for the transportation of goods.
- Each shipping company has its own Bill of lading.
- The exporter prepares the Bill of lading in the forms obtained from the shipping company or from the agents of the shipping company. The information continued in the bill of lading includes

10. The date and place of shipment
11. The name of the consignor

12. The name and destination of the vessel
13. Quantity and destination of the goods
14. The marks and numbers
15. The invoice number and the date of export
16. The gross weight and net weight
17. The number of packages
18. The amount of freight etc.

AIRWAY BILL

- An airway bill, also called an air consignment note is a receipt issued by an airline for the carriage of goods.
- As each shipping company has its own bill of lading each airline has its own airway bill.
- The same details as mentioned in Bill of Lading (for ship) are mentioned in the Airway bill.
- The goods may actually take off after a cooling period of 48 hours for Aircraft safety.

DOCUMENTS RELATED TO PAYMENT:

1. Letter of credit
2. Bill of exchange
3. Trust receipt
4. Letter of hypothecation
5. Bank certificate of payment

1. Letter of credit:

Letter of credit is the payment method commonly used in all international trade. A Letter of Credit is an arrangement whereby Bank acting at the request of a customer (Importer

/ Buyer), undertakes to pay for the goods/services, to a third party (Exporter / Beneficiary) by a given date, on documents being presented in compliance with the conditions laid down. It is also known as Documentary Credits.

2. Bill of Exchange:

The Bills of Exchange Act 1914 in Mauritius, defines a bill of exchange as "an unconditional order in writing, addressed by one person to another, signed by the person giving it, requiring the person to whom it is addressed to pay on demand, or at a fixed or determinable future time, a sum certain in money to or to the order of a specified person, or to bearer.

3. Trust receipt:

If the buyer fails or unable to pay for the goods arrived, the buyer's bank may make the goods available to the buyer by making an agreement after the buyer deal or sign with the trust receipt. By this agreement, the buyer is allowed to sell the goods and the buyer acts as an agent for the bank. The buyer didn't own the ownership of the goods until the full payment is done to the bank.

4. Letter of hypothecation:

If the importer (buyer) fails or refuses to pay for the shipped goods, the exporter may submit the document (letter of hypothecation) to the bank that the exporter gives full power to sell the goods and get paid from the bank.

5. Bank certification of payment:

This is issued by the negotiating bank of the exporter, proves that all bill covering the shipment has been negotiation and export procedures are completed by the exporter. The payment is received in accordance with exchange control regulations in the approved manner.

❖ Export documentation:

DOCUMENTS RELATED TO GOODS

(i) Preform invoice is the starting point of an export contract. As and when the exporter receives the trade inquiry from the importer, exporter submits the Preform invoice to the importer The Preform invoice contains details such as name and address of the exporter, name and address of the intending importer, nature of goods, mode of transportation, unit price in terms of internationally accepted quotation, name of the country of origin of goods, name of the country of final destination, period required for executing contract after receipt of confirmed order and finally signature of the exporter. Importance and Significance of Preform Invoice are Two-Fold.

(A) It forms basis of all trade transactions and further negotiation or contract is made on this basis.

(B) It helps the importer to obtain the import license, where required, and obtain Commercial Invoice

A commercial invoice is the seller's bill for merchandise or goods sold by him.
Invoice contains all the particulars and details in respect of name and address of seller (exporter), name and address of buyer (importer), date, exporter's reference number, importer's reference number, description of goods, price per unit at particular location, quantity, total value, packing specifications, terms of sale (FOB, CIF etc.), identification marks of the package,

total number of packages, name and number of the vessel or flight, bill of lading number, place and country of destination, country of origin of goods, reference to letter of credit, if opened, terms of payment, and finally signature of the exporter etc. From the details, it is clear that invoice is an important and basic export document. It is also known as 'DOCUMENT OF CONTENTS' as it contains all the important information necessary for the preparation of other export documents. For many countries, there are no prescribed special invoice forms. Exporters can use their normal invoices used for indigenous trade for exports made outside the country too and

show the particulars required by the importer in terms of the contract. However, there are special invoicing procedures in respect of exports to certain countries like Canada, Usmaan Australia. Some countries like Uganda, Mexico, Sudan and Tanzania require special customs invoices. Information about the special invoice forms required can be gathered from the respective Export Promotion Councils apart from the procedures of trade to be followed in respect of the importer's country. Any recognized Chamber of Commerce too can provide the information in this respect.

Significance of Commercial Invoice

(A) It is prima facie evidence of the contract of sale and purchase of goods. On the basis of the invoice, all the other documents, in the context of export, are prepared as it is the basic document.

(B) Invoice constitutes the main document for various export formalities such as pre-shipment inspection, quality, excise and customs procedures.

(C) It is useful for accounting purposes, both by the exporter and importer.

(D) This document is required in collection/negotiation of documents through the bank.

(E) For claiming incentives, this document is essential.

(iii) Consular Invoice

Some of the importing countries insist that the invoice is to business by the importing county's consular located in the exporter's country. Such invoices are known as consular invoice. The exporter has to pay a certain fee to obtain the certificate/invoice. Such charges/fees vary from country to country. The main purpose to obtain consular invoice is to secure authentication of information contained in the invoice. Once the invoice is signed by the consular of the country, the importer gets comfort and confidence in respect of accuracy of information in respect of quality, source of goods, volume and grade.

- **inspection and excisable goods.**

DOCUMENTS RELATING TO INSPECTION

Certificate of Inspection

It is a certificate issued by the Export Inspection Agency certifying that the consignment has been inspected under the Export (Quality Control and
Inspection) Act, 1963 and found that the requirements relating to quality control and inspection have been complied with, as applicable, and the goods are export worthy.

DOCUMENTS RELATED TO EXCISABLE GOODS (1) GP Forms

GP stands for Gate Pass. A GP form, gate pass, is issued for the removal of excisable goods from the factory or warehouse. Form GP1 is issued for the removal of excisable goods on payment of duty. GP2 is issued for the removal of excisable goods without payment of duty.

Form C

It is not to be confused with C form. Form C is used for applying for rebate of duty on excisable goods (other than vegetable, non-essential oils and tea) exported by sea. It is to be submitted, in triplicate, to the Collector of Central Excise.

(3) Forms AR4/AR4A

These forms are meant for removal of excisable goods for export by sea/post. Now, in their place, ARE-1 form is used.

Name of Course	**Bachelor of Computer Application (BCA)**
Semester	**III**
Name of Subject	**Database Management System**
Subject Code	**BCA-303**
Marks	**75 Marks**
Lectures	**50 Lectures**

Database Management System Objectives:

The objective of the course is to enable students to understand and use the concepts of Data, Database Systems, DBMS, and Applications of DBMS. Understand the structure of DBMS, various Data Models, designing relational database systems, using relational algebra and Normalization.

Database Management System Outcomes:

- Able to master the basic concepts and understand the applications of database systems.
- Able to construct an Entity-Relationship (E-R) model from specifications and to transform to relational model.
- Able to construct unary/binary/set/aggregate queries in Relational Algebra.
- Understand and apply database normalization principles.

UNIT - I

1. **Introduction to DBMS** **20 Lectures**
   - 1.1 Introduction, Definition & application area.
   - 1.2 Characteristics of DBMS.
   - 1.3 File processing system Vs DBMS.
   - 1.4 Advantages and Disadvantages of DBMS.
   - 1.5 Users of DBMS.
   - 1.6 Structure of DBMS.
   - 1.7 View of Data –Data abstraction, Instances and Schemas.
   - 1.8 Database Languages
   - 1.9 Index- Introduction, Types of Index
   - 1.10 File Organization-Introduction, Types of file organization.
   - 1.11 Data Models-Introduction, Types of Data Models.

UNIT - II

2. E-R Data Models: **10**

Lectures

2.1 Introduction

2.2 Basic Concepts-Entity and Entity Sets, Attributes and types of attributes

2.3 Relationship and relationship sets

2.4 Constraints, Keys

2.5 Entity-Relationship Diagram –Introduction and basic components of E-R diagram.

2.6 Extended E-R Features: Specialization, generalization, higher- and lower-level entity sets, attribute inheritance, and aggregation.

2.7 Example of E-R data base design.

UNIT - III

3. Relational Data Model: **10**

Lectures

3.1 Introduction to Relational Data Model.

3.2 Structure of Relational databases and Important Terms-Relation, Tuple, Attribute, Cardinality, Degree, Domain

3.3 Integrity constraints over relations

3.4 Logical Database Design: ER to Relational

3.5 The Relational Algebra: Select, Project, Union, Difference, Intersection, Cartesian Product, Natural Join

UNIT - IV

4. Relational Database Design: **10**

Lectures

4.1 Introduction

4.2 Anomalies of un-normalized database

4.3 Dependencies in Database

4.4 Normalization

4.5 Normal Form: 1NF, 2NF, 3NF, BCNF

References:

Name of Course	**Bachelor of Computer Application (BCA)**
Semester	**III**
Name of Subject	**Lab-Course : C++ Programming**
Subject Code	**BCA-306**
Marks	**50 Marks**

1) Program to demonstrate C++ programming Structure.
2) Program to demonstrate use of different data types.
3) Program to demonstrate Use of Scope Resolution Operator.
4) Program to demonstrate Use of arrays.
5) Program to demonstrate Use of conditional statements.
6) Program to demonstrate Use of looping statements.
7) Program to demonstrate Use of call by value.
8) Program to demonstrate Use of call by reference.
9) Program to demonstrate the concept of inline function
10) Program to demonstrate the concept of default arguments.
11) Program to demonstrate the concept of Function Overloading
12) Program to demonstrate the concept of static members.
13) Program to demonstrate the concept of constructor and destructor.
14) Program to demonstrate the concept of friend function.
15) Program to demonstrate unary operator overloading.
16) Program to demonstrate binary operator overloading.
17) Programs to demonstrate different types of inheritance.
18) Program to demonstrate concept of virtual base class.
19) Program to demonstrate different formatted and unformatted I/O operations.
20) Program to demonstrate reading from and writing in to the files.

Name of Course	**Bachelor of Computer Application (BCA)**
Semester	**III**
Lab/Practical	**Lab Course : DBMS**
Subject Code	**BCA-307**

Marks	50 Marks

This laboratory course will comprise as exercises to supplement what is learnt under paper BCA-303: Data Base Management System. Students are required to practices following:

1) Write procedure for creating database in MS-Access.
2) Create tables in MS ACCESS using different ways.
3) Perform Import data operation in MS ACCESS.
4) Perform export data operation in MS ACCESS.
5) Create queries in MS ACCESS for selection, projection, Cartesian product.
6) Create queries in MS ACCESS for union, intersection and difference.
7) Create queries in MS ACCESS for different types of joins.
8) Generate forms and add new records in MS-Access.
9) Generate the report in MS Access.
10) Generate the report in MS Access using Report Wizard.

Unit –I Introduction to DBMS

Basic concepts and definition:

With the growing use of computers the organisations are fast migrating from a manual system to a computer computerized information system for which the data within the organisation is a basic resource.

Therefore proper organisation and management of data is necessary to run the organisation efficiently the efficient use of data for planning, production control, marketing, invoicing, payroll accounting and other functions in an organisation have a major impact for its competitive edge.

So in this section we will see some formal definitions of the terms used in database.

What is Data?

Data may be defined as known facts that can be recorded and that have implicit meaning. Data raw or isolated facts from which the required information is produced. Data are distinct pieces of information usually formatted in a special way.

Data can exist in a variety of forms that have meaning in the user environment:

- Such as phone numbers or text on a piece of paper.
- Bits or bytes stored in computer memory.
- Facts stored in persons mind.
- Data can also be objects such as documents, photographic images and even video segments.

Examples of Data:

In Salesperson's view: Customer_name, Customer_account, address, telephone_numbers etc.

In Electricity suppliers view: Customer_name, Customer_nummber, address, telephone numbers,

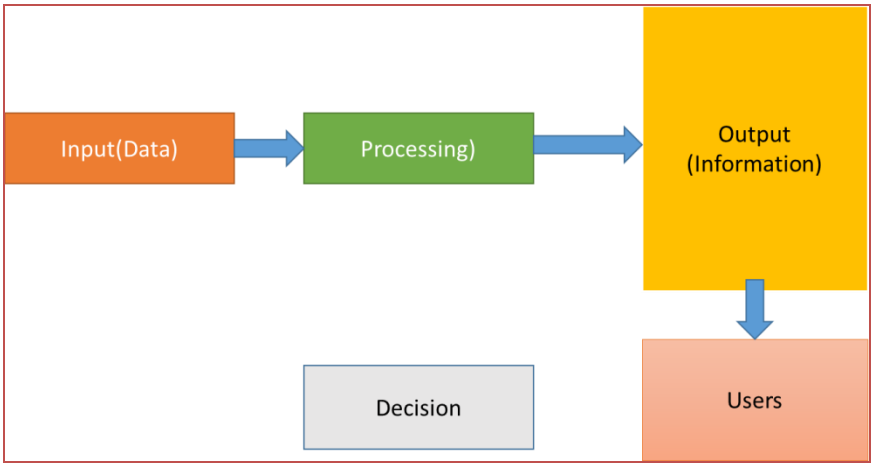
unit consumed, amount_payable

What is Information?

Data and information are closely related and are often used interchangeably. Information is processed, organized and summarized data.

It may be defined as collection of related data that will put together communicate meaningful and useful message to a recipient who uses it to make decisions or to interpret data to get the meaning.

Data process to create information which is meaningful to the recipient as shown in following:



What is Database?

Database is defined as a collection of logically related data stored together that is design to meet the information needs of an organisation. It is basically an electronic filing cabinet which contains computerized data files.

It can contain one data file (a very small database) or large number of data files (a large database) depending on organizational needs.

A database is organized in such a way that a computer program can quickly select the desired pieces of data.

Database can further be defined as, it

- Is a collection of interrelated data stored together without harmful or unnecessary redundancy.

- Serves multiple applications in which each user has its own view of data.
- Stores data independent of programs and changes in data storage structure or access strategy do not require changes in accessing programs or queries.
- Has structured data to provide a foundation for growth and controlled approach is used for adding new data, modifying and restoring data.

What is database management system?

A database system also called database management system is a generalized software system for manipulating data bases.
It is basically a computerized record keeping system, which stores information and allows users to add, delete, change, retrieve and update that information on demand.

It provides for simultaneous use of database by multiple users and tool for accessing and manipulating the data in the database. DBMS is also a collection of programs that enables users to create and maintain databases.

It is a general purpose software system that facilitates:

- The process of defining(specifying the data types, structures and constraints),
- Constructing(processing of storing data on storage media)
- And manipulating (queries to retrieve specific data, updating to reflect changes and generating reports from the data) for various applications.

Characteristics of DBMS

Following are some important characteristics Databases Management System:

1. Real World Entity
2. Self-Describing Nature
3. Support ACID Properties
4. Concurrent Use of Database
5. Insulation Between Data and Program
6. Transactions
7. Data Persistence
8. Backup and Recovery
9. Data Integrity
10. Multiple Views
11. Stores Any Kind of Data
12. Security
13. Represents Complex Relationship Between Data
14. Query Language

1. Real World Entity

DBMS these days is very realistic and real-world entities are used to design its architecture. Also behavior and attributes are used by DBMS.

To simplify it we can take an example of an organization database where employee is an entity and his employee id is an attribute.

2. Self-Describing Nature

Before DBMS, traditional file management system was used for storing information and data. There was no concept of definition in traditional file management system like we have in DBMS.

A DBMS should be of Self- Describing nature as it not only contains the database itself but also the metadata. A metadata (data about data) defines and describes not only the extent,

type, structure and format of all data but also relationship between data. This data represent itself that what actions should be taken on it.

3. **Support ACID Properties**

   Any DBMS is able to support ACID (Accuracy, Completeness, Isolation, and Durability) properties. It is made sure in every DBMS that the real purpose of data should not be lost while performing transactions like delete, insert and update.

   Let us take an example; if an employee name is updated then it should make sure that there is no duplicate data and no mismatch of employee information.

4. **Concurrent Use of Database**

   There are many chances that many users will be accessing the data at the same time. They may require altering the database system concurrently. At that time, DBMS supports them to concurrently use database without any problem. With the help of concurrency, economy of the system can be increased.

   For Example, employees of railway reservation system can book and access tickets for passengers concurrently. Every employee can see on his own interface that how many seats are available or bogie is fully booked.

5. **Insulation Between Data and Program**

   Program-data independence provides a big relief to database users. In traditional file management system, structure of data files was defined in the application programs so user had to change all the programs that are using that particular data file.

   But in DBMS, structure of data files is not stored in the program but it is stored in system catalogue. With the help of this, internal improvement of data efficiency or any changes in the data do not have any effect on application software.

6. **Transactions**

   Transactions are bunch of actions that are done to bring database from one consistent state to new consistent state. Traditional file-based system did not have this feature. Transaction is always atomic that means it can never be further divided. It can only be completed or uncompleted.

   For example, A person wants to credit money from his account to another person's account.

Then transaction will be complete if he sends money and other guy receives his money.

Anything other than this can lead to an inconsistent transaction.

7. **Data Persistence**

Persistence means if the data is not removed explicitly then all the data will be maintained in DBMS.

If any system failure happens then life span of data stored in the DBMS will be decided by the users directly or indirectly .Any data stored in the DBMS can never be lost. If system failure happens in between any transaction then it will be rolled back or fully completed, but data will never be at risk.

8. **Backup and Recovery**

There are many chances of failure of whole database. At that time no one will be able to get the database back and for sure company will be in a big loss.

The only solution is to take backup of database and whenever it is needed, it can be stored back. A database must have this characteristic to enable more effectiveness.

9. **Data Integrity**

This is one of the most important characteristics of database management system. Integrity ensures the quality and reliability of database system. It protects unauthorized access of database and makes it more secure. It brings only consistence and accurate data into the database.

10. **Multiple Views**

Users can have multiple views of database depending on their department and interest.

DBMS support multiple views of database to the users.

11. **Stores Any Kind of Data**

A database management system should be able to store any kind of data. It should not be restricted to employee name, salary and address. Any kind of data that exists in the real world can be stored in DBMS because we need to work with all kinds of data that is present around us.

12. Security

DBMS provides security to the data stored in it because all users have different rights to access database. Some of the user can access the whole database while other can access a small part of database.

For example, a computer network lecturer can only access files that are related to computer subjects but HOD of the department can access files of all subject that are related to their department.

13. Represents Complex Relationship Between Data

Data stored in a database is connected with each other and a relationship is made in between data. DBMS should be able to represent the complex relationship between data to make efficient and accurate use of data.

14. Query Language

Queries are used to retrieve and manipulate data but DBMS is armed by a strong query language that makes it more effective and efficient. Users have the power to retrieve any kind of data they want from database by applying different set of queries. File-Based system has not this luxury of query language.

Traditional File Processing System Vs Database System

Computer-based data processing systems where initially used for scientific and engineering calculations. With increased complexity of business requirements, gradually they were introduced into business applications.

In this discussion, we will discuss:

- Manual method of filing system
- Computer based filing system
- Database system approach

Manual filing system:

The manual method office filing systems of an organisation, such as to hold all internal and external correspondence relating to a project or activity, client, task, product, customer or employee, was maintaining different manual folders. These files or folders where labeled and stored in one or more cabinets or almirahs under lock and key for safety and security reason.

As and when required the concern person in the organisation used to search for a specific folder or file serially starting from the first entry. Alternatively files are indexed to help locate the file or folder more quickly. Ideally, the contents of each file folder where logically related.

For example, a file folder in a suppliers office might contain customer data; one file folder for each customer. All the data in that folder described only that customer's transaction.

Similarly a personal manager might organize personal data of employees by category of employment (for example, technical secretarial, sales, administrative and so on).Therefore, a file folder labeled **technical** would contain data pertaining to only those people whose duties where properly classified as technical.

The manual system worked well as data repository as long as the data collection where relatively small and the organisations managers had few reporting requirements. The manual system worked well as data repository as long as the data collection where relatively small and the organisations managers had few reporting requirements.

However as organisation grew and as the reporting requirements became more complex , it is difficult it became difficult in keeping track of data in manual file system. Also report generation from a manual file system could be slow and cumbersome. Thus, this manual filing system was replaced with a computer based filing system.

Computer-based filing system:

File oriented system where and early attempt to computerized the manual filing system that we are familiar. Because these systems performed normal record keeping functions, they were called data processing (DP) systems.

Rather than establish a centralized store for organisations operational data, a decentralized approach was taken where each department with the assistance of data processing department staff, stored and controlled its own data.

Following tables shows an example of file-oriented system of an organisation engaged in production distribution. Each table represents a file in the system, for example, PRODUCT file, CUSTOMER file, SALES file and so on. Each row in these files represents a record in the file.

PRODUCT file contains 6 records and each of these records contains data about different products.

PRODUCT (PRODUCT-ID,PRODUCT-DESC,MANUF-ID,UNIT-COST).

Table 1.5 File-oriented system

PRODUCT			
PRODUCT-ID	PRODUCT-DESC	MANUF-ID	UNIT-COST
A12345	Steel almirah	100	4000
B23412	Dryer	200	4500
B44332	Freeze	210	6000
A98765	Steel table	105	3500
A29834	Steel chair	110	4800
C11008	Iron moulding	444	5100

CUSTOMER file contains 5 records and each of these records contains data about customer.

CUSTOMER (CUST-ID,CUST-NAME,CUST-ADDRESS,COUNTRY,TEL-NO,BAL-AMT).

CUSTOMER

CUST-ID	CUST-NAME	CUST-ADDRESS	COUNTRY	TEL-NO	BAL-AMT
1001	Waterhouse Ltd.	Box 41, Mumbai	India	2147015	45,000
1000	KLY System	41, 1st Street, Chicago	USA	2000894	33,550
1005	Megapoints	C-12, Pataya, Goa	India	2222009	74,314
1010	Concept Shapers	32, Main Road, Ranchi	India	4598733	49,444
1006	Trinity Agencies	P.O. Box 266, Tokyo	Japan	2345678	55,542

SALES file contains 5 records and each of these records contains data about sales activities.

SALES(SALE-DATE,CUST-ID,PROD-ID,QTY,UNIT-PRICE).

SALES

SALE-DATE	CUST-ID	PROD-ID	QTY	UNIT-PRICE
02/12/02	1001	A12345	100	6,700
10/10/02	1000	B23412	250	4,000
12/12/03	1010	B44332	120	14,000
01/04/04	1005	A98765	110	5,500
30/02/04	1001	A29834	300	12,999

With the assistance of data processing department, the files where used for a number of different applications by the user departments, for example,

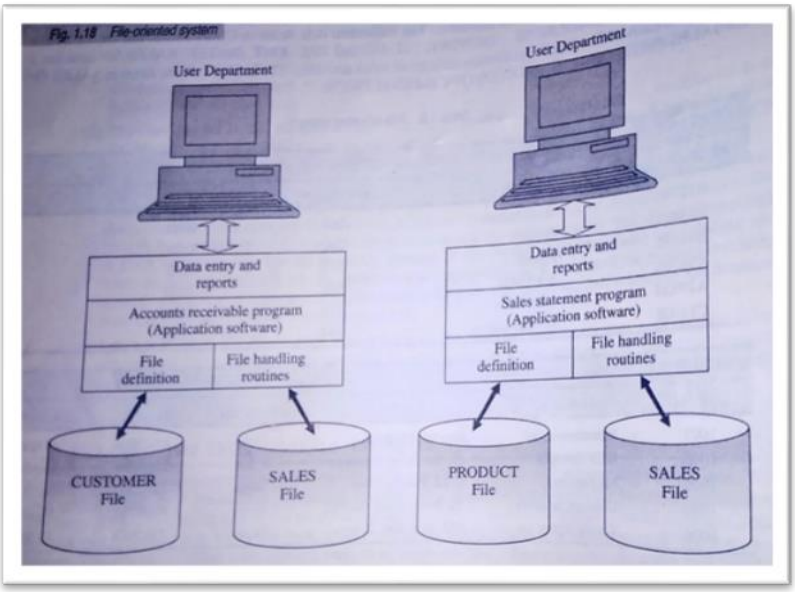
Account receivable program written to generate billing statements for customers. This program used the **CUSTOMER** and **SALES** files and these files were both stored in the computer in order by CUST-ID and were merged to create a printed statement.

Similarly, **sales statement generation program** which using **PRODUCT** and **SALES** files was written to generate product wise sales performance.

This type of program which accomplishes specific task of practical value in a business situation is called ***application program or application software****.*

Each application program that is developed is designed to meet the specific needs of particular requesting department for user group.

Following diagram illustrates structure in which application programs are written specifically for each user department for accessing their own files. Each set of departmental programs handles data entry file maintenance and the generation of fixed set of specific reports. Here the physical structure and storage of the data files and records are defined in the application program.



Here the physical structure and storage of the data files and records are defined in the application program.

For example,

- Sales department stores details relating to sales performance namely

  SALES (SALE-DATE, CUST-ID, PROD-ID, QTY, UNIT-PRICE).

- Customer department stores details relating to customer invoice realization summary, namely **CUSTOMER (CUST-ID, CUST-NAME, CUST-ADDRESS, COUNTRY, TEL-NO, BAL-AMT).**

- Product department stores details relating to product categorization summary, namely **PRODUCT (PRODUCT-ID, PRODUCT-DESC, MANUF-ID, UNIT-COST).**

It can be seen from the above examples that there is significant amount of **duplication** of data storage in different departments (for example **CUST-ID** and **PROD-ID**) which is generally true with file oriented system.

Advantages of file oriented system

Although the file oriented system is now largely obsolete following are several advantages of learning file based systems.

- It provides a useful historical perspective on how we handle data.
- Customer department stores details relating to customer invoice realization summary, namely CUSTOMER (CUST-ID,CUST-NAME,CUST-ADDRESS,COUNTRY,TEL-NO,BAL-AMT).
- Understanding the problems and knowledge of limitation inherent in the file based system helps to avoid the same problem when designing database system and thereby resulting in smooth transition.

Disadvantages of file oriented system:

Conventional file oriented system has the following disadvantages:

- **Data redundancy(or duplication)**

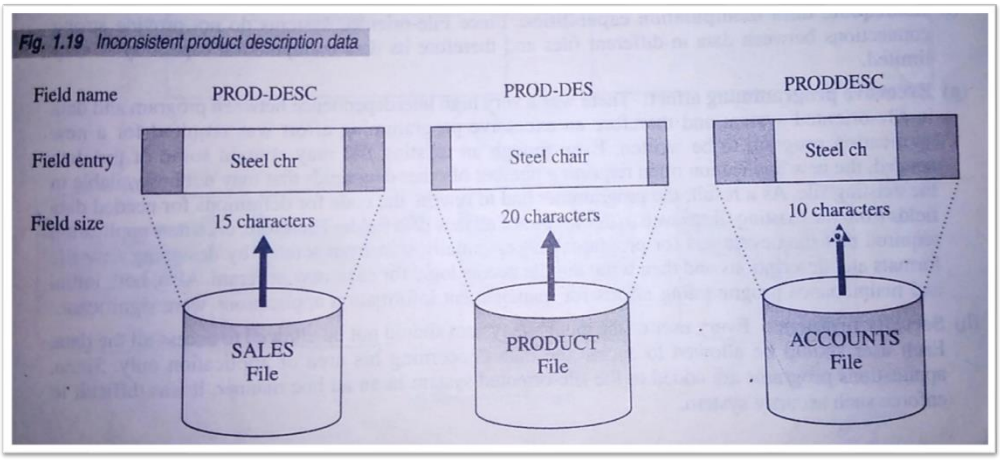
  Since a decentralized approach was taken each department used their own independent application programs and special files of data. This resulted into duplication of same data and information in several files, for example, duplication of PRODUCT-ID data in both PRODUCT and SALES files and CUST-ID data in both CUSTOMER and SALES files as shown in the table.

  This redundancy or duplication of data is wasteful and requires additional or higher storage space cost extra time and money and requires increased effort to keep all files up to date.

- **Data inconsistency (or loss of data integrity)**

  Data redundancy also leads to data inconsistency (or loss of data integrity), since other data formats may be inconsistent, or data values (various copies of the same data) may no longer agree or both.

Following diagram shows an example of data inconsistency in which field of product description is being shown by all three department files namely SALES, PRODUCT and ACCOUNTS.



- **Data inconsistency (or loss of data integrity)**

  It can be seen in this example that even though it was always the product description, the related field in all the three departments' files often had a different name for example Prod-DESC, PROD-DES, and PRODDESC.

  Also the same data field might have different length in various files, for example 15 characters in **SALES** files, 20 characters in **PRODUCT** file and 10 characters in **ACCOUNT** file.

  Furthermore suppose a product description was changed from "**Steel cabinet to Steel chair**". This duplication (or redundancy) of data increased the maintenance of overhead and storage cost. As shown in above diagram, the product description field might be immediately updated in the SALES file, updated incorrectly next week in PRODUCT file as well as ACCOUNT file.

  Over a period of time such discrepancies can cause serious degradation in the quality of information contained in the data files and can also affect the accuracy of reports.

- **Program-data dependence:**

  As we have seen file descriptions physical structure storage of the data files and records are defined within each application program that accesses a given file.

  For example, "**account receivable program**" exercise both **CUSTOMER** file and **SALES** file. Therefore this program contains a detailed file description for both these files.

As a consequence any change for a file structure, requiring changes to the file description for all programs that access the file. It can also be noticed that sales file that has been used in both "**account receivable program**" and "**sales statement program**". If it is decided to change **CUST-ID** field length from 4 characters to 6 characters, the file description in each program that is affected would have to be modified to confirm the new file structure.

It is often difficult to even locate all programs affected by such changes. It could be very time consuming and subject to error when making changes. These characteristics of file oriented system are known as **program-data dependence**.

- **Poor data control:**

  As shown in the above diagram file oriented system being decentralized in nature there was no centralized control at the data element (field) level. It could be very common for the data field to have multiple names defined the by various departments of an organisation and depending on the file it was in.This could lead to different meanings of a data frame in different context and conversely same meaning for different fields. This leads to poor data control resulting in a big confusion.

- **Limited data sharing:**

  There is limited data sharing opportunities with traditional file oriented system. Each application has its own private files and user has little opportunity to share data outside their own applications. To obtain data from several incompatible files in separate systems will require a major programming effort.

  In addition a major management effort may also be required since department Organisational units may own these different files.

- **Inadequate data manipulation capabilities**

  Since, file oriented systems do not provide strong connections between data in different files and therefore its data manipulation capability is very limited.

- **Excessive programming effort:**

  There was a very high interdependence between program and data in file oriented system and therefore and excessive programming effort was required for a new application program to be written. Even though an existing file may contain some of the data needed, the new application often requires a number of other data fields that may not be available in the existing file.

As a result, the programmer had to rewrite the code for definitions for needed data fields from the existing file as well as definitions for all new data fields. Therefore, each new application required that developers for programmers essentially start from scratch by designing new file formats and descriptions and then right the file access logic for each new program. Also both initial and maintenance programming efforts for management information applications were significant.

- **Security problems:**

  Every user of the database system should not be allowed to access all the data. Each user should be allowed to access the data concerning his area of application only. Since, application programs are added to the file oriented system in an Ad-Hoc manner it is very difficult to enforce such security system.

Database approach:

The problems inherent in file oriented systems make using the database system very desirable. Unlike file oriented system with its many separate and unrelated files the database system consists of logically related data stored in a single data dictionary.

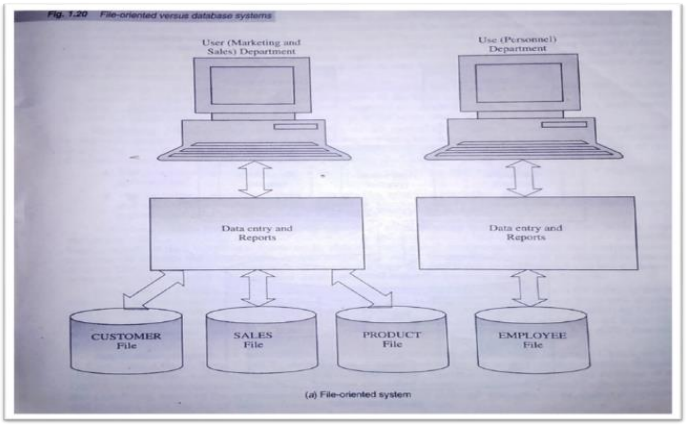
Therefore, the database approach represents the change in the way a user data are stored accessed and managed.

It emphasizes the integration and sharing of data throughout the organisation. Database systems overcome the disadvantages of file oriented system. The element problems related with data redundancy and data control by supporting and integrated and centralized data structure.

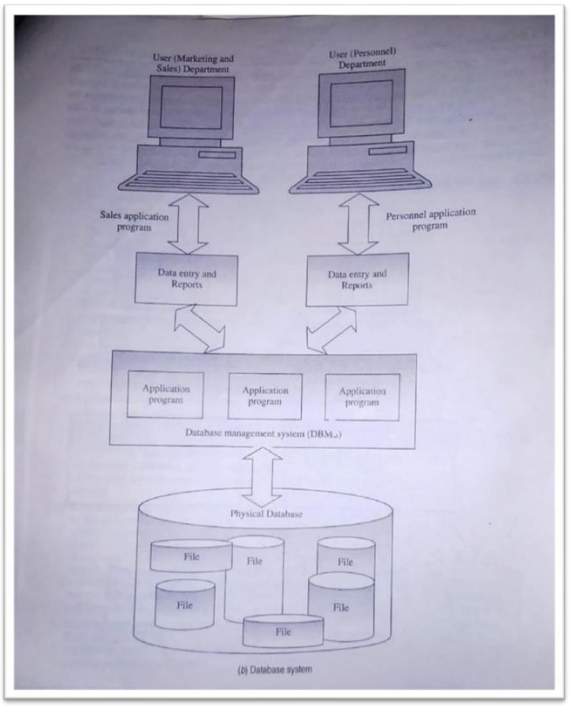
Data are control via data dictionary system which itself is controlled by database administrators.

Following diagram illustrates a comparison between file oriented and database systems:

File Oriented System



Database System



Advantages of Database Management System

Due to centralized management and control the database management system has numerous advantages. Some of these are as follows:

- **Minimal data redundancy**

  In database system views of different user groups i.e. data files are integrated during database design into a single logical centralized structure. By having a centralized database and centralized control of data by the database administrator the unnecessary duplication of data are avoided.

  Each primary fact is ideally recorded in only one place in the database. The total data storage requirement is effectively reduced. It also eliminates the extra processing to trace the required data in a large volume of data. Here all redundancy cannot be e eliminated completely. Some data is duplicated in various tables.

  In a database system however this redundancy can be carefully controlled. This is the DBMS is aware of it if it exist and assumes the responsibility of propagating updates and ensuring that the multiple copies are consistent.

- **Program data independence**

  The separation of Meta data (data description) from the application programs that used the data is called data independence. In the database environment it allows for changes at one level of database without affecting other levels.

  These changes are absorbed by the mapping between the levels. With database approach, Meta data are stored in a Central location called repository. This property of data systems allows organisations data to change and evolve (within limits) without changing the application programs that process the data.

- **Efficient data access**

  DBMS utilizes variety of sophisticated techniques to store and retrieve data efficiently.

  This feature is especially important if the data is stored on external storage devices.

- **Improved Data Sharing:**

Since, database system is a centralised repository of data belonging to the entire organisation including all departments it can be shared by all authorized users. Existing application programs can share the data in the database.

Furthermore new application programs can be developed on the existing data in the database to share the same data and add only that data that is not currently stored rather having to define all data requirements again. Therefore more users and applications can share more of the data.

- **Improved Data Consistency:**

  Inconsistency is the corollary to redundancy. In file oriented system when the data is duplicated and the changes made at once site are not propagated to the other side it results into inconsistency. Such database applies incorrect or contradictory information to its uses.

  So if the redundancy is removed or controlled chances of having inconsistent data is also remove and controlled. In database system such inconsistencies are avoided to some extent by making them known to DBMS. DBMS ensures that any change made to either of the two entries in the database is automatically applied to the other one as well. This process is known as propagating updates.

- **Improved Integrity:**

  Data integrity means that the data contained in the database is both accurate and consistent.

  Integrity is usually expressed in terms of constraints, which are consistency rules that the database system should not violet.

  For example: Mobile Number should be exact of 10 digits, only numbers are entered etc.

- **Improved Security**

  Database security is the protection of database from unauthorized users. The database administrator ensure that proper access procedure is followed including proper authentication schemes for access to the DBMS and additional checks before permitting access to sensitive data.

  A DBA can define which is in force by DBMS usernames and passwords to identify people authorized to use the database. Different levels of security could be implemented for various types of data and operations. The access of data by authorized user may be restricted for each type of access for example (retrieve insert modify update delete and so on) to each piece of information in the database.

The enforcement of security could be data **value dependent** (for example a works manager has access to the performance details of employees in his or her department only) as well as **data type dependent** but the manager cannot access the sensitive data such as salary details of any employees including those in his or her department.

- **Increased productivity of application development**

  The DBMS provides many of the standard functions that the application programmer would normally have to write in a file oriented application. It provides all the low-level file handling retains that are typical in application programs.

  The provision of these functions allows the application program to concentrate on the specific functionality required by the users without having to worry about the low level implementation details.

  DBMS also provide a high-level (4gl) environment consisting of productivity tools such as forms and report generator to automate some of the activities of database design and simplify the development of database applications. This results in increased productivity of the programmer and reduce development time and cost.

- **Economy of Scale:**

  Centralizing of all the organisations operational data into one database and creating a set of application programs that work on this source of data resulting in drastic cost savings. The DBMS approach permits consolidation of data and applications.

  This reduces the amount of a wasteful overlap between activities of data processing personal in different projects or departments.

  This enables the whole organisation to invest in more powerful processors storage devices or communication gear rather than having each department purchase its own low-end equipments. Thus a combined low cost budget is required instead of accumulated large budget that would be normally be allocated to each department for file oriented system for the maintenance and the development of the system.

  This reduces overall cost and up of operation and management leading to an economy of scale.

- **Enforcement of standards:**

  With the central control of the database a database administrator define stress and in forces the necessary standards. Application standards might include any or all of the following departmental, installation, Organisational, industry, corporate, national or international.

Standards can be defined for data formats to facilitate exchange of data between systems, naming conventions, display formats, report structures, terminology, documentation, standards, update procedures, access rules and so on.

This facilitates communication and cooperation among various departments' projects and users within the organisation. The data repository provides DBS wind power solar power full set of tools for developing and enforcing these standards.

- **Balance of conflicting requirements:**

  Knowing the overall requirements of the organisation instead of requirements of individual users, the DBA resolves the conflicting requirements of various users and applications. A DBA can structure the system to provide an overall service that is best for the organisation. A DBA can choose the best file structure and access method to get optimal performance for the response critical operations, while permitting less critical applications to continue to use the database with a relatively slower response.

  For example a physical representation can be chosen for the data in storage that gives fast access for the most important applications.

- **Improved data accessibility and responsiveness:**

  As a result of integration in database system data that crosses departmental boundaries is directly accessible to the end users. This provides a system with potentially much more functionality. Many DBMS provide query languages or report writers that allow users to ask ad hoc questions and to obtain the required information almost immediately at their terminal without requiring a programmer to write some software to extract this information from the database.

- **Increased Concurrency:**

  DBMS manage concurrent database access and prevents the problem of loss of information or loss of integrity.

- **Reduce program maintenance:**

  The problems of high maintenance efforts required in file oriented system are reduced in database system. In a file oriented environment the descriptions of data and the logic for processing data are built into individual application programs. As a result changes to data formats and access methods in inevitably result in the need to modify application programs.

  In database environment data are more independent of the application programs.

- **Improved backup and recovery services:**

  DBMS provides facilities for recovering from hardware or software failures through to its backup and recovery subsystem.

  For example if the computer system fails in the middle of complex update program the recovery system is responsible and makes sure that database is restored to the state it was in before the programme started executing.

  Alternatively the recovery subsystem insurance that the program is resumed from the point at which it was interrupted so that its full effect is recorded in the database.

- **Improved data quality:**

  The database system provides a number of tools and processes to improve data quality.

Disadvantages of Database Management System

In spite of the advantages the database approach entails some additional cost and risk that must be recognized and manage when implementing DBMS. Following are the disadvantages of using DBMS:

- **Increased complexity**

  Multi user DBMS become an extremely complex piece of software due to expected functionality from it.It became necessary for the database designers developers database administrators and the users to understand this functionality to full advantage of it.

  Failure to understand the system can lead to bad design decisions which can have serious consequences for an organisation.

- **Requirement of new and specialized manpower**

  Because of rapid changes in database technology and organisations business needs, the organisations need to hire, train and retain its manpower on regular basis to design and implement databases provide database administration services and manage a staff of new people. Therefore an organisation needs to maintain specialized skilled manpower.

- **Large size of DBMS**

  The large complexity and wide functionality makes the DBMS and extremely large piece of software. It occupies many gigabytes of storage disk space and requires substantial amount of main memory to run efficiently.

- **Increased installation and management cost**

  The large and complex DBMS software has high initial cost. It requires trained manpower to install and operate and also has substantial annual maintenance and support costs. Installing such a system also requires upgrades to the hardware software and data communications systems in the organisation.

  Substantial training of manpower is required on an ongoing basis to keep up with new releases and upgrades. Additional or more sophisticated and costly database software may be needed to provide security and to ensure proper concurrent updating of shared data.

- **Additional hardware cost**

  The cost of DBMS installation varies significantly depending on the environment and functionality size of the hardware.

For example, microcomputer, minicomputer or mainframe computer and the recurring annual maintenance cost of hardware and software.

- **Conversion cost**

  The cost of conversion both in terms of money and time from ledger system that is old file oriented and older database technology to modern DBMS environment is very high.

  In some situations the cost of DBMS and extra hardware maybe insignificant compared with the cost of conversion. This cost includes the cost of training manpower to use this new systems and cost of employing specialist manpower to help with the conversion and running of the system.

- **Need for explicit backup and recovery**

  For centralised shared database to be accurate and available all times a comprehensive procedure is required to be developed and use for providing backup copies of data and for restoring a database when damage occurs. Modern database normally automates many more of the backup and recovery task than a file oriented system.

- **Organisational conflict**

  A centralised and shared database which is the case with DBMS requires consensus on data definitions and ownership as well as responsibilities for accurate data maintenance. As per past history an experience sometimes there are conflicts on data definitions data formats and coding right to update shared data and associated issues which are frequent and often difficult to resolve.

  Organisational commitment to the database approach organizationally astute database administrators and a sound evolutionary approach to database development is required to handle these issues.

Users of Database Management System

This classification is based on the interaction of users to the database. Database system is made to store information and provide an environment for retrieving information.

- **End Users-Naive and Casual User**

  End users are those who access the database from the terminal end. They use the developed applications and they don't have any knowledge about the design and working of database. These are the second class of users and their main motto is just to get their task done.

  There are basically two types of end users that are as follows:

  1. **Naive User**

     Any user who does not have any knowledge about database can be in this category.

     Their task is to just use the developed application and get the desired results.

     For example: Clerical staff in any bank is a naïve user.

     They don't have any DBMS knowledge but they still use the database and perform their given task.

     Railway's ticket booking users are naive users.

     Also, persons using Automated Teller Machines (ATM's) fall under this category as he has access to limited portion of the database.

  2. **Casual User**

     These users have great knowledge of query language. Casual users access data by entering different queries from the terminal end. They do not write programs but they can interact with the system by writing queries. These Users occasionally access the database but may need different information each time.

     For example: High level Managers who access the data weekly or biweekly.

- **Sophisticated Users**

  Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database. They can develop their own data base applications according to their requirement. They don't write the program code but they interact the data base by writing SQL queries directly through the query processor.

- **System Analyst**

  System analyst is responsible for the design, structure and properties of database.

All the requirements of the end users are handled by system analyst.

Feasibility, economic and technical aspects of DBMS is the main concern of system analyst.

- **Application Programmers**

  Application programmers are the one who writes application programs that uses the database. These programs could be written in General Purpose Programming languages such as Visual Basic, Developer, C, FORTRAN, COBOL, Java and fourth generation language etc.

  These programs meet the user requirement and made according to user requirements.

  These application programs operate on the data to perform various operations such as retaining information, creating new informa-tion, deleting or changing existing information.

- **Database Administrator (DBA)**

  The DBA is a person or a group of persons who is responsible for the management of the database. Database Administrator (DBA) defines the schema and also controls the 3 levels of database. He makes the policies, strategies and provides technical supports.

  The DBA is responsible for:

  - For authorizing access to the database by grant and revoke permis-sions to the users.
  - For coordinating and monitoring its use.
  - For managing backups and repairing damage due to hardware and/or software failures.
  - For acquiring hardware and software resources as needed.

Application areas of DBMS:

Due the evolution of Database management system, there are various fields where a database management system is used. Following are some application areas which make use of the database management system:

- Enterprise Information
- Banking and Finance
- Educational Sector
- Travel Industry
- Telecommunication Sector

Enterprise Information:

1. **Sales Area:**

   In Sales area, DBMS is used to record/store following data:

   - customer information
   - Product information
   - Sales information
   - Purchase details
   - Invoice details.

2. **Accounting:**

   In Accounting field, DBMS is used to record/store following data:

   - customer information
   - Product information
   - Sales information
   - Purchase details
   - Invoice details.

3. **Human Resources:**

   In Human Resources field, DBMS is used to record/store following data:

   - Information about employees,
   - Salaries of employees,
   - payroll taxes, and benefits,
   - For generation of paychecks etc.

4. **Manufacturing:**

   In Manufacturing field, DBMS is used to record/store following data :

   - For management of the supply chain,
   - for tracking production of items in factories,
   - inventories of items in warehouses and stores,
   - orders for items

5. **Online retailers:**

   In Online retailers field, DBMS is used to record/store following data:

   - For sales data noted above plus online order tracking,
   - generation of recommendation lists
   - maintenance of online product evaluations

Banking and Finance

- **Banking**:

  In Banking field, DBMS is used to record/store following data:

- For customer information,
- Accounts details/Information
- Loans information
- banking transactions details- tracking day to day credit and debit transactions
- **Credit card transactions**:

  In **Credit card transactions** field, DBMS is used to record/store following data:

- For purchases on credit cards,

- generation of monthly statements
- Bills and invoices
- Transactions
- **Finance**:

  In Finance field, DBMS is used to record/store following data:

- For storing information about holdings, sales, and purchases of
- Financial instruments such as stocks and bonds
- For storing real-time market data to enable online trading by customers and automated trading by the firm.

Educational Sector

- **Universities, Colleges, Schools**:

  In **Universities, Colleges, Schools** field, DBMS is used to record/store following data:

- For student information/Details
- Course registrations/Details
- Staff Details
- Exam Details, Grades
- payroll data,
- attendance details,
- fees details

Travel Industry

Airlines, Railway, Travels Reservations:

In Travel Industry, DBMS is used to record/store following data:

- For reservations and schedule information
- Passengers Details, Journey Details
- Airlines were among the first to use databases in a geographically distributed manner.

Telecommunication Sector:

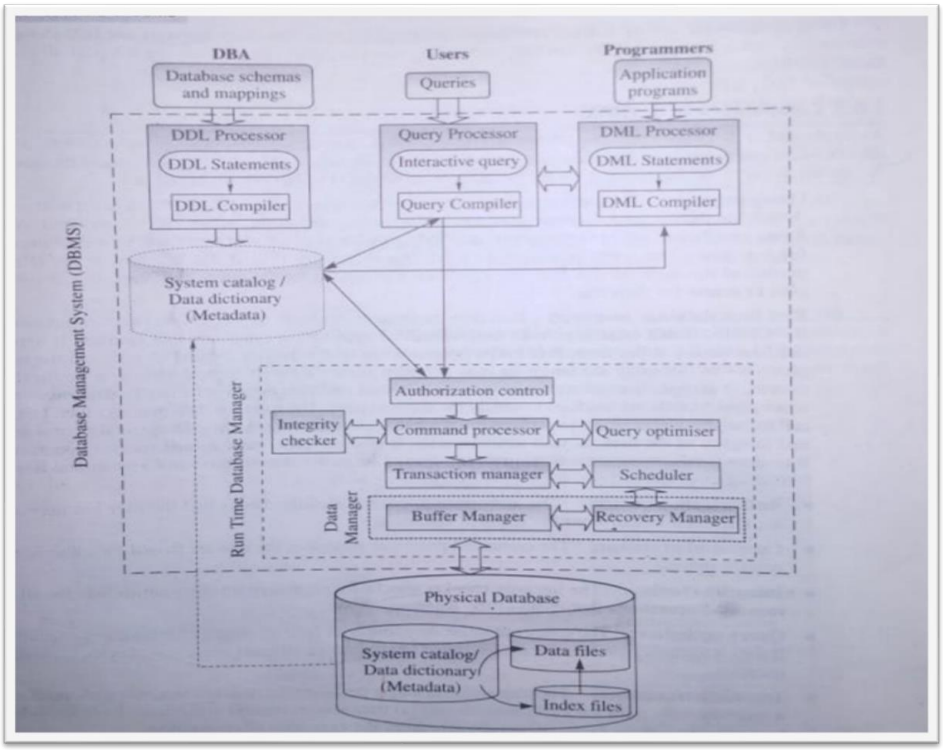
In Telecommunication Sector, DBMS is used to record/store following data:

- For keeping records of calls made
- network usage, customer details
- Generating monthly bills
- Maintaining balances on prepaid calling cards.
- Storing information about the communication networks.

Structure of DBMS

Database management system DBMS is highly complex and sophisticated software that handles access to the database. The structure of DBMS varies greatly from system to system and their forms a generalized component structure of DBMS is not possible to make.

A typical structure of DBMS with components and relationship between them is shown in the following structural diagram:



Components of a DBMS:

DBMS accepts the SQL commands generated from a variety of user interfaces produces query evaluation plans execute these plans against the database and return the answers.

1. **Query processor:**

   The query processor transforms user queries into a series of low level instructions directed to the runtime database manager.

   It is used to interpret the online users query and convert it into an efficient series of operations in a form capable of being saying to the runtime data manager for execution.

The query processor uses the data dictionary to find the structure of the relevant portion of the database and uses this information in modifying the query and repairing an optimal plan to access the database.

2. Run time database manager:

Runtime database manager is the central software component of the DBMS which interfaces with user submitted application programs and queries. It handles database access at runtime. It converts operations in user queries coming directly via the query processor or indirectly via an application program from the users logical view to the physical file system.

It accepts queries and examines the external and conceptual schema to determine what conceptual records are required to satisfy the user's request.

The run time data manager in places a call to the physical database to perform the request. It enforce constraints to maintain the consistency and integrity of the data as well as its security. It also performs backing and recovery operations.

Runtime database manager is sometimes referred to as database control system and has the following components:

- Authorization control
- Command processor
- Integrity checker
- Query optimizer
- Transaction manager
- Scheduler
- Data manager
- Recovery manager
- Buffer manager

- **Authorisation control:**

The authorisation control module text that the user has necessary authorisation to carry out the required operation.

- **Command processor**:

  The command processor processes the queries passed by authorisation control module.

- **Integrity checker**:

  The integrity checker checks for necessary integrity constraints for all the requested operations that changes the database.

- **Query optimizer:**

  The query optimizer determines and optimal strategy for the query execution. It uses information on how the data is stored to produce and efficient execution plan for evaluating query.

- **Transaction manager:**

  The transaction manager performs the required processing of operations it receives from transactions.

  It ensures that transactions request and release locks according to a suitable locking protocol Schedules the execution of transactions.

- **Scheduler:**

  The scheduler is responsible for ensuring that can current operations on the database proceed without conflicting with one another.

  It controls their relative order in which transaction operations are executed.

- **Data manager:**

  The data manager is responsible for the actual handling of data in the database.

  This module has the following two components:

- **Recovery manager:**

  The recovery manager ensures that the database remains in a consistent state in the presence of failures.

  It is responsible for Transaction, commit and abort operations, maintaining a log, restoring the system to a consistent state after a crash.

- **Buffer manager:**

  The buffer manager is responsible for the transfer of data between the main memory and secondary storage such as disk or tape, add links in pages from the disk to the main memory as needed in response to read user requests.

  Buffer manager is sometimes referred as cache manager.

3. DML processor:

Using a DML compiler the DML processor converts the DML statements embedded in an application program into standard function calls in the host language.

The DML compiler converts the DML statements written in a host programming language into object code for the database access.

The DML processor must interact with the query processor to generate the appropriate code.

4. DDL processor:

Using the DDL compiler the DDL processor converts the DDL statements into a set of tables containing metadata.

The tables contain the metadata concerning the database and are in a form that can be used by other components of the DBMS.

These tables are then stored in the system catalogue control information is store in data files headers.

The DDL compiler processes schema definitions specified in the DDL and stores description of schema metadata in the DBMS system catalog.

These the system catalogue includes information such as the names of the data files the Titans storage details of each data file mapping information amounts schemas and constraints.

View of Data

Abstraction is one of the main features of database systems. Hiding irrelevant details from user and providing abstract view of data to users helps in in easy and efficient user database interaction.

As per the three level of DBMS architecture the top level of that architecture is view level where data abstraction is highest. View level provides the views of data to the users and hides the relevant details such as data relationship, database schema, constraints, security etc. from the user.

To fully understand the view of the data you must have a basic knowledge of :

- Data abstraction
- Schema
- Instance

Data abstraction in DBMS:

Database systems are made up of complex data structures. To simplify user interaction with database, developer hides internal irrelevant details from users.

The process of hiding irrelevant details from user is called as data abstraction.

In DBMS, three are three levels of abstraction:

- Physical level/Internal level
- Logical level/ Conceptual Level
- View level or External Level

Data abstraction in physical level/internal level:

This is the lowest level of data abstraction. It describes how data is actually stored in database.

This level describes the physical storage of data. You can get complete data Structure details at this level.

Enrolment Number	9 digit number representation at offest 0
Student Name	25 character string at offset 9
Address	30 character string at offset 34
City	20 character string at offset 64
Pincode	.....
Telephone Number	.....
Highest Qualification	.....
Professional Skill	.....

Internal View

Data abstraction in Logical level/ Conceptual Level:

This is the middle level of three level data abstraction architecture. It describes what data is stored in the database. This level describes the organizational data which is actually to be stored in the database. This level contains information about entire database in terms of a small number of relatively simple structures.

Enrolment Number	Key : Numeric
Student Name	String
Address	String
City	String
Pincode	Numeric
Telephone Number	Numeric
Highest Qualification	String
Professional Skill	String

Conceptual View

Data abstraction in View level or External Level:

Highest level of data abstraction. This level describes the user interaction with database system. This is the abstraction for user. This level of abstraction generally describes only some small part of the database.

Student Name		Enrolment Number
Telephone Number		Student Name
Highest Qualification		Address
Professional Skill		Telephone Number
		Highest Qualification

DBMS Schema:

Definition of schema:

Design of a database is called the Schema.

The plan for formulation of skin off the database is known as schema. Schema gives the names of the entities and attributes. It specifies the relationship among them.

It is a framework into which the values of the data atoms or fields are fitted. The plans for the format of schema remains the same, but the values filled into this format changes from instance to instance. In other terms schema mean and overall plan of the data item or field types and record types stored in a database. Schema includes the definition of the database name the record type and the components that make up those records.

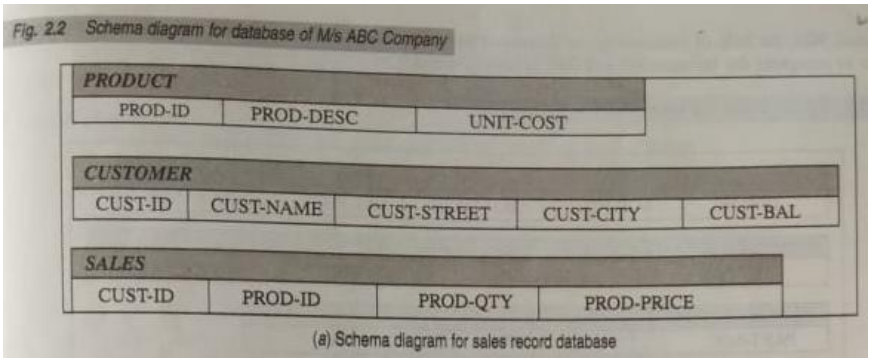
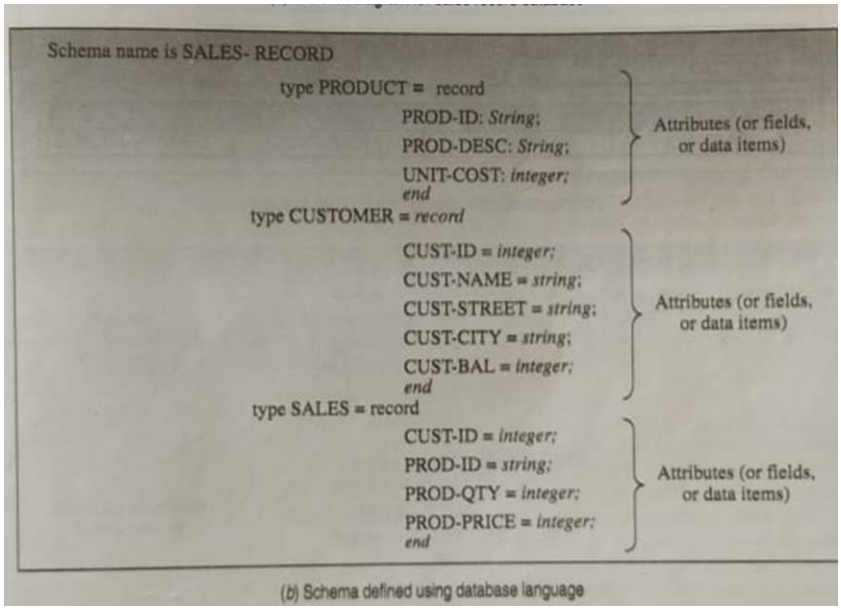


Fig. 2.2 Schema diagram for database of M/s ABC Company

(a) Schema diagram for sales record database



(b) Schema defined using database language

DBMS Schema is of three types:

- Physical Schema /Internal Schema
- Logical Schema / Conceptual Schema
- View Schema / External Schema

Physical Schema /Internal Schema

The design of database at physical level is called physical schema, how the data stored in blocks of storage is described at this level.

Logical Schema / Conceptual Schema

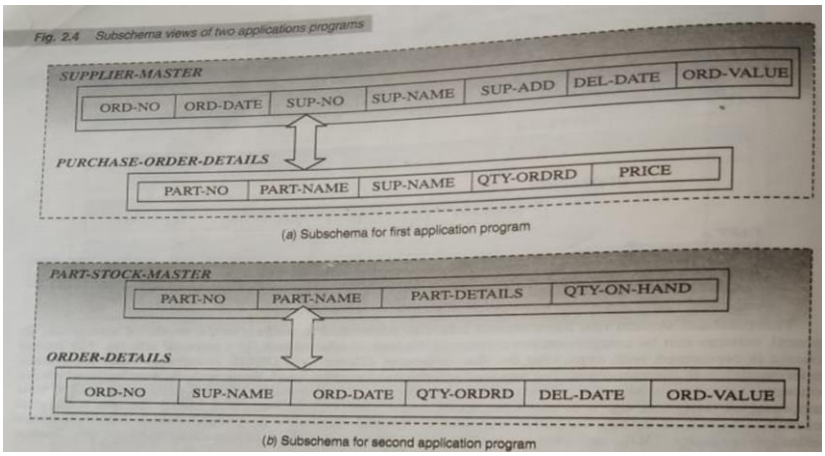
Design of database at logical level is called logical schema. Programmers and database administrators work at this level. Address label, data can be described as certain types of data records gets stored in data structures, however the internal details such as implementation of data structure is hidden at this level.

View Schema / External Schema

Design of database at view level is called view schema. This generally describes end user interaction with database systems. Conceptual level details and Internal level details are hidden at this level.

DBMS Sub schema:

A sub schema is a subset of schema and inherits the same property that a schema has. The plan or scheme for a view is often called sub schema. Subschema refers to an application programmer users view of the data atoms types and record types which he or she uses. Give the users a window through which he or she can view only that part of the database which is interest which is of interest to him. Different application programs can have different views of data.



DBMS Instance:

Definition of instance:

> *The data stored in the database at a particular moment of time is called instance of database.*

When the schema framework is field in the data atoms values or the contents of the database, at any point of time or current contents, is referred to as an instance of database. The term instance is also called as state of the database or snapshot. Each variable has a particular value at the given instant. The values of the variables in a program at a point in time corresponds to an instance of database schema. The difference between database schema and database state or instance is very distinct.

In case of a database schema, it is specified to DBMS when new database is defined, whereas at this point of time the corresponding database state is empty with no data in database. Once the database is first populated with the initial data, from then on, we get another database state whenever an update operation is applied to the database. At any point of time the current state of database is called the instance.

Difference between Schema and Instance:

SCHEMA	INSTANCE
It is the overall description of the database.	It is the collection of information stored in a database at a particular moment.
Schema is same for whole database.	Data in instances can be changed using addition, deletion, and updation.
Does not change Frequently.	Changes Frequently.
Defines the basic structure of the database i.e. how the data will be stored in the database.	It is the set of Information stored at a particular time.

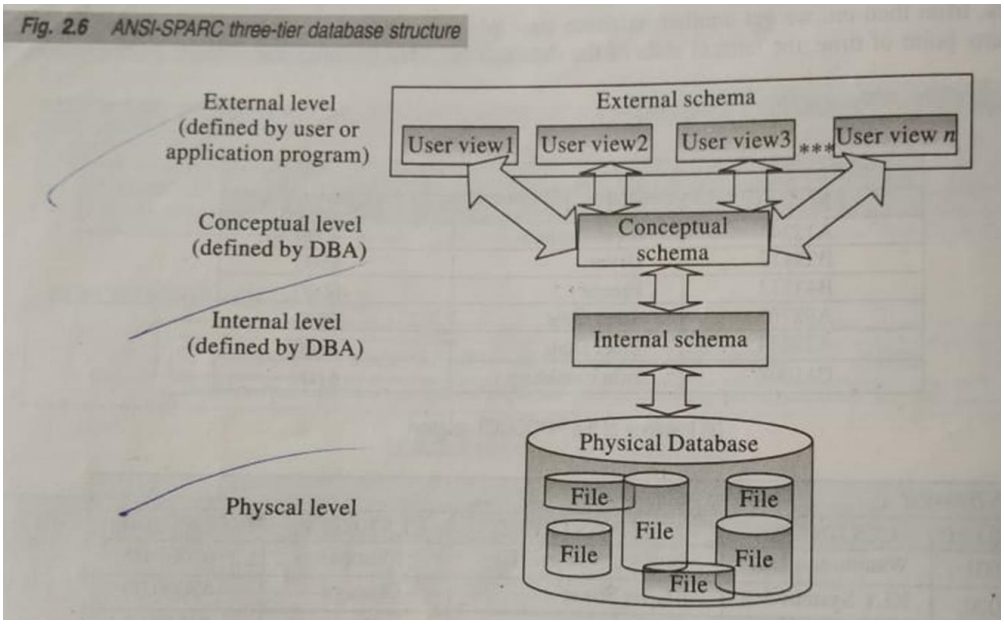
Three level database architecture

For the first time in 1971, database task group DBTG appointed by **Conference On Data Systems And Languages CODASYL**, produced a proposal for general architecture for database. The **DBTG** proposed two tier architecture with the system view called as schema and user views sub schemas. In 1975, ANSI SPARC **American National Standards Institute Standards Planning And Requirement** Committee producer 3 tier architecture system catalogue. The architecture of most commercial DBMS available today is based to some extent on this three level architecture of DBMS.

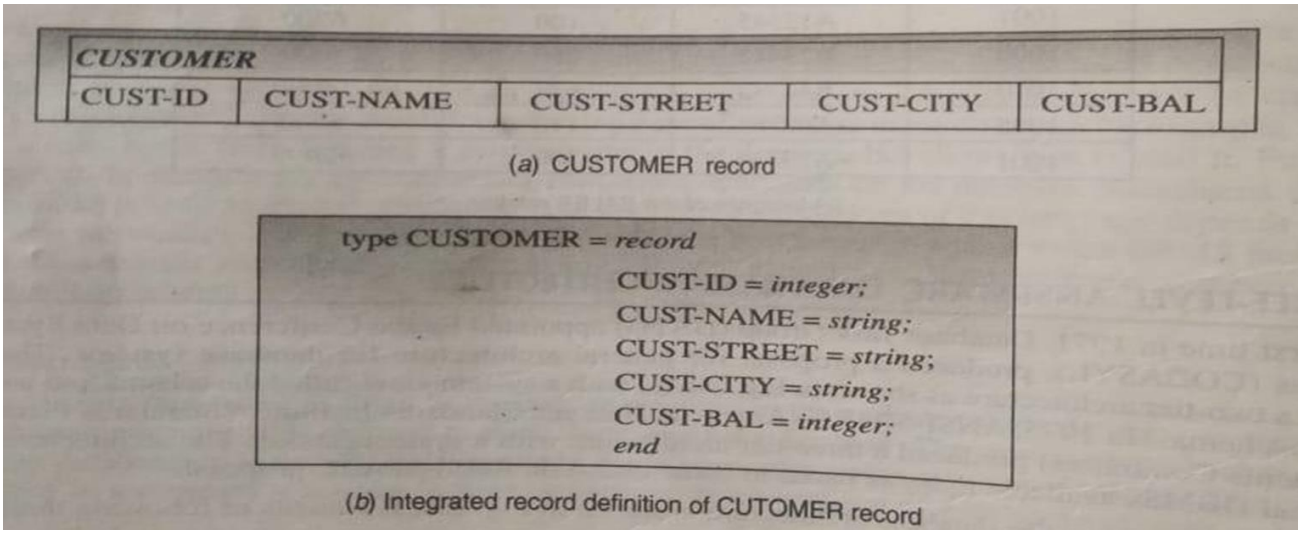
This ANSI SPARC 3 tier database architecture consists of following three levels.

- Internal level
- Conceptual level
- External level

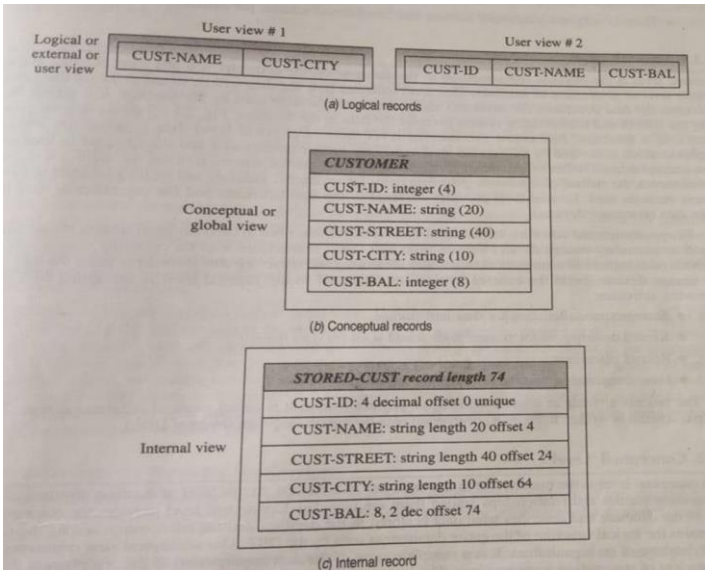
The views at each of the above labels is described by scheme or schema.



Let us take an example of customer record as shown in the following diagram. The integrated record definition of customer record is shown in the following diagram:



The data has been extracted in three levels corresponding to three views namely Internal, Conceptual and External views as shown in the following diagram:



- The lowest level of abstraction of data contains a description of the actual method of storing the data and is called the internal view.
- The second level of abstraction is the conceptual or global view.
- The third level is the highest level of abstraction seen by the user or application program and is called the external view or user view.

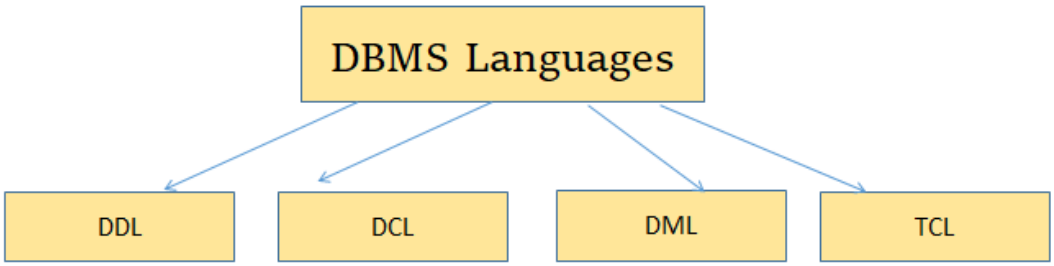
The conceptual view is the sum total of user or external view of data.

Database Languages

A DBMS has appropriate languages and interfaces to express database queries and updates. There are several such languages that can be used for this purpose; one of them is SQL (Structured Query Language).

Structured Query Language (SQL) as we all know is the database language by the use of which we can perform certain operations on the existing database and also we can use this language to create a database. SQL uses certain commands like Create, Drop, Insert etc. to carry out the required tasks.

Types of Database Language:



- Data Definition Language(DDL)
- Data Control Language(DCL)
- Data Manipulation Language(DML)
- Transaction Control Language(TCL)

Data Definition Language(DDL):

DDL stands for Data Definition Language. It is used to define database structure or pattern. It is used to create schema, tables, indexes, constraints, etc. in the database. Using the DDL statements, you can create the skeleton of the database.

Data definition language is used to store the information of metadata like the number of tables and schemas, their names, indexes, columns in each table, constraints, etc.

Examples of DDL commands:

Create:

It is used to create objects in the database.

Alter:

It is used to alter the structure of the database.

Drop:

It is used to drop objects from database such as tables.

Truncate:

It is used to remove all records from a table, including all spaces allocated for the records are removed.

Rename:

It is used to rename an object existing in the database.

Comment:

It is used to comment on the data dictionary.

These commands are used to update the database schema that's why they come under Data definition language.

Data Control Language (DCL):

DCL stands for Data Control Language. It is used to retrieve the stored or saved data. The DCL execution is transactional. It also has rollback parameters. DCL includes commands such as GRANT and REVOKE which mainly deals with the rights, permissions and other controls of the database system.

Examples of DCL commands:

GRANT-gives user's access privileges to database.

REVOKE-withdraw user's access privileges given by using the GRANT command.

Data Manipulation Language (DML):

DML stands for Data Manipulation Language. It is used for accessing and manipulating data in a database. It handles user requests.

Examples of DML:

Select: It is used to retrieve data from a database.

Insert: It is used to insert data into a table.

Update: It is used to update existing data within a table.

Merge: It performs UPSERT operation, i.e., insert or update operations.

Call: It is used to call a structured query language or a Java subprogram.

Explain Plan: It has the parameter of explaining data.

Lock Table: It controls concurrency.

Transaction Control Language (TCL):

TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction. TCL commands deals with the transaction within the database.

Examples of TCL commands:

Commit: It is used to save the transaction on the database.

Rollback: It is used to restore the database to original since the last Commit.

SAVEPOINT: It sets a savepoint within a transaction.

SET TRANSACTION: It specify characteristics for the transaction.

Index

Introduction:

Indexing is a way to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed. It is a data structure technique which is used to quickly locate and access the data in a database. A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and storage space to maintain the index data structure.

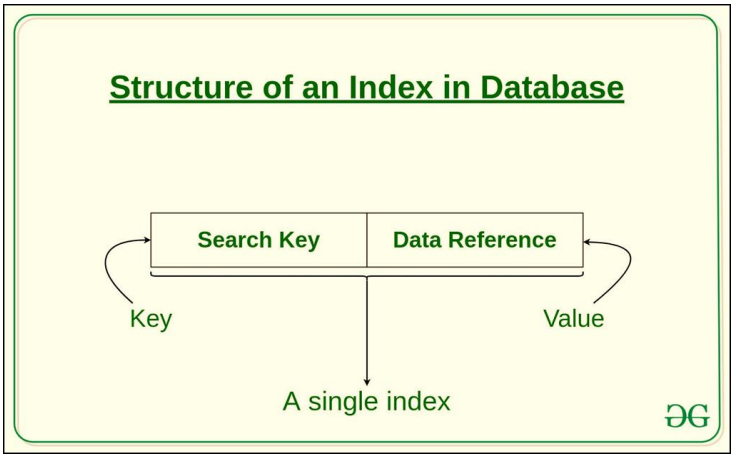
Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed.

Indexes are created using a few database columns.

The first column is the Search key that contains a copy of the primary key or candidate key of the table. These values are stored in sorted order so that the corresponding data can be accessed quickly.

Note: The data may or may not be stored in sorted order.

The second column is the Data Reference or Pointer which contains a set of pointers holding the address of the disk block where that particular key value can be found.



The indexing has various **attributes**:

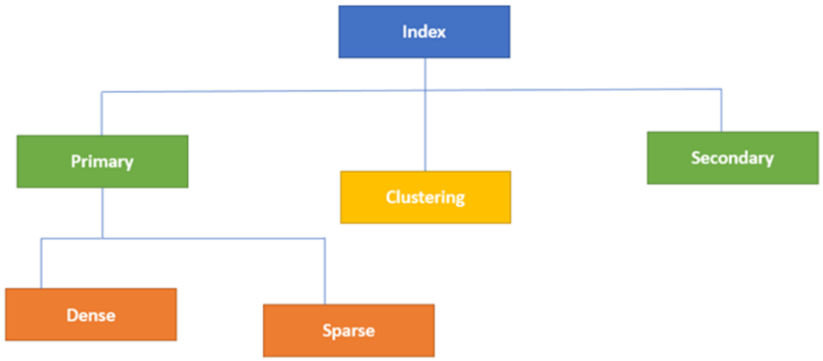
Access Types: This refers to the type of access such as value based search, range access, etc.

Access Time: It refers to the time needed to find particular data element or set of elements.

Insertion Time: It refers to the time taken to find the appropriate space and insert a new data.

Deletion Time: Time taken to find an item and delete it as well as update the index structure.

Space Overhead: It refers to the additional space required by the index.



Types of Index:

Database Indexing is defined based on its indexing attributes. Two main types of indexing methods are:

- Primary Indexing
- Secondary Indexing

Primary Indexing:

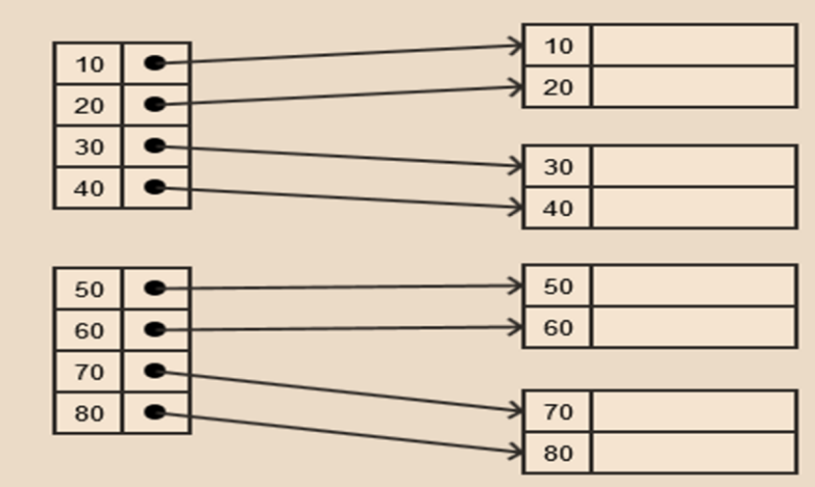
Primary Index is an ordered file which is fixed length size with two fields. The first field is the same a primary key and second, filed is pointed to that specific data block. In the primary Index, there is always one to one relationship between the entries in the index table.

The primary Indexing is also further divided into two types.

- Dense Index
- Sparse Index

Dense Index:

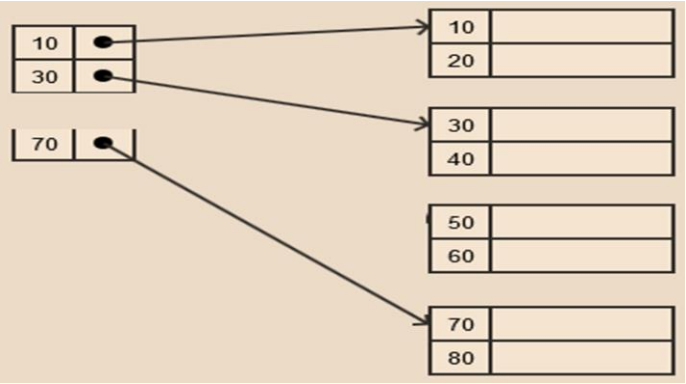
In a dense index, a record is created for every search key valued in the database. This helps you to search faster but needs more space to store index records. In this Indexing, method records contain search key value and points to the real record on the disk.



Sparse Index:

It is an index record that appears for only some of the values in the file. Sparse Index helps you to resolve the issues of dense Indexing. In this method of indexing technique, a range of index columns stores the same data block address, and when data needs to be retrieved, the block address will be fetched.

However, sparse Index stores index records for only some search-key values. It needs less space, less maintenance overhead for insertion, and deletions but It is slower compared to the dense Index for locating records.



Secondary Index:

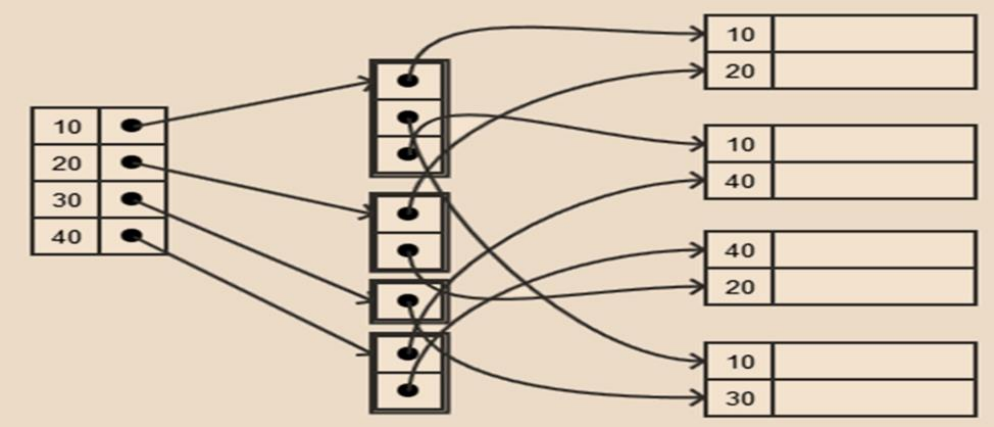
The secondary Index can be generated by a field which has a unique value for each record, and it should be a candidate key. It is also known as a non-clustering index. This two-level database indexing technique is used to reduce the mapping size of the first level.

For the first level, a large range of numbers is selected because of this; the mapping size always remains small.

Example of secondary Indexing:

In a bank account database, data is stored sequentially by acc_no; you may want to find all accounts in of a specific branch of ABC bank. Here, you can have a secondary index for every search-key.

Index record is a record point to a bucket that contains pointers to all the records with their specific search-key value.



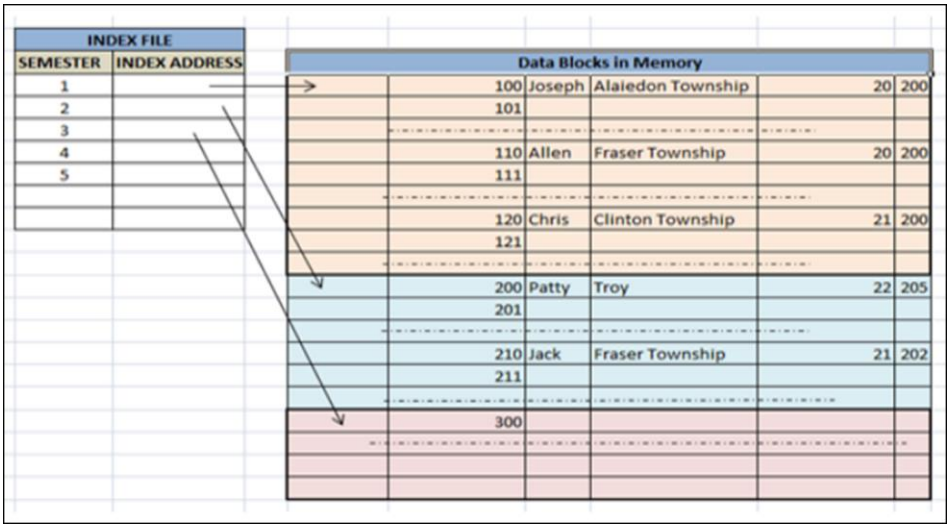
Clustering Index:

In a clustered index, records themselves are stored in the Index and not pointers. Sometimes the Index is created on non-primary key columns which might not be unique for each record. In such a situation, you can group two or more columns to get the unique values and create an index which is called clustered Index. This also helps you to identify the record faster.

When more than two records are stored in the same file these types of storing known as cluster indexing. By using the cluster indexing we can reduce the cost of searching reason being multiple records related to the same thing are stored at one place and it also gives the frequent joining of more than two tables(records).

Clustering index is defined on an ordered data file. The data file is ordered on a non-key field. In some cases, the index is created on non-primary key columns which may not be unique for each record. In such cases, in order to identify the records faster, we will group two or more columns together to get the unique values and create index out of them. This method is known as the clustering index.

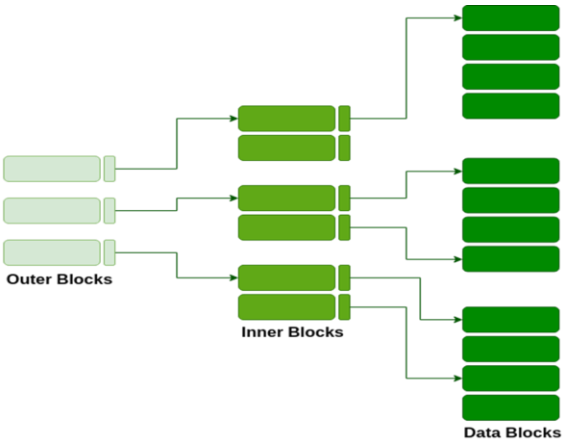
Basically, records with similar characteristics are grouped together and indexes are created for these groups. For example, students studying in each semester are grouped together. i.e. 1st Semester students, 2nd semester students, 3rd semester students etc. are grouped.

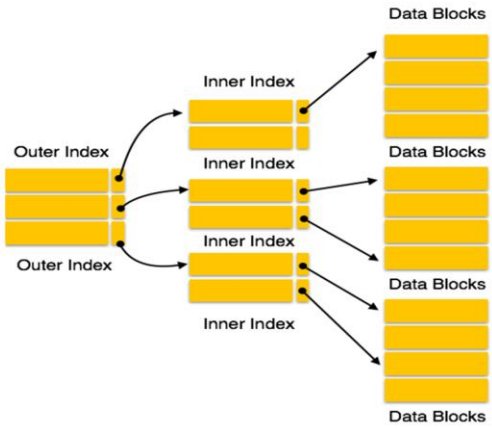


Clustered index sorted according to first name (Search key).

Multilevel Index:

With the growth of the size of the database, indices also grow. As the index is stored in the main memory, a single-level index might become too large a size to store with multiple disk accesses. Multi-level Index helps in breaking down the index into several smaller indices in order to make the outermost level so small that it can be saved in a single disk block, which can easily be accommodated anywhere in the main memory. The outer blocks are divided into inner blocks which in turn are pointed to the data blocks. This can be easily stored in the main memory with fewer overheads.





Advantages of Index:

- It helps you to reduce the total number of I/O operations needed to retrieve that data, so you don't need to access a row in the database from an index structure.
- Offers faster search and retrieval of data to users.
- Indexing also helps you to reduce table space as you don't need to link to a row in a table, as there is no need to store the ROWID in the Index. Thus you will able to reduce the table space.
- You can't sort data in the lead nodes as the value of the primary key classifies it.

Disadvantages of Index:

- To perform the indexing database management system, you need a primary key on the table with a unique value.
- You can't perform any other indexes on the Indexed data.
- You are not allowed to partition an index-organized table.
- SQL Indexing Decrease performance in INSERT, DELETE, and UPDATE query.

File Organization in DBMS

A database consists of a huge amount of data. The data is grouped within a table in RDBMS, and each table have related records. A user can see that the data is stored in form of tables, but in actual this huge amount of data is stored in physical memory in form of files.

File – A file is named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks.

What is File Organization?

File Organization refers to the logical relationships among various records that constitute the file, particularly with respect to the means of identification and access to any specific record.

In simple terms, Storing the files in certain order is called file Organization. In other words, file organization is a way of arranging records in a file when the file is stored on secondary storage.

The method of file organization chosen, determines how the data stored on the secondary storage device can be accessed. The file organization also affects the types of applications that can use the data and the time and cost necessary to do so.

Following operations are generally performed on a file:

- Scanning or fetching records from the file through the buffer pool.
- Scanning records that satisfy an equality selection (i.e., fetching a specific record).
- Searching records between a particular range.
- Inserting records into a file.
- Deleting a record from a file.

There are different types of file organizations that are used by applications. However, the operations to be performed and the selection of storage device influence the choice of a particular file organization.

Different types of file organization that are used in a database environment are as follows:

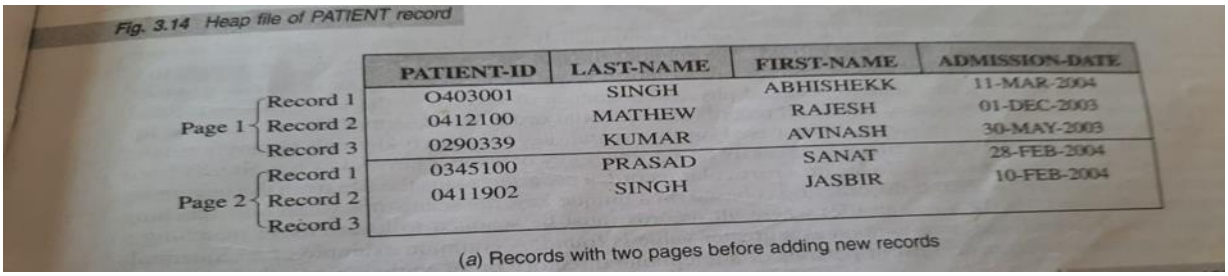
1. **Heap File Organization**
2. **Sequential File Organization**
3. **Indexed sequential File Organization**
4. **Direct or Hash file organization**

File Structure refers to the format of the label and data blocks and of any logical control record.

Heap File Organisation

In a heap file (also called a **pile** or **serial** file) organisation, records are collected in their arrival order. A heap file has no particular order. Therefore, it is equivalent to a file of unordered records or sequence Wherever there is an empty space available, any record can be placed in that block. Pointers are used to link the record blocks. If there is no space available to accommodate the inserted record, a new block is allocated and a new record is placed into it. Thus, a heap or serial file is generated by appending records at the end.

When the file is organised in pages, records are always added to the last page until there is insufficient room to hold a complete record. At this point a new page is allocated and the new record inserted. Fig. 1 shows adding records to heap file of hospital patient records. Fig. 1 (a) shows a PATIENT heap file having records with two pages. Page 1 has three records and is full, whereas Page 2 has two records and can accommodate one more.

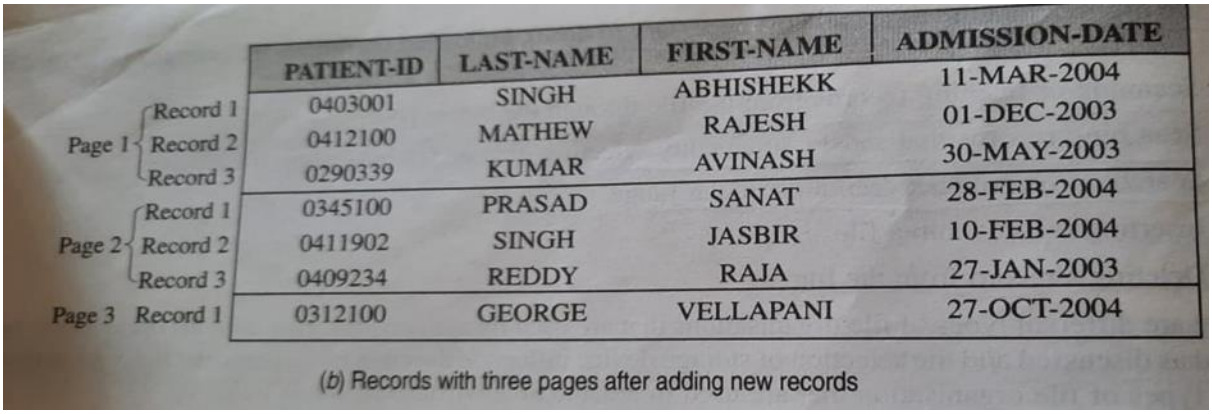

Fig. 3.14 Heap file of PATIENT record

		PATIENT-ID	LAST-NAME	FIRST-NAME	ADMISSION-DATE
Page 1	Record 1	O403001	SINGH	ABHISHEKK	11-MAR-2004
	Record 2	0412100	MATHEW	RAJESH	01-DEC-2003
	Record 3	0290339	KUMAR	AVINASH	30-MAY-2003
Page 2	Record 1	0345100	PRASAD	SANAT	28-FEB-2004
	Record 2	0411902	SINGH	JASBIR	10-FEB-2004
	Record 3				

(a) Records with two pages before adding new records

Fig. 1 (a)

Fig. 1 (b) shows the same file in which two new records have been added. Since Page 2 could accommodate only one record and is already filled, a new page (Page 3) was allocated to accommodate the next new record 3.



		PATIENT-ID	LAST-NAME	FIRST-NAME	ADMISSION-DATE
Page 1	Record 1	0403001	SINGH	ABHISHEKK	11-MAR-2004
	Record 2	0412100	MATHEW	RAJESH	01-DEC-2003
	Record 3	0290339	KUMAR	AVINASH	30-MAY-2003
Page 2	Record 1	0345100	PRASAD	SANAT	28-FEB-2004
	Record 2	0411902	SINGH	JASBIR	10-FEB-2004
	Record 3	0409234	REDDY	RAJA	27-JAN-2003
Page 3	Record 1	0312100	GEORGE	VELLAPANI	27-OCT-2004

(*b*) Records with three pages after adding new records

Fig. 1 (b)

If records are randomly appended, the logical ordering of the file with respect to a given key bears no Correspondence to the physical sequence. Updating an individual record or group of records can be done if it is assumed that the records are of fixed length and modifications do not change their size Retrieval of a particular record calls for searching of the file from the beginning to end.

In case of deletion of a record, records following the deleted record can be moved forward or the last record in the file can be brought into place vacated by the deleted record. However, these methods require many additional accesses up to the e end of the file. A more practical and better method is used in which a record is deleted logically and a (called deletion bit) flag is set whenever a record is deleted. The space of deleted record is reused by future insertion of records.

Advantages of a heap or serial file:

- Insertion of new record is less time consuming-
- It has a fill factor of 100 per cent as because each page is filled to capacity as new records are added.
- Space utilization is high, making the heap suitable for conserving space in large files.

Limitations of a heap or serial file:

- Slow retrieval of record.

- High updating cost of records.
- High cost of record retrieval
- Has restricted use.

Uses of a heap or serial file:

- Storing small files covering only a few pages
- Where data is difficult to organise.
- When data is collected prior to processing.
- Very efficient for bulk-loading large volumes of data

Sequential File Organisation:

Sequential file also called as an **ordered** file is a set of continuously stored records on a physical storage device such as magnetic disk tape or CD ROM.

A sequential file can be created by sorting the records in a heap file.

Sequential file organisation records are stored in sequential ascending or descending order onto the secondary storage media.

The logical and physical sequence of records is the same in a sequential file organisation.

Search key is used to sort all records in sequential order. A sequential file is organised for efficient processing of records in sorted order based on this search key. Search key can be any attribute field atom or set of attributes and need not necessarily be a primary key or super key.

To locate a particular record a program scans the file from the beginning until the desired record is located. If the ordering is based on unique key the scan stops when one matching record has been found. Unlike the heap file where all record must be scanned to locate those matching a search key the sequential scan stops as soon as a greater value is found.

Common example of sequential file organisation is the alphabetical list of persons of a telephone directory or and English/ Hindi directory and so on.

Following example is of sequential file that has been obtained after sorting of purchase file on primary key that is **SUP-NAME** in an ascending order.

Thus in a sequential file records are maintained in the logical sequence of their search (primary) key values.

	SUP-NAME	ORD-NO	ORD-VAL
Record 0	KLY System	P4-001	680000
Record 1	Price Waterhouse	P4-010	100000
Record 2	Concept Shapers	P4-008	450000
Record 3	Trinity Agency	P1-005	998000
Record 4	JUSCO Ltd	P1-007	124500
Record 5	Gujrat Rolls Ltd.	P1-120	50000
Record 6	Computer Associates	P2-210	88000

PURCHASE File

Fig. 3.15 Sequential file sorted in ascending order

	SUP-NAME	ORD-NO	ORD-VAL
Record 6	Computer Associates	P2-210	88000
Record 2	Concept Shapers	P4-008	450000
Record 5	Gujrat Rolls Ltd.	P1-120	50000
Record 4	JUSCO Ltd.	P1-007	124500
Record 0	KLY System	P4-001	680000
Record 1	Price Waterhouse	P4-010	100000
Record 3	Trinity Agency	P1-005	998000

PURCHASE File: Sequential File sorted in ascending order

In case of multiple key searches for sorting, first key is called a primary key while the others are called secondary keys.

Following tables a shows a simple EMPLOYEE file of an organisation, while fig. b shows the same file sorted on three keys in ascending order.

As shown in fig. b, the first key primary key is employees last name EMP-LNAME , second key is employees identification number EMP-ID, and third key is employees country (COUNTRY) to which they belong.

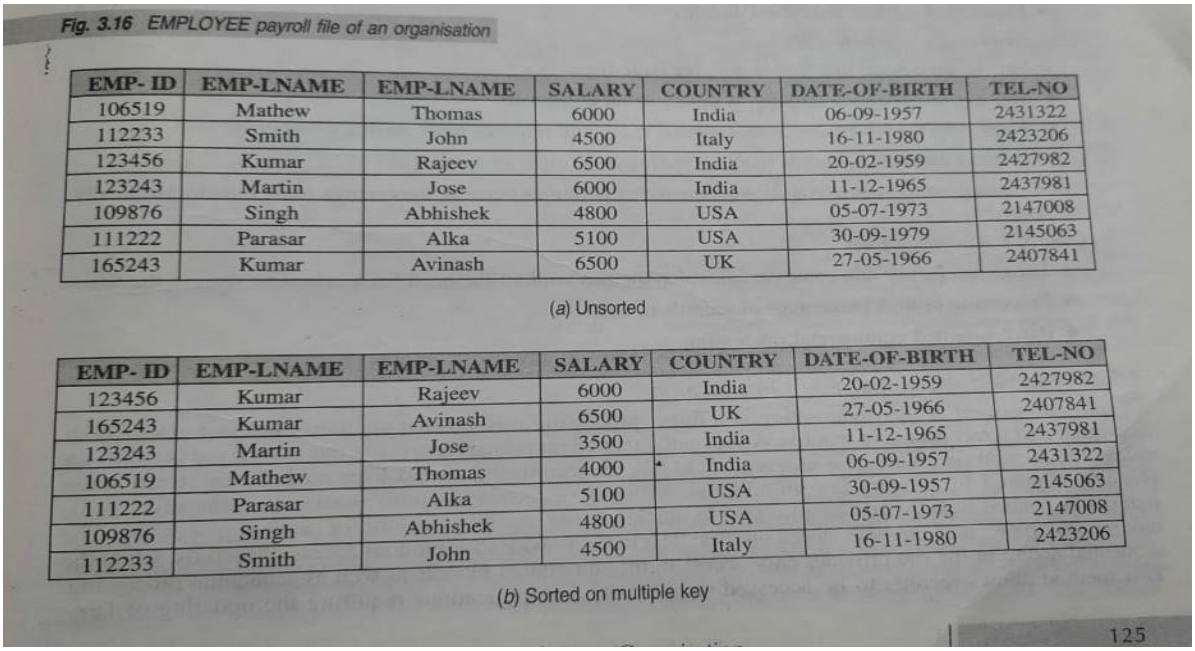


Fig. 3.16 EMPLOYEE payroll file of an organisation

EMP- ID	EMP-LNAME	EMP-LNAME	SALARY	COUNTRY	DATE-OF-BIRTH	TEL-NO
106519	Mathew	Thomas	6000	India	06-09-1957	2431322
112233	Smith	John	4500	Italy	16-11-1980	2423206
123456	Kumar	Rajeev	6500	India	20-02-1959	2427982
123243	Martin	Jose	6000	India	11-12-1965	2437981
109876	Singh	Abhishek	4800	USA	05-07-1973	2147008
111222	Parasar	Alka	5100	USA	30-09-1979	2145063
165243	Kumar	Avinash	6500	UK	27-05-1966	2407841

(*a*) Unsorted

EMP- ID	EMP-LNAME	EMP-LNAME	SALARY	COUNTRY	DATE-OF-BIRTH	TEL-NO
123456	Kumar	Rajeev	6000	India	20-02-1959	2427982
165243	Kumar	Avinash	6500	UK	27-05-1966	2407841
123243	Martin	Jose	3500	India	11-12-1965	2437981
106519	Mathew	Thomas	4000	India	06-09-1957	2431322
111222	Parasar	Alka	5100	USA	30-09-1957	2145063
109876	Singh	Abhishek	4800	USA	05-07-1973	2147008
112233	Smith	John	4500	Italy	16-11-1980	2423206

(*b*) Sorted on multiple key

125

Whenever an attribute (field item) or a set of attributes is added into the record, the entire file is reorganized to effect the addition of new attribute in each record of the file.

Therefore extra fields are always kept in the sequential file for future addition of items.

Sequential file organisation can exist on all types of secondary storage devices such as magnetic tapes magnetic disk and so on.

Due to the physical nature of magnetic tape storage the sequential file records stored on it are processed sequentially.

Accessing a particular record requires the accessing of all previous records in the file along the length of the tape.

When sequential file records are stored on magnetic disks, they are processing order sequentially or directly.

In case of sequential processing of records on disk while retrieving a particular record, all the records preceding it are to be processed first.

Thus the entire process of retrieval become slow if that target record is residing on the disc towards the end of the file.

The efficiency of a sequential file organisation depends upon the type of query.

If the query is for specific record identified by a record key for example EMP-ID, the file is search from the beginning until the record is found.

The retrieval of record from sequential file, on an average, requires access to the half the records in the file.

Thus, making enquiries in a sequential file is in efficient and very time-consuming for large files.

If the query is for the batch operation, the sorting is done in order of search key on the sequential file and then the processing of the sequential is done in a single pass.

Such a file containing the updates to be made to a sequential file, is also referred to as a transaction file.

The processing efficiency in the batch operation improves and cost of processing reduces.

Advantages of Sequential File Organisation

- Sequential retrieval of records on a primary key is very fast.
- On an average, queries can be performed in half the time taken for a similar queries on a heap file with a similar fill factor.
- Ease of access to the next record.
- Simplicity of organisation.
- Absence of auxiliary data structure.
- Creation of an automatic backup copy.

Limitations of Sequential File Organisation

- Simple queries are time-consuming for large files.
- Multiple key retrieval require scanning of an entire file.
- While deleting records it creates vested space and requires reorganizing of records.
- Insertion updates of records required creation (rewriting) of a new file.
- Inserting and deleting records are expensive operations because the records must remain physically ordered.

Use of Sequential File Organisation

- For range queries and some partial match curies without the need for a complete scan of the file.
- Processing of high percentage of records in a file.
- Batch oriented commercial processing.

Indexed-sequential File Organisation

An indexed-sequential file organisation is a direct processing method that combines the features of both sequential and direct access of records.

A sequential (sorted on primary keys) file that is indexed is called an indexed sequential file.

As in case of a sequential file, the records in indexed-sequential file are stored in the physical sequence by primary key. In addition, an index of record locations is stored on the disk.

Thus, Indexes associated with the file are provided to quickly locate any given record for random processing.

The indexes and the records are both stored on disk.

The index provides for random access of records, while the sequential nature of the file provides easy access to the subsequent records as well as sequential processing.

This method allows records to be accessed sequentially for applications requiring the updating of large numbers of records as well as providing the ability to access records directly in response to user queries.

An indexed-sequential file organisation consists of three contents namely:

(a**) Primary data storage.**

(b) Overflow area.

(C) Hierarchy of indices.

Primary data storage:

The primary data storage area (also called **prime area**) is an area in which records are written when an indexed-sequential file is originally created.

It contains the records written by the users programs.

The records are written in data blocks in ascending key sequence. These data blocks are in turn stored in ascending sequence in the primary data storage area. The data blocks are sequenced by the highest key of logical records contained in them.

The prime area is essentially a sequential file.

Overflow area:

The **overflow area** is essentially used to store new records, which cannot be otherwise inserted in the prime area without rewriting the sequential file.

It permits the addition of records to the file whenever a new record is inserted in the original logical block.
Multiple records belonging to the same logical area may be chained to maintain logical sequencing.
A pointer is associated with each record in the prime area which indicates that the next sequential record is stored in the overflow area.

Two types of overflow areas are generally used, which are known as:

(a) Cylinder overflow area.

(b) Independent overflow area.

Either or both of these overflow areas may be specified for a particular file.

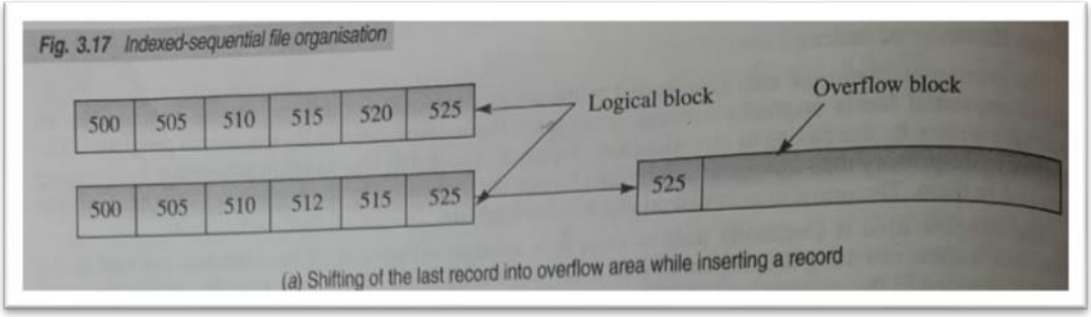
In **cylinder overflow area**, the spare tracks in every cylinder is reserved for accommodating the overflow records, whereas in an **independent overflow area**, overflow records from anywhere in the prime area may be placed.

In case of **random enquiry or update**, a hierarchy of indices are maintained that are accessed to get the physical location of the desired record.

- The data of the indexed-sequential files is stored on the cylinders, each or which is made up of a number of **track**s.
- Some of these tracks are reserved for primary data storage area and others are used for an overflow area associated with the primary data area on the cylinder.

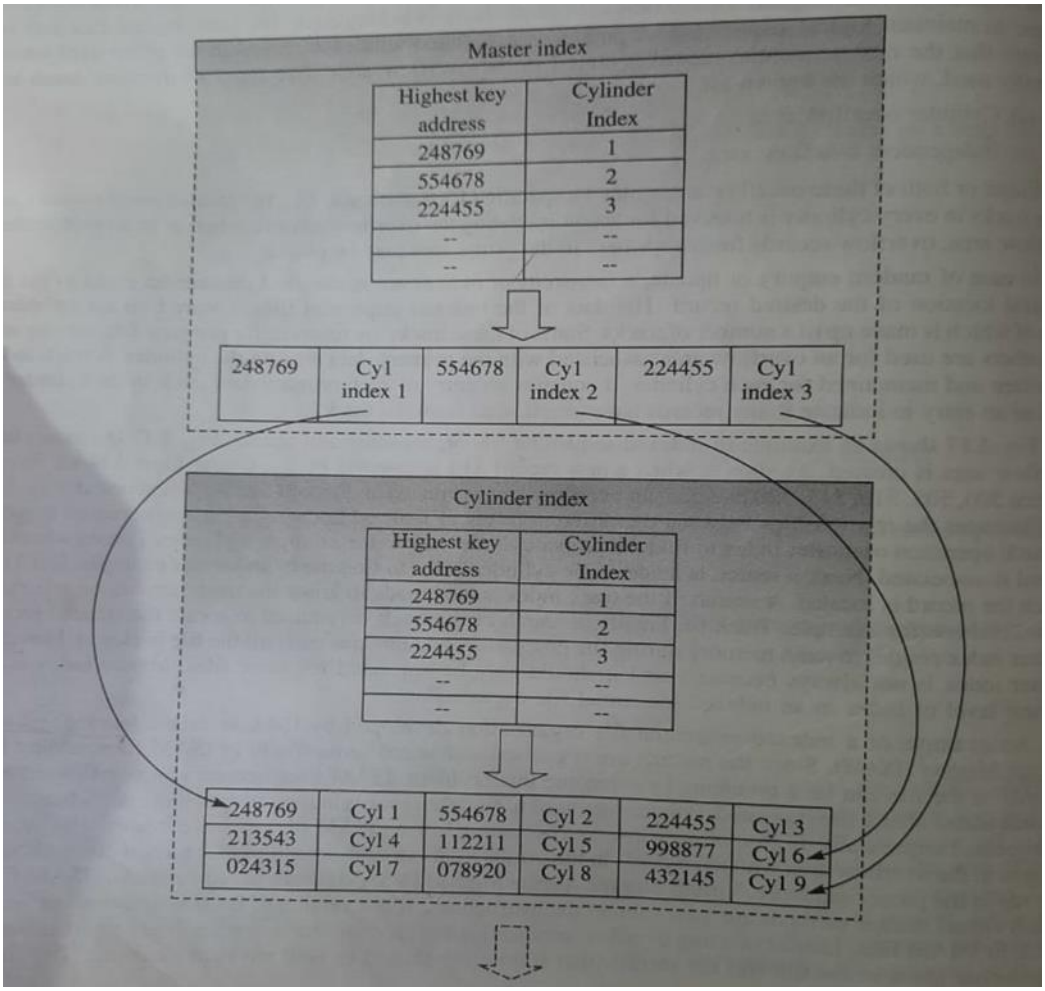
- A **track index** is written and maintained for each cylinder. It contains an entry of each primary data track in the cylinder as well as an entry to indicate if any records have overflowed from the track.

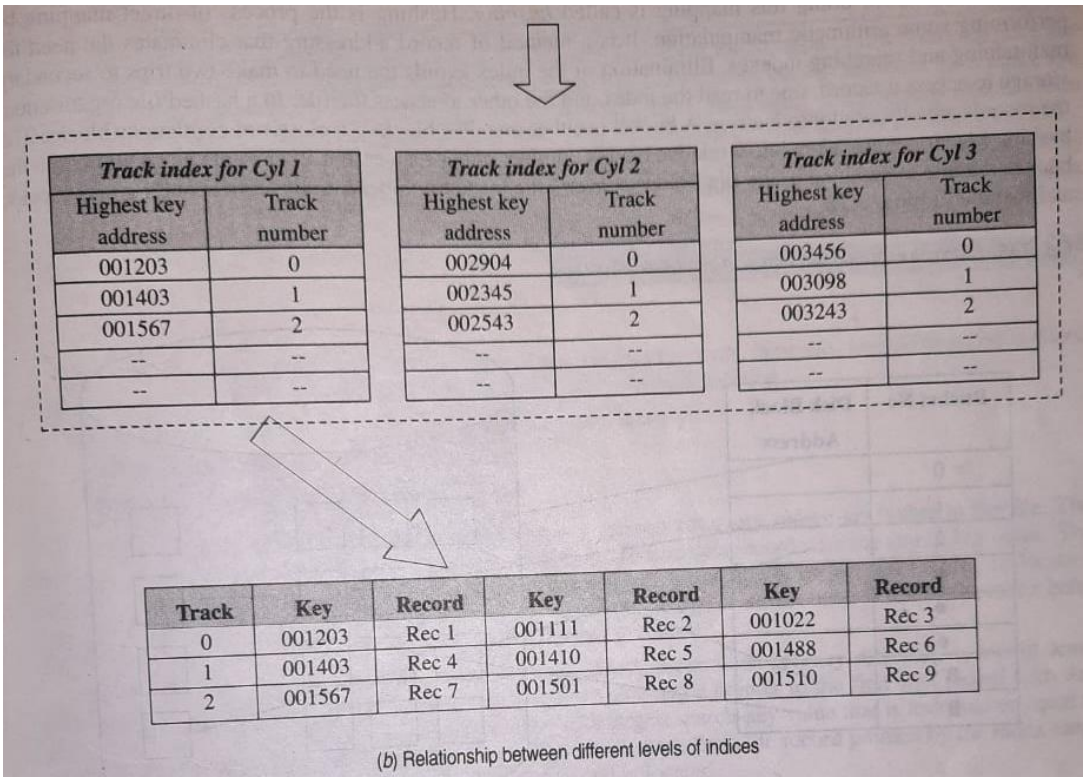
Fig. 3.17 shows an example of **indexed-sequential file organisation and access**. Fig. 3.17 (a) shows how **overflow area** is created.



- As shown, when a new record 512 is inserted in an existing logical block having records 50, 505, 510, 515, 520 and 525.
- An overflow area is created and record 525 is shifted into it.

Fig. 3.17(b) illustrates the relationships between the different levels of indices.





Track index for Cyl 1

Highest key address	Track number
001203	0
001403	1
001567	2
--	--
--	--

Track index for Cyl 2

Highest key address	Track number
002904	0
002345	1
002543	2
--	--
--	--

Track index for Cyl 3

Highest key address	Track number
003456	0
003098	1
003243	2
--	--
--	--

Track	Key	Record	Key	Record	Key	Record
0	001203	Rec 1	001111	Rec 2	001022	Rec 3
1	001403	Rec 4	001410	Rec 5	001488	Rec 6
2	001567	Rec 7	001501	Rec 8	001510	Rec 9

(*b*) Relationship between different levels of indices

Locating a particular record

- Locating a particular record involves a search operation of **master index** to find the **proper cylinder index** (for example Cyl index 1) with which the record is associated.
- Next, a search is made of the **cylinder index** to find the **cylinder** (for example, Cyl) on which the record is located.
- A search of the **track index** is then made to know the **track number** on which the record resides (for example, Track 0).
- Finally, a is required to locate the desired record. **search of the track**

Master index resides in **main memory** during file processing and remains their till the file is closed. However, master index is not always necessary and it should only be requested for large files.
Master index is the highest level of index in an indexed-sequential file organisation.

An example of an indexed-sequential file organisation developed by **IBM**, is called **Index Sequential Access Method (ISAM).**
Since the records are organised and stored sequentially in ISAM files, adding new records to the file can be a problem.
To overcome this problem, ISAM files maintain an overflow area for records added after a file is created.
Pointers are used to find the records in their proper sequence when the file is processed sequentially. In case of overflow area becoming full, an ISAM file can be reorganized by merging records in the overflow area with the records in the primary data storage area to produce a new file with all the records in the proper sequence.
Virtual Storage Access Method (VSAM) is advanced version of **ISAM** file in which virtual storage methods are used to enter the instructions. It is a version of B+ tree.
In **VSAM** files, instead of using overflow area for adding records, the new records are inserted into the appropriate place in the file and the records that follow are shifted to new physical locations. The shifted records are logically connected through pointers

located at the end of inserted records. Thus, **VSAM** file does not require reorganisation, as is the case with **ISAM** files. **VSAM** file method is much more efficient that **ISAM** files.

Advantages of an indexed-sequential file:

- Allows record to be processed efficiently in both sequential and random order, depending on the processing operations.
- Data can be accesses directly and quickly.
- Centrally maintain data can be kept up-to-date

Limitations of an indexed-sequential file:

- Only the key attributes determine the location of the record, and therefore retrieval operations involving non-key attributes may require searching of entire file.
- Lowers the computer system's efficiency.
- As file grows, performance deteriorates rapidly because of overflows and consequently reorganisation is required. This problem, however, is taken care by VSAM files.

Uses of an indexed-sequential file:

- Most popular in commercial data processing applications
- Applications where rate of insertion is very high.

Direct (or hash) File Organisation

In database management system, when we want to retrieve a particular data, it becomes very inefficient to search all the index values and reach the desired data. In this situation, **Hashing technique** comes into picture.

Hashing is an efficient technique to directly search the location of desired data on the disk without using index structure. It is a method of record addressing that eliminates the need for maintaining and searching indexes. Elimination of the index avoids the need to **make two trips to secondary storage to access a record**, **one to read the index** and the **other to access the file**.

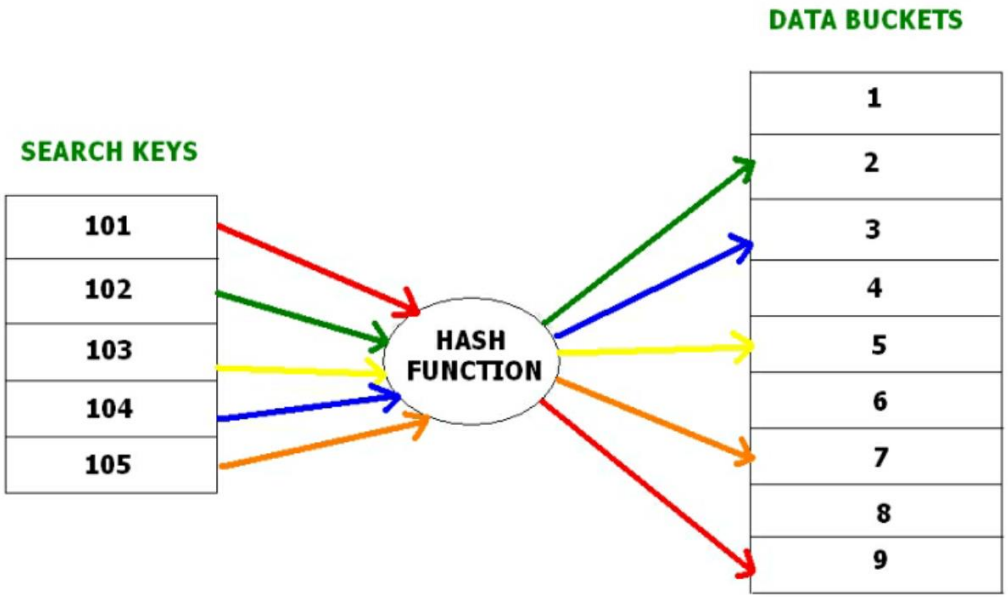
In a **direct (also called hash) file organisation**, mapping (also called transformation) of the search key value of a record is made directly to the address of the storage location at which that record is to reside in the file. One mechanism used for doing this mapping is called **hashing**. Hashing is the process of-direct mapping by performing some arithmetic manipulation.

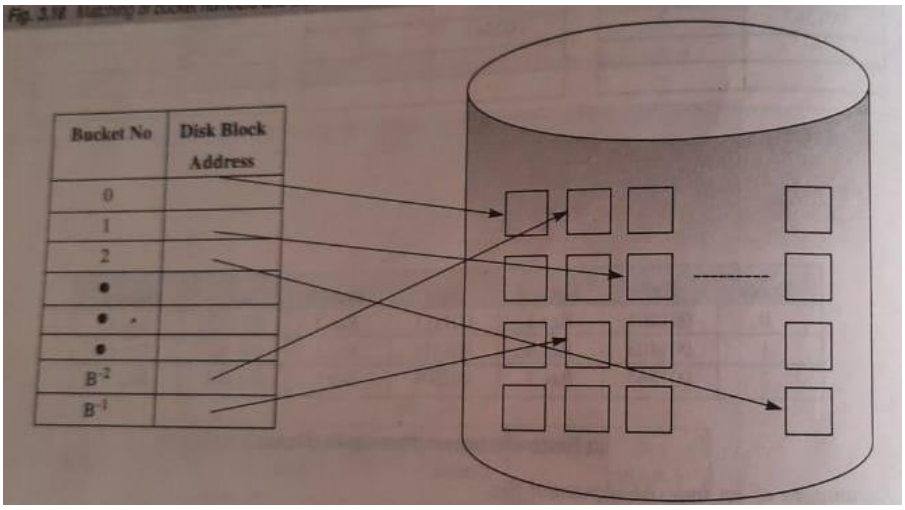
In a hashed file organisation, Data is stored at the **data blocks** whose address is generated by using **hash function**. The memory location where these records are stored is called as **data block** or **data bucket**.

A bucket is either one disk block or a cluster of contiguous blocks.

The hashing function maps a key into a relative bucket number, rather than assigning an absolute block address to the bucket.

A table maintained in the file header converts the bucket number into the corresponding disk blocks, as illustrated in Fig.





Data bucket –

- Data buckets are the **memory locations** where the records are stored.
- These buckets are also considered as Unit of Storage.

Hash Function –

- Hash function is a **mapping function** that maps all the set of search keys to actual record address.
- Generally, hash function uses **primary key** to generate the hash index – address of the data block.
- Hash function can be simple mathematical function to any complex mathematical function.

Hash Index-

- The prefix of an entire hash value is taken as a hash index.
- Every hash index has a depth value to signify how many bits are used for computing a hash function.
- These bits can address 2n buckets.
- When all these bits are consumed ? then the depth value is increased linearly and twice the buckets are allocated.

In a hash file, the data is scattered throughout the disk in a random order. The processing of a hash file is dependent on how the search key set for the records is transformed (or mapped) into the addresses of secondary storage device (for example, hard disk) to locate the desired record.

The search condition must be an equality condition on single files, called the **hash field** of the file.

In most cases, the hash field is also a **key field** of the file, in which case it is called **hash key**. In hashing operations, there is a function **h** called a hash function or randomizing function that is applied to the hash field value **v** of a record. This operation yields the address of the disk block in which the address is stored.

A search for the record within the block can be carried out in the main memory buffer. The function **h(v)** indicates the number of the bucket in which the record with key value v is to be found. It is desirable that **h** "hashes" **v**, that is, **h (v**) takes all its possible values with roughly equal to probability as **v** ranges over likely collections of values for the key.

Advantages of a direct file:

- Data can be accessed directly and quickly.
- Centrally maintained data can be kept up-to-date

Limitations of a direct file:

- Because all data must be stored on disks, the hardware is expensive.
- Because files are updated directly and no transaction files are maintained, there may not be any back in case a file is destroyed.

Data Models

Introduction:

A **model** is an abstraction process that concentrates or highlights essential and inherent aspects of the organisation applications while ignore or hides superfluous or accidental details.

It is a representation of the real world objects and events and their associations.

A **data model** also called **database model** is a mechanism that provides this abstraction for database application. It represents the organisation itself. It provides the basic concepts and notations to allow database designers and end users unambiguously and accurately communicate their understanding of the organizational data.

Data modeling is used for representing entities of interest and their relationships in the database. It allows the conceptualization of the association between various entities and attributes.

A data model is a conceptual method of structuring data. It provides mechanism to **structure data** (consisting of a set of rules according to which database can be constructed) for being modeled, **allow a set of manipulative operations** (for example, **updating or retrieving data from the database**) to be define on them, and **enforce set of constraints for integrity rules to ensure accuracy of data.**

We can say that a data model is a collection of mathematical e well defined concepts that help an enterprise to considered and express static and dynamic properties of data intensive applications. It consists of the following:

- Static properties, for example, objects, attributes and relationships
- Integrity rules over objects and operations
- Dynamic properties for example operations or rules defining New database States based on applied state changes

Data models can be broadly classified into following three categories:

1 Record based data models

2 Object based data models

3 Physical data models

Record based data models:

Record based data models are used to specify **the overall logical structures of the database**.

In the record based data models did the database consists o**f number of fixed format records** possibly of different types. Each record type define **a fixed number of fields each typically a fixed length.** Data integrity constraints cannot be explicitly specified using a record based data models.

There are three principal types of record based data models:

- Hierarchical data model
- Network data model
- Relational data model

Object based data models:

Object based data models are used to describe data and its relationships. **It uses concepts such as entities attributes and relationships.** It has flexible data structuring capabilities. Data integrity constraints can be quickly specified using object basic data model.

Following are the common types of object based data models:

- Entity relationship
- Semantic
- Functional
- Object oriented

The entity relationship data model is one of the main techniques for a database design and widely used in practice.

Physical data models

Physical data models are used for higher level description of storage structure and access mechanism.

Describe how data is stored in the computer presenting information such as record structure record orderings and access path.

It is possible to implement the database at system level using physical models therefore there are not as many e physical data models so far.

The most common physical data models are as follows:

- Unifying model
- Frame memory model

References:

1. Database Systems- Concepts, Design and Applications-By S.K. Singh, Pearson Education
2. Database Systems Concepts-Sixth Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, MC Graw Hill Publication

ENTITY-RELATIONSHIP (E-R) DATA MODEL

Introduction:

An entity relationship model is a logical database model which has a logical representation of data for an enterprise of business organisation. It was introduced by **P.P. Chen** in **1976**. Entity Relationship Model is a high-level conceptual data model diagram. ER model helps to systematically analyse data requirements to produce a well-designed database.

The ER Model represents **real-world entities and the relationships between them.** Creating an ER Model in DBMS is considered as a best practice before implementing your database.ER Modelling helps you to analyse data requirements systematically to produce a well-designed database.

It develops a conceptual design for the database. It also develops a very simple and easy to design view of data. In ER modelling, the database structure is portrayed as a diagram called an **entity-relationship diagram.**

Basic Concepts:

ER modelling is a high level conceptual data model developed to facilitate database design.

A conceptual data model is a set of concepts that describe the structure of a database and the associated retrieval and update transactions on the database.

E-R data model is a collection of objects of similar structures called an entity set. The relationship between entity sets is represented on the basis of number of entities from entity set that can be associated with the number of entities of another state such as one-to-one(1:1),one-to-many(1:n) or many-to-many (n:n) relationships.

ER model provides the following main semantic concepts to the database designers:

Entities: which are distinct objects in a user enterprise.

Relationships: which are meaningful interactions among the objects.

Attributes: which describe the entities and relationships. Each such attribute is associated with a value set also called domain and it can take a value from this value set.

Constraints: on the entities, relationships and attributes.

Entity (Entities):-

An entity is **an object for or a thing in real world with an independent existence and that it is distinguishable from other objects.**

Entities are the principal data objects about which information is to be collected.

An entity may be an object with a **physical existence** such as a person, car, house, employee or city. Or, it may be an object with the **conceptual existence** such as a company, an enterprise, a job or an event of informational interest.

Each entity has attributes.

Some of the examples of entity are given below:

Person: STUDENT, PATIENT, EMPLOYEE, DOCTOR, ENGINEER

Place: CITY, COUNTRY, STATE

Event: SEMINAR, SALE, RENEWAL, COMPETITION

Object: BUILDING, AUTOMOBILE, MACHINE, FURNITURE, TOY

Concept: COURSE, ACCOUNT, TRAINING CENTRE, WORK, CENTRE

In ER modelling, entities are considered as abstract but meaningful 'things' that exist in the user enterprise. Such things are modelled as entities that may be described by attributes.

They may also interact with one another in any number of relationships. Semantic net can be used to describe a model made up of number of entities. An entity is represented by a set of attributes. Each entity has a value for each of its attributes.

Entity Set:-

An entity set (also called as entity type) is a set of entities of the same type that share the same properties or attributes. In E-R modelling similar entities are grouped into entity (type) set. An entity (type) set it is a group of objects with the same properties. These are identified by the enterprise as having an independent existence. It can have objects with physical (real) existence or objects with conceptual (abstract) existence. Each entity type is identified by a name and a list of properties. A database normally contents many different entity types and not to a single entity occurrence.

In other words the word entity in E-R modelling corresponds to a table and not to a row in the relational environment. The E-R model refers to a specific table row as an entity instance or entity occurrence. An entity occurrence is a uniquely identifiable object of an entity type.

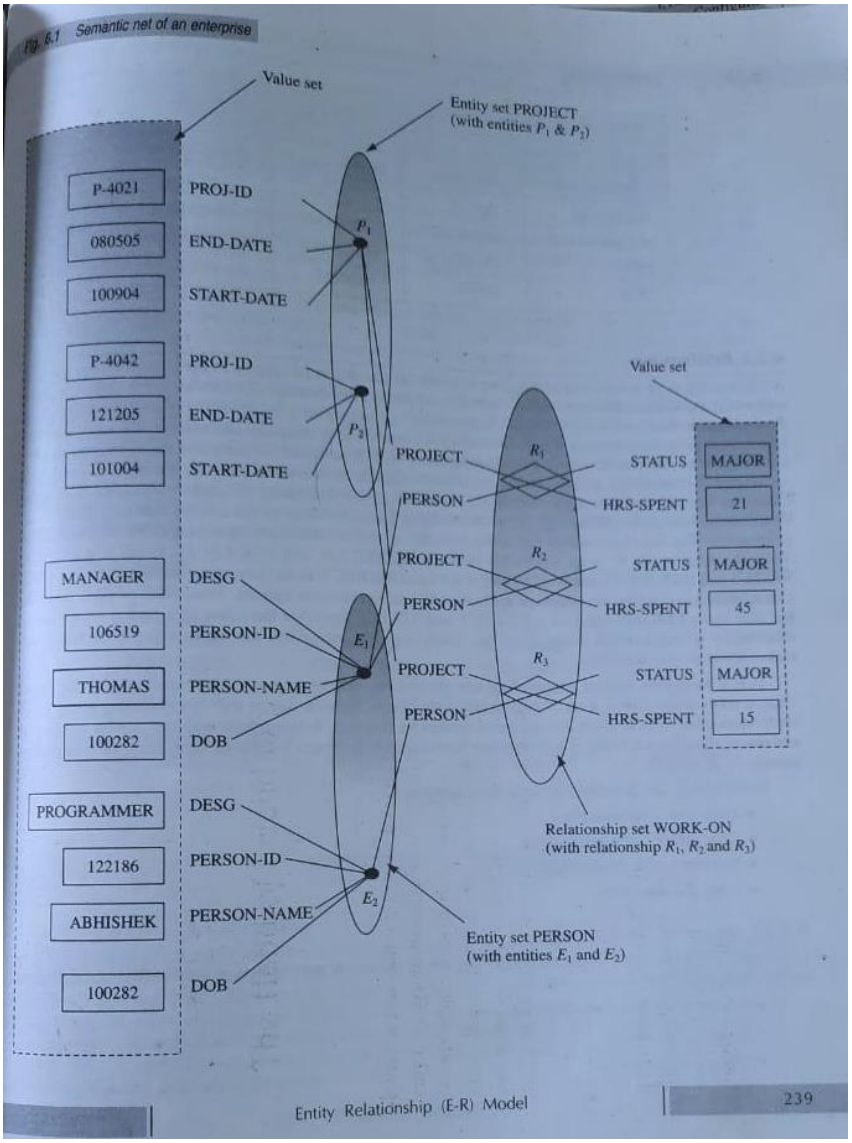


Fig. 6.1 Semantic net of an enterprise

Entity Relationship (E-R) Model 239

Fig. 6.2 Entity type with instances

Entity type: PERSON			
Attributes		Instances	
Attribute Name	Domain	First Instance	Second Instance
PERSON-ID	INT (6)	106519	122816
PERSON-NAME	CHAR (20)	THOMAS	ABHISHEK
DESG	CHAR (15)	MANAGER	PROGRAMMER
DOB	DATE	100282	220187

6.2.2 Relationship

A relationship is

Classification of Entity types:

Entity types can be classified as been **strong** or **weak** entity.

Strong entity:

An entity type that is not existence dependent on some other entity type is called strong entity type. The strong entity type has characteristics that have each entity occurrence is uniquely identifiable using the primary key attribute of that entity type. Strong entity types are sometimes referred to as parent owner or dominant entities.

Weak entity:

An entity type that is existence dependent on some other entity type is called weak entity type. Weak entity types are sometimes referred to as child dependent or for subordinate entities. The weak entity type has characteristics that each entity occurrence cannot be uniquely identifiable using only the attributes associated with that entity type.

Attributes:

An attribute is a property of an entity or a relationship type. **An entity is described using a set of attributes.** All entities in a given entity type have the same or similar attributes.

For example: An **EMPLOYEE** entity type could use **NAME**, social security number (**SSN**), date of birth (**DOB**) and so on as characteristics.

Each attribute is associated with a set of values called a **domain**. A domain of possible values identifies each attribute associated with an entity type. The domain defines the potential values that an attribute may hold and is similar to the domain concept in relational model.

An attributes can be classified into the following categories:

- Simple attribute
- Composite attribute
- Single valued attribute
- Multi valued attribute
- Derived attribute

Simple attributes

Simple attribute is an attribute composed of a single component with an independent existence. Simple attribute cannot be subdivided or broken down into smaller components. Simple attributes are sometimes called **atomic** attributes.

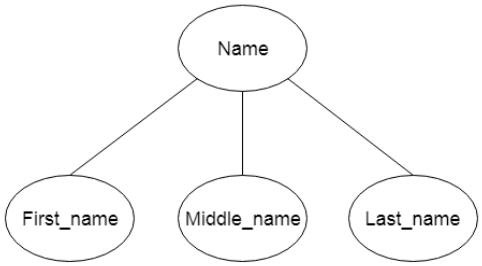
For example: **EMP-ID**, **salary** of the **EMPLOYEE** entity are the example of simple attributes.

Composite attributes:

A Composite attributes is an attribute composed of multiple components, each with an independent existence. Some attributes can be further broken down or divided into smaller components with an independent existence of their own.

For example, **EMP-NAME** attribute of EMPLOYEE entity can further divided as **FIRST- NAME** and **LAST -NAME.**

Another example is EMPLOYEE address that is **Street- Address** can be subdivided into 3 simple attributes namely **Street –Number**, **Street -Name** and **Apartment- Number**.



Single valued attribute

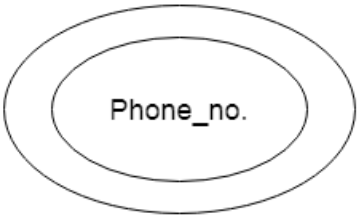
A single valued attribute is an attribute that holds a single value for each occurrence of an entity type.

For example, each occurrence of **EMPLOYEE** entity as a single value for the **EMP-ID** number employee ID attributes for example '**106519**' and therefore the **EMP-ID** attribute is referred to as being single value. The majority of attributes are single valued.

Multivalued attributes

Multivalued attribute is an attribute that holds multiple values for each occurrence of an entity type. That means multivalued attributes can take more than one value.

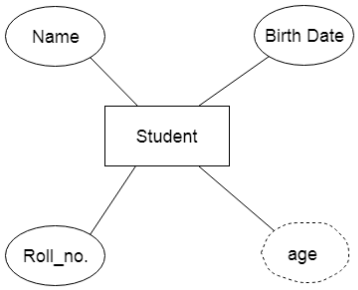
For example, a student can have more than one phone number.



Derived attributes

Attribute is an attribute that represents a value that is derivable from the value of a related attribute or set of attributes not essentially the same entities set. Therefore the value held by some attributes are derived from two or more attribute values. For example, the value for the **PROJ-DURN** attribute of the entity **PROJECT** can be calculated from the project **START- DATE** and project **END-DATE** attributes. The project duration attribute is referred to as **derived attribute**.

For example, A person's **age** changes over time and can be derived from another attribute like Date of birth.

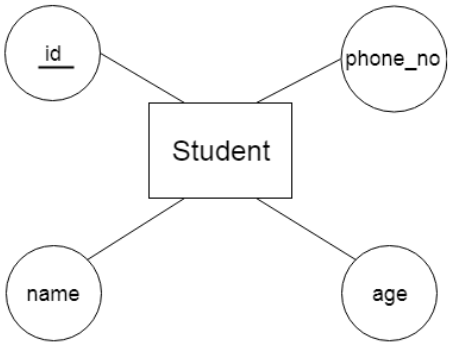


Identifier attributes

Each entity is required to be identified uniquely in a particular entity set. Using one or more entity attributes as an entity identifier this identification is done. These attributes are known as **identifier attributes**.

Therefore an identifier is an attribute or combination of attributes that uniquely identifies individual instances of an entity type.

For example, **PER-ID** can be a identifier attribute in case of **PERSON** or **PROJ-ID** in case of **PROJECTS**.



Relationship:

Relationship is an association among two or more entities that is of interest to the enterprise. Entities take part in relationships. We can often identify relationships with verbs or verb phrases.

For example, an employee **works_at** a department, a student **enrolls** in a course. Here, **Works_at** and **Enrolls** are called relationships.

Relationship as such has no physical or conceptual existence other than that which depends upon their entity association. Particular occurrence of relationship is called a relationship instance for relational occurrence.

A relationship is used to describe the relation between entities.



Relationship Set:

In E-R modelling, similar relationships are grouped into relationship sets also called relationship type. Thus, relationship type is a set of meaningful associations between one or more participating entity types. Each relationship type is given a name that describes its function. Relationships with the same attribute fall into one relationship set.

A set of relationships of similar type is called a relationship set. Like entities, a relationship too can have attributes. These attributes are called descriptive attributes.

Degree of Relationship

The number of participating entities in a relationship defines the degree of the relationship.

- Recursive or unary relationship
- Binary relationship
- Ternary relationship
- n-ary relationship

Recursive relationship:

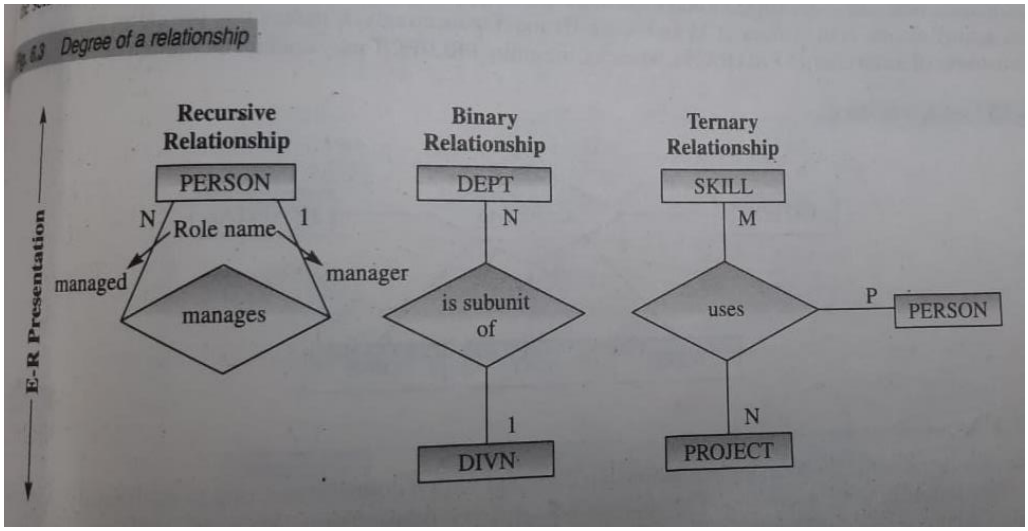
Recursive relationship is a relationship between the instances of a single entity type. It is a relationship type in which the same entity type is associated with more than once in different roles.

Binary relationship:

The association between the **two** entities is called binary relationship. Binary relationship is the most common type relationship and its degree of relation is **two**.

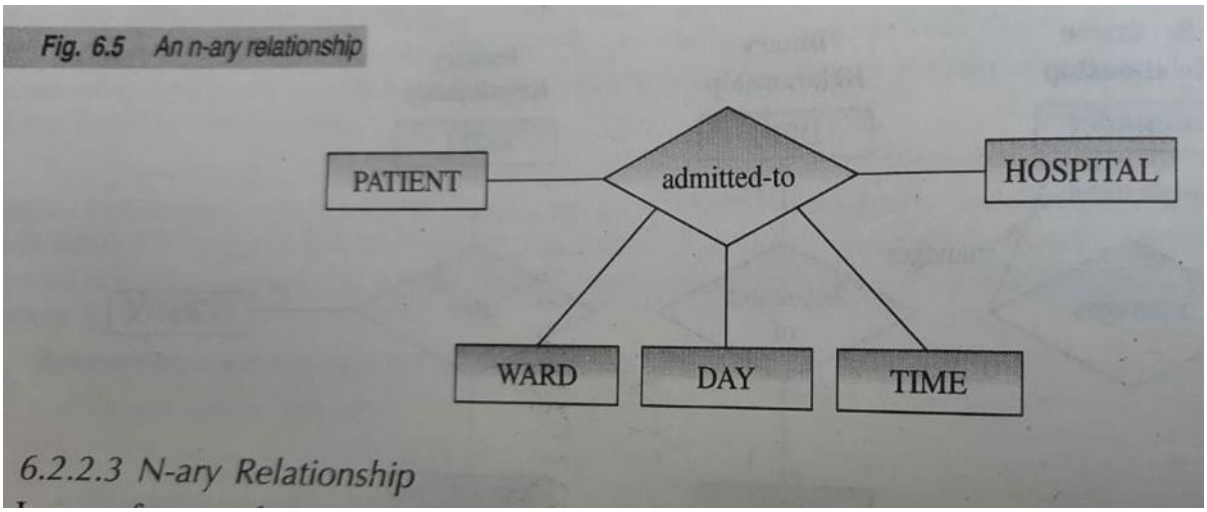
Ternary relationship:

A ternary relationship is an association among **three** entities and its degree of relationship is **three** (3).



n-ary relationship:

When there are n entities set participating in a relation, the relationship is called as n-ary relationship.



Constraints:

An E-R enterprise schema may define certain constraints to which the contents of a database must conform. In this section, we examine mapping cardinalities and participation constraints, which are two of the most important types of constraints.

Mapping Cardinality:

The number of times an entity of an entity set participates in a relationship set is known as cardinality.

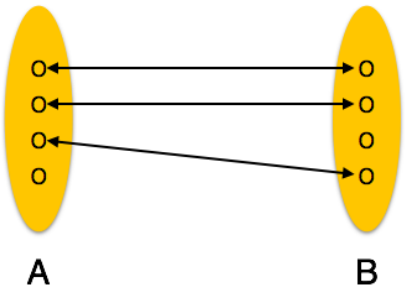
Mapping cardinalities are most useful in describing binary relationship sets, although they can contribute to the description of relationship sets that involve more than two entity sets.

Cardinality can be of different types:

- One to one
- One-to-many
- Many to one
- Many to many

One to one:

One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

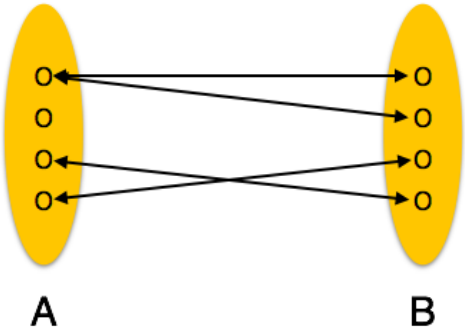


Example: One student can register for numerous courses. However, all those courses have a single line back to that one student.



One-to-many:

One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.

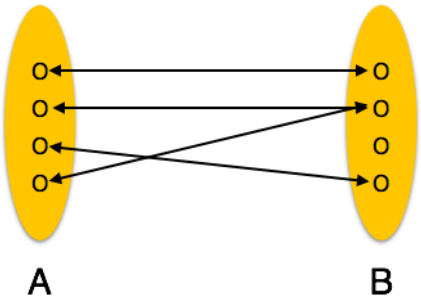


For example, one class is consisting of multiple students.

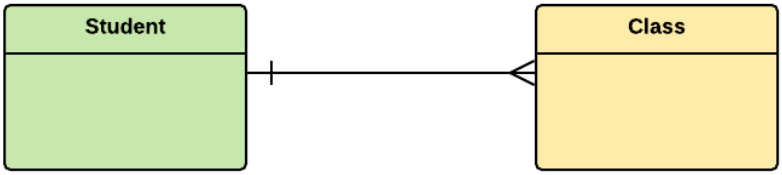


Many-to-One:

More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.

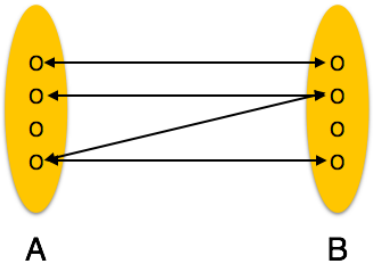


For example, many students belong to the same class.



Many to many:

One entity from A can be associated with more than one entity from B and vice versa.



For example, Students as a group are associated with multiple faculty members, and faculty members can be associated with multiple students.



The appropriate mapping cardinality for a particular relationship set obviously depends on the real-world situation that the relationship set is modelling.

As an illustration, consider the **borrower** relationship set. If, in a particular bank, a **loan** can belong to only **one customer**, and a **customer** can have **several loans**, then

the **relationship set from customer to loan is one to many**. If a loan can belong to several customers (as can loans taken jointly by several business partners), the relationship set is **many to many.**

Participation Constraints

The participation of an entity set **E** in a relationship set **R** is said to be **total** if every entity in **E** participates in at least one relationship in **R.** If only some entities in **E** participate in relationships in **R**, the participation of entity set **E** in relationship **R** is said to be **partial**. For example, we expect every loan entity to be related to at least one customer through the **borrower** relationship. Therefore the participation of loan in the relationship set borrower is total. In contrast, an individual can be a bank customer Whether or not she has a loan with the bank. Hence, it is possible that only some of the customer entities are related to the loan entity set through the borrower relationship, and the participation of customer in the borrower relationship set is therefore **partial**.

Entity Relationship Diagram:

An Entity–relationship model (**ER model**) describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**.

ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships. An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an **entity** is a **table** or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

ER Diagrams contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships.

Components of E-R Diagram:

The E-R diagram has three main components:

1. **Entity**:

   An Entity can be any object, place, person or class.

   In E-R Diagram, an entity is represented using **rectangles**. Consider an example of an **Organisation**. **Employee**, **Manager**, **Department**, **Product** and many more can be taken as entities from an Organisation.

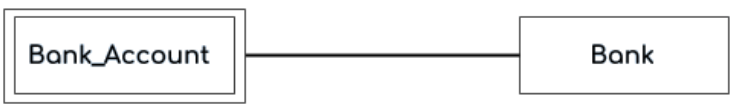


   In this example, **Employee** and **Department** are the entities.

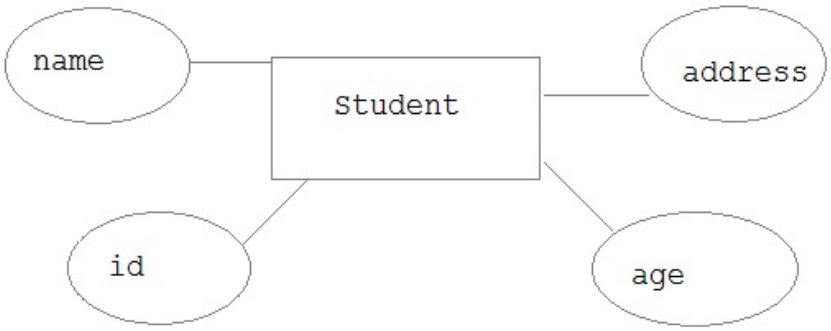
Weak Entity:

An entity that cannot be uniquely identified by its own attributes and relies on the relationship with other entity is called **weak entity**.

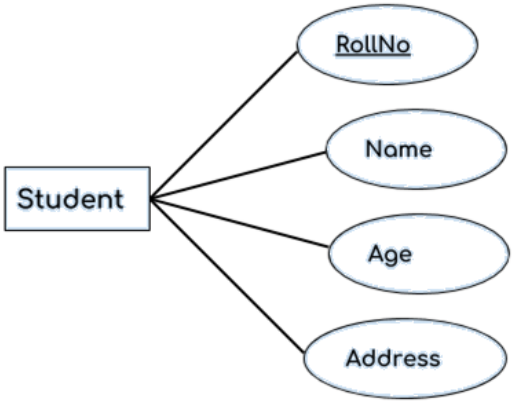
The weak entity is represented by a **double rectangle**. For example – a **bank account** cannot be uniquely identified without knowing the **bank** to which the account belongs, so bank account is a weak entity.



2. **Attribute**:
   An Attribute describes a property or characteristic of an entity.
   An attribute is represented using **eclipse/Oval**.
   For example, **Name**, **Age**, **Address** etc. can be **attributes** of a Student.

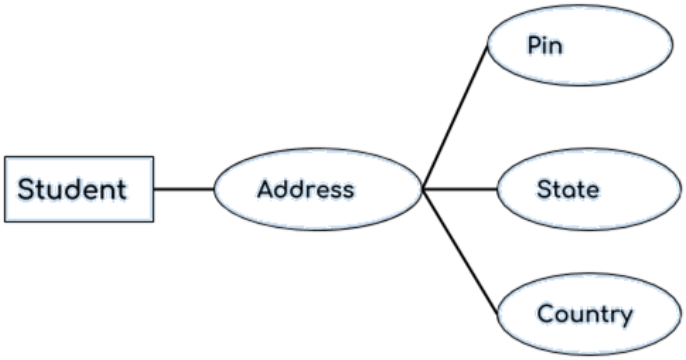


Key Attributes:
A key attribute can uniquely identify an entity from an entity set. It is used to represent **Primary** key.
For example, **student roll** number can uniquely identify a **student** from a **set of students**. Key attribute is represented by oval same as other attributes however the **text of key attribute is underlined**.



Composite attribute:

An attribute that is a combination of other attributes is known as **composite attribute**. For example, in **student** entity, the student **address** is a **composite** attribute as an address is composed of other attributes such as **pin code**, **state**, **country**.



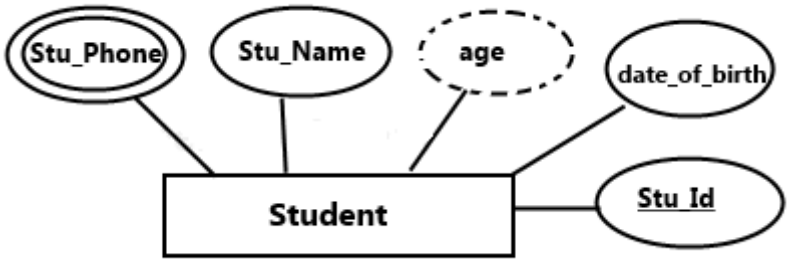
Multivalued attribute:

An attribute that can hold **multiple values** is known as **multivalued** attribute.
It is represented with **double ovals** in an ER Diagram.
For example – A **person** can have more than one **phone numbers** so the phone number attribute is multivalued.



Derived attribute:

A derived attribute is one whose value is dynamic and derived from another attribute.
It is represented by **dashed oval** in an ER Diagram.
For example – **Person age** is a derived attribute as it changes over time and can be derived from another attribute (Date of birth).



3. **Relationship**:
   A Relationship describes relations between entities. Relationship is represented using diamonds.

   There are four types of relationships:

   - **One to One Relationship**:
     When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship. For example, a person has only one passport and a passport is given to one person.



- **One to Many Relationship**:
  When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship. For example – a customer can place many orders but a order cannot be placed by many customers.



- **Many to One Relationship**:
  When more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship. For example – many students can study in a single college but a student cannot study in many colleges at the same time.



- **Many to Many Relationship**:
  When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship. For example, a can be assigned to many projects and a project can be assigned to many students.



Extended E-R features: Generalization, Specialization and Aggregation

Generalization, Specialization and Aggregation in ER model are used for data abstraction in which abstraction mechanism is used to hide details of a set of objects.

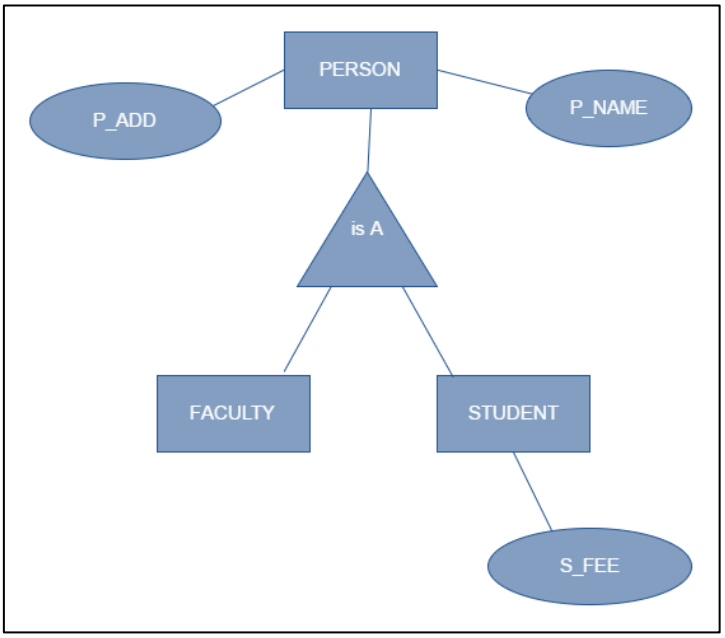
Generalization:

Generalization is the process of extracting common properties from a set of entities and create a generalized entity from it.

It is a bottom-up approach in which two or more entities can be generalized to a higher level entity if they have some attributes in common.

For Example, **STUDENT** and **FACULTY** can be generalized to a higher level entity called **PERSON** as shown in Figure 1.

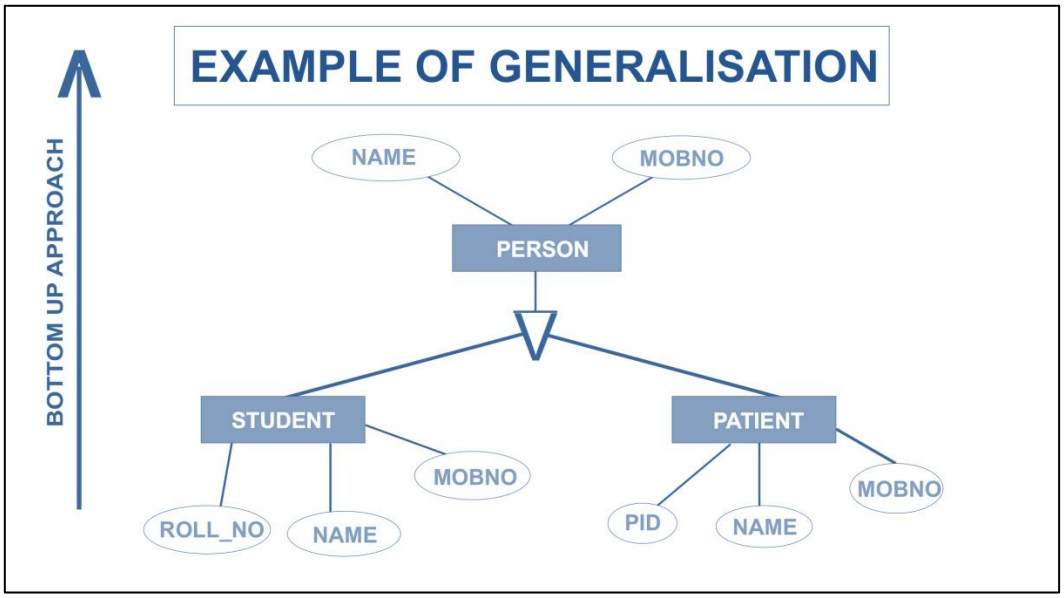
In this case, common attributes like **P_NAME, P_ADD** become part of higher entity (**PERSON**) and specialized attributes like **S_FEE** become part of specialized entity (**STUDENT**).



It works on the principle of **bottom up approach**. In Generalization lower level functions are combined to form higher level function which is called as entities. This process is repeated further to make advanced level entities.

In the Generalization process properties are drawn from particular entities and thus we can create generalized entity. We can summarize Generalization process as it combines subclasses to form superclass.

Consider two entities **Student** and **Patient**. These two entities will have some characteristics of their own. For example **Student** entity will have **Roll_No**, **Name** and **Mob_No** while **patient** will have **PId**, **Name** and **Mob_No** characteristics. Now in this example **Name** and **Mob_No** of both **Student** and **Patient** can be combined as a **Person** to form one higher level entity and this process is called as **Generalization** Process.



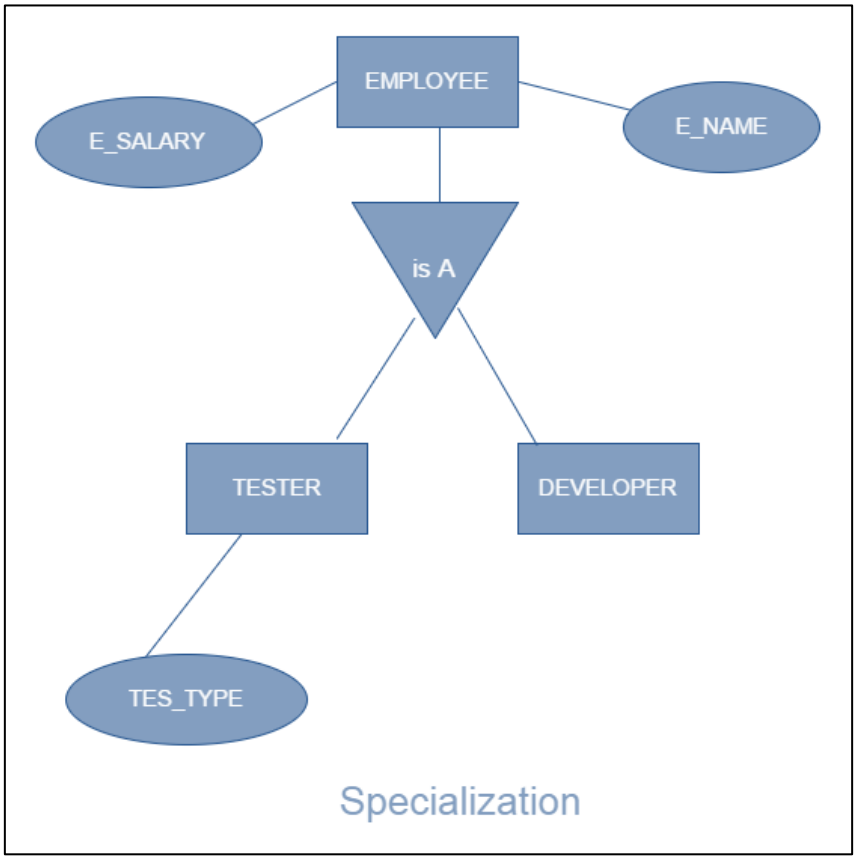
Specialization:

In specialization, an entity is divided into sub-entities based on their characteristics.

It is a **top-down approach** where higher level entity is specialized into two or more lower level entities.

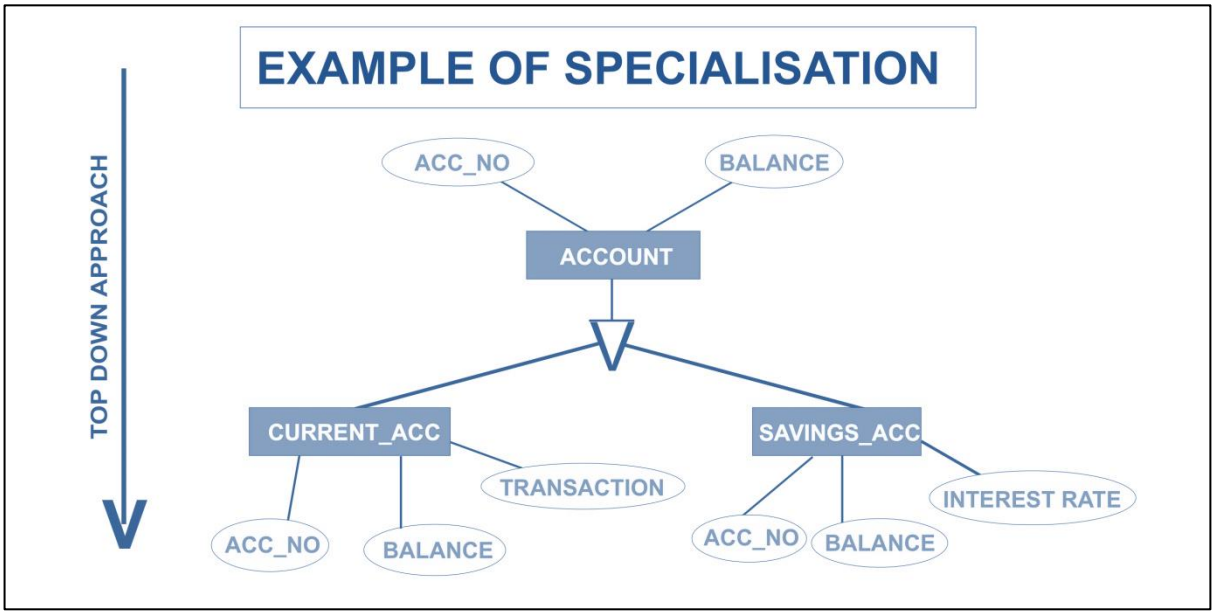
For Example, **EMPLOYEE** entity in an **Employee management system** can be specialized into **DEVELOPER**, **TESTER** etc. as shown in Figure 2.

In this case, common attributes like **E_NAME**, **E_SAL** etc. become part of higher entity (**EMPLOYEE**) and specialized attributes like **TES_TYPE** become part of specialized entity (**TESTER**).



We can say that Specialization is opposite of Generalization. In Specialization things are broken down into smaller things to simplify it further. We can also say that in Specialization a particular entity gets divided into sub entities and it's done on the basis of it's characteristics. Also in Specialization Inheritance takes place.

Consider an entity **Account**. This will have some attributes consider them **Acc_No** and **Balance**. **Account** entity may have some other attributes like **Current_Acc** and **Savings_Acc**. Now **Current_Acc** may have **Acc_No**, **Balance** and **Transactions** while **Savings_Acc** may have **Acc_No**, **Balance** and **Interest_Rate** henceforth we can say that specialized entities inherits characteristics of higher level entity.



Difference between Generalization and Specialization:

GENERALIZATION	SPECIALIZATION
Generalization works in Bottom-Up approach.	Specialization works in top-down approach.
In Generalization, size of schema gets reduced.	In Specialization, size of schema gets increased.
Generalization is normally applied to group of entities.	We can apply Specialization to a single entity.
Generalization can be defined as a process of creating groupings from various entity sets	Specialization can be defined as process of creating subgrouping within an entity set
In Generalization process, what actually happens is that it takes the union of two or more lower-level entity sets to produce a higher-level entity sets.	Specialization is reverse of Generalization. Specialization is a process of taking a subset of a higher level entity set to form a lower-level entity set.
Generalization process starts with the number of entity sets and it creates high-level entity with the help of some common features.	Specialization process starts from a single entity set and it creates a different entity set by using some different features.
In Generalization, the difference and similarities between lower entities are	In Specialization, a higher entity is split to form lower entities.

ignored to form a higher entity.	
There is no inheritance in Generalization.	There is inheritance in Specialization.

Attribute Inheritance:

A crucial property of the higher- and lower-level entities created by specialization and generalization is **attribute inheritance**.

The attributes of the higher-level entity sets are said to be **inherited** by the lower-level entity sets.

For example, **student** and **employee** inherit the attributes of person. Thus, student is described by its **ID**, **name**, and **address** attributes, and additionally a **tot_cred** attribute; **employee** is described by its **ID**, **name**, and **address** attributes, and additionally a **salary** attribute.

Whether a given portion of an E-R model was arrived at by specialization or

generalization, the outcome is basically the same:

- A higher-level entity set with attributes and relationships that apply to all of its lower-level entity sets.
- Lower-level entity sets with distinctive features that apply only within a particular lower-level entity set.

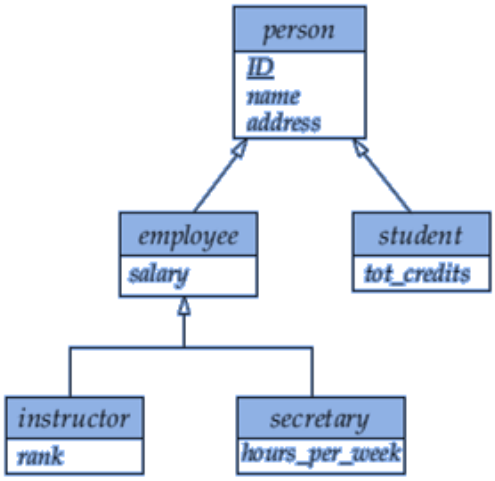


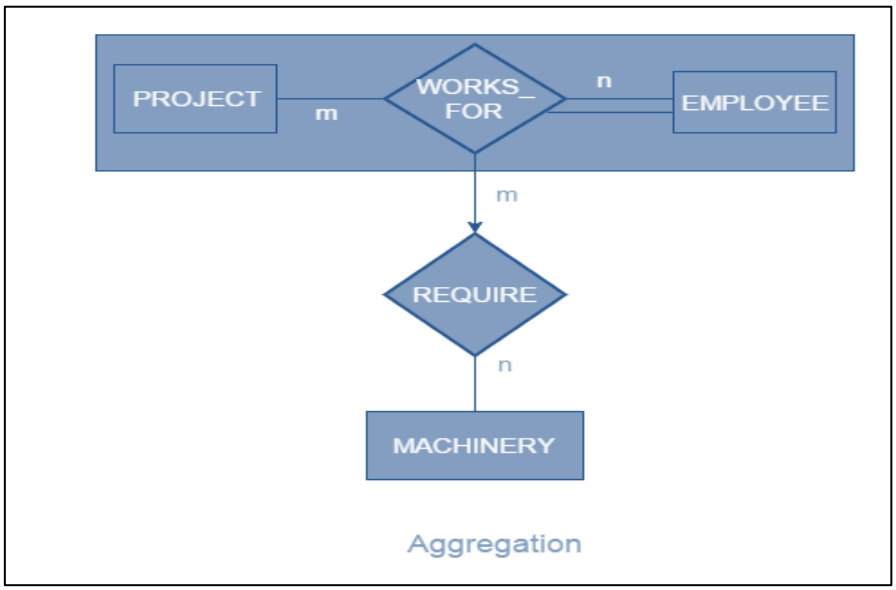
Figure 7.21 Specialization and generalization.

Figure 7.21 depicts a hierarchy of entity sets. In the figure, employee is a lower-level entity set of **person** and a higher-level entity set of the **instructor** and **secretary** entity sets. In a hierarchy, a given entity set may be involved as a lower level entity set in only **one ISA relationship**; that is, entity sets in this diagram have only **single inheritance**. If an entity set is a lower-level entity set in **more than one ISA relationship**, then the entity set has **multiple inheritance**, and the resulting structure is said to be a **lattice**.

Aggregation:

An ER diagram is not capable of representing relationship between an entity and a relationship which may be required in some scenarios. In those cases, a relationship with its corresponding entities is aggregated into a higher level entity.

For Example, **Employee** working for a project may require some machinery. So, **REQUIRE** relationship is needed between relationship **WORKS_FOR** and entity **MACHINERY**. Using aggregation, **WORKS_FOR** relationship with its entities **EMPLOYEE** and **PROJECT** is aggregated into single entity and relationship **REQUIRE** is created between aggregated entity and **MACHINERY**.



Introduction to the Relational Model

The relational model is today the primary data model for commercial data processing applications. It attained its primary position because of its simplicity, which eases the job of the programmer, compared to earlier data models such as the network model or the hierarchical model. Relational Model was proposed by **E.F. Codd** to model data in the form of **relations** or **tables**. After designing the conceptual model of Database using ER diagram, we need to convert the conceptual model in the relational model which can be implemented using any RDBMS languages like Oracle SQL, MySQL etc. So we will see what Relational Model is.

Relational Model (RM) represents the **database** as a collection of **relations**. A **relation** is nothing but a **table** of values. Every **row** in the **table** represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.

Some popular Relational Database management systems are:

- DB2 and Informix Dynamic Server - **IBM**
- Oracle and RDB – **Oracle**
- SQL Server and Access - **Microsoft**

Structure of Relational Databases:

A relational database consists of a collection of **tables**, each of which is assigned a unique name. For example, consider the ***instructor*** table of Figure 2.1, which stores information about instructors.

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Figure 2.1 The *instructor* relation.

It stores information about instructors. The table has four column headers: **ID**, **name**, **dept**

name, and **salary**. Each row of this table records information about an instructor, consisting of the instructor's ID, name, dept name, and salary.

In general, a **row** in a table represents a **relationship** among a set of values. Since a table is a collection of such relationships, there is a close correspondence between the concept of table and the mathematical concept of relation, from which the relational data model takes its name. In mathematical terminology, a **tuple** is simply a sequence (or list) of values. A relationship between **n values** is represented mathematically by an **n-tuple** of values, i.e., a tuple with n values, which corresponds to a row in a table.

Thus, in the relational model the term **relation** is used to refer to a **table**, while the term **tuple** is used to refer to a **row**. Similarly, the term **attribute** refers to a **column** of a table.

Relational Databases Terms:

- **Attribute**: Each **column** in a Table. Attributes are the properties which define a relation. e.g., Student_Rollno, NAME, etc.
- **Relation/Table** – In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.
- **Tuple**: It is nothing but a single row of a table, which contains a single record.
- **Relation Schema**: A relation schema represents the name of the relation with its attributes.

  Instructor(ID,Name,Dept,Salary)
- **Degree**: The total number of attributes which in the relation is called the degree of the relation.
- **Cardinality**: Total number of rows present in the Table.
- **Column**: The column represents the set of values for a specific attribute.
- **Relation instance** – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.
- **Relation key** - Every row has one, two or multiple attributes, which is called relation key.
- **Attribute domain** – Every attribute has some pre-defined value and scope which is known as attribute domain.

DBMS Keys: Candidate, Super, Primary, Foreign

What are Keys in DBMS?

A **KEY** in **DBMS** is **an attribute or set of attributes** which **helps you to identify a row (tuple) in a relation (table).**

They allow you to find the relation between two tables.

Keys help you uniquely identify a row in a table by a combination of one or more columns in that table.

Key is also helpful for finding unique record or row from the table. Database key is also helpful for finding unique record or row from the table.

Example:

Employee ID	FirstName	LastName
11	Andrew	Johnson
22	Tom	Wood
33	Alex	Hale

In the above-given example, **Employee ID** is a **primary key** because it **uniquely** identifies an employee record. In this table, no other employee can have the same employee ID.

Why we need a Key?

Here are some reasons for using sql key in the DBMS system.

Keys help you to identify any row of data in a table. In a real-world application, a table could contain thousands of records. Moreover, the records could be duplicated. Keys ensure that you can uniquely identify a table record despite these challenges.

Allows you to establish a relationship between and identify the relation between tables

Help you to enforce identity and integrity in the relationship.

Types of Keys in Database Management System

There are mainly **seven** different types of Keys in DBMS and each key has it's different functionality:

- **Super Key -** A super key is a group of **single or multiple keys** which identifies rows in a table.
- **Primary Key -** is a column or group of columns in a table that **uniquely** identify every row in that table.
- **Candidate Key -** is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes.
- **Alternate Key -** is a column or group of columns in a table that uniquely identify every row in that table.

- **Foreign Key -** is a column that creates a relationship between two tables. The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity.
- **Compound Key -** has two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique by itself within the database.
- **Composite Key -** COMPOSITE KEY is a combination of two or more columns that uniquely identify rows in a table.
- **Surrogate Key -** An artificial key which aims to uniquely identify each record is called a surrogate key. These kinds of key are unique because they are created when you don't have any natural primary key.

What is the Super key?

A **superkey** is a group of single or multiple keys which identifies rows in a table.

A Super key may have additional attributes that are not needed for unique identification.

Example:

EmpSSN	EmpNum	Empname
9812345098	AB05	Shown
9876512345	AB06	Roslyn
199937890	AB07	James

In the above-given example, **EmpSSN** and **EmpNum** name are **superkeys**.

What is a Primary Key?

PRIMARY KEY is a column or group of columns in a table that **uniquely** identify every row in that table. The Primary Key can't be a duplicate meaning the same value can't appear more than once in the table. A table cannot have more than one primary key.

Rules for defining Primary key:

- Two rows can't have the same primary key value
- It must for every row to have a primary key value.
- The primary key field cannot be null.
- The value in a primary key column can never be modified or updated if any foreign key refers to that primary key.

Example:

In the following example, <code>StudID</code> is a Primary Key.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

What is the Alternate key?

ALTERNATE KEYS is a column or group of columns in a table that uniquely identify every row in that table. A table can have multiple choices for a primary key but only one can be set as the primary key. All the keys which are not primary key are called an Alternate Key.

Example:

In this table, **StudID**, **Roll No**, **Email** are qualified to become a **primary** key. But since StudID is the primary key, Roll No, Email becomes the **alternative key**.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

What is a Candidate Key?

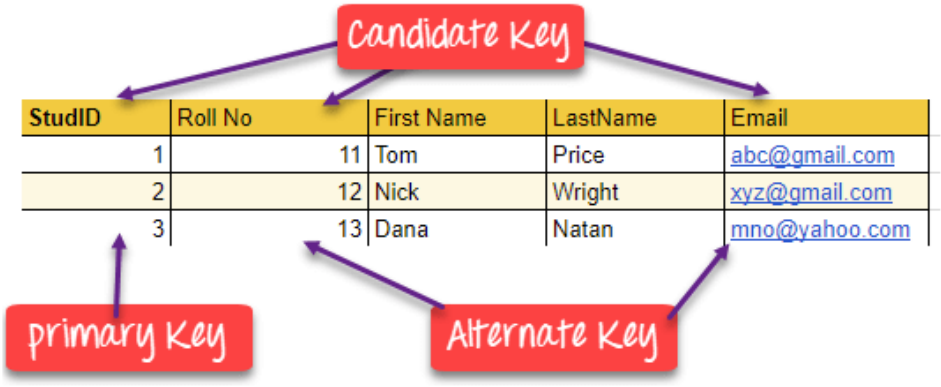
CANDIDATE KEY is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes. The Primary key should be selected from the candidate keys. Every table must have at least a single candidate key. A table can have multiple candidate keys but only a single primary key.

Properties of Candidate key:

- It must contain unique values
- Candidate key may have multiple attributes
- Must not contain null values
- It should contain minimum fields to ensure uniqueness
- Uniquely identify each record in a table

Example: In the given table Stud ID, Roll No, and email are candidate keys which help us to uniquely identify the student record in the table.

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com



What is the Foreign key?

FOREIGN KEY is a column that creates a relationship between two tables. The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity. It acts as a cross-reference between two tables as it references the primary key of another table.

Example:

DeptCode	DeptName
001	Science
002	English
005	Computer

Teacher ID	Fname	Lname
B002	David	Warner
B017	Sara	Joseph
B009	Mike	Brunton

In this key in dbms example, we have two table, teach and department in a school. However, there is no way to see which search work in which department.

In this table, adding the foreign key in **Deptcode** to the Teacher name, we can create a relationship between the two tables.

Teacher ID	DeptCode	Fname	Lname
B002	002	David	Warner
B017	002	Sara	Joseph
B009	001	Mike	Brunton

This concept is also known as Referential Integrity.

What is the Compound key?

COMPOUND KEY has two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique by itself within the database. However, when combined with the other column or columns the combination of composite keys become unique. The purpose of the compound key in database is to uniquely identify each record in the table.

Example:

OrderNo	PorductID	Product Name	Quantity
B005	JAP102459	Mouse	5
B005	DKT321573	USB	10
B005	OMG446789	LCD Monitor	20
B004	DKT321573	USB	15
B002	OMG446789	Laser Printer	3

In this example, OrderNo and ProductID can't be a primary key as it does not uniquely identify a record. However, a compound key of **Order ID** and **Product ID** could be used as it uniquely identified each record.

What is the Composite key?

COMPOSITE KEY is a combination of two or more columns that uniquely identify rows in a table. The combination of columns guarantees uniqueness, though individually uniqueness is not guaranteed. Hence, they are combined to uniquely identify records in a table.

The difference between **compound** and the **composite** key is that any part of the compound key can be a foreign key, but the composite key may or maybe not a part of the foreign key.

What is a Surrogate key?

SURROGATE KEYS is An artificial key which aims to uniquely identify each record is called a surrogate key. This kind of partial key in dbms is unique because it is created when you don't have any natural primary key. They do not lend any meaning to the data in the table. Surrogate key is usually an integer. A surrogate key is a value generated right before the record is inserted into a table.

Fname	Lastname	Start Time	End Time
Anne	Smith	09:00	18:00
Jack	Francis	08:00	17:00
Anna	McLean	11:00	20:00
Shown	Willam	14:00	23:00

Above, given example, shown shift timings of the different employee. In this example, a surrogate key is needed to uniquely identify each employee.

Surrogate keys in sql are allowed when

- No property has the parameter of the primary key.
- In the table when the primary key is too big or complicated.

Difference between Primary key & Foreign key

Primary Key	Foreign Key
Helps you to uniquely identify a record in the table.	It is a field in the table that is the primary key of another table.
Primary Key never accept null values.	A foreign key may accept multiple null values.
Primary key is a clustered index and data in the DBMS table are physically organized in the sequence of the clustered index.	A foreign key cannot automatically create an index, clustered or non-clustered. However, you can manually create an index on the foreign key.
You can have the single Primary key in a table.	You can have multiple foreign keys in a table.

Summary:

- A key in SQL is an attribute or set of attributes which helps you to identify a row(tuple) in a relation(table)
- DBMS keys allow you to establish a relationship between and identify the relation between tables
- Seven Types of DBMS keys are Super, Primary, Candidate, Alternate, Foreign, Compound, Composite, and Surrogate Key.
- A super key is a group of single or multiple keys which identifies rows in a table.
- A column or group of columns in a table which helps us to uniquely identifies every row in that table is called a primary key
- All the keys which are not primary key are called an alternate key
- A super key with no repeated attribute is called candidate key
- A compound key is a key which has many fields which allow you to uniquely recognize a specific record
- A key which has multiple attributes to uniquely identify rows in a table is called a composite key
- An artificial key which aims to uniquely identify each record is called a surrogate key
- Primary Key never accepts null values while a foreign key may accept multiple null values.

Constraints on Relational database model

On modelling the design of the relational database we can put some restrictions like

- What values are allowed to be inserted in the relation,
- What kind of modifications and deletions are allowed in the relation.

These are the restrictions we impose on the relational database.

In models like ER models, we did not have such features.

Constraints in the databases can be categorized into 3 main categories:

- Constraints that are applied in the data model is called **Implicit constraints**.
- Constraints that are directly applied in the schemas of the data model, by specifying them in the DDL(Data Definition Language). These are called as **schema-based constraints** or **Explicit constraints**.
- Constraints that cannot be directly applied in the schemas of the data model. We call these Application based or semantic constraints.

There are many types of Integrity Constraints in DBMS. Constraints on the Relational database management system are mostly divided into 4 types:

1. **Domain constraints**
2. **Key constraints**
3. **Entity Integrity constraints**
4. **Referential integrity constraints**

Domain constraints :

1. Every domain must contain atomic values (smallest indivisible units) it means **composite** and **multi-valued attributes** are not allowed.
2. We perform data type check here, which means when we assign a data type to a column we limit the values that it can contain. Eg. If we assign the data type of attribute age as **int**, we can't give it values other than **int** data type.

Example:

EmpID	Name	Phone
01	Pravin Kumar	1234567890 0123456789

Explanation:

In the above relation, **Name** is a **composite** attribute and **Phone** is a **multi-values** attribute, so it is violating domain constraint.

Key Constraints or Uniqueness Constraints:

1. These are called uniqueness constraints since it ensures that **every tuple in the relation should be unique.**
2. A relation can have multiple keys or candidate keys(minimal superkey), out of which we choose one of the keys as primary key, we don't have any restriction on choosing the primary key out of candidate keys, but it is suggested to go with the candidate key with less number of attributes.
3. **Null values are not allowed in the primary key**, hence Not Null constraint is also a part of key constraint.

EmpID	Name	Phone
01	Pravin Kumar	1234567890
02	Vijay Kumar	0123456789
01	Ajay Kumar	2345657799

Explanation:

In the above table, **EmpID** is the primary key, and first and the last tuple has the same value in **EmpID** ie 01, so it is violating the key constraint.

Entity Integrity Constraints:

1. Entity Integrity constraints says that **no primary key can take NULL value**, since using primary key we identify each tuple uniquely in a relation.

EmpID	Name	Phone
01	Pravin Kumar	1234567890
02	Vijay Kumar	0123456789
NULL	Ajay Kumar	2345657799

Explanation:

In the above relation, **EmpID** is made primary key, and the primary key can't take NULL values but in the third tuple, the primary key is null, so it is a violating Entity Integrity constraints.

Referential Integrity Constraints:

1. The Referential integrity constraints is specified between two relations or tables and used to maintain the consistency among the tuples in two relations.
2. This constraint is enforced through **foreign key**, when an attribute in the foreign key of relation R1 have the same domain(s) as the primary key of relation R2, then the foreign key of R1 is said to reference or refer to the primary key of relation R2.
3. The values of the foreign key in a tuple of relation R1 can either take the values of the primary key for some tuple in relation R2, or can take NULL values, but can't be empty.

EmpID	**Name**	**Dept No**
01	Pravin Kumar	12
02	Vijay Kumar	22
04	Ajay Kumar	14

Dept No	**Place**
12	Jaipur
13	Mumbai
14	Delhi

Explanation:

In the above, **Dept No** of the first relation is the foreign key, and **Dept No** in the second relation is the primary key. **Dept No** = 22 in the foreign key of the first table is not allowed since **Dept No** = 22 is not defined in the primary key of the second relation. Therefore Referential integrity constraints is violated here

Relational Algebra in DBMS

RELATIONAL ALGEBRA is a widely used **procedural query language**. It gives a step by step process to obtain the result of the query. It uses **operators** to perform queries. It collects instances of relations as input and gives occurrences of relations as output.

SQL Relational algebra query operations are performed recursively on a relation. The output of these operations is a **new relation**, which might be formed from one or more input relations.

Basic SQL Relational Algebra Operations:

Relational Algebra divided in various groups:

Unary Relational Operations

- SELECT (symbol: σ)
- PROJECT (symbol: π)
- RENAME (symbol: ρ)

Relational Algebra Operations from Set Theory

- UNION (υ)
- INTERSECTION ( ),
- DIFFERENCE (-)
- CARTESIAN PRODUCT (x)

Binary Relational Operations

- JOIN

<u>Unary Relational Operations:</u>

1. **SELECT (σ):**

   The **SELECT** operation is used for selecting a subset of the tuples according to a given selection condition. **Sigma (σ)** Symbol denotes it. It is used as an expression to choose **tuples** which meet the selection condition. Select operator selects tuples that satisfy a given predicate.

   Syntax: σp(r)

   σ is the predicate

   r stands for relation which is the name of the table

   p is prepositional logic formula which may use connectors like: AND OR and NOT. These relational can use as relational operators like =, ≠, ≥, <, >, ≤.

Example 1:

$$\sigma \text{ topic = "Database" (Tutorials)}$$

Output - Selects tuples from **Tutorials** where **topic = 'Database'**.

Example 2:

$$\sigma \text{ topic = "Database" and author = "ABC"(Tutorials)}$$

Output - Selects tuples from Tutorials where the topic is 'Database' and 'author' is ABC.

For example: LOAN Relation

BRANCH_NAME	LOAN_NO	AMOUNT
Downtown	L-17	1000
Redwood	L-23	2000
Perryride	L-15	1500
Downtown	L-14	1500
Mianus	L-13	500
Roundhill	L-11	900
Perryride	L-16	1300

Input:

$$\sigma \text{ BRANCH_NAME="perryride" (LOAN)}$$

Output:

BRANCH_NAME	LOAN_NO	AMOUNT
Perryride	L-15	1500
Perryride	L-16	1300

2. **Projection(π):**

The projection eliminates all attributes of the input relation except those mentioned in the projection list. This operation shows the list of those attributes that we wish to appear in the result. Rest of the attributes are eliminated from the table. The projection method defines a relation that contains a vertical subset of Relation.

∏ (pi) symbol is used to choose attributes from a relation. This operator helps you to keep specific columns from a relation and discards the other columns.

Example of Projection:

Consider the following table

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

Here, the projection of **CustomerName** and status will give

Π CustomerName, Status (Customers)

CustomerName	Status
Google	Active
Amazon	Active
Apple	Inactive
Alibaba	Active

3. Rename (ρ):

The rename operation is used to rename the output relation. It is denoted by rho (ρ).

Example: We can use the rename operator to rename **STUDENT** relation to **STUDENT1**.

ρ (STUDENT1, STUDENT)

Relational Algebra Operations from Set Theory

1. **UNION (υ):**

Suppose there are two tuples R and S. The union operation contains all the tuples that are either in R or S or both in R & S.

It eliminates the duplicate tuples. It is denoted by ∪**.**

Notation: **R ∪ S**

A union operation must hold the following condition:

- R and S must have the attribute of the same number.
- Duplicate tuples are eliminated automatically.

Example:

DEPOSITOR RELATION

CUSTOMER_NAME	ACCOUNT_NO
Johnson	A-101
Smith	A-121
Mayes	A-321
Turner	A-176
Johnson	A-273
Jones	A-472
Lindsay	A-284

BORROW RELATION

CUSTOMER_NAME	LOAN_NO
Jones	L-17
Smith	L-23
Hayes	L-15
Jackson	L-14
Curry	L-93
Smith	L-11
Williams	L-17

Input:

$$\prod \text{CUSTOMER_NAME (BORROW)} \cup \prod \text{CUSTOMER_NAME (DEPOSITOR)}$$

Output:

CUSTOMER_NAME
Johnson
Smith
Hayes
Turner
Jones
Lindsay
Jackson
Curry
Williams
Mayes

Example 2

Consider the following tables.

Table A		Table B	
column 1	column 2	column 1	column 2
1	1	1	1
1	2	1	3

Input:

A ∪ B gives

Output:

Table A ∪ B	
column 1	column 2
1	1
1	2
1	3

2. **INTERSECTION ( )**

Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in both R & S.

It is denoted by intersection ∩.

Notation: **R ∩ S**

Example: Using the above DEPOSITOR table and BORROW table

DEPOSITOR RELATION:

CUSTOMER_NAME	ACCOUNT_NO
Johnson	A-101
Smith	A-121
Mayes	A-321
Turner	A-176
Johnson	A-273
Jones	A-472
Lindsay	A-284

BORROW RELATION:

CUSTOMER_NAME	LOAN_NO
Jones	L-17
Smith	L-23
Hayes	L-15
Jackson	L-14
Curry	L-93
Smith	L-11
Williams	L-17

Input:

∏ CUSTOMER_NAME (BORROW) ∩ ∏ CUSTOMER_NAME (DEPOSITOR)

Output:

CUSTOMER_NAME
Smith
Jones

3. **DIFFERENCE (-):**

Suppose there are two tuples **R** and S**. The set difference operation contains all tuples that are in R but not in S.**

It is denoted by **intersection minus (-).**

Notation: **R - S**

Example: Using the above **DEPOSITOR** table and **BORROW** Relation

Input:

∏ CUSTOMER_NAME (BORROW) - ∏ CUSTOMER_NAME (DEPOSITOR)

Output:

CUSTOMER_NAME
Jackson
Hayes
Willians
Curry

4. **CARTESIAN PRODUCT (x):**

The Cartesian product is used to combine each row in one table with each row in the other table. It is also known as a **cross product**.

It is denoted by **X**.

Notation: **E X D**

Example:

EMPLOYEE Relation

EMP_ID	EMP_NAME	EMP_DEPT
1	Smith	A
2	Harry	C
3	John	B

DEPARTMENT Relation

DEPT_NO	DEPT_NAME
A	Marketing
B	Sales
C	Legal

Input:

EMPLOYEE X DEPARTMENT

Output:

EMP_ID	EMP_NAME	EMP_DEPT	DEPT_NO	DEPT_NAME
1	Smith	A	A	Marketing
1	Smith	A	B	Sales
1	Smith	A	C	Legal
2	Harry	C	A	Marketing
2	Harry	C	B	Sales
2	Harry	C	C	Legal
3	John	B	A	Marketing
3	John	B	B	Sales
3	John	B	C	Legal

Join Operations

A Join operation **combines related tuples** from different relations, if and only if a given **join condition** is satisfied. Join operation is essentially a **Cartesian product followed by a selection criterion**.

JOIN operation also allows joining variously related tuples from different relations.

It is denoted by ⋈.

Types of Join operations:

Various forms of join operation are:

Inner Joins:

- Theta join
- EQUI join
- Natural join

Outer join:

- Left Outer Join
- Right Outer Join
- Full Outer Join

Inner Join:

In an inner join, **only those tuples that satisfy the matching criteria are included**, while the rest are excluded. Let's study various types of Inner Joins:

1. **Theta Join**:
   THETA JOIN allows you to merge two tables based on the condition represented by theta. Theta joins work for all comparison operators.

   It is denoted by **symbol θ.**

   The general case of JOIN operation is called a Theta join.

   Notation: $A \bowtie_\theta B$

   Theta join can use any conditions in the selection criteria.
   Consider the following tables.

Student

SID	Name	Std
101	Alex	10
102	Maria	11

Subjects

Class	Subject
10	Math
10	English
11	Music
11	Sports

For example: Student_Detail −

STUDENT ⋈Student.Std = Subject.Class SUBJECT

Student_detail

SID	Name	Std	Class	Subject
101	Alex	10	10	Math
101	Alex	10	10	English
102	Maria	11	11	Music
102	Maria	11	11	Sports

2. **EQUI Join:**

EQUI JOIN is done when a Theta join uses only the **equivalence** condition.
Consider the following tables.

Student

SID	Name	Std
101	Alex	10
102	Maria	11

Subjects

Class	Subject
10	Math
10	English
11	Music
11	Sports

For example: Student_Detail −

STUDENT ⋈Student.Std = Subject.Class SUBJECT

Student_detail

SID	Name	Std	Class	Subject
101	Alex	10	10	Math
101	Alex	10	10	English
102	Maria	11	11	Music
102	Maria	11	11	Sports

3. **Natural Join**:

NATURAL JOIN does not utilize any of the comparison operators. In this type of join, the attributes should have the same name and domain. In Natural Join, there should be at least one common attribute between two relations.

It performs selection forming equality on those attributes which appear in both relations and eliminates the duplicate attributes.

Example:

Consider the following two tables

Relation C

Num	Square
2	4
3	9

Relation D

Num	Cube
2	8
3	27

Input:

$$C \bowtie D$$

Output:

Num	Square	Cube
2	4	8
3	9	27

Outer Join:

The outer join operation is an extension of the join operation. It is used to deal with missing information.

Theta Join, Equijoin, and Natural Join are called inner joins. An inner join includes only those tuples with matching attributes and the rest are discarded in the resulting relation.

Therefore, we need to use outer joins to include all the tuples from the participating relations in the resulting relation. There are three kinds of outer joins − left outer join, right outer join, and full outer join.

An outer join is basically of three types:

1. Left outer join
2. Right outer join
3. Full outer join

a) Left outer join: $R \supset\!\bowtie S$

In the left outer join, operation allows keeping all tuple in the left relation. However, if there is no matching tuple is found in right relation, then the attributes of right relation in the join result are filled with null values.



Consider the following two tables:

Relation **A**

Num	Square
2	4
3	9
4	16

Relation B

Num	Cube
2	8
3	18
5	75

Input:

$A \bowtie B$

Output:

Num	Square	Cube
2	4	8
3	9	18
4	16	-

b) Right Outer Join: (A ⟖ B):

In the right outer join, operation allows keeping all tuple in the right relation. However, if there is no matching tuple is found in the left relation, then the attributes of the left relation in the join result are filled with null values.

Notation: $A \bowtie B$



Consider the above two tables **A** and **B**:

Input:

$$A \bowtie B$$

Output:

Num	Cube	Square
2	8	4
3	18	9
5	75	-

c) Full Outer Join: (A ⋈ B):

In a full outer join, all tuples from both relations are included in the result, irrespective of the matching condition.

Notation: $\mathbf{A \bowtie B}$

Consider the above two tables A and B:

Input:

$$\mathbf{A \bowtie B}$$

Relation A

Num	**Square**
2	4
3	9
4	16

Relation B

Num	**Cube**
2	8
3	18
5	75

Output:

Num	**Square**	**Cube**
2	4	8
3	9	18
4	16	-
5	-	75

Relational Database Design

Introduction

Relational database was proposed by **Edgar Codd** (of **IBM Research**) around 1969. It has since become the dominant database model for commercial applications (in comparison with other database models such as hierarchical, network and object models). Today, there are many commercial Relational Database Management System (**RDBMS**), such as **Oracle**, **IBM DB2** and **Microsoft SQL Server**. There are also many free and open-source RDBMS, such as **MySQL, mSQL (mini-SQL)** and the **embedded JavaDB** (Apache Derby).

A relational database organizes **data in tables (or relations)**. A table is made up of **rows** and **columns**. A row is also called a **record** (or **tuple**). A column is also called a **field** (or **attribute**). A database table is similar to a spread sheet. However, the relationships that can be created among the tables enable a relational database to efficiently store huge amount of data, and effectively retrieve selected data.

Example: A table of students engaged in sports activities, where a student is allowed to participate in at most one activity.

StudentID	Activity	Fee
100	Skiing	200
150	Swimming	50
175	Squash	50
200	Swimming	50

A language called **SQL** (Structured Query Language) was developed to work with relational databases.

Database Design Objective

A well-designed database shall:

- **Eliminate Data Redundancy**: the same piece of data shall not be stored in more than one place. This is because duplicate data not only waste storage spaces but also easily lead to inconsistencies.
- **Ensure Data Integrity and Accuracy**:

Relational Database Design Process

Database design is more **art** than science, as you have to make many decisions. Databases are usually customized to suit a particular application. No two customized applications are alike, and hence, no two databases are alike. Guidelines (usually in terms of what not to do instead of what to do) are provided in making these design decision, but the choices ultimately rest on the you - the designer.

Step 1: Define the Purpose of the Database (Requirement Analysis)

Gather the requirements and define the objective of your database, e.g. ...

Drafting out the sample input forms, queries and reports, often helps.

Step 2: Gather Data, Organize in tables and Specify the Primary Keys

Once you have decided on the purpose of the database, gather the data that are needed to be stored in the database. Divide the data into subject-based tables.

Choose one column (or a few columns) as the so-called **primary key**, which **uniquely identify** the each of the rows.

Step 3: Create Relationships among Tables

A database consisting of independent and unrelated tables serves little purpose (you may consider to use a spread sheet instead). The power of relational database lies in the relationship that can be defined between tables. The most crucial aspect in designing a relational database is to identify the relationships among tables. The relationships between the various tables are established with the help of **foreign keys**. Foreign keys are attributes occurring in a table that are primary keys of another table.

The types of relationship include:

- One-to-One
- One-to-Many
- Many-to-One
- Many-to-Many

Step 4: Refine & Normalize the Design

This is the process of optimizing the database structure. Normalization simplifies the database design to avoid **redundancy** and **confusion**.

For example,

- adding more columns,
- create a new table for optional data using one-to-one relationship,
- split a large table into two smaller tables,
- Others.

The different normal forms are as follows:

- First normal form
- Second normal form
- Third normal form
- Boyce-Codd normal form
- Fifth normal form

By applying a set of rules, a table is normalized into the above normal forms in a linearly progressive fashion. The efficiency of the design gets better with each higher degree of normalization.

Primary Keys:

- A primary key is an attribute or a collection of attributes whose value(s) uniquely identify each row in a relation.
- A primary key should be minimal: it should not contain unnecessary attributes.

StudentID	Activity	Fee
100	Skiing	200
150	Swimming	50
175	Squash	50
200	Swimming	50

- We assume that a student is allowed to participate in at most one activity.
- The only possible primary key in the above table is **StudentID**.
- Sometimes there is more than one possible choice; each possible choice is called a candidate key.
- What if we allow the students to participate in more than one activity?

StudentID	Activity	Fee
100	Skiing	200
100	Golf	65
175	Squash	50
175	Swimming	50
200	Swimming	50
200	Golf	65

- Now the only possible primary key is the combined value of (StudentID, Activity),
- Such a multi-attribute primary key is called a **composite** key or **concatenated** key.

Composite Keys:

- A table can only have one primary key.
- But sometimes the primary key can be made up of several fields.
- Concatenation means putting two things next to one another.

Consider the following table of cars:

LicensePlate	State	Make	Model	Year
LVR120	NJ	Honda	Accord	2003
BCX50P	NJ	Buick	Regal	1998
LVR120	CT	Toyota	Corolla	2002
908HYY	MA	Ford	Windstar	2001
UHP33X	NJ	Nissan	Altima	2006

- **LicensePlate** is not a possible primary key, because two different cars can have the same license plate number if they're from different states.
- But if we concatenate **LicensePlate** and **State**, the resulting value of (**LicensePlate**, **State**) must be unique:

Example: "LVR120NJ" and "LVR120CT"

- Therefore, (**LicensePlate**, **State**) is a possible primary key (a candidate key).

Foreign Keys:

- A foreign key is an attribute or a collection of attributes whose value are intended to match the primary key of some related record (usually in a different table).

  Example: the **STATE** and **CITY** table below:

STATE table:

State Abbrev	StateName	Union Order	StateBird	State Population
CT	Connecticut	5	American robin	3,287,116
MI	Michigan	26	robin	9,295,297
SD	South Dakota	40	pheasant	696,004
TN	Tennessee	16	mocking bird	4,877,185
TX	Texas	28	mocking bird	16,986,510

CITY table:

State Abbrev	CityName	City Population
CT	Hartford	139,739
CT	Madison	14,031
CT	Portland	8,418
MI	Lansing	127,321
SD	Madison	6,257
SD	Pierre	12,906
TN	Nashville	488,374
TX	Austin	465,622
TX	Portland	12,224

- Primary key in **STATE** relation: **StateAbbrev**
- Primary key in **CITY** relation: (**StateAbbrev, CityName**)
- Foreign key in **CITY** relation: **StateAbbrev**

Dependencies in DBMS:

Dependencies in DBMS is a relation between two or more attributes. It has the following types in DBMS:

1. Functional Dependency
2. Partial Dependency
3. Transitive Dependency

Functional Dependency:

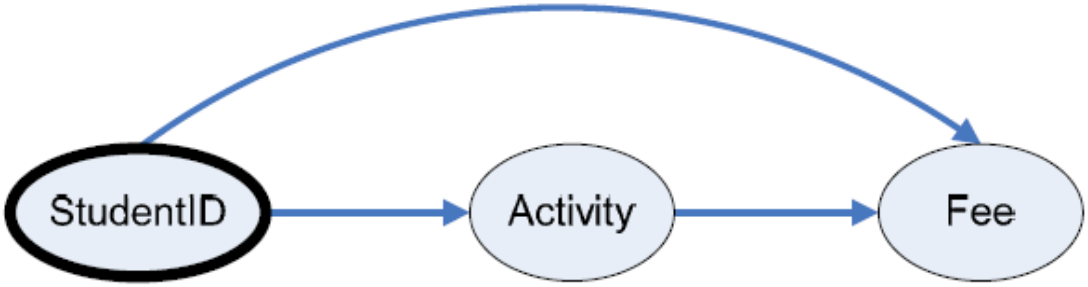
- Attribute B is functionally dependent on attribute A if given a value of attribute A, there is only one possible corresponding value of attribute B.
- That is, any two rows with the same value of A must have the same value for B.
- Attribute A is the determinant of attribute B if attribute B is functionally dependent on attribute A.
    - In the **STATE** relation above, **StateAbbrev** is a determinant of all other attributes.
    - In the STATE relation, the attribute **StateName** is also a determinant of all other attributes
    - So, **StateAbbrev** and **StateName** are both candidate keys for STATE.
    - In the **CITY** relation above, the attributes (**StateAbbrev**, **CityName**) together are a determinant of the attribute **CityPopulation**.
    - In the **CITY** relation, the attribute **CityName** is not a determinant of the attribute **CityPopulation** because multiple cities in the table may have the same name.

Dependency Diagrams:

- A dependency diagram or bubble diagram is a pictorial representation of functional dependencies.
    - an attribute is represented by an oval
    - You draw an arrow from A to B when attribute A is a determinant of attribute B.

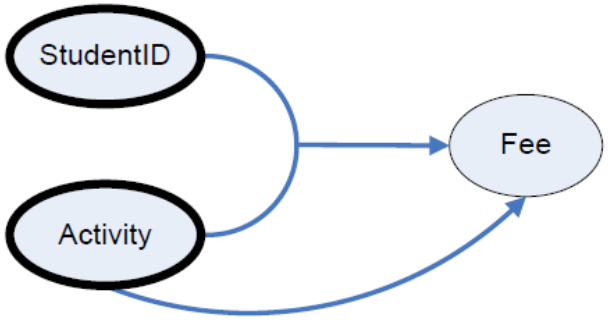
Example: When students were only allowed one sports activity, we have

ACTIVITY (StudentID, Activity, Fee)



Example: when students can have multiple activities, we have

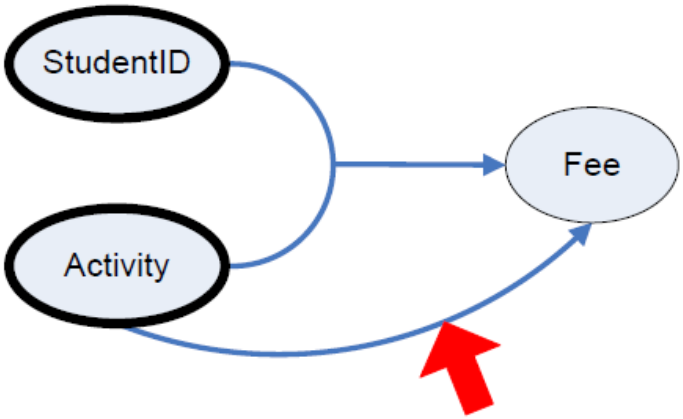
ACTIVITY (StudentID**,** Activity, Fee)



Partial Dependencies:

- A partial dependency is a functional dependency whose determinant is part of the primary key (but not all of it).

Example: **ACTIVITY** (StudentID, Activity, Fee)

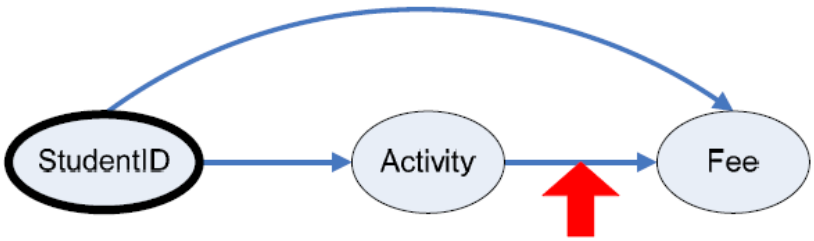


In above example, Primary keys are (**StudentID, Activity**) and non-prime attribute **Fee** is dependent on **Activity**, which is part of the primary key (but not all of it). Hence, partial dependency is there.

Transitive Dependencies:

- A transitive dependency is a functional dependency whose determinant is not the primary key, part of the primary key, or a candidate key.

Example: **ACTIVITY** (StudentID, Activity, Fee)



In above example, Primary key is **StudentID** and non-prime attribute are Activity and Fee. Fee is dependent on Activity, which is not the primary key, part of the primary key, or a candidate key. Hence, transitive dependency is there.

Anomalies of un-normalized database

Anomalies are problems caused by bad database design. Anomalies are caused when there is too much redundancy in the database's information. Anomalies can often be caused when the tables that make up the database suffer from poor construction. So, what does "poor construction" mean? Poor table design will become evident if, when the designer creates the database, he doesn't identify the entities that depend on each other for existence, like the rooms of a hotel and the hotel, and then minimize the chance that one would ever exist independent of the other.

The normalization process was created largely in order to reduce the negative effects of creating tables that will introduce anomalies into the database.

There are three types of Data Anomalies:

1. Insertion Anomalies ,
2. Update Anomalies,
3. Deletion Anomalies.

Example: Consider following relation:

StudentID	Activity	Fee
100	Skiing	200
100	Golf	65
175	Squash	50
175	Swimming	50
200	Swimming	50
200	Golf	65

ACTIVITY (StudentID, Activity, Fee)

Insertion Anomalies:

An insertion anomaly occurs when a row cannot be added to a relation, because not all data are available (or one has to invent “dummy” data).

Example: we want to store that **scuba diving** costs $175, but have no place to put this information until a student takes up scuba-diving (unless we create a fake student).

Update Anomalies:

An update anomaly occurs when one must make many changes to reflect the modification of a single datum.

Example: If the cost of swimming changes, then all entries with swimming Activity must be changed too.

Deletion Anomalies:

A deletion anomaly occurs when data is deleted from a relation, and other critical data are unintentionally lost.

Example: If we delete the record with StudentID = 100, we forget that skiing costs $200.

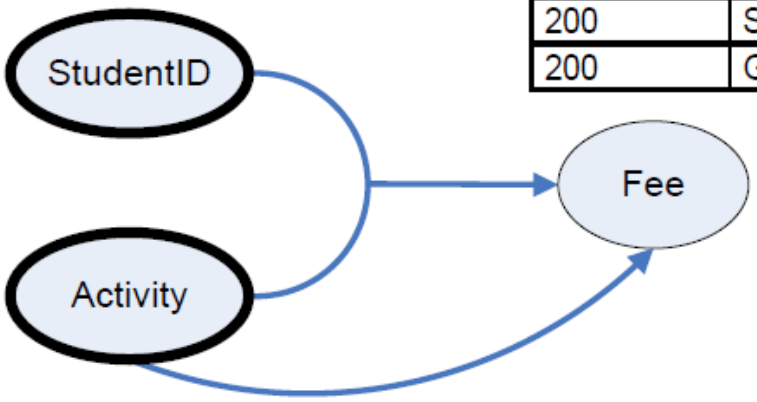
Cause of Anomalies:

Anomalies are primarily caused by:

1. Data redundancy: replication of the same field in multiple tables, other than foreign keys.
2. Functional dependencies whose determinants are not candidate keys, including partial dependency, transitive dependency.

In the above example **ACTIVITY** (StudentID, Activity, Fee):

StudentID	Activity	Fee
100	Skiing	200
100	Golf	65
175	Squash	50
175	Swimming	50
200	Swimming	50
200	Golf	65



Activity by itself is **not a candidate key**, so we get anomalies (in this case, from a **partial dependency**).

Fixing Anomalies (Normalizing):

Break up tables so all dependencies are from primary (or candidate) keys.

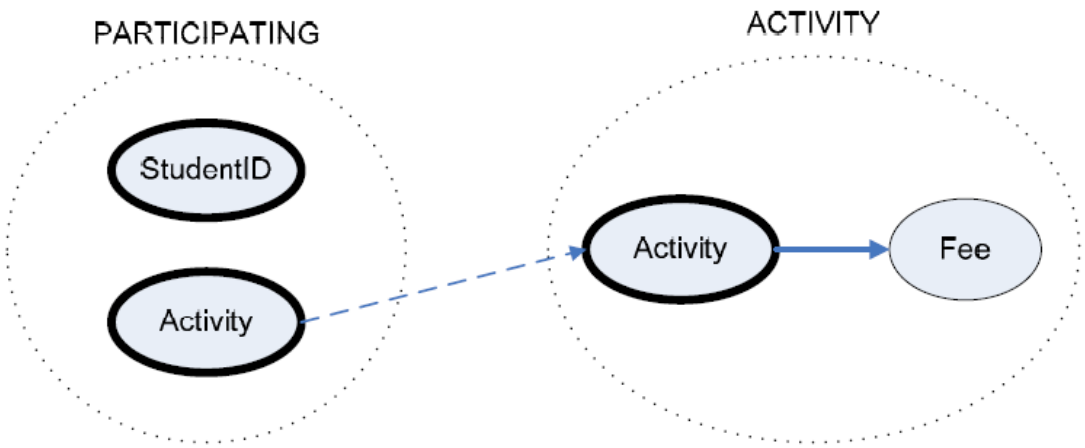
PARTICIPATING (StudentID, Activity)

StudentID	Activity
100	Skiing
100	Golf
150	Swimming
175	Squash
175	Swimming
200	Swimming
200	Golf

ACTIVITY (Activity, Fee)

Activity	Fees
Skiing	200
Golf	65
Swimming	50
Squash	50
ScubaDiving	200

Activity foreign key to **ACTIVITIES**



The above relations do not have any of the anomalies:

•we can add the cost of diving in **ACTIVITIES** even though no one has taken it in **STUDENTS.**

• If **StudentID 100** drops **Skiing**, no skiing-related data will be lost.

• If the **cost** of **swimming** changes, that cost need only be changed in one place only (the **ACTIVITIES** table).

• The **Activity** field is in both tables, but that's needed to relate ("**join**") the information in the two tables.

Good Database Design Principles:

1. **No redundancy**:
    - A field is stored in only one table, unless it happens to be a foreign key.
    - Replication of foreign keys is permissible, because they allow two tables to be joined together.
2. **No "bad" dependencies**:

- In the dependency diagram of any relation in the database, the determinant should be the whole primary key, or a candidate key. Violations of this rule include:
    - partial dependencies
    - transitive dependencies

Normalization is the process of eliminating "bad" dependencies by splitting up tables and linking them with foreign keys.

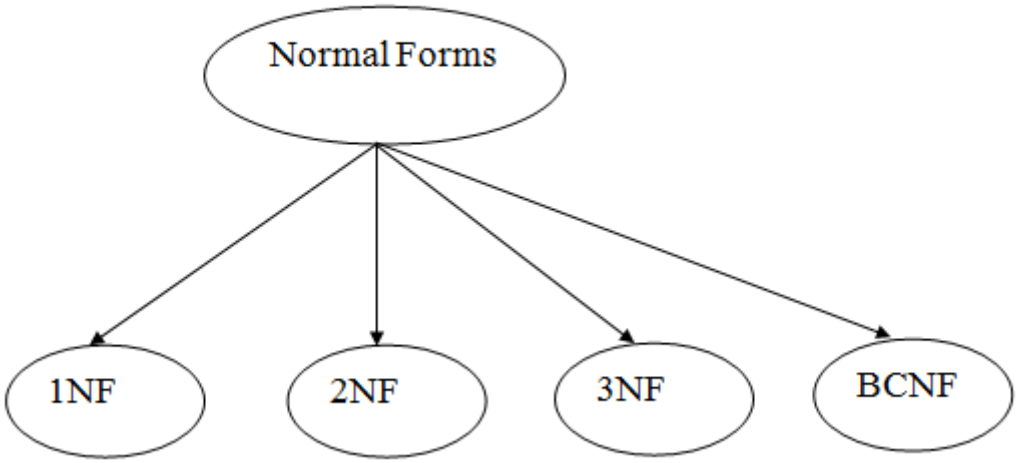
- "**Normal forms**" are categories that classify how completely a table has been normalized.
- There are six recognized normal forms (NF):
    1. First Normal Form (1NF)
    2. Second Normal Form (2NF)
    3. Third Normal Form (3NF)
    4. Boyce Codd Normal Form (BCNF)
    5. Fourth Normal Form (4NF)
    6. Fifth Normal Form (5NF)

Normalization in DBMS

Introduction:

Normalization is the process of organizing the data in the database. Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

Types of Normal Forms:



Normal Form	Description
1NF	A relation is in 1NF if it contains an atomic value.
2NF	A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.
3NF	A relation will be in 3NF if it is in 2NF and no transition dependency exists.
4NF	A relation will be in 4NF if it is in Boyce Codd normal form and has no multi-valued dependency.
5NF	A relation is in 5NF if it is in 4NF and not contains any join dependency and joining should be lossless.

First Normal Form (1NF):

- A relation will be 1NF if it contains an **atomic** value.
- It states that an **attribute** of a table **cannot hold multiple values**. It must hold only **single-valued attribute**.
- First normal form disallows the multi-valued attribute, composite attribute, and their combinations.

Example: Relation **EMPLOYEE** is not in **1NF** because of multi-valued attribute **EMP_PHONE**.

EMPLOYEE table:

EMP_ID	EMP_NAME	EMP_PHONE	EMP_STATE
14	John	7272826385, 9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389, 8589830302	Punjab

The decomposition of the **EMPLOYEE** table into **1NF** has been shown below:

EMP_ID	EMP_NAME	EMP_PHONE	EMP_STATE
14	John	7272826385	UP
14	John	9064738238	UP
20	Harry	8574783832	Bihar
12	Sam	7390372389	Punjab
12	Sam	8589830302	Punjab

Second Normal Form (2NF)

- In the **2NF**, relational must be in **1NF**.
- In the second normal form, all **non-key attributes** are **fully functional dependent on the primary key**.

Example: Let's assume, a **school** can store the data of **teachers** and the **subjects** they teach. In a school, a teacher can teach more than one subject.

TEACHER table

TEACHER_ID	SUBJECT	TEACHER_AGE
25	Chemistry	30
25	Biology	30
47	English	35
83	Math	38
83	Computer	38

In the given table, non-prime attribute **TEACHER_AGE** is dependent on **TEACHER_ID** which is a proper **subset of a candidate key**. That's why it violates the rule for 2NF.

To convert the given table into **2NF**, we decompose it into two tables:

TEACHER_DETAIL table:

TEACHER_ID	TEACHER_AGE
25	30
47	35
83	38

TEACHER_SUBJECT table:

TEACHER_ID	SUBJECT
25	Chemistry
25	Biology
47	English
83	Math
83	Computer

Now the tables comply with Second normal form (**2NF**).

Third Normal Form (3NF)

- A relation will be in 3NF if it is in **2NF** and not contain any **transitive partial dependency**.
- **3NF** is used to reduce the data duplication. It is also used to achieve the data integrity.
- If there is **no transitive dependency** for **non-prime attributes**, then the relation must be in third normal form.

In other words **3NF** can be explained like this: A table is in 3NF if it is in **2NF** and for each functional dependency **X-> Y** at least one of the following conditions hold:

- **X** is a super key of table
- **Y** is a prime attribute of table

Example:

EMPLOYEE_DETAIL table:

EMP_ID	EMP_NAME	EMP_ZIP	EMP_STATE	EMP_CITY
222	Harry	201010	UP	Noida
333	Stephan	02228	US	Boston
444	Lan	60007	US	Chicago
555	Katharine	06389	UK	Norwich
666	John	462007	MP	Bhopal

Super key in the table above:

{EMP_ID}, {EMP_ID, EMP_NAME}, {EMP_ID, EMP_NAME, EMP_ZIP}....so on

Candidate key: {EMP_ID}

Non-prime attributes: In the given table, all attributes except **EMP_ID** are non-prime.

Here, **EMP_STATE** & **EMP_CITY** dependent on **EMP_ZIP** and **EMP_ZIP** dependent on **EMP_ID**. The non-prime attributes (**EMP_STATE, EMP_CITY**) transitively dependent on super key (**EMP_ID**). It violates the rule of third normal form.

That's why we need to move the **EMP_CITY** and **EMP_STATE** to the new <**EMPLOYEE_ZIP**> table, with **EMP_ZIP** as a **Primary key**.

EMPLOYEE table:

EMP_ID	EMP_NAME	EMP_ZIP
222	Harry	201010
333	Stephan	02228
444	Lan	60007
555	Katharine	06389
666	John	462007

EMPLOYEE_ZIP table:

EMP_ZIP	EMP_STATE	EMP_CITY
201010	UP	Noida
02228	US	Boston
60007	US	Chicago
06389	UK	Norwich
462007	MP	Bhopal

Boyce Codd normal form (BCNF)

- BCNF is the advance version of 3NF. It is stricter than 3NF.
- A table is in BCNF if every **functional dependency X → Y**, **X** is the **super key** of the table.
- For BCNF, the table should be in 3NF, and for every **FD**, **LHS** is **super key**.

Example: Let's assume there is a company where **employees work** in **more than one department**.

EMPLOYEE table:

EMP_ID	EMP_COUNTRY	EMP_DEPT	DEPT_TYPE	EMP_DEPT_NO
264	India	Designing	D394	283
264	India	Testing	D394	300
364	UK	Stores	D283	232
364	UK	Developing	D283	549

In the above table **Functional dependencies** are as follows:

EMP_ID → EMP_COUNTRY

EMP_DEPT → {DEPT_TYPE, EMP_DEPT_NO}

Candidate key: {EMP-ID, EMP-DEPT}

The table is **not in BCNF** because neither **EMP_DEPT** nor **EMP_ID** alone are keys.

To convert the given table into **BCNF**, we decompose it into three tables:

EMP_COUNTRY table:

EMP_ID	EMP_COUNTRY
264	India
264	India

EMP_DEPT table:

EMP_DEPT	DEPT_TYPE	EMP_DEPT_NO
Designing	D394	283
Testing	D394	300
Stores	D283	232
Developing	D283	549

EMP_DEPT_MAPPING table:

EMP_ID	EMP_DEPT
D394	283
D394	300
D283	232
D283	549

Functional dependencies:

EMP_ID → EMP_COUNTRY

EMP_DEPT → {DEPT_TYPE, EMP_DEPT_NO}

Candidate keys:

For the first table: EMP_ID

For the second table: EMP_DEPT

For the third table: {EMP_ID, EMP_DEPT}

Now, this is in **BCNF** because left side part of both the functional dependencies is a **key**.

Name of Course	**Bachelor of Computer Application (BCA)**
Semester	**III**
Name of Subject	**Elective - Introduction to Multimedia**
Subject Code	**BCA-304 B**
Marks	**75 Marks**
Lectures	**50 Lectures**

Introduction to Multimedia Objectives

Multimedia is content that uses a combination of different content forms such as text, audio, images, animations, video and interactive content. Multimedia contrasts with media that use only rudimentary computer displays such as text-only or traditional forms of printed or hand-produced material.

Introduction to Multimedia Outcomes

Student will learn the different content forms of Multimedia such as text, audio, images, animations, video and interactive content.

UNIT – I

1. Introduction to Multimedia **15 Lectures**

1.1 Introduction to Multimedia

1.2 Definition of Multimedia elements

1.3 Multimedia Elements

1.4 Multimedia Applications

1.5 Global structure of Multimedia

1.6 Data Compression

1.7 Basic compression techniques (Run length & Huffman encoding)

1.8 Introduction to compression techniques: JPEG, MPEG

UNIT - II

2. Optical Storage Media & Audio File Formats **15 Lectures**

2.1 Optical Storage Media & Retrieval Technologies

2.2. Basic Technology

2.3 Video Disk & other WORMS

2.4 CD-ROM and Multimedia Highway

2.5 DVD- ROM

2.6 Basic Concept of Sound

2.7 MIDI

2.8 Digital audio

2.9 Audio file formats

UNIT - III

3. Image and Graphics **10 Lectures**

3.1 Making Still Images: BITMAPS, Vector Drawing

3.2 Image Formats

3.3 Graphics Format

3.4 Image file format: BMP, JPEG, TIFF, PNG

UNIT - IV

4. Video & Animation **10 Lectures**

4.1 Basic concepts of Video

4.2 Broadcast Video Standards

4.3 Television: Conventional systems, Enhanced definition systems, High Definition system

4.4 Computer based Animation

References:

1. Multimedia System Design By By P. K. Andleigh, Kiran Thakrar, Dhanpat Rai Publications
2. Multimedia : Computing Communications & Applications, By Ralf Steinmetz And Klara Nehrstedt, Pearson Education

Elective: Introduction to Multimedia
Unit-1

Introduction to Multimedia

Multimedia is an interactive media and provides multiple ways to represent information to the user in a powerful manner. It provides an interaction between users and digital information. It is a medium of communication. Some of the sectors where multimedia's is used extensively are education, training, reference material, business presentations, advertising and documentaries.

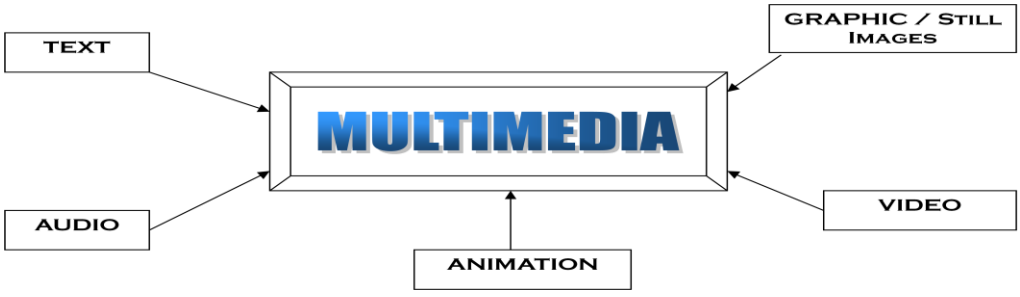
Definition of Multimedia

By definition Multimedia is a representation of information in an attractive and interactive manner with the use of a combination of text, audio, video, graphics and animation. In other words we can say that Multimedia is a computerized method of presenting information combining textual data, audio, visuals (video), graphics and animations. For examples: E-Mail, Yahoo Messenger, Video Conferencing, and Multimedia Message Service (MMS).

Multimedia as name suggests is the combination of Multi and Media that is many types of media (hardware/software) used for communication of information.

Multimedia Elements

Following are the basic Elements of Multimedia:



1. Text :

A broad term for something that contains words to express something. Text is the most basic element of multimedia. A good choice of words could help convey the intended message to the users (keywords).It is used in contents, menus, and navigational buttons.

2. Graphics:

Graphics make the multimedia application attractive. In many cases people do not like reading large amount of textual matter on the screen. Therefore, graphics are used more often than text to explain a concept, present background information etc.

3. Audio:

A multimedia application may require the use of speech, music and sound effects. These are called audio or sound element of multimedia. Speech is also a perfect way for teaching. Audio are of analog and digital types. Analog audio or sound refers to the original sound signal. Computer stores the sound in digital form. Therefore, the sound used in multimedia application is digital audio.

4. **Video**:

The term video refers to the moving picture, accompanied by sound such as a picture in television. Video element of multimedia application gives a lot of information in small duration of time. Digital video is useful in multimedia application for showing real life objects. Video have highest performance demand on the computer memory and on the bandwidth if placed on the internet. Digital video files can be stored like any other files in the computer and the quality of the video can still be maintained. The digital video files can be transferred within a computer network. The digital video clips can be edited easily.

5. **Animation**:

Animation is a process of making a static image look like it is moving. An animation is just a continuous series of still images that are displayed in a sequence. The animation can be used effectively for attracting attention. Animation also makes a presentation light and attractive. Animation is very popular in multimedia application.

Multimedia Applications:

Following are the common areas of applications of multimedia:

1. Business
2. Education
3. Entertainment
4. Home
5. Public Places

Multimedia in Business-

Multimedia can be used in many applications in a business. The multimedia technology along with communication technology has opened the door for information of global wok groups. Today the team members may be working anywhere and can work for various companies. Thus the work place will become global. The multimedia network should support the following facilities:

- Voice Mail
- Electronic Mail
- Multimedia based FAX
- Office Needs
- Employee Training
- Sales and Other types of Group Presentation
- Records Management

Multimedia in Education-

Many computer games with focus on education are now available. Consider an example of an educational game which plays various rhymes for kids. The child can paint the pictures, increase reduce size of various objects etc apart from just playing the rhymes. Several other multimedia packages are available in the market which provide a lot of detailed information and playing capabilities to kids.

Multimedia in Entertainment-

By using multimedia marketing of new products can be greatly enhanced. Multimedia boost communication on an affordable cost opened the way for the marketing and advertising personnel. Presentation that have flying banners, video transitions, animations, and sound effects are some of the elements used in composing a multimedia based advertisement to appeal to the consumer in a way never used before and promote the sale of the products.

Multimedia Services in Home

The advancement of high computing abilities, communication ways and relevant standards has started the beginning of an era where you will be provided with multimedia facilities at home. These services may include:

- Basic Television Services
- Interactive entertainment

- Digital Audio
- Video on demand
- Home shopping
- Financial Transactions
- Interactive multiplayer or single player games
- Digital multimedia libraries
- E-Newspapers, e-magazines

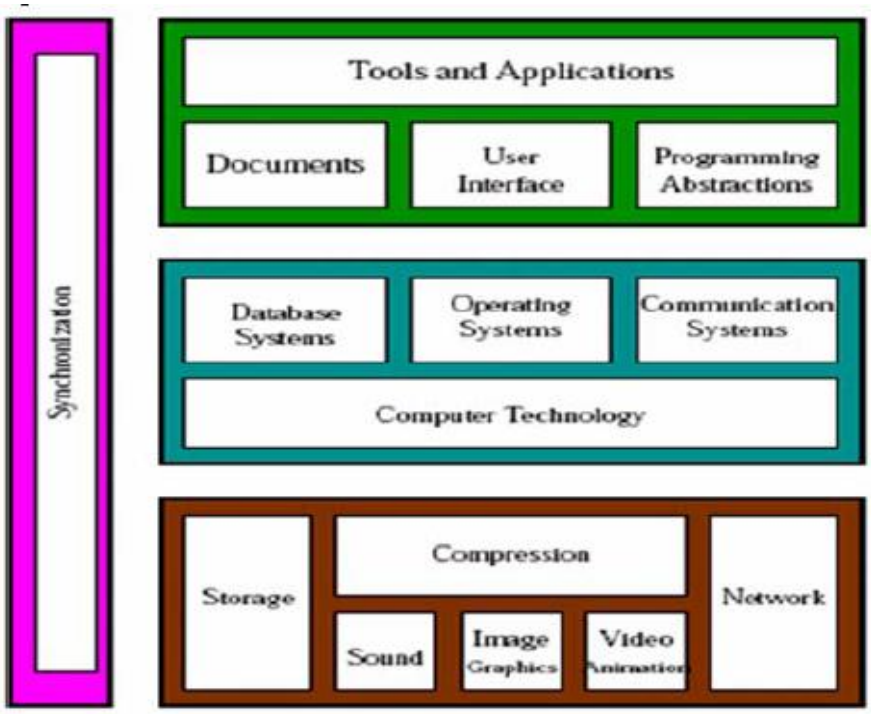
Multimedia in public places:

In the public places have hotels, train station, shopping malls, museums, libraries and grocery stores. We make the multimedia in public places is for the people provide 24 hours a day also can use the multimedia in them life.

Global Structure OF Multimedia:

It is divided into 4 part as follows-

1. Device Domain
2. System Domain
3. Application Domain
4. Cross Domain



1. Device Domain

- Device domain basic concepts and skill for processing various multimedia elements and for handling physical device.
- Basic concepts for the processing of **images, animation, audio** and **video** data are based on digital signal processing.
- The audio techniques section includes music (MIDI) and speech processing.
- The understanding of video techniques is built mainly on TV development, including digital representation and HDTV.
- The current quality requirements and available technology, corresponding compression methods.

2. System Domain

- The interface between the device domain and the system domain is specified by the **computer technology** . To utilize the device domain, several system services are needed
- System domain including all supports for using the functions of the device domain that is operating systems, communication systems (networking) and database systems.

  Database System allows a structured access to data a management of large database.
- **Operating System** interface between computer hardware with system software and all other software components. (network, memory, processor, input/output devices etc)
- **Communication System** data transmission according to the timing and reliability requirements of the networked multimedia application.

3. Application Domain

- The services of the system domain are offered to the application domain through proper **programming abstractions** .
- Application domain provides functions to the user to develop and present multimedia projects. This includes Software tools, and multimedia projects development methodology.
- A document consists of a set of structural information, represented in different media, and generated or recorded at the time of presentation.
- Many functions of **document handling** and other **applications** are accessible and presented to the user through a **user interface** .

4. Cross Domain

- Compositions must allow any type of logical structure besides those ones for synchronization (presentation) purposes.
- Multimedia involves multiple elements of text, audio, images, drawings, animation, and video.
- provides the ability to manually or automatically access or transfer information between two or more different domains.
- Synchronization are difficult to locate in one or two components or domains.
- Synchronization the temporal relationship among various media, relates to many components across all domains.

Data Compression:

The process of reducing the volume of data by applying a compression technique is called compression & the resulting data is called compressed data.

Data compression is used everywhere. Many different file types use compressed data. Without data compression a 3-minute song would be over 100Mb in size, while a 10-minute video would be over 1Gb in size. Data compression shrinks big files into much smaller ones. It does this by getting rid of unnecessary data while retaining the information in the file.

Data compression can be expressed as a decrease in the number of bits required to illustrate data. Compressing data can conserve storage capacity, accelerate file transfer, and minimise costs for hardware storage and network capacity.

Need of Compression:

- ✓ Reduce File Size
- ✓ Save disk space
- ✓ Increase transfer speed at a given data rate.
- ✓ Allow real-time transfer at a given data rate.

Basic Compression technique :

Runlength Coding:

This method replaces the consecutive occurrences of a given symbol with only one copy of the symbol along with a count of how many times that symbol occurs. Hence the names 'run length'.

For example, the string AAABBCDDDD would be encoded as 3A2BIC4D.

A real life example where run-length encoding is quite effective is the fax machine. Most faxes are white sheets with the occasional black text. So, a run-length encoding scheme can take each line and transmit a code for while then the number of pixels, then the code for black and the number of pixels and so on.

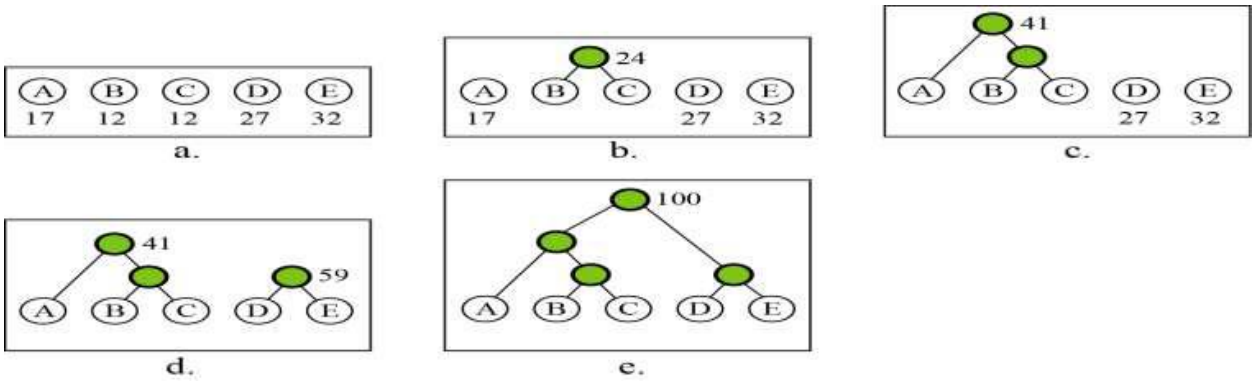
This method of compression must be used carefully. If there is not a lot of repetition in the data then it is possible the run length encoding scheme would actually increase the size of a file.

Huffman Coding:

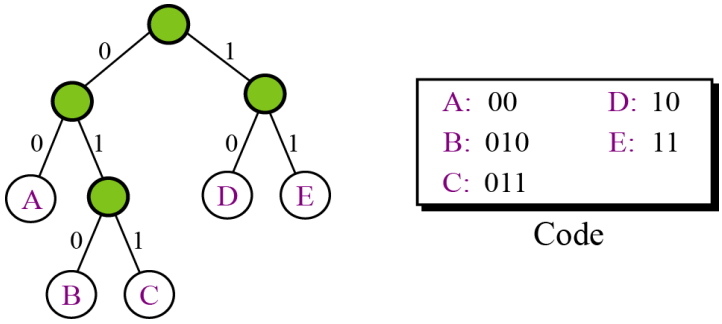
Huffman coding assigns shorter codes to symbols that occur more frequently and longer codes to those that occur less frequently.

For example, imagine we have a text file that uses only five characters (A, B, C, D, E).

Before we can assign bit patterns to each character, we assign each character a weight based on its frequency of use.



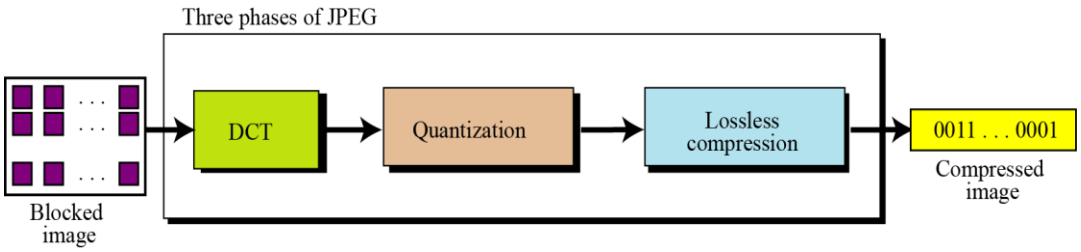
A character's code is found by starting at the root and following the branches that lead to that character. The code itself is the bit value of each branch on the path, taken in sequence.



Introduction to Compression technique :

JPEG Compression:

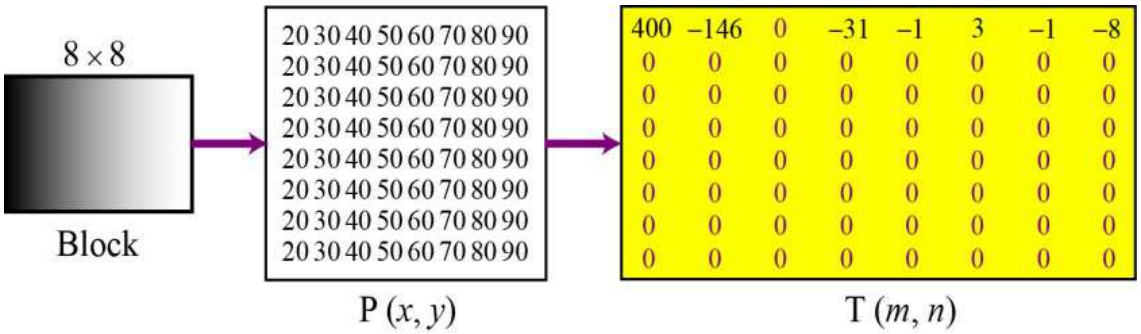
- An image can be represented by a two-dimensional array (table) of picture elements (pixels).
- A grayscale picture of 307,200 pixels is represented by 2,457,600 bits, and a color picture is represented by 7,372,800 bits.
- In JPEG, a grayscale picture is divided into blocks of 8 × 8 pixel blocks to decrease the number of calculations because, as we will see shortly, the number of mathematical operations for each picture is the square of the number of units.
- For Example – Steps:
- Change the picture into a linear (vector) sets of numbers that reveals the redundancies.
- The redundancies are then removed by one of lossless.



DCT: Discrete Concise Transform.

DCT transforms the 64 values in 8x8 pixel block in a way that the relative relationships between pixels are kept but the redundancies are revealed.

Example: A gradient grayscale



Quantization & Compression:

Quantization:

After T table is created, the values are quantized to reduce the number of bits needed for encoding.

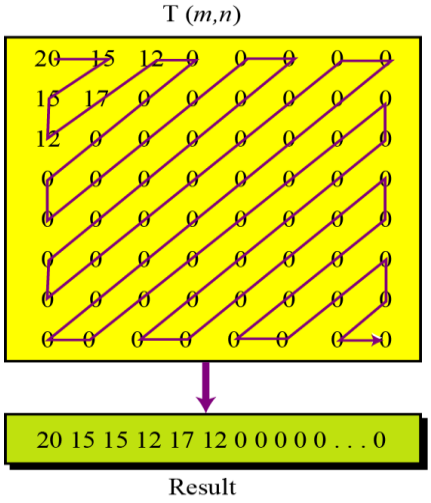
Quantization divides the number of bits by a constant, then drops the fraction. This is done to optimize the number of bits and the number of 0s for each particular application.

Compression:

Quantized values are read from the table and redundant 0s are removed.

To cluster the 0s together, the table is read diagonally in an zigzag fashion. The reason is if the table doesn't have fine changes, the bottom right corner of the table is all 0s.JPEG usually uses lossless run-length encoding at the compression phase

JPEG Encoding:



MPEG Compression:

The Moving Picture Experts Group (MPEG) method is used to compress video. In principle, a motion picture is a rapid sequence of a set of frames in which each frame is a picture.

Used to compress video.

Basic Concept:

Each video is a rapid sequence of a set of frames. Each frame is a spatial combination of pixels, or a picture.

Compressing video = spatially compressing each frame + temporally compressing a set of frames.

Types of MPEG compression:

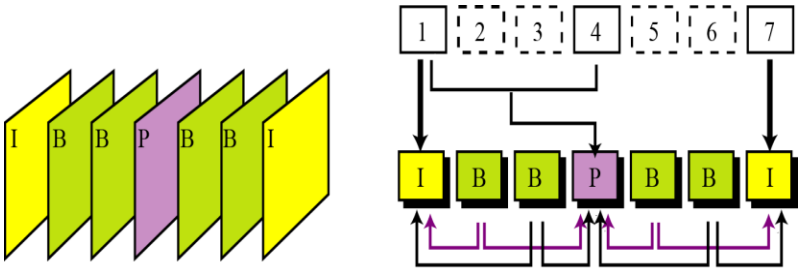
1.Spatial Compression

Each frame is spatially compressed by JPEG.

2.Temporal Compression

Redundant frames are removed.

For example, in a static scene in which someone is talking, most frames are the same except for the segment around the speaker's lips, which changes from one frame to the next.



a. Frames b. Frame construction

Unit-II

Optical Storage Media & Audio File Formats

Optical Storage Media & Retrieval Technologies:

Optical storage devices are flat, round disks that spins around its center. The difference with magnetic storage device is that in optical storage devices LASER light is used to read and write data in disks. Examples are CD, DVD etc.
There are two types of optical disks; one that can be written for one time only and another which can be used to write data more than one time, these are called re-writable disks. Some optical disks can store data on both sides of the disk.
Optical disks are cheaper than magnetic storage devices. But data stability of these disks depends on their use and maintenance. Optical disks do not provide huge storing capacity like magnetic devices. However, now days, Blue-ray disks are coming up with nearly 20 GB of space which is huge. Common DVDs provide nearly 4.7 GB of spaces and others provide below this. These disks require drives to operate. Such as DVD need DVD driver. Again to write on disks the driver needs to be able to write on to disks, Such as to write on writable or re-writable DVD disks will need DVD writer. Optical disks are easy to remove from the drives and can be used of data travelling.
Optical disks are most commonly used for storing and sharing audio, video, games, software etc. though these disks can be used for storing any kind of data.

Optical storage media offer a higher storage density at a lower cost. The Audio Compact Disk the successor to Long Play Disks (LPs) is a commercially successful product in the entertainment industry. The computer industry has profited from this development, especially when audio and video should be stored digitally in the computer. This technology has been the main catalyst for the whole development of multimedia in computing because it is used in multimedia external devices. For example, external devices such as video recorders and DAT recorders (Digital Audio Tape) can be used for multimedia systems.
The video disk in the form of the Video Long Play (VLP) has been available for quite some time and was described in detail by 1973. The read-only video disk has not become a commercial success although there are several Write-Once (WO) optical disks of different sizes and formats available on the market. These initial developments were based on analog techniques with the highest quality requirements at a moderate cost.

Basic Technology:

1. In principle, optical storage media use the intensity of reflected laser light as an information source. A laser beam of approximately 780 nm wave lengths can be focused at approximately 1um.

2. In a polycarbonate substrate layer we encounter holes, corresponding to the coded data, which are called pits.
3. The areas between these pits are called lands. Figure 1.1 shows a cut through an optical disk along a track. In the middle of the figure, the lands and pits are schematically presented.
4. The substrate layer is covered with a thin reflective layer. The laser beam is focused on the reflective layer from the substrate layer. Therefore, the reflected beam has a strong intensity at the lands.
5. The pits have a depth of 0.12 um (from the substrate surface). The laser beam is lightly scattered at the pits, meaning it is reflected with a weak intensity.
6. The signal, shown in Figure 1.1, denotes schematically the intensity of the reflected beam - a horizontal line is drawn as the threshold value. Hence, according to Figure 1.1, a compact disk consists of:

A. The label
B. The protective layer
C. The reflective layer
D. The substrate layer

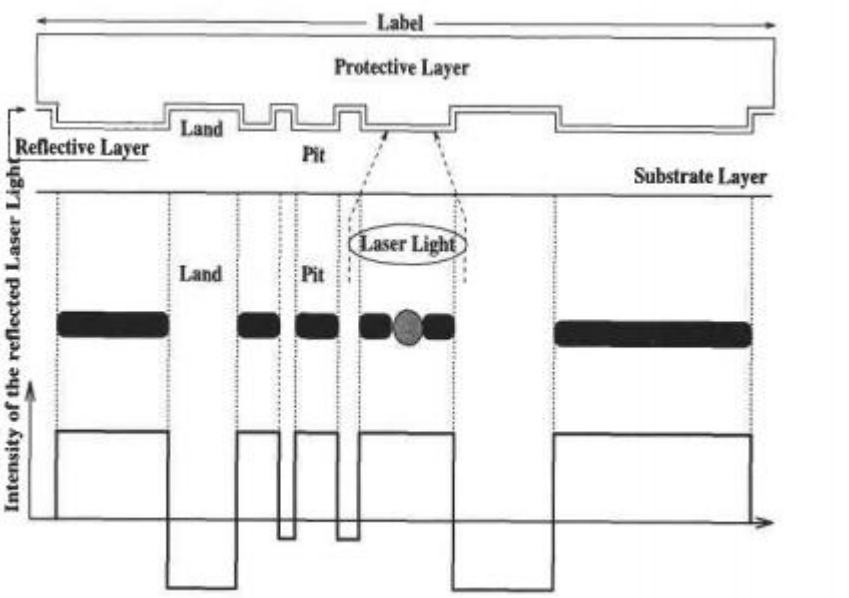


Figure 1.1: Cut through an optical disk along the data trace. A schematic presentation with the layers (above), the "lands" and the "pits" (in the middle), and the signal (below).
An optical disk consists of a sequential order of these pits and lands allocated in one track. Figure 1.2 shows an enlarged cut of such a structure. In contrast to floppy disks and other conventional secondary storage media, the entire optical disk information is stored in one track. Thus, the stored information can be easily played back at a continuous data rate. This has advantages for audio and video data, as they are continuous data streams. The track is a spiral. In the case of a CD, the distance between the tracks is 1.6 um. The

track width of each pit is 0.6 um. The pits themselves have different lengths. Using these measurements, the main advantage of the optical disk in comparison to magnetic disks is that on the former 1.66 data bits per pm car be stored. This results in a data density of 1,000,000 bits per mm^2, which implies 16,000 tracks per inch. In comparison, a floppy disk has 96 tracks per inch.

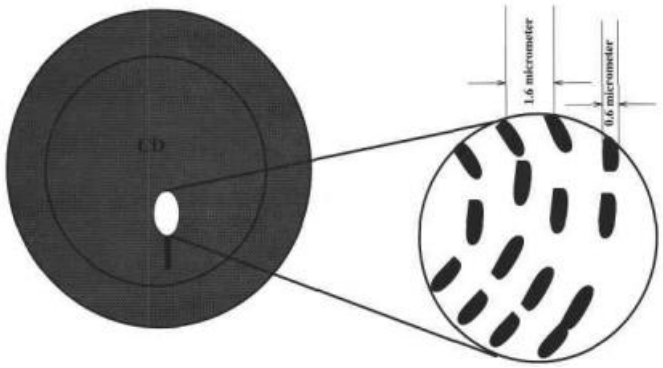


Figure 1.2: Data on a CD as an example of an optical disk (track with "lands' and "pits').

While magnetization can decrease over time and in the case of tapes, for example, cross talk can occur, these effects are unknown in optical disks. Hence, this medium is very good for long-term storage. Only a decomposition or change of the material can cause irreparable damage. According to current knowledge, such effects will not occur.

The light source of the laser can be positioned at a distance of approximately one millimeter from the disk surface, and hence, it does not have to be positioned directly on the disk, respectively near the surface, as is the case with magnetic hard disks. This approach reduces friction and increases the life span of the involved components.

Video Disks and Other WORMS:

The video disk, in the form of Laser Vision, serves as the output of motion pictures and audio. The data are stored in an analog-coded format on the disk; the reproduced data meet the highest quality requirements. The Laser Vision disk has a diameter of approximately 30 cm and stores approximately 2.6 Gigabytes.

Due to the similarities to LP records for audio information, the video disk was originally called the Video Long Play disk. It was described for the first time in Phillips' 1973 Technical Review.

The motion picture on the video disk is encoded as frequency modulation, and the audio signal is mixed with the video signal. Figure 1.1 shows the principle of the recorded data. The main information of the mixed audio-video signal is the time at which the signal has the value zero.

Hence, each zero cross-point corresponds to a change between a pit and a land on the disk. Such a change can occur at any time, and is written to the disk in a non-quantized form, i.e., the pit length is not quantized. Therefore, this method is time-continuous and. analog.

Since the video disk was designed as Read Only Memory (ROM), many different write-once optical storage systems have come out, known as the Write Once Read, Many (WORM) disk. An example is the Interactive Video Disk. This disk is played at a Constant Angular Velocity (CAV). On each side, 36 minutes of audio and video at a rate of 30 frames per second can be stored and retrieved. One can also store around 54,000 studio quality images per side.

Write-once storage media have a capacity between 600 MB and approximately 8 Gigabytes. The diameter of the disks is between 3.5 and 14 inches. The main advantage of a WORM disk, compared to other mass storage media, is the ability to store large amounts of data which may not be changed later, i.e., an archive which is secure. To increase capacity, juke-boxes are available, which allow the stocking of several disks and lead to capacities of over 20 Gigabytes.

Compact Disc Read Only Memory & Multimedia Highway:

Since compact discs store audio in a digital format, they are suitable for storing other information that can be represented in a digital form. In 1984, Philips and Sony released the Compact Disc Read Only Memory (CD-ROM).Short for Compact Disc Read-Only Memory; a CD-ROM is an optical disc that contains audio or software data whose memory is read-only. A CD-ROM Drive or optical drive is the device used to read them. CD-ROM drives have speeds ranging from 1x to 72x, meaning it reads the CD roughly 72 times faster than the 1x version. As you would imagine, these drives are capable of playing audio CDs and reading data CDs, including CD-R and CD-RW discs.

Now that telecommunications are global, information can be received online as distributed resources on a data highway, where we will pay to acquire and use multimedia based information. Any computer data can be pre-recorded on a CD-ROM.

The specification of a CD-ROM disc is summarized in following table

Parameter	Value	Comments
Data capacity	656 MB	For 74 minutes (688 x 10^6 bytes)
Raw data bitrate	1.41Mbits/s	Includes all bytes in sector
User data rate	150 kB/s	At 1x speed
Block (sector) size	2,352 bytes	
User data per sector	2048	With full error correction
Sector rate	75 sectors/s	At 1x speed reading
Sector Modes	1 or 2	See 3.1
Sector Forms	1 or 2	Mode 2 only

CD-ROM discs are read by CD-ROM drives, which have been standard components of personal computers and some games consoles for a number of years. DVD-ROM drives, which are now replacing CD-ROM drives will also read CD-ROM discs. A CD-ROM has several advantages over other forms of data storage, and a few disadvantages.

• Capacity of a CD-ROM is about 700 MB of data.

• The data on a CD-ROM can be accessed much faster than a tape, but CD-ROMs are slower than hard discs.

• Like audio CDs you cannot write to a pre-recorded CD-ROM but only to recordable versions.

CD-ROM discs differ from CD audio discs in two important ways.

• The data on a CD-ROM disc are divided into sectors, which contain both user data and other data for control and error protection.

• The data on a CD-ROM are contained in files. All CD-ROMs therefore need a file system to enable the computers to access the required file easily and quickly.

The Multimedia Highway:

Now, that telecommunications networks are global, and when information providers and content owners determines the worth of theirs products and how to charge money for them, information elements will ultimately link up online's as distributed resources on a data highway (actually more like a toll road). Where you will pay to acquire and use multimedia-based information.

Curiously, the actual glass Fiber cables that makes up much of the physical backbone of the data highway are, in many cases. Owned by railroad and pipelines companies who simply buried the cables on existing rights of way where no special permits and environmental reports are necessary. One railroad in the United States invested more than a million dollars in a special cable laying trenching car; in the United Kingdom, there is talk of placing a fiber-optic cables backbone along the decaying 19th century canal and barge system. Bandwidth on these lines is leased to other, so competing retailers such as AT&T, MCI, and Sprint may even share the same cable. Full-text content from books and magazines is accessible by modem and electronic link; features movies are played at home; real-time new reports from anywhere on earth are available; lectures from participating universities are monitored for education credits; street maps of any city are view-able with recommendations for restaurants, in any language-and online travelogues include testimonials and video tracks. This is not science fiction; it is happing now. For each of these interfaces or gateways to information is a Multimedia projects just waiting to be developed.

Digital Versatile Disc-Read Only Memory (DVD-ROM):

Digital versatile disc-read only memory (DVD-ROM) is a read-only digital versatile disc (DVD) commonly used for storing large software applications. It is similar to a compact disk-read only memory (CD-ROM) but has a larger capacity. A DVD-ROM stores around 4.38 GB of data. A CD-ROM usually stores 650 MB of data.

A DVD-ROM permanently stores data files which cannot be changed, written over or erased. A personal computer (PC) with a DVD-ROM or a DVD-RAM drive is designed to read a DVD-ROM disc. Generally a DVD-ROM disc is not equipped to be used with a DVD drive connected to a home theater system or television. But many DVD-ROM drives can generally read a DVD movie disc.

A DVD-ROM is one of the various types of DVDs. A blank DVD is generally a DVD-R or DVD+R, which has a read-write format. The +R or -R references the format standards and is a rewritable or recordable DVD.

Compared to a CD-ROM, a DVD-ROM has the same 5 inch diameter and 1.2 millimeter (mm) thickness. But because a DVD-ROM uses a shorter wavelength laser with tighter compacted pits, the disc capacity is increased. In fact, the smallest DVD-ROM can store approximately 7 times more data than a CD-ROM. This term is also known as digital video disc ROM.

The DVD-ROM was first introduced in 1996 by the DVD Forum, a group of ten international companies using and developing DVD and HD DVD formats for media, software and hardware. The DVD Forum consists of the founding companies plus over 220 other members. Japan produced the first DVD-ROMs in November 1996. By March 1997 it was introduced in the United States. A typical DVD-ROM can hold up to 17 GB/s of data if both sides of the disc are writable.

The DVD-ROM is comprised of two 0.6 millimeters (mm) acrylic layers bonded together. The double-sided disc consists of two recordable grooved sides. With two layers, a DVD's laser beam only has to go through 0.06 mm to reach the recording layer. Having a thin layer allows the lens to focus the beam to a smaller spot size, which in turn writes smaller pits for more data. The data is encoded in the form of spiral pits that are merely nanometers apart. The spiral path begins at the center of the disc and coils numerous times until it reaches the outer edge. With a double-layered disc the path continues to the second layer. A double-sided disc needs to be manually turned over and the path resumes in the center.

Basic concepts of Sound :

- Sound is the best way to attract attention.
- Often audio provides the only effective way to convey an idea, elicit an emotion, or dramatize point.

- Sounds also can be combined in a multimedia presentation to provide information and enhance the other media being presented.

Why Sound is Important in Multimedia?

- To reinforce message or theme
- To set the mood
- To catch the interest of the audience
- To alert the audience
- To include narration: effective for training and educational application.

Types of Sound

• Music
– Background
– Attention grabber
– Sound effect

• Speech
– Narration
– Instruction

Sound Wave Characteristics:

- Two main characteristic in sound waves are Frequency and Amplitude.
- Frequency in the number of cycles a sound wave creates in one second.
- A cycle is measured from one wave peak to another.
- The standard measurement for frequency is called HERTZ (Hz).
- Amplitude is the volume or loudness a particular sound makes.
- The louder the sound, the higher the amplitude will be.
- The unit of measure for loudness or volume is decibel (dB).

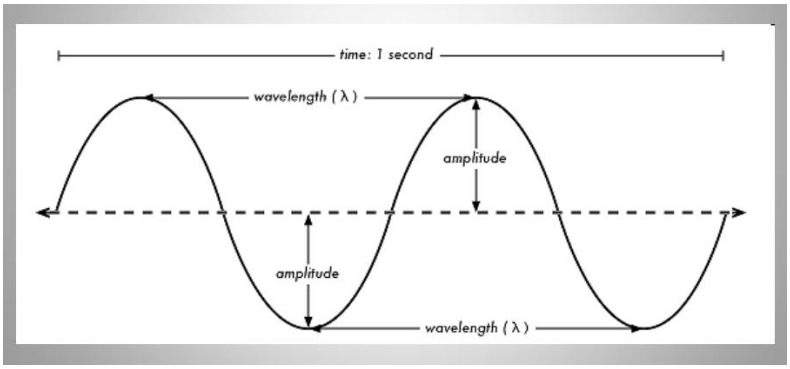


Fig: Basic concept of sound

Refers to the reproduction and transmission of sound stored in a digital format. The digitizing and storage of sound or music on computer or compact disc.

MIDI:

Stands for "Musical Instrument Digital Interface." MIDI is a connectivity standard for transferring digital instrument data. It is primarily used by computers, synthesizers, and electronic keyboards. However, MIDI is supported by several other instruments, such as electronic drums, beat boxes, and even digital stringed instruments like guitars and violins.

MIDI data includes several types of information. For example, pressing a single key on a synthesizer transmits the note played, the velocity (how hard the note is pressed), and how long the note is held. If multiple notes are played at once, the MIDI data is transmitted for all the notes simultaneously. Other data that may be sent over a MIDI connection includes the instrument ID, sustain pedal timings, and controller information, such as pitch bend and vibrato.

When a synthesizer is hooked up to a computer via a MIDI connection, the notes played can be recorded by DAW software in MIDI format. The MIDI data can be played back by sending the recorded MIDI notes to the keyboard, which outputs them as audio samples, such as a piano or strings. Most DAW software supports MIDI editing, allowing you to adjust the timing and velocity of individual notes, change their pitch, delete notes, or add

new ones. MIDI data is often displayed in a digital format, with lines representing each note played. Many programs can convert MIDI data to a musical score as well.

A MIDI recording simply contains instrument information and the notes played. The actual sound is played back using samples (individual recordings) from real instruments. For example, a MIDI track recorded as a song for piano can be played back with a guitar sound by simply changing the output instrument — though it might not sound very realistic.

Originally, MIDI connections used MIDI cables, which connected to a 5-pin MIDI port on each device. Now most MIDI devices have standard computer interfaces, such as USB or Thunderbolt ports. These modern interfaces provide more bandwidth than traditional MIDI ports, allowing more tracks with more data to be transmitted at once.

Digital audio:

Digital audio is a representation of sound recorded in, or converted into, digital form. In digital audio, the sound wave of the audio signal is typically encoded as numerical samples in a continuous sequence. For example, in CD audio, samples are taken 44,100 times per second, each with 16-bit sample depth. Digital audio is also the name for the entire technology of sound recording and reproduction using audio signals that have been encoded in digital form. Following significant advances in digital audio technology during the 1970s and 1980s, it gradually replaced analog audio technology in many areas of audio engineering and telecommunications in the 1990s and 2000s.

In a digital audio system, an analog electrical signal representing the sound is converted with an analog-to-digital converter (ADC) into a digital signal, typically using pulse-code modulation (PCM). This digital signal can then be recorded, edited, modified, and copied using computers, audio playback machines, and other digital tools. When the sound engineer wishes to listen to the recording on headphones or loudspeakers (or when a consumer wishes to listen to a digital sound file), a digital-to-analog converter (DAC) performs the reverse process, converting a digital signal back into an analog signal, which is then sent through an audio power amplifier and ultimately to a loudspeaker.

Digital audio systems may include compression, storage, processing, and transmission components. Conversion to a digital format allows convenient manipulation, storage, transmission, and retrieval of an audio signal. Unlike analog audio, in which making copies of recording results in generation loss and degradation of signal quality, digital audio allows an infinite number of copies to be made without any degradation of signal quality.

Types of Audio File Formats:

There are many different types of digital audio file formats that have resulted from working with different computer platforms and software. Some of the better known formats include:

WAV

WAV is the Waveform format. It is the most commonly used and supported format on the Windows platform. Developed by Microsoft, the Wave format is a subset of RIFE RIFF is capable of sampling rates of 8 and 16 bits. With Wave, there are several different encoding methods to choose from including Wave or PCM format. Therefore, when developing sound for the Internet, it is important to make sure you use the encoding method that the player you're recommending supports.

AU

AU is the Sun Audio format. It was developed by Sun Microsystems to be used on UNIX, NeXT and Sun Sparc workstations. It is a 16-bit compressed audio format that is fairly prevalent on the Web. This is probably because it plays on the widest number of platforms.

RA

RA is Progressive Networks RealAudio format. It is very popular for streaming audio on the Internet because it offers good compression up to a factor of 18. Streaming technology enables a sound file to begin playing before the entire file has been downloaded.

AIFF

AIFF or AFF is Apple's Audio Interchange File Format. This is the Macintosh waveform format. It is also supported on IBM compatibles and Silicon Graphics machines. The AIFF format supports a large number of sampling rates up to 32 bits.

MPEG

MPEG and MPEG2 are the Motion Picture Experts Group formats. They are a compressed audio and video format. Some Web sites use these formats for their audio because their compression capabilities offer up to a factor of at least 14:1. These formats will probably become quite widespread as the price of hardware based MPEG decoders continues to go down and as software Notes decoders and faster processors become more main stream. In addition, MPEG is a standard format.

MIDI

MIDI (MID, MDI, MFF) is an internationally accepted file format used to store Musical InstrumentDigital Interface (MIDI) data. It is a format used to represent electronic music produced by a MIDI device (such as a synthesizer or electronic keyboard). This format provides instructions on how to replay music, but it does not actually record the waveform. For this reason, MIDI files are small and efficient, which is why they are often used on the Web.

SND

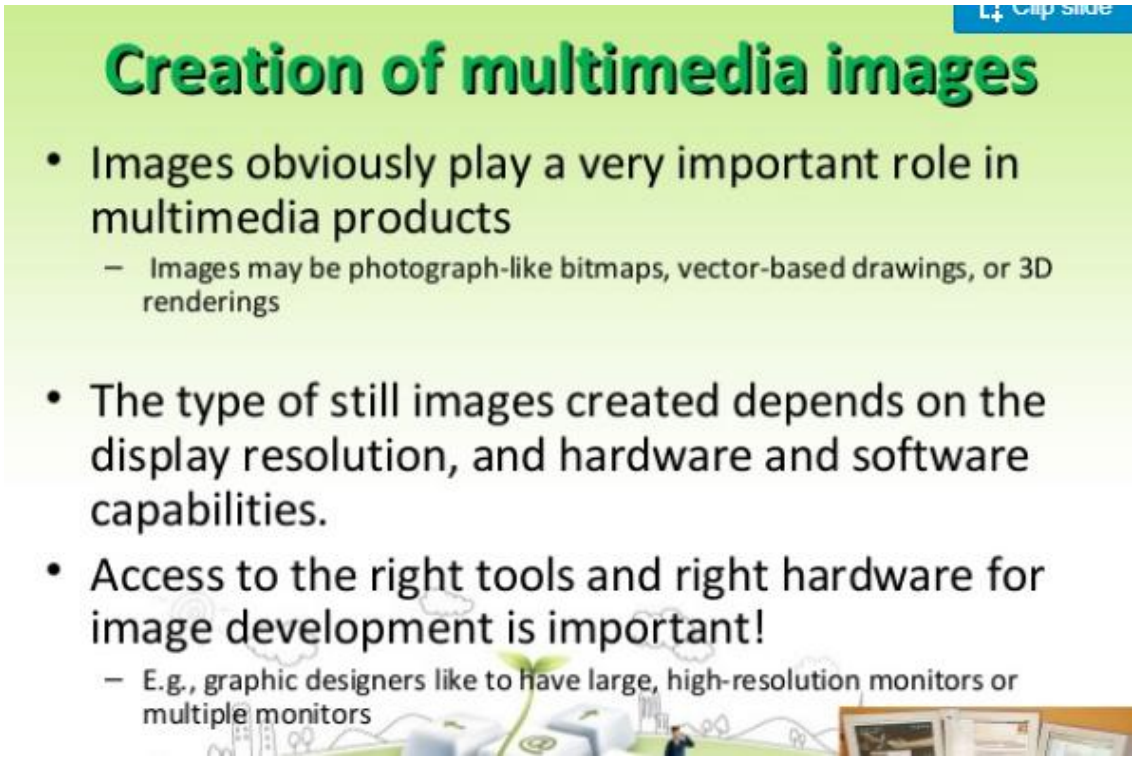
SND is the Sound file format developed by Apple. It is used mainly within the operating system and has a limited sampling rate of eight bits.

UNIT-III
IMAGE AND GRAPHICS

Bitmap (BMP):

Bitmap (BMP) is an image file format that can be used to create and store computer graphics. A bitmap file displays a small dots in a pattern that, when viewed from afar, creates an overall image. A bitmap image is a grid made of rows and columns where a specific cell is given a value that fills it in or leaves it blank, thus creating an image out of the data.

To create a bitmap, an image is broken into the smallest possible units (pixels) and then the color information of each pixel (color depth) is stored in bits that are mapped out in rows and columns. The complexity of a bitmap image can be increased by varying the color intensity of each dot or by increasing the number of rows and columns used to create the image. However, when a user magnifies a bitmap image enough, it eventually becomes pixalated as the dots resolve into tiny squares of color on a grid.



Types of Still Images

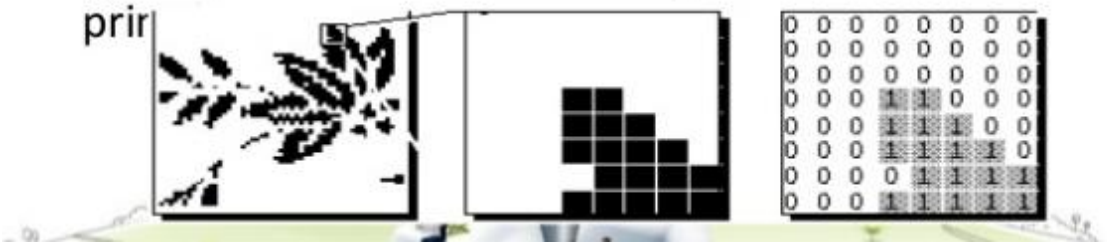
Still images are generated in two ways:

- Bitmaps (or *raster-based*) .
- Vector-drawn graphics.

Bitmaps

- Bitmap is derived from the words 'bit', which means the simplest element in which only two digits are used, and 'map', which is a two-dimensional matrix of these bits.

- A bitmap is a data matrix describing the individual dots of an image that are the smallest elements (pixels) of resolution on a computer screen or prir



Bitmaps

- Bitmaps are an image format suited for creation of:
  - Photo-realistic images.
  - Complex drawings.
  - Images that require fine detail.
- Bitmapped images are known as paint graphics.
- Bitmapped images can have varying bit and color depths.

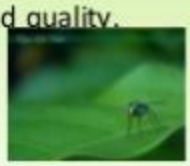
Bitmaps

- More bits provide more color depth, hence more photo-realism;
  - but require more memory and processing power

Bit Depth	Number of Colors Possible	Available Binary Combinations for Describing a Color
1-bit	2	0, 1
2-bit	4	00, 01, 10, 11
4-bit	16	0000, 0001, 0011, 0111, 1111, 0010, 0100, 1000, 0110, 1100, 1010, 0101, 1110, 1101, 1001, 1011

Available binary Combinations for Describing a Color

- Bitmaps picture and their suitability of use:-
  - Use the native Microsoft bmp format as a raw image that will later be processed. It faster to process.
  - Use JPEG, for photo sharing on the web because of its size and quality.



  - GIF is normally used for diagrams, buttons, etc., that have a small number of colours
    - It is also suitable for simple animation because it supports interlaced images.



  - PNG is almost equal to gif except that it didn't support the animation format.

Bitmaps

Bitmaps can be inserted by:

- Using clip art galleries.
- Using bitmap software.
- Capturing and editing images.
- Scanning images.

Clip Art



Drawn





Scan



Capture

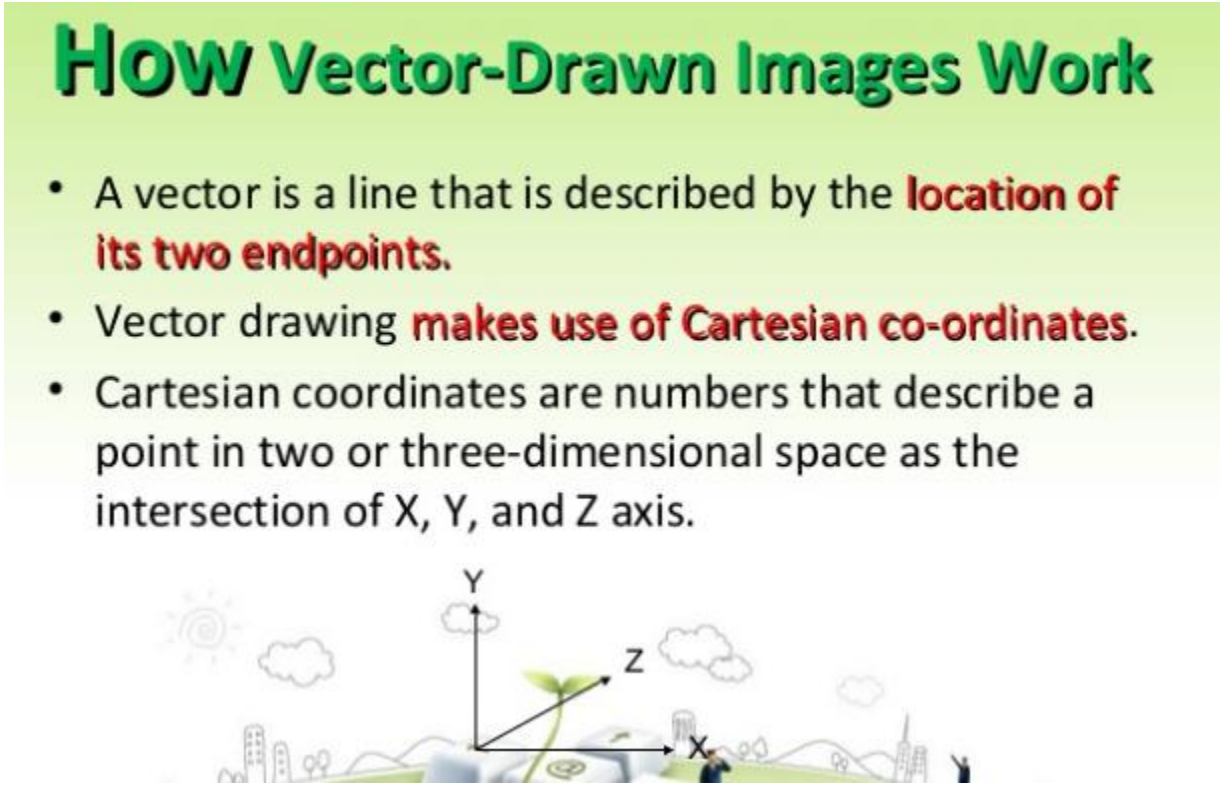
VECTOR DRAWING:-

Vector graphics are graphics that computers render by using mathematical formulas. This allows the image to be scaled up many times without distortion, unlike something in MS Paint which uses raster graphics (graphics made up of pixels).

Vector graphics are primarily used for logos, icons on your computer, and wallpapers due to their scalability.

In computer graphics, a **vector graphic** or **vector image** is an image made up paths (lines with a starting and ending points). These paths, all based on mathematical expressions, consist of basic geometric shapes, such as lines, polygons, and curves. The key advantage of using vector graphics as opposed to raster images is their ability to scale without compromising image quality. For example, you could use a vector graphic to help make a banner or sign.

There are many different file formats (file extensions) used to create a vector graphic. The most common vector file types include .ai, .eps, .pdf, and .svg. One of the most popular programs used to view and create vector images is Adobe Illustrator.



Applications of Vector-Drawn Images

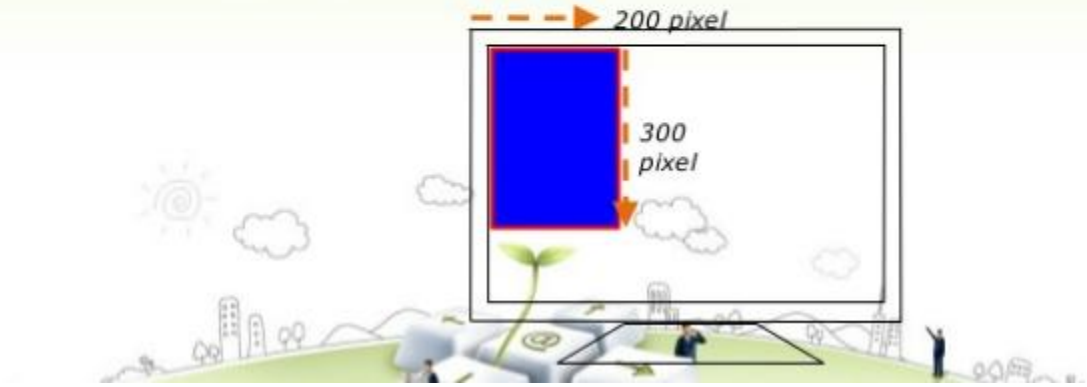
- **Vector-drawn images** - created from **geometric objects** such as lines, rectangles, ovals, polygons using mathematical formulas

- Vector-drawn images are used in the following areas:
  - Computer-aided design (CAD) programs.
  - Graphic artists designing for the print media.
  - 3-D animation programs.
  - Applications requiring drawing of graphic shapes.

- Example

RECT 0,0,200,300,RED,BLUE says

  - *"Draw a rectangle starting at **0,0** (upper left corner of screen) going **200** pixels horizontally right and **300** pixels downward, with a **RED** boundary and filled with **BLUE**."*



Vector-Drawn Images V/S Bitmaps

- Vector images use **less memory space** and have a **smaller file size** (.svg) as compared to bitmaps.

- For the Web, pages that use vector graphics in plug-ins download faster, and when used for animation, draw faster than bitmaps.

Vector-Drawn Images V/S Bitmaps

- Vector images **cannot be used for photorealistic images.**

- Vector images require a **plug-in for Web-based display.**

- Bitmaps are not easily scalable and resizable.

- Bitmaps can be converted to vector images using autotracing.



Image/Graphic File Formats:-

Graphic images are stored digitally using a small number of standardized graphic file formats, including bit map, TIFF, JPEG, GIF, PNG; they can also be stored as raw, unprocessed data.

There are likely billions of graphic images available on the World Wide Web, and with few exceptions, almost any user can view any of them with no difficulty. This is because all those images are stored in what amounts to a handful of file formats. Before discussing the principal graphics file formats, however, we need to review the two fundamental types of graphics: raster and vector.

- Raster graphics, also called bitmap graphics, are digital images that are composed of tiny rectangular pixels. They are arranged in a grid or raster of x and y coordinates in such a way that it forms an image. It is also referred to as bitmap since it has information that is mapped directly to the display's grid.
- The file size of a raster image depends also on the size of the image, which is determined by the number of pixels being used in the image. This means that an image with a 1280x720 resolution will contain 921,600 pixels while a full HD 1920x1080 image will have 2,073,600 pixels.
- Some examples of Raster Graphic File Formats are as follows:

JPEG (JOINT PHOTOGRAPHIC EXPERTS GROUP)

- **A JPEG is an image file format used for compressing image files. The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality.**

PROS

- Retains up to 16,000,000 colours.
- Suitable for images, high details & quality pictures.
- It is the most used graphic file format.
- Approved as standard in 1994.



CONS

- It does not support transparency.
- File size larger than GIF because of colour information.

GIF (GRAPHICS INTERCHANGE FORMAT)

- **A GIF is a lossless file format for image files that supports both animated and static images.**

PROS

- 256 number of colours.
- Uses lossless compression.
- Support for transparency.
- Small file format.

CONS

- The oldest format for web – 1989.
- In most cases it has a bigger file size than PNG.
- Loss of colour variation.



PNG (PORTABLE NETWORK GRAPHICS)

- **Portable Network Graphics (PNG) is a raster graphics file format that supports lossless data compression. PNG was created as an improved replacement for GIF files and is the most widely used lossless image compression format on the internet.**

PROS

- 256 number of colours.
- Uses lossless compression.
- Suitable for flat areas of colours, logos, transparent or semitransparent images
- In most cases has a smaller file size than GIF
- It has alpha channel transparency which is more advanced than GIF's transparency.

CONS

- Relatively large file format.
- In different situations it has bigger file sizes than JPG.



TIFF (TAGGED IMAGE FILE FORMAT)

- **Tagged Image File Format (TIFF) is a computer file format for storing raster images, popular among the publishing industry and photographers. The TIFF format is widely supported by image manipulation applications and publishing and page layout applications.**

PROS

- No image data is lost.
- Better image quality than even the JPEG fine quality.
- Good for images that will be heavily manipulated in a photo editing program.

CONS

- File size is very large.
- Still need to make sure that exposure, white balance and colour saturation are properly set because fixing these in the photo editing program will degrade the image to a certain degree.

VECTOR GRAPHICS

- Unlike other formats, vector graphics are not made up of a grid of pixels. Instead, they are comprised of paths, which are defined by a start and end point, along with other points, curves, and angles along the way. A path can be a line, square, triangle, or a curvy shape. These paths can be used to create simple drawings or complex diagrams.
- Some examples of popular vector graphic file formats are as follows:

AI (ADOBE ILLUSTRATOR ARTWORK)

- **Adobe Illustrator Artwork (AI) is a file format developed by Adobe Systems for representing single-page vector-based drawings in either the EPS or PDF formats. The .ai filename extension is used by Adobe Illustrator.**

PROS

- No loss of image quality.
- Can be scaled to any size.

CONS

- Not as portable as EPS files.



PSD (PHOTOSHOP DOCUMENT)

- **A PSD file is a layered image file used in Adobe Photoshop. It is the default format that Photoshop uses for saving data. PSD is a proprietary file that allows the user to work with the images' individual layers even after the file has been saved. When an image is complete, Photoshop allows the user to flatten the layers and convert the flat image into a JPG, TIFF, PNG or other file format so it can be shared.**

PROS

- Allows user to change images with good control over individual layers, channels and paths.

CONS

- Only works in certain applications/software programs.
- Not useful for web display.



PDF (PORTABLE DOCUMENT FORMAT)

- **The Portable Document Format (PDF) is a file format used to present documents in a manner independent of application software and hardware. Each PDF file encapsulates a complete description of a fixed-layout flat document, including the text, fonts, graphics and other information needed to display it.**

PROS

- PDF can be opened on any device with any operating system in exactly the same form in which it was created.
- Easy for viewing
- Takes up very little storage space.



CONS

- It's not free to edit PDF files.
- It is difficult to work with text in PDF files because they are perceived as pictures.

Image file format:-

BMP

The simplest way to define a raster graphic image is by using color-coded information for each pixel on each row. This is the basic bit-map format used by Microsoft Windows. The disadvantage of this type of image is that it can waste large amounts of storage. Where there's an area with a solid color, for example, we don't need to repeat that color information for every new contiguous pixel. Instead, we can instruct the computer to repeat the current color until we change it. This type of space-saving trick is the basis of compression, which allows us to store the graphic using fewer bytes. Most Web graphics today are compressed so that they can be transmitted more quickly. Some compression techniques will save space yet preserve all the information that's in the image. That's called "lossless" compression. Other types of compression can save a lot more space, but the price you pay is degraded image quality. This is known as "lossy" compression.

TIFF

Most graphics file formats were created with a particular use in mind, although most can be used for a wide variety of image types. Another common bit-mapped image type is Tagged Image File Format, which is used in faxing, desktop publishing and medical imaging. TIFF is actually a "container" that can hold bit maps and JPEGs and allows (but doesn't require) various types of compression.

JPEG

The Joint Photographic Experts Group created the JPEG standard in 1990 for the efficient compression of photographic images. JPEG allows varying levels of lossy compression, letting you trade off quality against file size. Progressive JPEG is a way to rearrange the graphic data to permit a rough view of the entire image even when only a small portion of the file has been downloaded. The JPEG standard includes 29 distinct coding processes, but not all of them need to be used. If an image has flat areas of single color that transition sharply to contiguous areas, JPEG doesn't work as well as GIF.

JPEG 2000 is a wavelet-based standard designed to supersede the original. It offers improved compression, including lossless compression, and supports multiple resolutions in a single file, but it has only limited support in current Web browsers.

GIF

The Graphic Interchange Format takes an image and re-creates it using a palette of no more than 256 colors. These palettes can be totally different for different images. GIF is a very efficient format that achieves very good compression for non photographic images. GIF also permits the creation of animated images by allowing a file to contain several different frames (each with its own palette) and to switch between them with a specified delay. In addition, GIF images are one of the few types that can have a transparent background, meaning that there's no need to always display a rectangular area.

PNG

Portable Network Graphics is a standard developed in 1996 as an alternative to and improvement on GIF, but without the patent issues and palette restrictions. PNG can compress an image more than GIF and supports improved background transparency/opacity but allows only single images, without animation.

UNIT-IV
VIDEO & ANIMATION

Basics Concepts of Video :

Video is a combination of image and audio. It consists of a set of still images called frames displayed to the user one after another at a specific speed, known as the frame rate measured in number of frames per second (fps), If displayed fast enough our eye cannot distinguish the individual frames, but because of persistence of vision merges the individual frames with each other thereby creating an illusion of motion. The frame rate should range between 20 and 30 for perceiving smooth realistic motion. Audio is added and synchronized with the apparent movement of images. The recording and editing of sound has long been in the domain of the PC. Doing so with motion video has only recently gained acceptance. This is because of the enormous file size required by video.

Example: One second of 24-bit, 640 × 480 mode video and its associated audio requires 27 MB of space, i.e. (640 × 480 pixels) (24 bits/pixel) (30 frames/sec). Thus, a 20 minute dip fills up 32 GB of disk space.

In all the multimedia elements, video places the highest performance demand on your computer and its memory and storage. Consider that a high-quality color still image on a computer screen could require as much as a megabyte of storage memory. Multiply this by 30-the number of times per second that still picture is replaced to provide the appearance of motion-and you would need 30 megabytes of storage to play your video for one second, or 1.8 gigabytes of storage for a minute. Just moving these entire pictures from computer memory to the screen at that rate would challenge the processing capability of a super- computer. Multimedia technologies and re-search efforts today deal with compressing digital video image data into manageable streams of information so that a massive amount of image can be squeezed into a comparatively small data file that still delivers a good viewing experience on the intended viewing platform during playback.
Carefully planned, well-executed video clips can make a dramatic difference in a multimedia project.

Analog Video

When light reflected from an object passes through a video camera lens that light is converted into electronic signals by a special sensor called charged couple device (CCD). Top-quality broadcast cameras may have as many as three CCDs (one for each color of red, green, and blue) to enhance the resolution of the camera. The output of the CCD is processed by the camera into a signal containing three channels of color information and synchronization pulses (sync): There are several video standards for managing CCD output, each dealing with the amount of separation between the components of the signal. The more separation of the color information in the signal results in higher quality of the image (and the more expensive the equipment).

Digital Video

Analog video has been used for years in recording / editing studios and television broadcasting. For the purpose of incorporating video content in multimedia production video needs to be converted into the digital format.
It has already been mentioned that processing digital video on personal computers was very difficult initially, firstly because of the huge file sizes required, and secondly of the large bit rate and processing power required. Full screen video only became a reality after advent of the Pentium-II processor together with fast disks capable of delivering the required output. Even with these powerful resources delivering video files was difficult until the reduction in prices of compression hardware and software. Compression helped to reduce the size of video files to a great extent which required a lower bit-rate to transfer them over communication buses. Nowadays video is rarely viewed in; the uncompressed form unless there is specific reason for doing so, e.g. to maintain the high quality, as for medical analysis.

Using Video On PC

Analog video needs to be converted to the digital format before it can be displayed on a PC screen. The procedure for conversion involves two types of devices- source devices and capture devices.
The source and source device can be one of the following:
camcorder with pre-recorded video tape
VCP with pre-recorded video tape
Video camera with live footage.
We need Video capture card to convert analog signal to digital signal along with video capture Software such as AVI capture, AVI to MPEG Converter, MPEG capture, DAT to MPEG Converter or MPEG Editor.

Video Formats

The AVI Format
The AVI (Audio Video Interleave) format was developed by Microsoft.The AVI format is supported by all computers running Windows, and by all the most popular web browsers. It is a very common format on the Internet, but not always possible to play on non-Windows computers. Videos stored in the AVI format have the extension .avi.
The Windows Media Format
The Windows Media format is developed by Microsoft. Windows Media is a common format on the Internet, but Windows Media movies cannot be played on non-Windows computer without an extra (free) component installed. Some later Windows Media movies cannot play at all on non-Windows computers because no player is available. Videos stored in the Windows Media format have the extension .wmv.
The MPEG Format
The MPEG (Moving Pictures Expert Group) format is the most popular format on the Internet. It is cross-platform, and supported by all the most popular web browsers. Videos stored in the MPEG format have the extension .mpg or .mpeg.
The QuickTime Format

The QuickTime format is developed by Apple. QuickTime is a common format on the Internet, but QuickTime movies cannot be played on a Windows computer without an extra (free) component installed. Videos stored in the QuickTime format have the extension .mov.

The RealVideo Format

The RealVideo format was developed for the Internet by Real Media. The format allows streaming of video (on-line video, Internet TV) with low bandwidths. Because of the low bandwidth priority, quality is often reduced.

Videos stored in the RealVideo format have the extension .rm or .ram.

The Shockwave (Flash) Format

The Shockwave format was developed by Macromedia. The Shockwave format requires an extra component to play. This component comes preinstalled with the latest versions of Netscape and Internet Explorer.

Videos stored in the Shockwave format have the extension .swf.

Broadcast video standards:

Three analog broadcast video standards are commonly in use around the world: NTSC, PAL, and SECAM. In the United States, the NTSC standard is being phased out, replaced by ATSC digital television standard. Because these standards and formats are not easily interchangeable, it is important to know where your multimedia project will be used. A video cassette recorded in USA which uses NTSC will not play on a television set in any European country (which uses either PAL or SECAM), even thought the recording method and style of the cassette is "VHS". Likewise, tapes recorded in European PAL or SECAM formats will not play back on an NTSC video cassette recorder. Each system is based on a different standard that defines the way information is encoded to produce the electronics signal that ultimately creates a television picture. Multi-format VCRs can play back all three standards but typically cannot dub from one standard to another; dubbing between standards still require high-end specialized equipment.

National Television Standard Committee (NTSC)

The United States, Canada, Mexico, Japan, and many other countries use a system for broadcasting and displaying video that is based upon the specifications set forth by the 1952 National Television Standards Committee. These standards define a method for encoding information into the electronic signal that ultimately creates a television picture. As specified by the NTSC standard, a single frame of video is made up of 525 horizontal scan lines drawn onto the inside face of a phosphor-coated picture tube every 30th of a second by a fast-moving electron beam. The drawing occurs so fast that your eye perceives the image as stable. The electron beam actually makes two passes as it draws a single video frame, first laying down all the odd-numbered lines, then all the even-numbered lines. Each of these passes (which happen at a rate of 60 per second, or 60 Hz) paints a field, and the two fields are combined to create a single frame at a rate of 30 frames per second (fps). (Technically, the speed is actually 29.97 Hz.) This process of building a single frame from two fields is called interlacing, a technique that helps to prevent flicker on television screens. Computer monitors use a different progressive-

scan technology, and draw the lines of an entire frame in a single pass, without interlacing them and without flicker.

Phase Alternate Line (PAL)

The Phase Alternate Line (PAL) system is used in the United Kingdom, Western Europe, Australia, South Africa, China, and South America. PAL increases the screen resolution to 625 horizontal lines, but slows the scan rate to 25 frames per second. As with NTSC, the even and odd lines are interlaced, each field taking 1/50th of a second to draw (50 Hz).

Sequential Color and Memory (SECAM)

The Sequential Color and Memory is used in France, Eastern Europe, the former USSR, and a few other countries. Although SECAM is a 625-line, 50 Hz system, it differs greatly from both the NTSC and the PAL color systems in its basic technology and broadcast method. Often, however, TV sets sold in Europe utilize dual components and can handle both PAL and SECAM systems.

Advanced Television Standard Committee (ATSC) and Digital Television (DTV)

What started as the High Definition Television (HDTV) initiative of the Federal Communications Commission in the 1980s, changed first to the Advanced Television (ATV) initiative and then finished as the Digital Television (DTV) initiative by the time the FCC announced the change in 1996. This standard, slightly modified from the Digital Television Standard and Digital Audio Compression Standard, moves U.S. television from an analog to digital standard and provides TV stations with sufficient bandwidth to present four or five Standard Television signals (STV, providing the NTSC's resolution of 525 lines with a 3:4 aspect ratio, but in a digital signal) or one HDTV signal (providing 1,080 lines of resolution with a movie screen's 16:9 aspect ratio). More significantly for multimedia producers, this emerging standard allows for transmission of data to computers and for new ATV interactive services. As of May 2003, 1,587 TV stations in the United States (94 percent) had been granted a DTV construction permit or license. Among those, 1,081 stations were actually broadcasting a DTV signal, almost all simultaneously-casting their regular TV signal. According to the current schedule, all the stations are to cease broadcasting on their analog channel and completely switch to a digital signal by 2006.

High Definition Television (HDTV)

It provides high resolution in 16: 9 aspect ratios. This aspect ratio allows the viewing of Cinemascope and Panavision movies. There is contention between the broadcast and computer industries about whether to use interlacing or progressive-scan technologies. The broadcast industry has promulgate an ultra-high-resolution, 1920xl080 interlaced format to become the cornerstone of a new generation of high-end entertainment centers, but the computer industry would like to settle on a 1280x720 progressive-scan system for HDTV. While the 1920xl080 format provides more pixel than the 1280x720 standard, the refresh rates are quite different. This

higher-resolution interlaced format delivers only half the picture every 1/60 of a second, and because of the interlacing, on highly detailed images there is a great deal of screen flicker at 30 Hz. The computer people argue that the picture quality at 1280x720 is superior and steady. Both formats have been included in the HDTV standard by the Advanced Television System!

Today's multimedia monitors typically use a screen pixel ratio of 4:3 (800x600), but the new HDTV standard specifies a ratio of 16:9 (1280x720), much wider than tall. There is no easy way to stretch and shrink existing graphics material to this new aspect ratio, so new multimedia design and interface principles will need to be developed for HDTV presentations.

TELEVISION:

Conventional System:

Analog television is the original television technology that uses analog signals to transmit video and audio. In an analog television broadcast, the brightness, colors and sound are represented by amplitude, phase and frequency of an analog signal.

Analog signals vary over a continuous range of possible values which means that electronic noise and interference may be introduced. Thus with analog, a moderately weak signal becomes snowy and subject to interference. In contrast, picture quality from a digital television (DTV) signal remains good until the signal level drops below a threshold where reception is no longer possible or becomes intermittent.

Analog television may be wireless (terrestrial television and satellite television) or can be distributed over a cable network as cable television.

All broadcast television systems used analog signals before the arrival of DTV. Motivated by the lower bandwidth requirements of compressed digital signals, since the 2000s a digital television transition is proceeding in most countries of the world, with different deadlines for the cessation of analog broadcasts.

In an analogue system, the output of the CCD is processed by the camera into three channels of colour information and synchronization pulses (sync) and the signals are recorded onto magnetic tape. There are several video standards for managing analogue CCD output, each dealing with the amount of separation between the components—the more separation of the colour information, the higher the quality of the image (and the more expensive the equipment). If each channel of colour information is transmitted as a separate signal on its own conductor, the signal output is called component (separate red, green, and blue channels), which is the preferred method for higher-quality and professional video work. Lower in quality is the signal that makes up Separate Video (S-Video), using two channels that carry luminance and chrominance information. The least separation (and thus the lowest quality for a video signal) is composite, when all the signals are mixed together and carried on a single cable as a composite of the three colour channels and the sync signal. The composite signal yields less-precise colour definition, which cannot be manipulated or colour-corrected as much as S-Video or component signals.

Enhanced Definition System:

Digital television (DTV) is the transmission of television signals using digital rather than conventional analog methods.
Advantages of DTV over analog TV include:

- Superior image resolution (detail) for a given bandwidth
- Smaller bandwidth for a given image resolution
- Compatibility with computers and the Internet
- Interactivity
- Superior audio quality
- Consistency of reception over varying distances

Digital television is not the same thing as HDTV (high-definition television). HDTV describes a new television format (including a new aspect ratio and pixel density), but not how the format will be transmitted. Digital television can be either standard or high definition.
In the United States, analog television broadcasts will stop in 2009. Analog television sets will require digital set-top boxes to convert transmissions. Most satellite and cable subscribers already have converter boxes. However, people who previously used antennas for "over the air" transmission and cable customers who now plug the cable directly into their sets will need converters.

High-definition System

The high-definition television, also known as HDTV (High Definition Television) is a television system with a resolution significantly higher than in the traditional formats (NTSC, SECAM, PAL).
The HDTV is transmitted digitally and therefore its implementation generally coincides with the introduction of digital television (DTV), technology that was launched during the 1990s.
Although several patterns of high-definition television have been proposed or implemented, the current HDTV standards are defined by ITU-R BT.709 as 1080i (interlaced), 1080p (progressive) or 720p using the 16:9 screen format.
The term “high definition” can refer to the specification of the resolution itself or, more generally, the mídia capable of such a definition as the video media support or the television set.
What will be of interest in the near future is high definition video, through the successors of the DVD, HD DVD and Blu-Ray (is expected that the last one will be adopted as a standard) and, consequently, the projectors and LCD and plasma televisions sets as well as retro projectors and video recorders with 1080p resolution/definition.
High-definition television (HDTV) yields a better-quality image than standard television does, because it has a greater number of line resolution.
The visual information is some 2 to 5 times sharper because the gaps between the scan lines are narrower or invisible to the naked eye.
The larger the size of the television the HD picture is viewed on, the greater the improvement in picture quality. On smaller televisions there may be no noticeable improvement in picture quality.

The lower-case “i” appended to the numbers denotes interlaced; the lower-case “p” denotes progressive: With the interlaced scanning method, the 1,080 lines of resolution are divided into pairs, the first 540 alternate lines are painted on a frame and then the second 540 lines are painted on a second frame; the progressive scanning method simultaneously displays all 1,080 lines on every frame, requiring a greater bandwidth.

Computer Based Animation:

Animation is the process of designing, drawing, making layouts and preparation of photographic sequences which are integrated in the multimedia and gaming products. Animation involves the exploitation and management of still images to generate the illusion of movement. A person who creates animations is called animator. He / she use various computer technologies to capture the still images and then to animate these in desired sequence.

Multimedia is the term used to represent combination of visual and audio materials gathered from various resources and then added into one single combination. A multimedia product can be sets of texts, graphic arts, sounds, animations and videos. Precisely, term multimedia is used to refer visual and audio materials into a single common presentation which can be played in a computer including CD ROM or digital video, internet or web technology, streaming audio or video and data projection system etc.

Modern entertainment industry i.e. film and television has gained new heights because of advances in animation, graphics and multimedia. Television advertisements, cartoons serials, and presentation and model designs - all use animation and multimedia techniques.

Animation Techniques

Animators have invented and used a variety of different animation techniques. Basically there are six animation technique which we would discuss one by one in this section.

Traditional Animation frame by frame

Traditionally most of the animation was done by hand. All the frames in an animation had to be drawn by hand. Since each second of animation requires 24 frames filmfilm, the amount of efforts required to create even the shortest of movies can be tremendous.

Keyframing

In this technique, a storyboard is laid out and then the artists draw the major frames of the animation. Major frames are the ones in which prominent changes take place. They are the key points of animation. Key framing requires that the animator specifies critical or key positions for the objects. The computer then automatically fills in the missing frames by smoothly interpolating between those positions.

Procedural

In a procedural animation, the objects are animated by a procedure − a set of rules − not by key framing. The animator specifies rules and initial conditions and runs simulation. Rules are often based on physical rules of the real world expressed by mathematical equations.

Behavioral

In behavioral animation, an autonomous character determines its own actions, at least to a certain extent. This gives the character some ability to improvise, and frees the animator from the need to specify each detail of every character's motion.

Performance Based Motion Capture Motion Capture

Another technique is Motion Capture, in which magnetic or vision-based sensors record the actions of a human or animal object in three dimensions. A computer then uses these data to animate the object.

This technology has enabled a number of famous athletes to supply the actions for characters in sports video games. Motion capture is pretty popular with the animators mainly because some of the commonplace human actions can be captured with relative ease. However, there can be serious discrepancies between the shapes or dimensions of the subject and the graphical character and this may lead to problems of exact execution.

Physically Based Dynamics Dynamics

Unlike key framing and motion picture, simulation uses the laws of physics to generate motion of pictures and other objects. Simulations can be easily used to produce slightly different sequences while maintaining physical realism. Secondly, real-time simulations allow a higher degree of interactivity where the real person can maneuver the actions of the simulated character.

In contrast the applications based on key-framing and motion select and modify motions form a pre-computed library of motions. One drawback that simulation suffers from is the expertise and time required to handcraft the appropriate controls systems.

Key Framing

A keyframe is a frame where we define changes in animation. Every frame is a keyframe when we create frame by frame animation. When someone creates a 3D animation on a computer, they usually don’t specify the exact position of any given object on every single frame. They create keyframes.

Keyframes are important frames during which an object changes its size, direction, shape or other properties. The computer then figures out all the in-between frames and saves an extreme amount of time for the animator. The following illustrations depict the frames drawn by user and the frames generated by computer.

Name of Course	Bachelor of Computer Application (BCA)
Semester	IV
Name of Subject	RDBMS
Subject Code	BCA- 403
Marks	75 Marks
Lectures	50 Lectures

RDBMS Objectives

The objective of this course is to expose the students to the fundamentals & basic concepts in relational Data Base Management Systems. This course discusses architecture of Database Systems with concept of relational model & ER model. The course discusses the SQL statements, functions and views. Concepts of PL/SQL also discussed.

RDBMS Outcomes

The course will demonstrate an understanding of the basic & advanced features of RDBMS. The course will demonstrate the various database tables and joins them using SQL commands, able to develop structured query language (SQL) queries to create, read, update, and delete relational database data.

UNIT – I

1. Introduction and Basic Concepts 10

Lectures

1.1 Introduction to RDBMS

1.2 Characteristics of RDBMS

1.3 Applications and Advantages of RDBMS

1.4 Data models – Entity Relationship (ER), Mapping ER Model to Relational Mode, Network.

1.5 Relational and Object Oriented Data Models

UNIT - II

2. SQL Statements & Working with Tables 15

Lectures

2.1 Introduction to SQL
2.2 SQL Commands and its types (DDL, DML, DQL, DCL, Transaction Control Commands)

2.3 Data types in SQL

2.4 Creating Tables, Selecting from tables, WHERE Clause

2.5 Selecting from tables, DISTINCT Clause, Column aliasing

2.6 Manipulation Table data

2.7 Altering Table structure

2.8 Data Constraints: Unique, Not Null, Primary Key, Foreign Key, Check, Default Constraint

UNIT - III

3. Operators & SQL Functions & Views 15

Lectures

3.1 Arithmetic Operators, Relational Operators

3.2 Comparison Operators BETWEEN, IN, LIKE, IS NULL

3.3 LOGICAL Operators: AND OR NOT

3.4 SQL Functions: Single, Multiple Row Functions

3.5 Single Row Character, Single Row Number, Single Row Date, Single Row Conversion, Single Row General Functions

3.6 Multiple Row Functions

3.7 Views

UNIT - IV

4. Sorting, Grouping, Joining and PL/ SQL 10

Lectures

4.1 Introduction to Sorting

4.2 Order by Clause, Group by Clause

4.3 Join, Types of Join: Equi Joins, Non Equi Join, Outer Join: Left, Right, Full, Self Join, Cross Join, Joining three tables

4.4 Subqueries & its types

4.5 Overview of PL / SQL

4.5.1 Declarations Section

4.5.2 Executable Commands Section

4.5.3 Exception Handling Section

References:

3. Database System Concepts by Sudarshan, Korth (McGraw-Hill Education)
4. An introduction to Database System – Bipin Desai, Galgotia Publications
5. Database System: concept, Design & Application by S.K.Singh (Pearson Education)
6. Database Modeling and Design: Logical Design by Toby J. Teorey, Sam S. Lightstone, and Tom Nadeau, 4thEdition, 2005, Elsevier India Publications, New Delhi
7. SQL, PL/SQL the programming language of ORACLE 4th Edition by Ivan Bayross ISBN-817656964-X
8. "Oracle Database 10g PL/SQL Programming" by Scott Urman , Ron Hardman, MichaleMc Laughlin, Oracle Press, TMH, ISBN-0-07-059779-0.

CHAPTER 1

INTRODUCTION TO RDBMS

1.0 Objectives
1.1 Introduction
1.2 What is RDBMS ?
1.3 Difference between DBMS & RDBMS
1.4 Summary
1.5 Check your Progress – *Answers*
1.6 Questions for Self – Study
1.7 Suggested Readings

1.0 OBJECTIVES

After reading this chapter you will be able to,

- Describe what RDBMS is
- State the difference between DBMS & RDBMS

1.1 INTRODUCTION

Most of the problems faced at the time of implementation of any system are outcome of a poor database design. In many cases it happens that system has to be continuously modified in multiple respects due to changing requirements of users. It is very important that a proper planning has to be done.
A relation in a relational database is based on a relational schema, which consists of number of attributes.
A relational database is made up of a number of relations and corresponding relational database schema.
The goal of a relational database design is to generate a set of relation schema that allows us to store information without unnecessary redundancy and also to retrieve information easily.
One approach to design schemas that are in an appropriate normal form. The normal forms are used to ensure that various types of anomalies and inconsistencies are not introduced into the database.

1.2 WHAT IS RDBMS?

RDBMS stands for Relational Database Management System. RDBMS data is structured in database tables, fields and records. Each RDBMS table consists of database table rows. Each database table row consists of one or more database table fields.
RDBMS store the data into collection of tables, which might be related by common fields (database table columns). RDBMS also provide relational operators to manipulate the data stored into the database tables. Most RDBMS use SQL as database querylanguage.
The most popular RDBMS are MS SQL Server, DB2, Oracle and MySQL.
The relational model is an example of record-based model. Record based models are so named because the database is structured in fixed format records of several types. Each table contains records of a particular type. Each record type defines a fixed number of fields, or attributes. The columns of the table correspond to the attributes of the record types. The relational data model is the most widely used data model, and a vast majority of current database systems are based on the relational model.
The relational model was designed by the IBM research scientist and mathematician, Dr. E.F.Codd. Many modern DBMS do not conform to the Codd's definition of a RDBMS, but nonetheless they are still considered to be RDBMS.
Two of Dr.Codd's main focal points when designing the relational model were to further reduce data redundancy and to improve data integrity within database systems.

The relational model originated from a paper authored by Dr.codd entitled “A Relational Model of Data for Large Shared Data Banks”, written in 1970. This paper included the following concepts that apply to database management systems for relational databases.
The relation is the only data structure used in the relational data model to represent both entities and relationships between them.
Rows of the relation are referred to as **tuples** of the relation and columns are its **attributes**. Each attribute of the column are drawn from the set of values known as **domain**. The domain of an attribute contains the set of values that the attribute may assume.
From the historical perspective, the relational data model is relatively new .The first database systems were based on either network or hierarchical models .The relational data model has established itself as the primary data model for commercial data processing applications. Its success in this domain has led to its applications outside data processing in systems for computer aided design and other environments.

1.3 DIFFERENCE BETWEEN DBMS & RDBMS

A DBMS has to be persistent, that is it should be accessible when the program created the data ceases to exist or even the application that created the data restarted. A DBMS also has to provide some uniform methods independent of a specific application for accessing the information that is stored.
RDBMS is a Relational Data Base Management System Relational DBMS. This adds the additional condition that the system supports a tabular structure for the data, with enforced relationships between the tables. This excludes the databases that don't support a tabular structure or don't enforce relationships between tables.
You can say DBMS does not impose any constraints or security with regard to data manipulation it is user or the programmer responsibility to ensure the ACID PROPERTY of the database whereas the RDBMS is more with this regard because RDBMS define the integrity constraint for the purpose of holding ACID PROPERTY.

1.1,1.2, and 1.3 Check your progress

Fill in the blanks

1) A relation in a relational database is based on a relational schema, which consists of number of ..... .
2) .....is a Relational Data Base Management System.
3) Rows of the relation are referred to as ..... of the relation
4) The relational model was designed by the IBM research scientist and mathematician, Dr. .....
5) The ..... is the only data structure used in the relational data model to represent both entities and relationships between them.

State true or false

1) The normal forms never removes anomalies.
2) Each attribute of the column are drawn from the set of values known as domain.
3) The first database systems were based on either network or hierarchical models .
4) Most RDBMS use SQL as database query language.
5) Relational database design makes data retrieval difficult.

1.4 SUMMARY

The goal of a relational database design is to generate a set of relation schema that allows us to store information without unnecessary redundancy and also to retrieve information easily.
A database system is an integrated collection of related files, along with details of interpretation of the data contained therein. DBMS is a s/w system that allows access to data contained in a database. The objective of the DBMS is to provide a convenient and effective method of defining, storing and retrieving the information contained in the database.

Chapter 2: Entity-Relationship Model

- Entity Sets
- Relationship Sets
- Design Issues
- Mapping Constraints
- Keys
- E–R Diagram
- Extended E-R Features
- Design of an E-R Database Schema
- Reduction of an E-R Schema to Tables

Entity Sets

- A *database* can be modeled as:
  - a collection of entities,
  - relationships among entities.
- An *entity* is an object that exists and is distinguishable from other objects.

  Example: specific person, company, event, plant
- An *entity set* is a set of entities of the same type that share the same properties.

  Example: set of all persons, companies, trees, holidays

Attributes

- An entity is represented by a set of attributes, that is, descriptive properties possessed by all members of an entity set.

  Example:

  customer = (*customer-name*, *social-security*, *customer-street*, *customer-city*)
  account = (*account-number*, *balance*)

- *Domain* – the set of permitted values for each attribute
- Attribute types:
  - *Simple* and *composite* attributes.
  - *Single-valued* and *multi-valued* attributes.
  - *Null* attributes.
  - *Derived* attributes.

Relationship Sets

- A *relationship* is an association among several entities

  Example:

  Hayes — *depositor* — A-102

  customer entity — relationship set — *account* entity

- A *relationship set* is a mathematical relation among $n \geq 2$ entities, each taken from entity sets

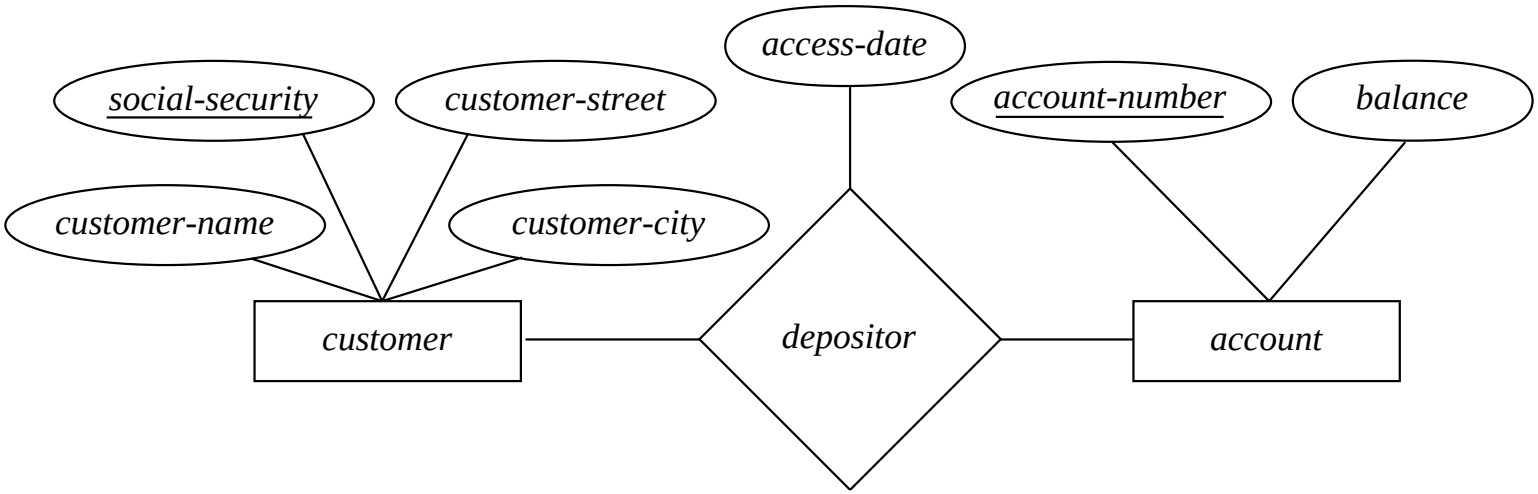
$$\{(e_1, e_2, ..., e_n) \mid e_1 \in E_1, e_2 \in E_2, ..., e_n \in E_n\}$$

  where $(e_1, e_2, ..., e_n)$ is a relationship

  - Example: (Hayes, A-102) $\in$ *depositor*

Relationship Sets (Cont.)

- An *attribute* can also be a property of a relationship set.
  For instance, the *depositor* relationship set between entity sets *customer* and *account* may have the attribute *access-date*



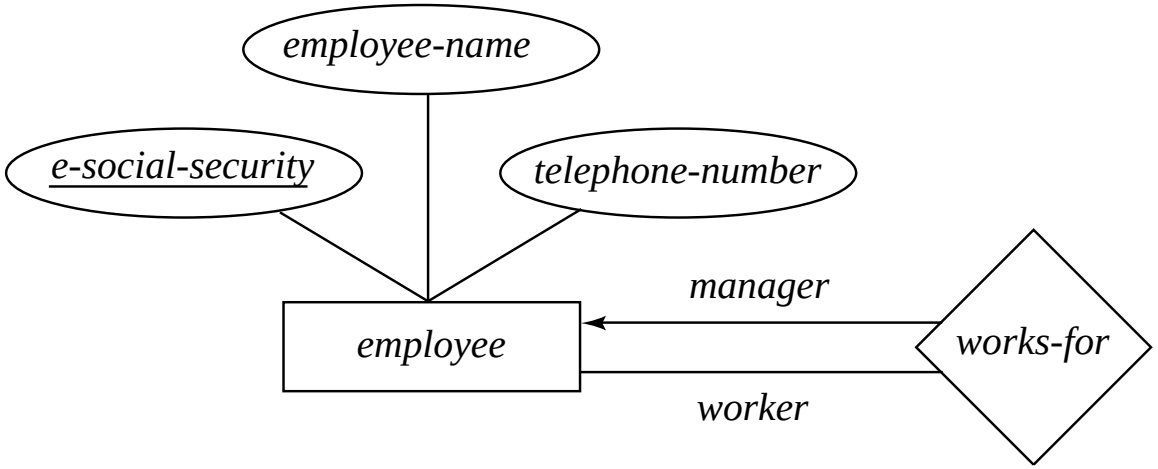
Degree of a Relationship Set

- Refers to number of entity sets that participate in a relationship set.
- Relationship sets that involve two entity sets are *binary* (or degree two). Generally, most relationship sets in a database system are binary.
- Relationship sets may involve more than two entity sets.

  The entity sets *customer*, *loan*, and *branch* may be linked by the ternary (degree three) relationship set *CLB*.

Roles

Entity sets of a relationship need not be distinct



- The labels “manager” and “worker” are called *roles*; they specify how employee entities interact via the works-for relationship set.
- Roles are indicated in E-R diagrams by labeling the lines that connect diamonds to rectangles.
- Role labels are optional, and are used to clarify semantics of the relationship

Design Issues

- Use of entity sets vs. attributes

  Choice mainly depends on the structure of the enterprise being modeled, and on the semantics associated with the attribute in question.

- Use of entity sets vs. relationship sets

  Possible guideline is to designate a relationship set to describe an action that occurs between entities

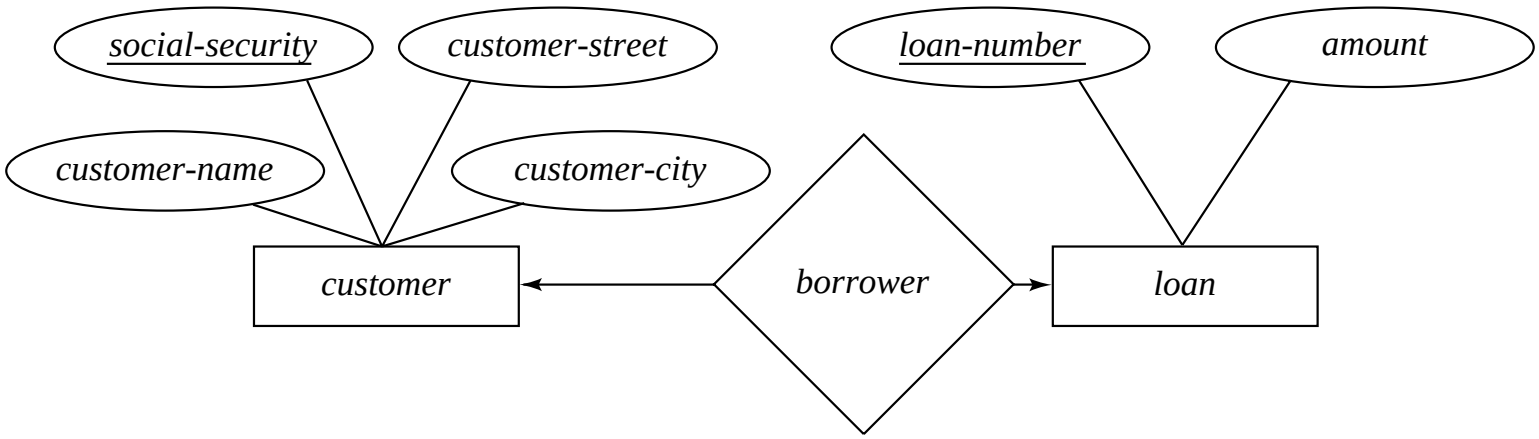
- Binary versus n-ary relationship sets

  Although it is possible to replace a nonbinary (n-ary, for $n > 2$) relationship set by a number of distinct binary relationship sets, a n-ary relationship set shows more clearly that several entities participate in a single relationship.

Mapping Cardinalities

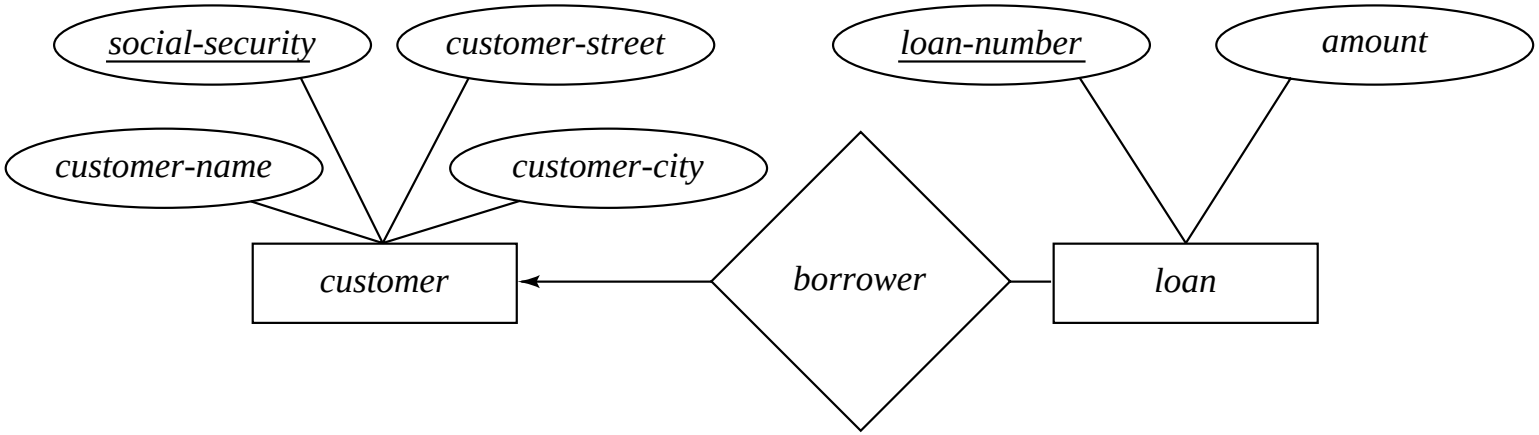
- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
  - One to one
  - One to many
  - Many to one
  - Many to many
- We distinguish among these types by drawing either a directed line ($\rightarrow$), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

One-To-One Relationship

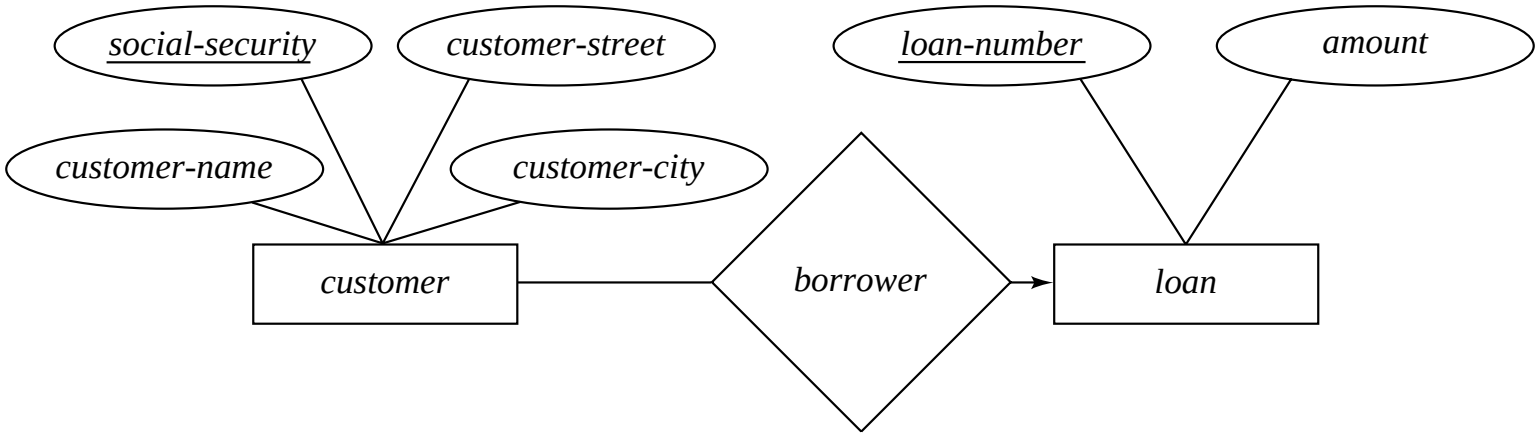


- A customer is associated with at most one loan via the relationship *borrower*
- A loan is associated with at most one customer via *borrower*

One-To-Many and Many-to-One Relationships



(a)

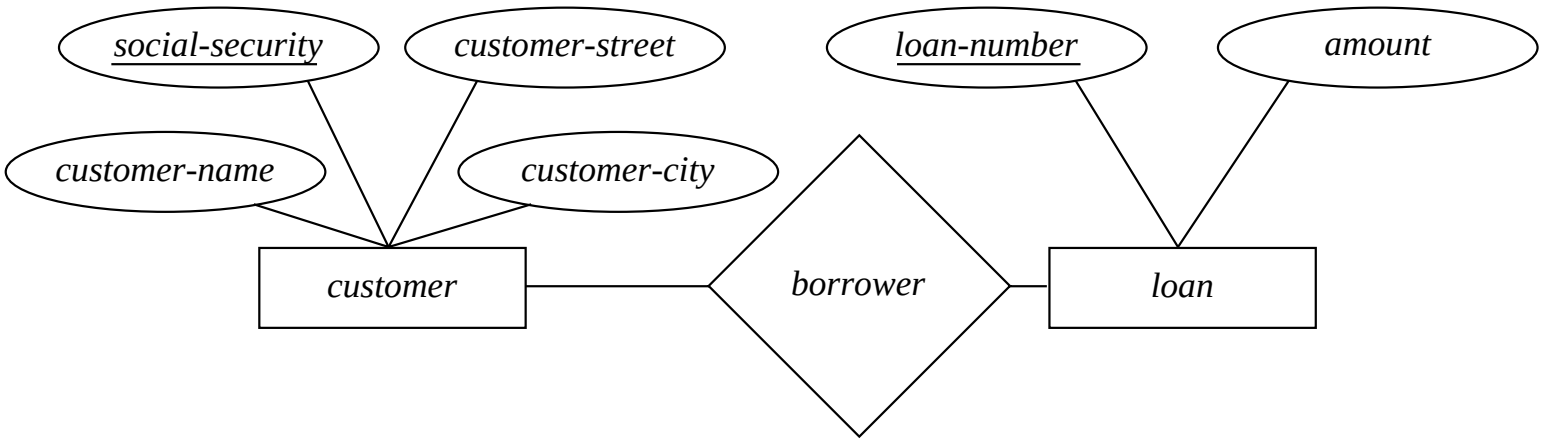


(b)

One-To-Many and Many-to-One (Cont.)

- In the one-to-many relationship (a), a loan is associated with at most one customer via *borrower*; a customer is associated with several (including 0) loans via *borrower*
- In the many-to-one relationship (b), a loan is associated with several (including 0) customers via *borrower*; a customer is associated with at most one loan via *borrower*

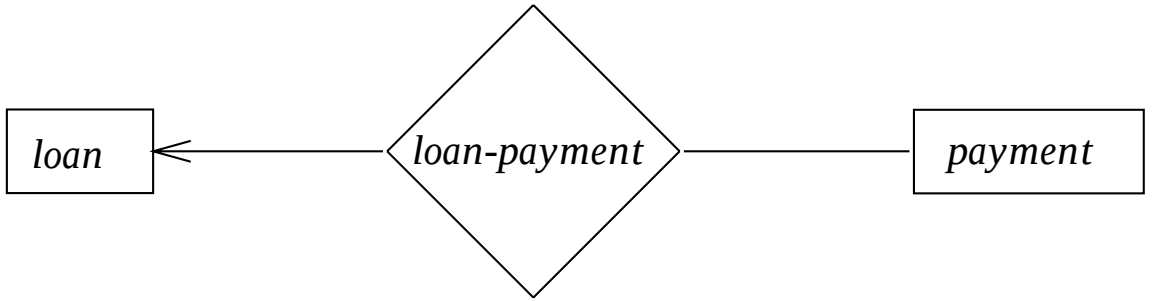
Many-To-Many Relationship



- A customer is associated with several (possibly 0) loans via borrower
- A loan is associated with several (possibly 0) customers via borrower

Existence Dependencies

- If the existence of entity *x* depends on the existence of entity *y*, then *x* is said to be *existence dependent* on *y*.
  - *y* is a *dominant entity* (in example below, *loan*)
  - *x* is a *subordinate entity* (in example below, *payment*)



- If a *loan* entity is deleted, then all its associated *payment* entities must be deleted also.

Keys

- A *super key* of an entity set is a set of one or more attributes whose values uniquely determine each entity
- A *candidate key* of an entity set is a minimal super key
  - *social-security* is candidate key of *customer*
  - *account-number* is candidate key of *account*
- Although several candidate keys may exist, one of the candidate keys is selected to be the *primary key*.
- The combination of primary keys of the participating entity sets forms a candidate key of a relationship set.
  - must consider the mapping cardinality and the semantics of the relationship set when selecting the *primary key*.
  - (*social-security, account-number*) is the primary key of *depositor*

E-R Diagram Components

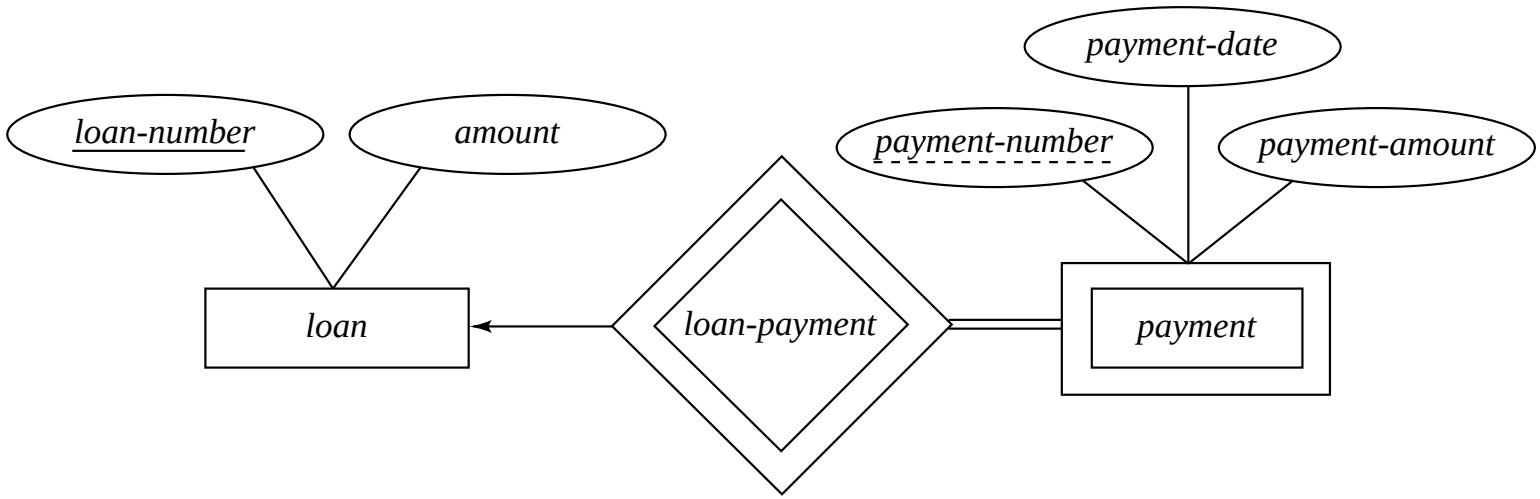
- **Rectangles** represent entity sets.
- **Ellipses** represent attributes.
- **Diamonds** represent relationship sets.
- **Lines** link attributes to entity sets and entity sets to relationship sets.
- **Double ellipses** represent multivalued attributes.
- **Dashed ellipses** denote derived attributes.
- Primary key attributes are underlined.

Weak Entity Sets

- An entity set that does not have a primary key is referred to as a *weak entity set*.
- The existence of a weak entity set depends on the existence of a strong entity set; it must relate to the strong set via a one-to-many relationship set.
- The *discriminator* (or *partial key*) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.
- The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's discriminator.

Weak Entity Sets (Cont.)

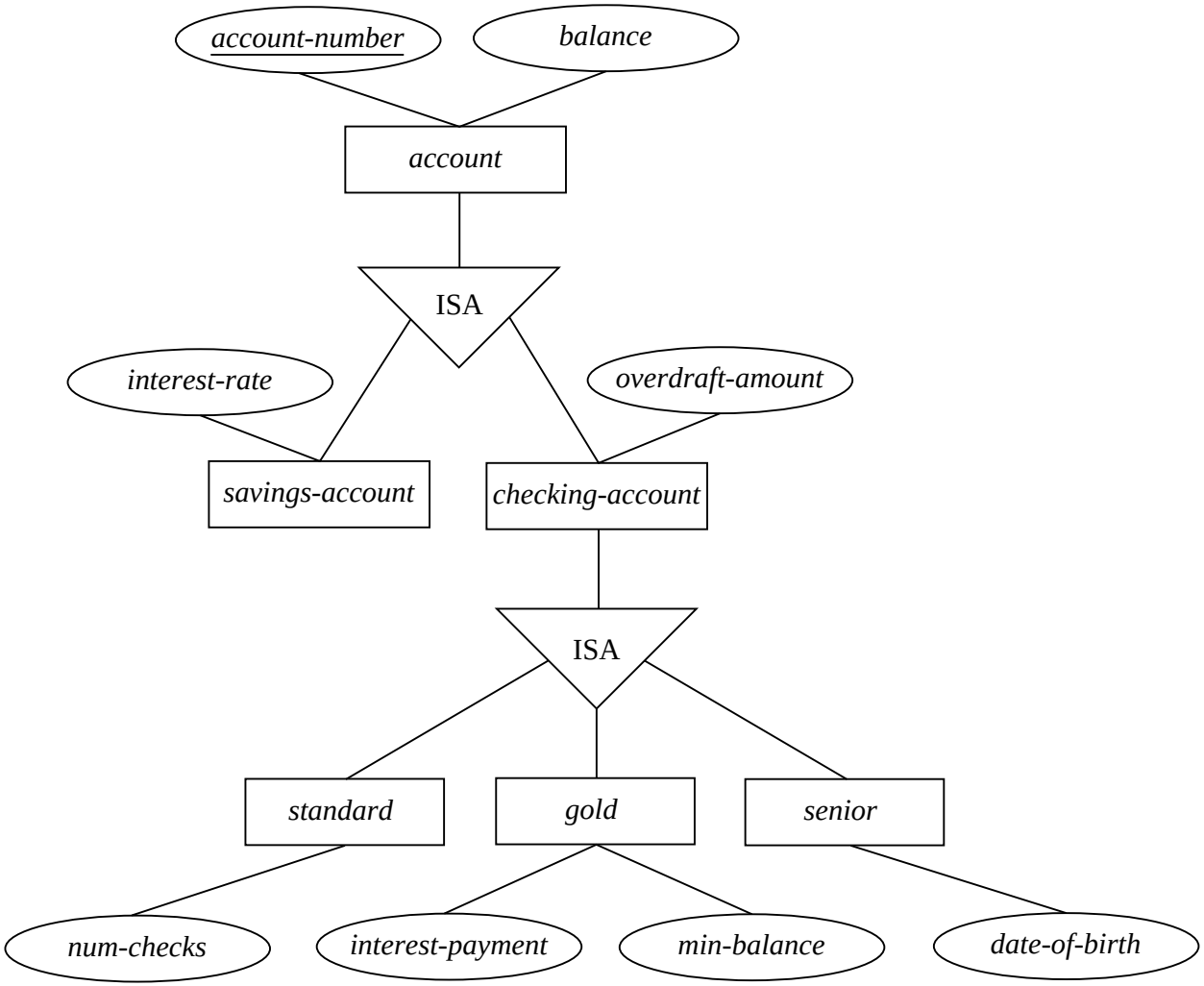
- We depict a weak entity set by double rectangles.
- We underline the discriminator of a weak entity set with a dashed line.
- *payment-number* – discriminator of the *payment* entity set
- Primary key for *payment* – (*loan-number, payment-number*)



Specialization

- Top-down design process; we designate subgroupings within an entity set that are distinctive from other entities in the set.
- These subgroupings become lower-level entity sets that have attributes or participate in relationships that do not apply to the higher-level entity set.
- Depicted by a *triangle* component labeled ISA (i.e., *savings-account* "is an" *account*)

Specialization Example



Generalization

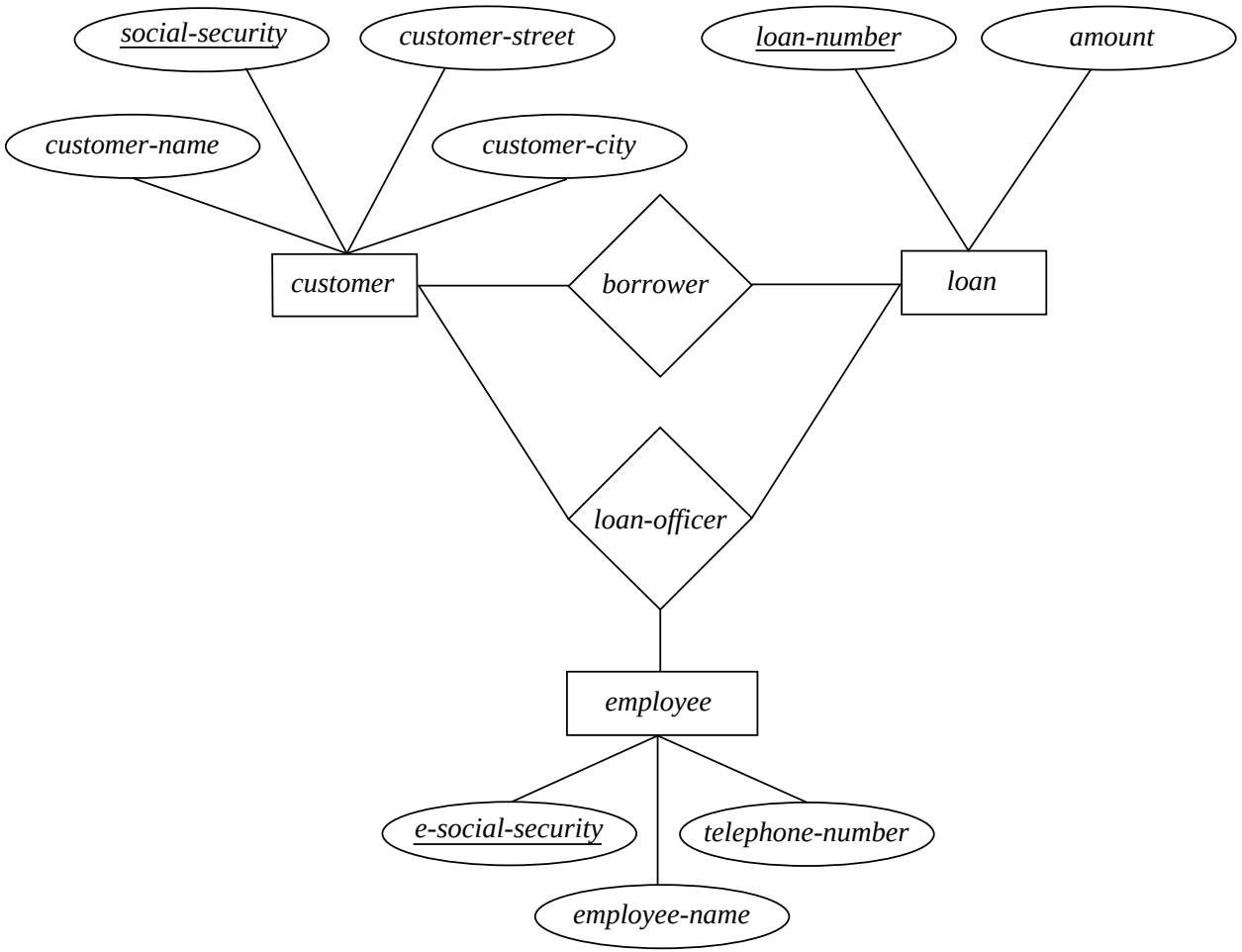
- A bottom-up design process – combine a number of entity sets that share the same features into a higher-level entity set
- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.
- **Attribute Inheritance** – a lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked.

Design Constraints on a Generalization

- Constraint on which entities can be members of a given lower-level entity set.
  - condition-defined
  - user-defined
- Constraint on whether or not entities may belong to more than one lower-level entity set within a single generalization.
  - disjoint
  - overlapping
- Completeness constraint – specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets within a generalization.
  - total
  - partial

Aggregation

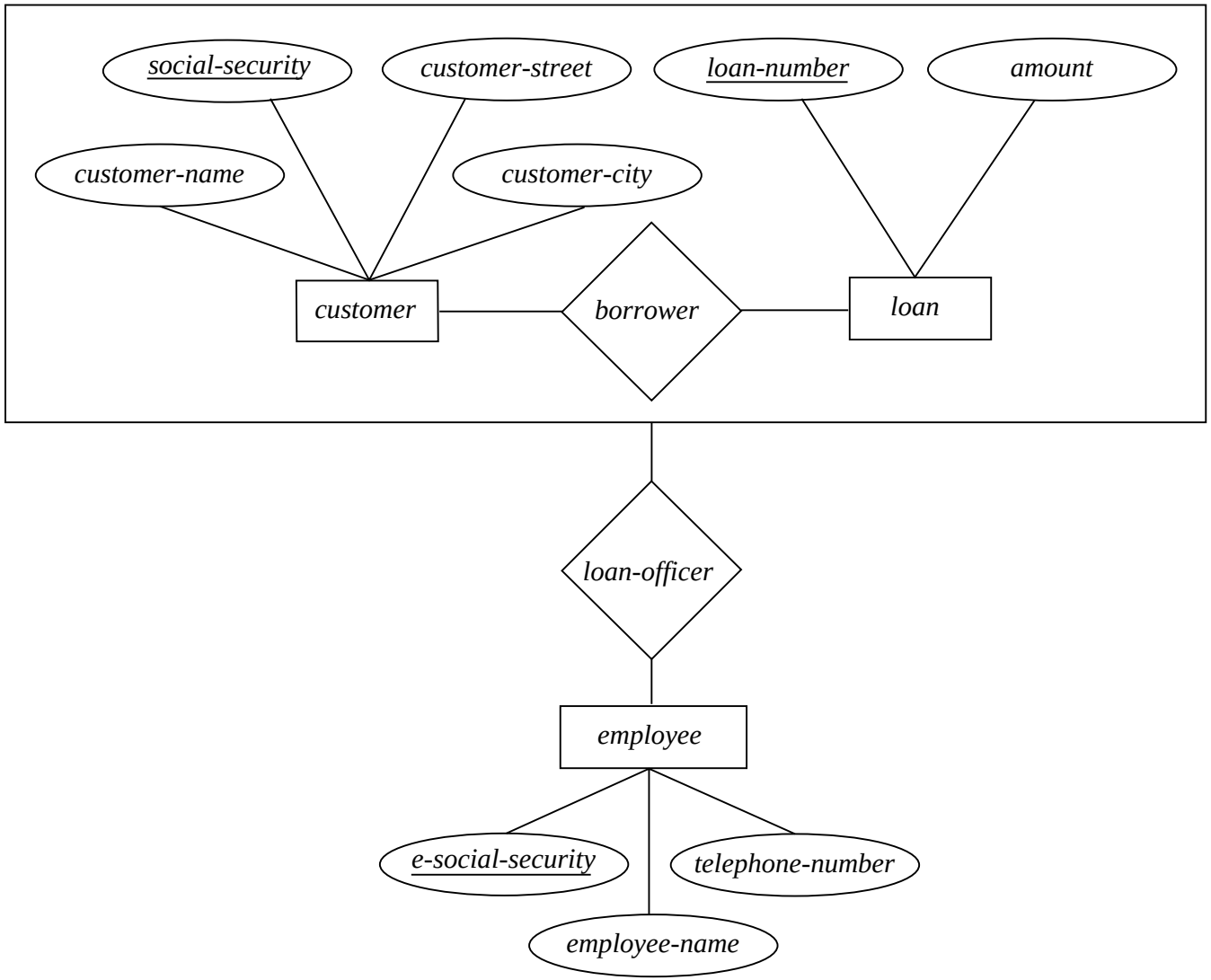
- Loan customers may be advised by a loan-officer.



Aggregation (Cont.)

- Relationship sets *borrower* and *loan-officer* represent the same information
- Eliminate this redundancy via *aggregation*
  - Treat relationship as an abstract entity
  - Allows relationships between relationships
  - Abstraction of relationship into new entity
- Without introducing redundancy, the following diagram represents that:
  - A customer takes out a loan
  - An employee may be a loan officer for a *customer-loan* pair

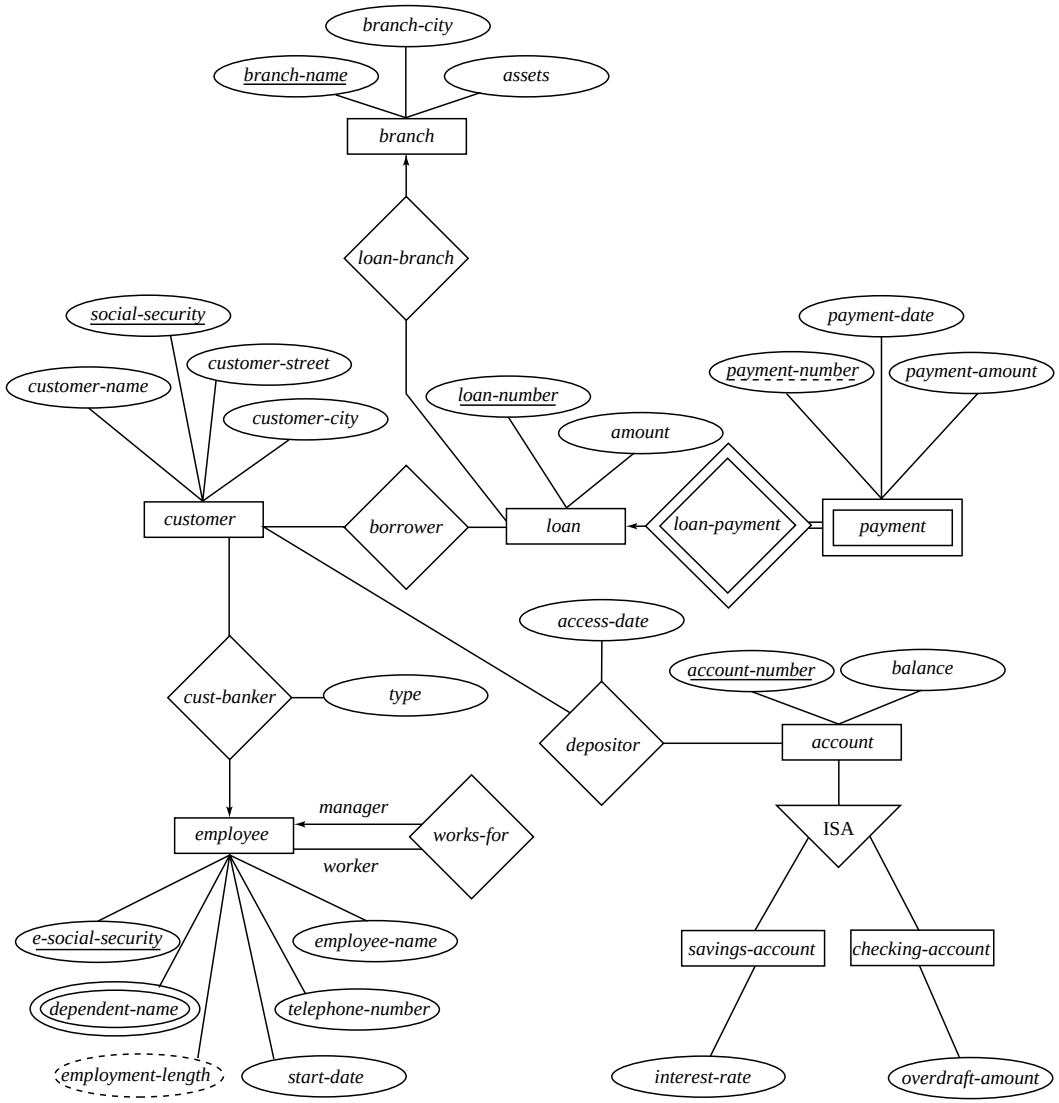
Aggregation Example



E-R Design Decisions

- The use of an attribute or entity set to represent an object.
- Whether a real-world concept is best expressed by an entity set or a relationship set.
- The use of a ternary relationship versus a pair of binary relationships.
- The use of a strong or weak entity set.
- The use of generalization – contributes to modularity in the design.
- The use of aggregation – can treat the aggregate entity set as a single unit without concern for the details of its internal structure.

E-R Diagram for Banking Enterprise



Reduction of an E-R Schema to Tables

- Primary keys allow entity sets and relationship sets to be expressed uniformly as *tables* which represent the contents of the database.
- A database which conforms to an E-R diagram can be represented by a collection of tables.
- For each entity set and relationship set there is a unique table which is assigned the name of the corresponding entity set or relationship set.
- Each table has a number of columns (generally corresponding to attributes), which have unique names.
- Converting an E-R diagram to a table format is the basis for deriving a relational database design from an E-R diagram.

Representing Entity Sets as Tables

- A strong entity set reduces to a table with the same attributes.

customer-name	*social-security*	*c-street*	*c-city*
Jones	321-12-3123	Main	Harrison
Smith	019-28-3746	North	Rye
Hayes	677-89-9011	Main	Harrison

The *customer* table

- A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set.

loan-number	*payment-number*	*payment-date*	*payment-amount*
L-17	5	10 May 1996	50
L-23	11	17 May 1996	75
L-15	22	23 May 1996	300

The *payment* table

Representing Relationship Sets as Tables

- A many-to-many relationship set is represented as a table with columns for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.

social-security	*account-number*	*access-date*
...	...	...

The *depositor* table

- The table corresponding to a relationship set linking a weak entity set to its identifying strong entity set is redundant.

  The *payment* table already contains the information that would appear in the *loan-payment* table (i.e., the columns *loan-number* and *payment-number*).

Representing Generalization as Tables

- Method 1: Form a table for the generalized entity *account*
  Form a table for each entity set that is generalized (include primary key of generalized entity set)

table	table attributes
account	*account-number, balance, account-type*
savings-account	*account-number, interest-rate*
checking-account	*account-number, overdraft-amount*

- Method 2: Form a table for each entity set that is generalized

table	table attributes
savings-account	*account-number, balance, interest-rate*
checking-account	*account-number, balance, overdraft-amount*

Method 2 has no table for generalized entity *account*

Relations Corresponding to Aggregation

customer

customer-name	*cust-social-security*	*customer-street*	*customer-city*

loan

loan-number	*amount*

borrower

cust-social-security	*loan-number*

employee

emp-social-security	*employee-name*	*phone-number*

loan-officer

emp-social-security	*cust-social-security*	*loan-number*

Introduction of RDBMS

1. What is RDBMS

RDBMS is a database management system based on relational model defined by E.F.Codd. Data is stored in the form of rows and columns. The relations among tables are also stored in the form of the table.

2. Differentiate between DBMS and RDBMS?

Difference between DBMS vs RDBMS

Parameter	**DBMS**	**RDBMS**
Storage	DBMS stores data as a file.	Data is stored in the form of tables.
Database structure	DBMS system, stores data in either a navigational or hierarchical form.	RDBMS uses a tabular structure where the headers are the column names, and the rows contain corresponding values
Number of Users	DBMS supports single user only.	It supports multiple users.
ACID	In a regular database, the data may not be stored following the ACID model. This can develop inconsistencies in the database.	Relational databases are harder to construct, but they are consistent and well structured. They obey ACID (Atomicity, Consistency, Isolation, Durability).
Type of program	It is the program for managing the databases on the computer networks and the system hard disks.	It is the database systems which are used for maintaining the relationships among the tables.
Hardware and	Low software and hardware	Higher hardware and software

Parameter	DBMS	RDBMS
software needs.	needs.	need.
Integrity constraints	DBMS does not support the integrity constants. The integrity constants are not imposed at the file level.	RDBMS supports the integrity constraints at the schema level. Values beyond a defined range cannot be stored into the particular RDMS column.
Normalization	DBMS does not support Normalization	RDBMS can be Normalized.
Distributed Databases	DBMS does not support distributed database.	RBMS offers support for distributed databases.
Ideally suited for	DBMS system mainly deals with small quantity of data.	RDMS is designed to handle a large amount of data.
Dr. E.F. Codd Rules	Dbms satisfy less than seven of Dr. E.F. Codd Rules	Dbms satisfy 8 to 10 Dr. E.F. Codd Rules
Client Server	DBMS does not support client-server architecture	RDBMS supports client-server architecture.
Data Fetching	Data fetching is slower for the complex and large amount of data.	Data fetching is rapid because of its relational approach.
Data Redundancy	Data redundancy is common in this model.	Keys and indexes do not allow Data redundancy.
Data Relationship	No relationship between data	Data is stored in the form of tables which are related to each other with the help of foreign keys.

Parameter	DBMS	RDBMS
Security	There is no security.	Multiple levels of security. Log files are created at OS, Command, and object level.
Data Access	Data elements need to access individually.	Data can be easily accessed using SQL query. Multiple data elements can be accessed at the same time.
Examples	Examples of DBMS are a file system, XML, Windows Registry, etc.	Example of RDBMS is MySQL, Oracle, SQL Server, etc.

3. Explain four objects of Oracle .

- Relation
  - Table - A table has a specified number of columns, but can have any number of rows.
  - Column –n a relational database, a column is a set of data values of a particular simple type, one value for each row of the database.
  - Row - In relational database, a row—also called a tuple—represents a single, implicitly structured data item in a table.
- View - A database view is a searchable object in a database that is defined by a query. Though a view doesn't store data, some refer to a views as “virtual tables,” you can query a view like you can a table. A view can combine data from two or more table, using joins, and also just contain a subset of information.
- Transaction - A *transaction* is a single logical unit of work which accesses and possibly modifies the contents of a database.
- Transaction log - In the field of databases in computer science, a transaction log is a history of actions executed by a database

management system used to guarantee ACID properties over crashes or hardware failures.

- Trigger – a **trigger** is a **SQL** procedure that initiates an action (i.e., fires an action) when an event (INSERT, DELETE or UPDATE) occurs. Since **triggers** are event-driven specialized procedures, they are stored in and managed by the **DBMS**. ... Each **trigger** is attached to a single, specified table in the database.

- Index - An **index** is a copy of selected columns of data from a table that can be searched very efficiently that also includes a low-level disk block address or direct link to the complete row of data it was copied from.
- Stored Procedure - A **stored procedure** is a set of Structured Query Language (**SQL**) statements with an assigned name, which are **stored** in a **relational database** management system as a group, so it can be reused and shared by multiple programs.
- Cursor - A **cursor** is a temporary work area created in the system memory when a **SQL** statement is executed. A **cursor** contains information on a select statement and the rows of data accessed by it. This temporary work area is used to store the data retrieved from the database, and manipulate this data.
- Partition - **Partitioning** a stored **procedure means** that the **procedure** executes within a unique **partition** of the database.

4. Write characteristic of RDBMS.

- Data abstraction
- Self-describing data
- Concurrency
- Support for multiple views
- Security
- The use of keys. Each row of data in a table is identified by a unique "key", called the primary key.
- Avoiding data redundancy.
- Constraining the input.

- Maintaining data integrity.
- Rights.
- Structured Query Language (SQL) .
- Portability.

5. Explain features of Oracle.

- Scalability and Performance feature.
    - Concurrency.
    - Read Consistency
    - Locking Mechanism
    - Quiescence database
    - Real Application cluster
    - Portability
- Manageability Features
    - Self-Managing database
    - Oracle Enterprise Manager
    - Automatic Storage Management
    - Scheduler
    - Database Resource Manager
- Database Backup and recovery Failure .
- High availability feature .
- Business Intelligence features
    - Data Warehousing
    - Extraction , Transformation and loading
    - Data mining
- Content Management Features
    - XML in Oracle
    - LOBs
    - Oracle Text
    - Oracle Spatial
- Security Features
    - System Security
    - Data security
    -

6. What is RDBMS ? Explain features of RDBMS.

RDBMS is a database management system based on relational model defined by E.F.Codd. Data is stored in the form of rows and columns. The relations among tables are also stored in the form of the table.

Features:

- Provides data to be stored in tables
- Persists data in the form of rows and columns
- Provides facility primary key, to uniquely identify the rows
- Creates indexes for quicker data retrieval
- Provides a virtual table creation in which sensitive data can be stored and simplified query can be applied.(views)
- Sharing a common column in two or more tables(primary key and foreign key)
- Provides multi user accessibility that can be controlled by individual users.Explain Product of RDBMS.

7. Explain advantages and disadvantages of RDBMS.

Benefits or advantages of RDBMS

Following are the benefits or **advantages of RDBMS**:

➡It is easy to use.
➡It is secured in nature.
➡The data manipulation can be done.
➡It limits redundancy and replication of the data.
➡It offers better data integrity.
➡It provides better physical data independence.
➡It offers logical database independence i.e. data can be viewed in different ways by the different users.
➡It provides better backup and recovery procedures.
➡It provides multiple interfaces.
➡Multiple users can access the database which is not possible in DBMS.

Drawbacks or disadvantages of RDBMS

Following are the **disadvantages of RDBMS**:
➡Software is expensive.
➡Complex software refers to expensive hardware and hence increases overall cost to avail the RDBMS service.
➡It requires skilled human resources to implement.
➡Certain applications are slow in processing.
➡It is difficult to recover the lost data.

8. Explain Products od RDBMS .

- Oracle

  Oracle database (**Oracle** DB) is a **relational database** management system (**RDBMS**) from the **Oracle** Corporation. ... The system is built around a **relational database** framework in which data objects may be directly accessed by users (or an application front end) through structured query language (**SQL**).

- MS Access

  Microsoft Access is a database management system (**DBMS**) from **Microsoft** that combines the relational **Microsoft** Jet Database Engine with a graphical user interface and software-development tools.It is a **Relational Database** Management System (**RDBMS**) that allows a user to store database in one or more tables.

- SQL Server

  SQL Server is Microsoft's **relational database** management system (**RDBMS**). It is a full-featured database primarily designed to compete against competitors Oracle Database (DB) and MySQL. ... **SQL Server** is sometimes referred to as **MSSQL** and Microsoft **SQL Server**.

- MySQL

  MySQL is an open-source **relational database** management system (**RDBMS**). Its name is a combination of "My", the

name of co-founder Michael Widenius's daughter, and "**SQL**", the abbreviation for Structured Query Language.

- DB2

  DB2 is a database product from IBM. It is a **Relational Database** Management System (**RDBMS**). **DB2** is designed to store, analyze and retrieve the data efficiently. **DB2** product is extended with the support of Object-Oriented features and non-relational structures with XML.

- PostgreSQL

  PostgreSQL, also known as **Postgres**, is a free and open-source **relational database** management system (**RDBMS**) emphasizing extensibility and technical standards compliance. It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users.

Relational Model in DBMS

Relational Model was proposed by E.F. Codd to model data in the form of relations or tables. After designing the conceptual model of Database using ER diagram, we need to convert the conceptual model in the relational model which can be implemented using any RDBMS languages like Oracle SQL, MySQL etc. So we will see what Relational Model is.

What is Relational Model?

Relational Model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables). Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE and AGE shown in Table 1.

STUDENT

ROLL_NO	NAME	ADDRESS	PHONE	AGE
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI		18

Attribute: Attributes are the properties that define a relation. e.g.; ROLL_NO, NAME

Relation Schema: A relation schema represents name of the relation with its attributes. e.g.; STUDENT (ROLL_NO, NAME, ADDRESS, PHONE and AGE) is relation schema for STUDENT. If a schema has more than 1 relation, it is called Relational Schema.

Tuple: Each row in the relation is known as tuple. The above relation contains 4 tuples, one of which is shown as:

1 RAM DELHI 945512345 18

Relation Instance: The set of tuples of a relation at a particular instance of time is called as relation instance. Table 1 shows the relation instance of STUDENT at a particular time. It can change whenever there is insertion, deletion or updation in the database.

Degree: The number of attributes in the relation is known as degree of the relation. The STUDENT relation defined above has degree 5.

Cardinality: The number of tuples in a relation is known as cardinality. The STUDENT relation defined above has cardinality 4.

Column: Column represents the set of values for a particular attribute. The column ROLL_NO is extracted from relation STUDENT.

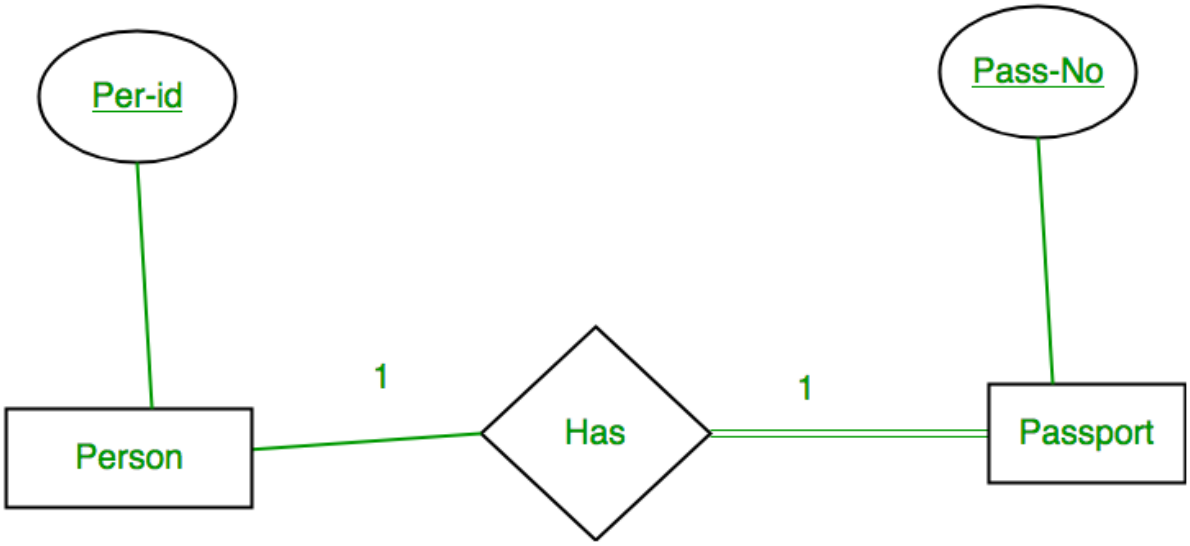
ROLL_NO
1
2
3
4

NULL Values: The value which is not known or unavailable is called NULL value. It is represented by blank space. e.g.; PHONE of STUDENT having ROLL_NO 4 is NULL.

Mapping from ER Model to Relational Model

After designing the ER diagram of system, we need to convert it to Relational models which can directly be implemented by any RDBMS like Oracle, MySQL etc. In this article we will discuss how to convert ER diagram to Relational Model for different scenarios.

Binary Relationship with 1:1 cardinality with total participation of an entity



A person has 0 or 1 passport number and Passport is always owned by 1 person. So it is 1:1 cardinality with full participation constraint from Passport.

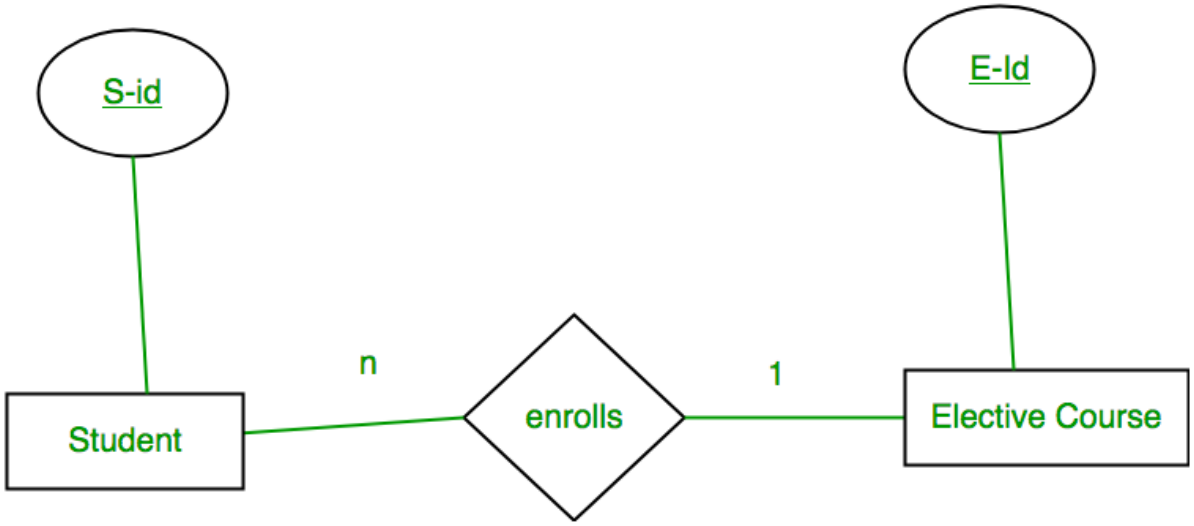
First Convert each entity and relationship to tables. Person table corresponds to Person Entity with key as Per-Id. Similarly Passport table corresponds to Passport Entity with key as Pass-No. Has Table represents relationship between Person and Passport (Which person has which passport). So it will take attribute Per-Id from Person and Pass-No from Passport.

Person			Has			Passport	
Per-Id	**Other Person Attribute**		**Per-Id**	**Pass-No**		**Pass-No**	**Other Passport Attribute**
PR1	–		PR1	PS1		PS1	–
PR2	–		PR2	PS2		PS2	–
PR3	–						

As we can see from Table 1, each Per-Id and Pass-No has only one entry in Has table. So we can merge all three tables into 1 with attributes shown in Table 2. Each Per-Id will be unique and not null. So it will be the key. Pass-No can't be key because for some person, it can be NULL.

Per-Id Other Person Attribute Pass-No Other PassportAttribute

Binary Relationship with n: 1 cardinality



In this scenario, every student can enroll only in one elective course but for an elective course there can be more than one student. First Convert each entity and relationship to tables. Student table corresponds to Student Entity with key as S-Id. Similarly Elective_Course table corresponds to Elective_Course Entity with key as E-Id. Enrolls Table represents relationship between Student and Elective_Course (Which student enrolls in which course). So it will take attribute S-Id from and Student E-Id from Elective_Course.

Student			**Enrolls**			**Elective_Course**	
S-Id	Other Student Attribute		S-Id	E-Id		E-Id	Other Elective CourseAttribute
S1	–		S1	E1		E1	–

S2	–		S2	E2		E2	–
S3	–		S3	E1		E3	–
S4	–		S4	E1			

As we can see from ABOVE Table , S-Id is not repeating in Enrolls Table. So it can be considered as a key of Enrolls table. Both Student and Enrolls Table's key is same; we can merge it as a single table. The resultant tables are shown in Table 7 and Table 8. Primary Keys have been underlined.

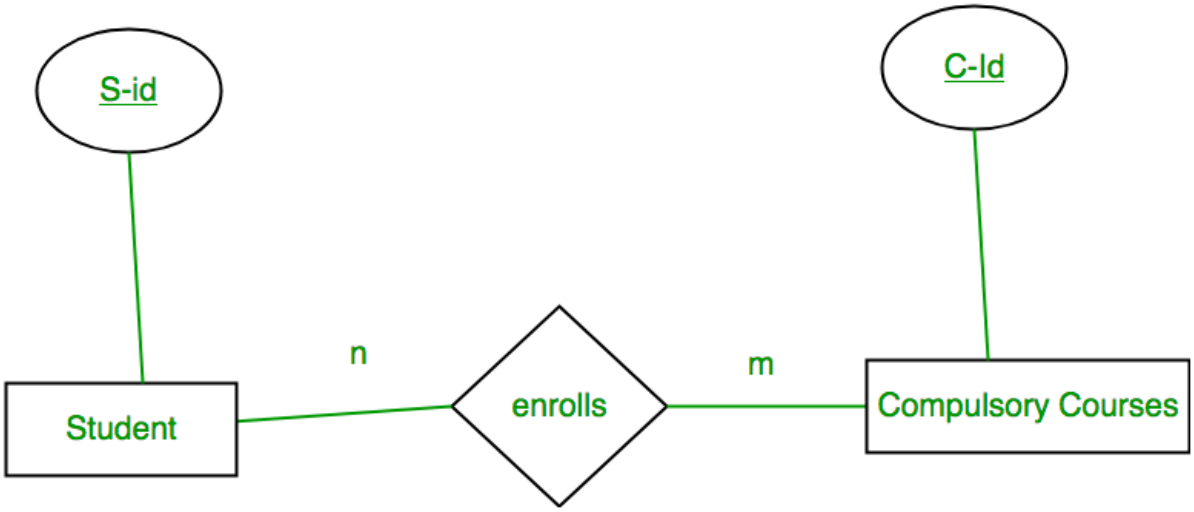
S-Id	Other Student Attribute	E-Id

Table 7

E-Id	Other Elective CourseAttribute

Table 8

Binary Relationship with m: n cardinality



In this scenario, every student can enroll in more than 1 compulsory course and for a compulsory course there can be more than 1 student. First Convert each entity and relationship to tables. Student table corresponds to Student Entity with key as S-Id. Similarly Compulsory_Courses table corresponds to Compulsory Courses Entity with key as C-Id. Enrolls Table represents relationship between Student and Compulsory_Courses (Which student enrolls in which course). So it will take attribute S-Id from Person and C-Id from Compulsory_Courses.

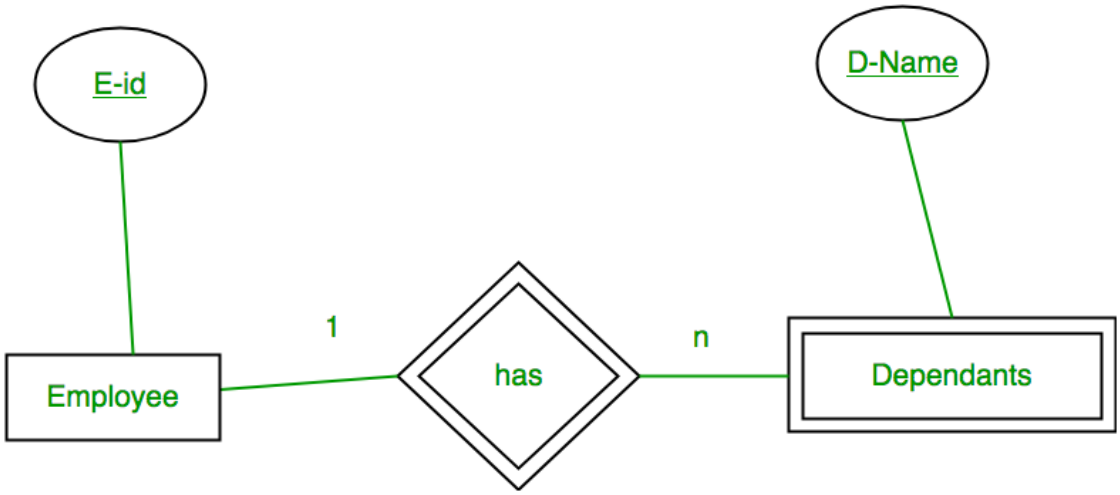
Student			**Enrolls**			**Compulsory_Courses**	
S-Id	Other Student Attribute		S-Id	C-Id		C-Id	Other Compulsory CourseAttribute
S1	–		S1	C1		C1	–
S2	–		S1	C2		C2	–
S3	–		S3	C1		C3	–
S4	–		S4	C3		C4	–

			S4	C2			
			S3	C3			

Table 9

As we can see from Table 9, S-Id and C-Id both are repeating in Enrolls Table. But its combination is unique; so it can be considered as a key of Enrolls table. All tables' keys are different, these can't be merged. Primary Keys of all tables have been underlined.

Binary Relationship with weak entity



In this scenario, an employee can have many dependants and one dependant can depend on one employee. A dependant does not have any existence without an employee (e.g; you as a child can be dependant of your father in his company). So it will be a weak entity and its participation will always be total. Weak Entity does not have key of its own. So its key will be combination of key of its identifying entity (E-Id of Employee in this case) and its partial key (D-Name).

First Convert each entity and relationship to tables. Employee table corresponds to Employee Entity with key as E-Id. Similarly Dependants table corresponds to Dependant Entity with key as D-Name and E-Id. Has Table represents relationship between Employee and Dependants (Which employee has which dependants). So it will take attribute E-Id from Employee and D-Name from Dependants.

Employee			**Has**			**Dependants**		
E-Id	Other Employee Attribute		E-Id	D-Name		D-Name	E-Id	Other Dependants Attribute
E1	–		E1	RAM		RAM	E1	–
E2	–		E1	SRINI		SRINI	E1	–
E3	–		E2	RAM		RAM	E2	–
			E3	ASHISH		ASHISH	E3	–

Table 10

As we can see from Table 10, E-Id, D-Name is key for **Has** as well as Dependants Table. So we can merge these two into 1. So the resultant tables are shown in Tables 11 and 12. Primary Keys of all tables have been underlined.

E-Id	Other Employee Attribute

Table 11

D-Name	E-Id	Other DependantsAttribute

Table 12

Chapter 3: Relational Model

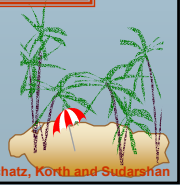
- Structure of Relational Databases
- Relational Algebra
- Tuple Relational Calculus
- Domain Relational Calculus
- Extended Relational-Algebra-Operations
- Modification of the Database
- Views



©Silberschatz, Korth and Sudarshan

Example of a Relation

account-number	*branch-name*	*balance*
A-101	Downtown	500
A-102	Perryridge	400
A-201	Brighton	900
A-215	Mianus	700
A-217	Brighton	750
A-222	Redwood	700
A-305	Round Hill	350



©Silberschatz, Korth and Sudarshan

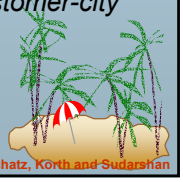
Basic Structure

- Formally, given sets D_1, D_2, …. D_n a **relation** r is a subset of D_1 x D_2 x … x D_n
  Thus a relation is a set of n-tuples (a_1, a_2, …, a_n) where each $a_i \in D_i$
- Example: if

  customer-name = {Jones, Smith, Curry, Lindsay}
  customer-street = {Main, North, Park}
  customer-city = {Harrison, Rye, Pittsfield}

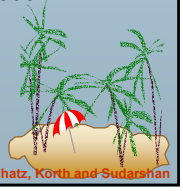
  Then r = { (Jones, Main, Harrison),
  (Smith, North, Rye),
  (Curry, North, Rye),
  (Lindsay, Park, Pittsfield)}

  is a relation over *customer-name x customer-street x customer-city*



Attribute Types

- Each attribute of a relation has a name
- The set of allowed values for each attribute is called the **domain** of the attribute
- Attribute values are (normally) required to be **atomic**, that is, indivisible
  - E.g. multivalued attribute values are not atomic
  - E.g. composite attribute values are not atomic
- The special value *null* is a member of every domain
- The null value causes complications in the definition of many operations
  - we shall ignore the effect of null values in our main presentation and consider their effect later



Relation Schema

- $A_1, A_2, \ldots, A_n$ are *attributes*
- $R = (A_1, A_2, \ldots, A_n)$ is a *relation schema*

  E.g. *Customer-schema* = (*customer-name, customer-street, customer-city*)
- $r(R)$ is a *relation* on the *relation schema R*

  E.g. *customer (Customer-schema)*



Relation Instance

- The current values (*relation instance*) of a relation are specified by a table
- An element *t* of *r* is a *tuple*, represented by a *row* in a table

attributes (or columns)

customer-name	*customer-street*	customer-city
Jones	Main	Harrison
Smith	North	Rye
Curry	North	Rye
Lindsay	Park	Pittsfield

tuples (or rows)

customer



Relations are Unordered

- Order of tuples is irrelevant (tuples may be stored in an arbitrary order)
- E.g. *account* relation with unordered tuples

account-number	*branch-name*	*balance*
A-101	Downtown	500
A-215	Mianus	700
A-102	Perryridge	400
A-305	Round Hill	350
A-201	Brighton	900
A-222	Redwood	700
A-217	Brighton	750



©Silberschatz, Korth and Sudarshan

Database

- A database consists of multiple relations
- Information about an enterprise is broken up into parts, with each relation storing one part of the information

  E.g.: *account* : stores information about accounts
  depositor : stores information about which customer owns which account
  customer : stores information about customers

- Storing all information as a single relation such as
  bank(*account-number, balance, customer-name*, ..)
  results in
  - repetition of information (e.g. two customers own an account)
  - the need for null values (e.g. represent a customer without an account)
- Normalization theory (Chapter 7) deals with how to design relational schemas



©Silberschatz, Korth and Sudarshan

The *customer* Relation

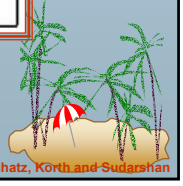
customer-name	*customer-street*	*customer-city*
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton



©Silberschatz, Korth and Sudarshan

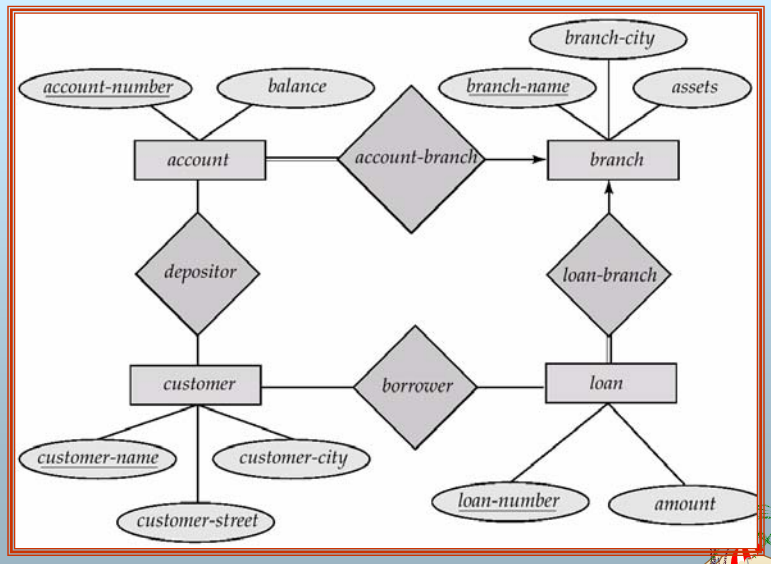
The *depositor* Relation

customer-name	*account-number*
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305



©Silberschatz, Korth and Sudarshan

E-R Diagram for the Banking Enterprise



Keys

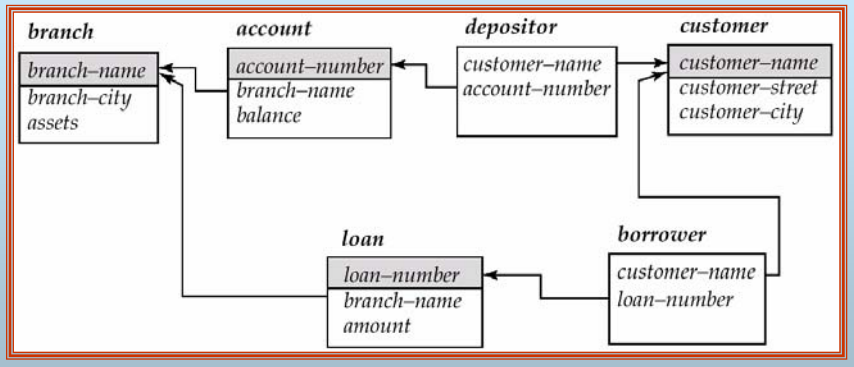
- Let K $\subseteq$ R
- *K* is a ***superkey*** of *R* if values for *K* are sufficient to identify a unique tuple of each possible relation *r(R)*
  - by "possible *r*" we mean a relation *r* that could exist in the enterprise we are modeling.
  - Example: {*customer-name, customer-street*} and {*customer-name*}
    are both superkeys of *Customer*, if no two customers can possibly have the same name.
- *K* is a ***candidate key*** if *K* is minimal
  Example: {*customer-name*} is a candidate key for *Customer*, since it is a superkey (assuming no two customers can possibly have the same name), and no subset of it is a superkey.

Determining Keys from E-R Sets

- **Strong entity set**. The primary key of the entity set becomes the primary key of the relation.
- **Weak entity set**. The primary key of the relation consists of the union of the primary key of the strong entity set and the discriminator of the weak entity set.
- **Relationship set**. The union of the primary keys of the related entity sets becomes a super key of the relation.
  - For binary many-to-one relationship sets, the primary key of the "many" entity set becomes the relation's primary key.
  - For one-to-one relationship sets, the relation's primary key can be that of either entity set.
  - For many-to-many relationship sets, the union of the primary keys becomes the relation's primary key

©Silberschatz, Korth and Sudarshan

Schema Diagram for the Banking Enterprise



©Silberschatz, Korth and Sudarshan

Query Languages

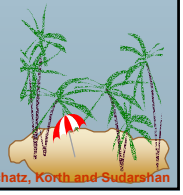
- Language in which user requests information from the database.
- Categories of languages
  - procedural
  - non-procedural
- "Pure" languages:
  - Relational Algebra
  - Tuple Relational Calculus
  - Domain Relational Calculus
- Pure languages form underlying basis of query languages that people use.



©Silberschatz, Korth and Sudarshan

Relational Algebra

- Procedural language
- Six basic operators
  - select
  - project
  - union
  - set difference
  - Cartesian product
  - rename
- The operators take one or more relations as inputs and give a new relation as a result.



©Silberschatz, Korth and Sudarshan

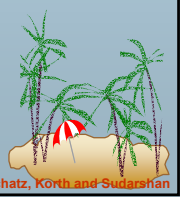
Select Operation – Example

- Relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

- $\sigma_{A=B \wedge D > 5}(r)$

A	B	C	D
α	α	1	7
β	β	23	10



©Silberschatz, Korth and Sudarshan

Select Operation

- Notation: $\sigma_p(r)$
- p is called the selection predicate
- Defined as:

$$\sigma_p(\mathbf{r}) = \{t \mid t \in r \textbf{ and } p(t)\}$$

Where p is a formula in propositional calculus consisting of terms connected by : $\wedge$ (**and**), $\vee$ (**or**), $\neg$ (**not**)
Each term is one of:

<attribute> *op* <attribute> or <constant>

where *op* is one of: =, $\neq$, >, $\geq$. <. $\leq$

- Example of selection:

$\sigma_{\textit{branch-name}=\text{“Perryridge”}}(\textit{account})$



©Silberschatz, Korth and Sudarshan

Project Operation – Example

- Relation *r*:

A	*B*	*C*
α	10	1
α	20	1
β	30	1
β	40	2

- $\Pi_{A,C}$ (*r*)

A	*C*
α	1
α	1
β	1
β	2

=

A	*C*
α	1
β	1
β	2



Project Operation

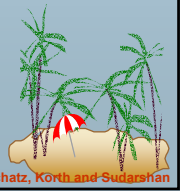
- Notation:

$$\Pi_{A1, A2, \ldots, Ak}\ (r)$$

where A_1, A_2 are attribute names and *r* is a relation name.

- The result is defined as the relation of *k* columns obtained by erasing the columns that are not listed
- Duplicate rows removed from result, since relations are sets
- E.g. To eliminate the *branch-name* attribute of *account*

$$\Pi_{account\text{-}number,\ balance}\ (account)$$



Union Operation – Example

- Relations *r, s:*

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

r ∪ s:

A	B
α	1
α	2
β	1
β	3



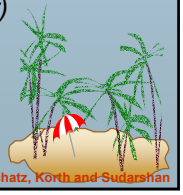
Union Operation

- Notation: $r \cup s$
- Defined as:

$$r \cup s = \{t \mid t \in r \text{ or } t \in s\}$$

- For $r \cup s$ to be valid.
  1. *r, s* must have the *same arity* (same number of attributes)
  2. The attribute domains must be *compatible* (e.g., 2nd column of *r* deals with the same type of values as does the 2nd column of *s*)
- E.g. to find all customers with either an account or a loan

$$\Pi_{customer\text{-}name}(depositor) \cup \Pi_{customer\text{-}name}(borrower)$$



Set Difference Operation – Example

- Relations *r, s:*

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

r – s:

A	B
α	1
β	1

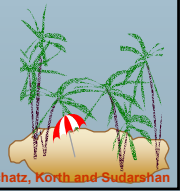


Set Difference Operation

- Notation $r - s$
- Defined as:

$$r - s = \{t \mid t \in r \textbf{ and } t \notin s\}$$

- Set differences must be taken between *compatible* relations.
  - *r* and *s* must have the *same arity*
  - attribute domains of *r* and *s* must be compatible



Cartesian-Product Operation-Example

Relations *r, s*:

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

r x *s*:

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b



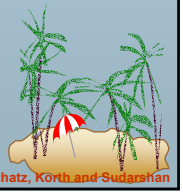
©Silberschatz, Korth and Sudarshan

Cartesian-Product Operation

- Notation *r* x *s*
- Defined as:

$$r \times s = \{t\, q \mid t \in r \textbf{ and } q \in s\}$$

- Assume that attributes of r(R) and s(S) are disjoint. (That is, $R \cap S = \varnothing$).
- If attributes of *r(R)* and *s(S)* are not disjoint, then renaming must be used.



©Silberschatz, Korth and Sudarshan

Composition of Operations

- Can build expressions using multiple operations
- Example: $\sigma_{A=C}(r \times s)$
- $r \times s$

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

- $\sigma_{A=C}(r \times s)$

A	B	C	D	E
α	1	α	10	a
β	2	β	20	a
β	2	β	20	b



©Silberschatz, Korth and Sudarshan

Rename Operation

- Allows us to name, and therefore to refer to, the results of relational-algebra expressions.
- Allows us to refer to a relation by more than one name.

Example:

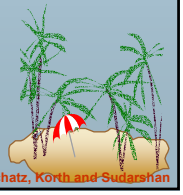
$$\rho_X(E)$$

returns the expression E under the name X

If a relational-algebra expression E has arity n, then

$$\rho_{X(A1, A2, \ldots, An)}(E)$$

returns the result of expression E under the name X, and with the attributes renamed to $A1, A2, \ldots, An$.



©Silberschatz, Korth and Sudarshan

Banking Example

branch (branch-name, branch-city, assets)

customer (customer-name, customer-street, customer-only)

account (account-number, branch-name, balance)

loan (loan-number, branch-name, amount)

depositor (customer-name, account-number)

borrower (customer-name, loan-number)



©Silberschatz, Korth and Sudarshan

Example Queries

- Find all loans of over $1200

$$\sigma_{amount > 1200}(loan)$$

- Find the loan number for each loan of an amount greater than $1200

$$\Pi_{loan\text{-}number}(\sigma_{amount > 1200}(loan))$$



©Silberschatz, Korth and Sudarshan

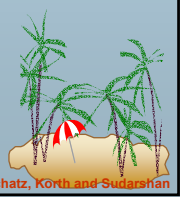
Example Queries

- Find the names of all customers who have a loan, an account, or both, from the bank

$$\Pi_{customer\text{-}name}(borrower) \cup \Pi_{customer\text{-}name}(depositor)$$

- Find the names of all customers who have a loan and an account at bank.

$$\Pi_{customer\text{-}name}(borrower) \cap \Pi_{customer\text{-}name}(depositor)$$



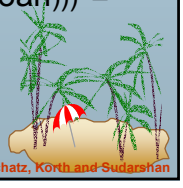
Example Queries

- Find the names of all customers who have a loan at the Perryridge branch.

$$\Pi_{customer\text{-}name}(\sigma_{branch\text{-}name=\text{“Perryridge”}}(\sigma_{borrower.loan\text{-}number = loan.loan\text{-}number}(borrower \times loan)))$$

- Find the names of all customers who have a loan at the Perryridge branch but do not have an account at any branch of the bank.

$$\Pi_{customer\text{-}name}(\sigma_{branch\text{-}name = \text{“Perryridge”}}(\sigma_{borrower.loan\text{-}number = loan.loan\text{-}number}(\text{borrower} \times \text{loan}))) - \Pi_{customer\text{-}name}(\text{depositor})$$



Example Queries

- Find the names of all customers who have a loan at the Perryridge branch.
  - Query 1

$$\Pi_{\text{customer-name}}(\sigma_{\text{branch-name} = \text{“Perryridge”}} (\sigma_{\text{borrower.loan-number} = \text{loan.loan-number}}(\text{borrower} \times \text{loan})))$$

  - Query 2

$$\Pi_{\text{customer-name}}(\sigma_{\text{loan.loan-number} = \text{borrower.loan-number}}((\sigma_{\text{branch-name} = \text{“Perryridge”}}(\text{loan})) \times \text{borrower}))$$

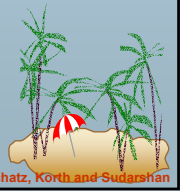


Example Queries

Find the largest account balance

- Rename *account* relation as *d*
- The query is:

$$\Pi_{balance}(account) - \Pi_{account.balance} (\sigma_{account.balance < d.balance} (account \times \rho_{d} (account)))$$



Formal Definition

- A basic expression in the relational algebra consists of either one of the following:
  - A relation in the database
  - A constant relation
- Let E_1 and E_2 be relational-algebra expressions; the following are all relational-algebra expressions:
  - $E_1 \cup E_2$
  - $E_1 - E_2$
  - $E_1 \times E_2$
  - $\sigma_p(E_1)$, P is a predicate on attributes in E_1
  - $\Pi_s(E_1)$, S is a list consisting of some of the attributes in E_1
  - $\rho_x(E_1)$, x is the new name for the result of E_1

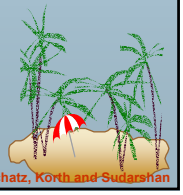


©Silberschatz, Korth and Sudarshan

Additional Operations

We define additional operations that do not add any power to the relational algebra, but that simplify common queries.

- Set intersection
- Natural join
- Division
- Assignment



©Silberschatz, Korth and Sudarshan

Set-Intersection Operation

- Notation: $r \cap s$
- Defined as:
- $r \cap s = \{ t \mid t \in r \text{ and } t \in s \}$
- Assume:
  - r, s have the *same arity*
  - attributes of r and s are compatible
- Note: $r \cap s = r - (r - s)$



Set-Intersection Operation - Example

- Relation r, s:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- r ∩ s

A	B
α	2



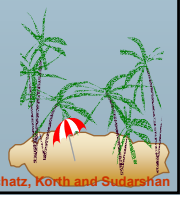
Natural-Join Operation

- Notation: r ⋈ s
- Let *r* and *s* be relations on schemas *R* and *S* respectively.
  Then, r ⋈ s is a relation on schema $R \cup S$ obtained as follows:
  - Consider each pair of tuples t_r from *r* and t_s from *s*.
  - If t_r and t_s have the same value on each of the attributes in $R \cap S$, add a tuple *t* to the result, where
    - *t* has the same value as t_r on *r*
    - *t* has the same value as t_s on *s*
- Example:

  R = (*A*, *B*, *C*, *D*)

  S = (*E*, *B*, *D*)
  - Result schema = (*A*, *B*, *C*, *D*, *E*)
  - *r* ⋈ *s* is defined as:

$$\Pi_{r.A,\ r.B,\ r.C,\ r.D,\ s.E}\ (\sigma_{r.B = s.B\ \wedge\ r.D = s.D}\ (r \times s))$$



Natural Join Operation – Example

- Relations r, s:

A	*B*	*C*	*D*
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	*D*	*E*
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

s

r ⋈ *s*

A	*B*	*C*	*D*	*E*
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ



Division Operation

$$r \div s$$

- Suited to queries that include the phrase “for all”.
- Let *r* and *s* be relations on schemas R and S respectively where
  - $R = (A_1, \ldots, A_m, B_1, \ldots, B_n)$
  - $S = (B_1, \ldots, B_n)$

  The result of r ÷ s is a relation on schema

  $R - S = (A_1, \ldots, A_m)$

$$r \div s = \{ t \mid t \in \prod_{R-S}(r) \wedge \forall u \in s \, (tu \in r) \}$$



Division Operation – Example

Relations *r, s*:

A	B
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
∈	6
∈	1
β	2

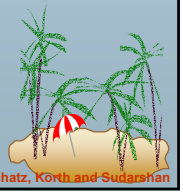
r

B
1
2

s

r ÷ *s*:

A
α
β



Another Division Example

Relations *r, s*:

A	*B*	*C*	*D*	*E*
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

r

D	*E*
a	1
b	1

s

$r \div s$:

A	*B*	*C*
α	a	γ
γ	a	γ



©Silberschatz, Korth and Sudarshan

Division Operation (Cont.)

- Property
  - Let $q - r \div s$
  - Then q is the largest relation satisfying $q \times s \subseteq r$
- Definition in terms of the basic algebra operation
  Let *r(R)* and *s(S)* be relations, and let $S \subseteq R$

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

To see why

- $\Pi_{R-S,S}(r)$ simply reorders attributes of r
- $\Pi_{R-S}(\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$ gives those tuples t in $\Pi_{R-S}(r)$ such that for some tuple $u \in s$, $tu \notin r$.



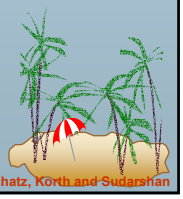
©Silberschatz, Korth and Sudarshan

Assignment Operation

- The assignment operation ($\leftarrow$) provides a convenient way to express complex queries.
  - Write query as a sequential program consisting of
    - a series of assignments
    - followed by an expression whose value is displayed as a result of the query.
  - Assignment must always be made to a temporary relation variable.
- Example: Write $r \div s$ as

$$temp1 \leftarrow \Pi_{R-S}(r)$$
$$temp2 \leftarrow \Pi_{R-S}((temp1 \times s) - \Pi_{R-S,S}(r))$$
$$result = temp1 - temp2$$

  - The result to the right of the $\leftarrow$ is assigned to the relation variable on the left of the $\leftarrow$.
  - May use variable in subsequent expressions.



Example Queries

- Find all customers who have an account from at least the "Downtown" and the Uptown" branches.

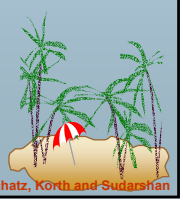
Query 1

$$\Pi_{CN}(\sigma_{BN=\text{"Downtown"}}(depositor \bowtie account)) \cap \Pi_{CN}(\sigma_{BN=\text{"Uptown"}}(depositor \bowtie account))$$

where *CN* denotes customer-name and *BN* denotes *branch-name*.

Query 2

$$\Pi_{customer\text{-}name,\ branch\text{-}name}(depositor \bowtie account) \div \rho_{temp(branch\text{-}name)}(\{(\text{"Downtown"}), (\text{"Uptown"})\})$$



Example Queries

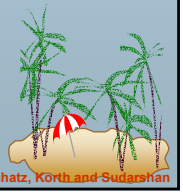
- Find all customers who have an account at all branches located in Brooklyn city.

$$\Pi_{customer\text{-}name,\ branch\text{-}name}\ (depositor \bowtie account)$$
$$\div\ \Pi_{branch\text{-}name}\ (\sigma_{branch\text{-}city\ =\ \text{“Brooklyn”}}\ (branch))$$



Extended Relational-Algebra-Operations

- Generalized Projection
- Outer Join
- Aggregate Functions



Generalized Projection

- Extends the projection operation by allowing arithmetic functions to be used in the projection list.

$$\Pi_{F1, F2, \ldots, Fn}(E)$$

- E is any relational-algebra expression
- Each of $F_1, F_2, \ldots, F_n$ are are arithmetic expressions involving constants and attributes in the schema of E.
- Given relation *credit-info(customer-name, limit, credit-balance)*, find how much more each person can spend:

$$\Pi_{customer\text{-}name,\ limit - credit\text{-}balance}\ (credit\text{-}info)$$



Aggregate Functions and Operations

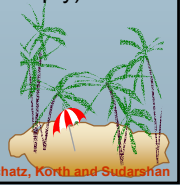
- **Aggregation function** takes a collection of values and returns a single value as a result.

  avg: average value
  min: minimum value
  max: maximum value
  sum: sum of values
  count: number of values

- **Aggregate operation** in relational algebra

$$G1, G2, \ldots, Gn\ \mathcal{G}_{F1(A1), F2(A2), \ldots, Fn(An)}\ (E)$$

  - E is any relational-algebra expression
  - $G_1, G_2 \ldots, G_n$ is a list of attributes on which to group (can be empty)
  - Each F_i is an aggregate function
  - Each A_i is an attribute name



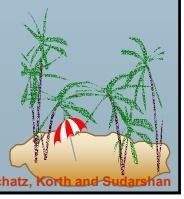
Aggregate Operation – Example

- Relation *r*:

A	*B*	*C*
α	α	7
α	β	7
β	β	3
β	β	10

$\mathcal{G}_{\mathbf{sum(c)}}(r)$

sum-C
27



©Silberschatz, Korth and Sudarshan

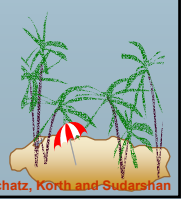
Aggregate Operation – Example

- Relation *account* grouped by *branch-name*:

branch-name	*account-number*	*balance*
Perryridge	A-102	400
Perryridge	A-201	900
Brighton	A-217	750
Brighton	A-215	750
Redwood	A-222	700

$branch\text{-}name\ \mathcal{G}_{sum(balance)}\ (account)$

branch-name	*balance*
Perryridge	1300
Brighton	1500
Redwood	700



©Silberschatz, Korth and Sudarshan

Aggregate Functions (Cont.)

- Result of aggregation does not have a name
  - Can use rename operation to give it a name
  - For convenience, we permit renaming as part of aggregate operation

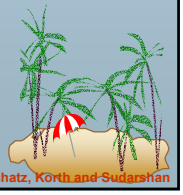
$$branch\text{-}name\ \mathcal{G}\ \textbf{sum}(balance)\ \textbf{as}\ sum\text{-}balance\ (account)$$



©Silberschatz, Korth and Sudarshan

Outer Join

- An extension of the join operation that avoids loss of information.
- Computes the join and then adds tuples form one relation that do not match tuples in the other relation to the result of the join.
- Uses *null* values:
  - *null* signifies that the value is unknown or does not exist
  - All comparisons involving *null* are (roughly speaking) **false** by definition.
    - Will study precise meaning of comparisons with nulls later



©Silberschatz, Korth and Sudarshan

Outer Join – Example

- Relation *loan*

loan-number	*branch-name*	*amount*
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

- Relation *borrower*

customer-name	*loan-number*
Jones	L-170
Smith	L-230
Hayes	L-155



©Silberschatz, Korth and Sudarshan

Outer Join – Example

- **Inner Join**

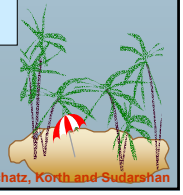
loan ⋈ *Borrower*

loan-number	*branch-name*	*amount*	*customer-name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

- **Left Outer Join**

loan ⟕ *Borrower*

loan-number	*branch-name*	*amount*	*customer-name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	*null*



©Silberschatz, Korth and Sudarshan

Outer Join – Example

- **Right Outer Join**

  loan ⟖ *borrower*

loan-number	*branch-name*	*amount*	*customer-name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	*null*	*null*	Hayes

- **Full Outer Join**

  loan ⟗ *borrower*

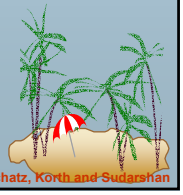
loan-number	*branch-name*	*amount*	*customer-name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	*null*
L-155	*null*	*null*	Hayes



©Silberschatz, Korth and Sudarshan

Null Values

- It is possible for tuples to have a null value, denoted by *null*, for some of their attributes
- *null* signifies an unknown value or that a value does not exist.
- The result of any arithmetic expression involving *null* is *null.*
- Aggregate functions simply ignore null values
  - Is an arbitrary decision. Could have returned null as result instead.
  - We follow the semantics of SQL in its handling of null values
- For duplicate elimination and grouping, null is treated like any other value, and two nulls are assumed to be the same
  - Alternative: assume each null is different from each other
  - Both are arbitrary decisions, so we simply follow SQL



©Silberschatz, Korth and Sudarshan

Null Values

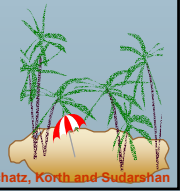
- Comparisons with null values return the special truth value *unknown*
  - If *false* was used instead of *unknown*, then *not (A < 5)* would not be equivalent to *A >= 5*
- Three-valued logic using the truth value *unknown*:
  - OR: (*unknown* **or** *true*) = *true*,
    (*unknown* **or** *false*) = *unknown*
    (*unknown* **or** *unknown*) = *unknown*
  - AND: (*true* **and** *unknown*) = *unknown*,
    (*false* **and** *unknown*) = *false*,
    (*unknown* **and** *unknown*) = *unknown*
  - NOT: (**not** *unknown*) = *unknown*
  - In SQL "*P* **is unknown**" evaluates to true if predicate *P* evaluates to *unknown*
- Result of select predicate is treated as *false* if it evaluates to *unknown*



©Silberschatz, Korth and Sudarshan

Modification of the Database

- The content of the database may be modified using the following operations:
  - Deletion
  - Insertion
  - Updating
- All these operations are expressed using the assignment operator.



©Silberschatz, Korth and Sudarshan

Deletion

- A delete request is expressed similarly to a query, except instead of displaying tuples to the user, the selected tuples are removed from the database.
- Can delete only whole tuples; cannot delete values on only particular attributes
- A deletion is expressed in relational algebra by:

$$r \leftarrow r - E$$

where r is a relation and E is a relational algebra query.



©Silberschatz, Korth and Sudarshan

Deletion Examples

- Delete all account records in the Perryridge branch.

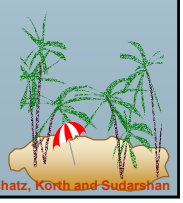
$$account \leftarrow account - \sigma_{branch\text{-}name = \text{“Perryridge”}}(account)$$

- Delete all loan records with amount in the range of 0 to 50

$$loan \leftarrow loan - \sigma_{amount \geq 0 \text{ and } amount \leq 50}(loan)$$

- Delete all accounts at branches located in Needham.

$$r_1 \leftarrow \sigma_{branch\text{-}city = \text{“Needham”}}(account \bowtie branch)$$
$$r_2 \leftarrow \Pi_{branch\text{-}name,\ account\text{-}number,\ balance}(r_1)$$
$$r_3 \leftarrow \Pi_{customer\text{-}name,\ account\text{-}number}(r_2 \bowtie depositor)$$
$$account \leftarrow account - r_2$$
$$depositor \leftarrow depositor - r_3$$



©Silberschatz, Korth and Sudarshan

Insertion

- To insert data into a relation, we either:
  - specify a tuple to be inserted
  - write a query whose result is a set of tuples to be inserted
- in relational algebra, an insertion is expressed by:

$$r \leftarrow r \cup E$$

  where r is a relation and E is a relational algebra expression.
- The insertion of a single tuple is expressed by letting E be a constant relation containing one tuple.



Insertion Examples

- Insert information in the database specifying that Smith has $1200 in account A-973 at the Perryridge branch.

  $account \leftarrow account \cup \{(\text{“Perryridge”}, \text{A-973}, 1200)\}$

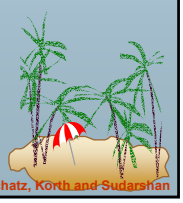
  $depositor \leftarrow depositor \cup \{(\text{“Smith”}, \text{A-973})\}$

- Provide as a gift for all loan customers in the Perryridge branch, a $200 savings account. Let the loan number serve as the account number for the new savings account.

  $r_1 \leftarrow (\sigma_{branch\text{-}name = \text{“Perryridge”}}(borrower \bowtie loan))$

  $account \leftarrow account \cup \prod_{branch\text{-}name,\ account\text{-}number,200}(r_1)$

  $depositor \leftarrow depositor \cup \prod_{customer\text{-}name,\ loan\text{-}number}(r_1)$

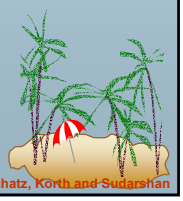


Updating

- A mechanism to change a value in a tuple without charging *all* values in the tuple
- Use the generalized projection operator to do this task

$$r \leftarrow \prod_{F1, F2, \ldots, Fl,} (r)$$

- Each F_i is either
  - the *i*th attribute of *r*, if the *i*th attribute is not updated, or,
  - if the attribute is to be updated F_i is an expression, involving only constants and the attributes of *r*, which gives the new value for the attribute



©Silberschatz, Korth and Sudarshan

Update Examples

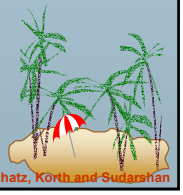
- Make interest payments by increasing all balances by 5 percent.

$$account \leftarrow \prod_{AN, BN, BAL * 1.05} (account)$$

where *AN*, *BN* and *BAL* stand for *account-number*, *branch-name* and *balance*, respectively.

- Pay all accounts with balances over $10,000 6 percent interest and pay all others 5 percent

$$account \leftarrow \prod_{AN, BN, BAL * 1.06} (\sigma_{BAL > 10000} (account)) \cup \prod_{AN, BN, BAL * 1.05} (\sigma_{BAL \leq 10000} (account))$$



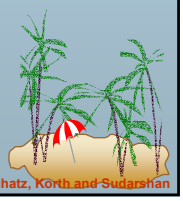
©Silberschatz, Korth and Sudarshan

Views

- In some cases, it is not desirable for all users to see the entire logical model (i.e., all the actual relations stored in the database.)
- Consider a person who needs to know a customer's loan number but has no need to see the loan amount. This person should see a relation described, in the relational algebra, by

$$\Pi_{customer\text{-}name,\ loan\text{-}number}(borrower \bowtie loan)$$

- Any relation that is not of the conceptual model but is made visible to a user as a "virtual relation" is called a **view**.



©Silberschatz, Korth and Sudarshan

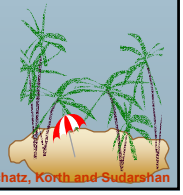
View Definition

- A view is defined using the **create view** statement which has the form

create view *v* **as** <query expression

where <query expression> is any legal relational algebra query expression. The view name is represented by *v.*

- Once a view is defined, the view name can be used to refer to the virtual relation that the view generates.
- View definition is not the same as creating a new relation by evaluating the query expression
  - Rather, a view definition causes the saving of an expression; the expression is substituted into queries using the view.



©Silberschatz, Korth and Sudarshan

View Examples

- Consider the view (named *all-customer*) consisting of branches and their customers.

  create view *all-customer* **as**

  $$\Pi_{\textit{branch-name, customer-name}} (\textit{depositor} \bowtie \textit{account}) \cup \Pi_{\textit{branch-name, customer-name}} (\textit{borrower} \bowtie \textit{loan})$$

- We can find all customers of the Perryridge branch by writing:

  $$\Pi_{\textit{customer-name}} (\sigma_{\textit{branch-name} = \text{“Perryridge”}} (\textit{all-customer}))$$



Updates Through View

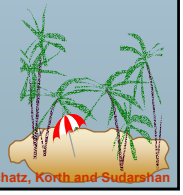
- Database modifications expressed as views must be translated to modifications of the actual relations in the database.
- Consider the person who needs to see all loan data in the *loan* relation except *amount.* The view given to the person, *branch-loan,* is defined as:

  create view *branch-loan* **as**

  $$\Pi_{\textit{branch-name, loan-number}} (\textit{loan})$$

- Since we allow a view name to appear wherever a relation name is allowed, the person may write:

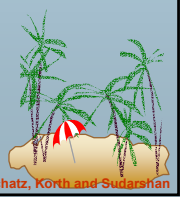
  $$\textit{branch-loan} \leftarrow \textit{branch-loan} \cup \{(\text{“Perryridge”}, \text{L-37})\}$$



Updates Through Views (Cont.)

- The previous insertion must be represented by an insertion into the actual relation *loan* from which the view *branch-loan* is constructed.
- An insertion into *loan* requires a value for *amount*. The insertion can be dealt with by either.
  - rejecting the insertion and returning an error message to the user.
  - inserting a tuple (“L-37”, “Perryridge”, *null*) into the *loan* relation
- Some updates through views are impossible to translate into database relation updates
  - create view v as $\sigma_{branch\text{-}name = \text{“Perryridge”}}(account))$

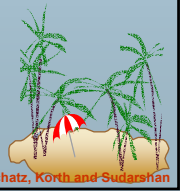
    $v \leftarrow v \cup$ (L-99, Downtown, 23)
- Others cannot be translated uniquely
  - *all-customer* $\leftarrow$ *all-customer* $\cup$ {(“Perryridge”, “John”)}
    - Have to choose loan or account, and create a new loan/account number!



©Silberschatz, Korth and Sudarshan

Views Defined Using Other Views

- One view may be used in the expression defining another view
- A view relation v_1 is said to *depend directly* on a view relation v_2 if v_2 is used in the expression defining v_1
- A view relation v_1 is said to *depend on* view relation v_2 if either v_1 depends directly to v_2 or there is a path of dependencies from v_1 to v_2
- A view relation v is said to be *recursive* if it depends on itself.



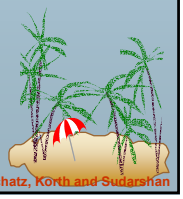
©Silberschatz, Korth and Sudarshan

View Expansion

- A way to define the meaning of views defined in terms of other views.
- Let view v_1 be defined by an expression e_1 that may itself contain uses of view relations.
- View expansion of an expression repeats the following replacement step:

  repeat
  Find any view relation v_i in e_1
  Replace the view relation v_i by the expression defining v_i
  until no more view relations are present in e_1

- As long as the view definitions are not recursive, this loop will terminate



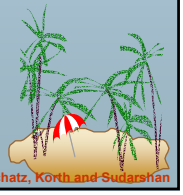
©Silberschatz, Korth and Sudarshan

Tuple Relational Calculus

- A nonprocedural query language, where each query is of the form

  $$\{t \mid P(t)\}$$

- It is the set of all tuples t such that predicate P is true for t
- t is a *tuple variable*, $t[A]$ denotes the value of tuple t on attribute A
- $t \in r$ denotes that tuple t is in relation r
- P is a *formula* similar to that of the predicate calculus



©Silberschatz, Korth and Sudarshan

Predicate Calculus Formula

1. Set of attributes and constants
2. Set of comparison operators: (e.g., $<$, $\leq$, $=$, $\neq$, $>$, $\geq$)
3. Set of connectives: and ($\wedge$), or (v), not ($\neg$)
4. Implication ($\Rightarrow$): x $\Rightarrow$ y, if x if true, then y is true

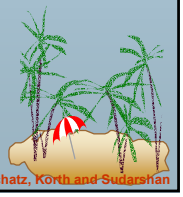
$$x \Rightarrow y \equiv \neg x \vee y$$

5. Set of quantifiers:
   - $\exists t \in r\ (Q(t)) \equiv$ "there exists" a tuple in t in relation r such that predicate $Q(t)$ is true
   - $\forall t \in r\ (Q(t)) \equiv Q$ is true "for all" tuples t in relation r



Banking Example

- *branch (branch-name, branch-city, assets)*
- *customer (customer-name, customer-street, customer-city)*
- *account (account-number, branch-name, balance)*
- *loan (loan-number, branch-name, amount)*
- *depositor (customer-name, account-number)*
- *borrower (customer-name, loan-number)*



Example Queries

- Find the *loan-number, branch-name,* and *amount* for loans of over $1200

$$\{t \mid t \in loan \wedge t\,[amount] > 1200\}$$

- Find the loan number for each loan of an amount greater than $1200

$$\{t \mid \exists\, s \in \text{loan}\ (t[loan\text{-}number] = s[loan\text{-}number] \wedge s\,[amount] > 1200)\}$$

Notice that a relation on schema [*loan-number*] is implicitly defined by the query



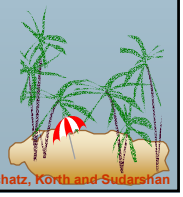
Example Queries

- Find the names of all customers having a loan, an account, or both at the bank

$$\{t \mid \exists s \in borrower(\ t[customer\text{-}name] = s[customer\text{-}name]) \vee \exists u \in depositor(\ t[customer\text{-}name] = u[customer\text{-}name])$$

- Find the names of all customers who have a loan and an account at the bank

$$\{t \mid \exists s \in borrower(\ t[customer\text{-}name] = s[customer\text{-}name]) \wedge \exists u \in depositor(\ t[customer\text{-}name] = u[customer\text{-}name])$$



Example Queries

- Find the names of all customers having a loan at the Perryridge branch

$$\{t \mid \exists s \in \textit{borrower}(t[\textit{customer-name}] = s[\textit{customer-name}] \wedge \exists u \in \textit{loan}(u[\textit{branch-name}] = \text{“Perryridge”} \wedge u[\textit{loan-number}] = s[\textit{loan-number}]))\}$$

- Find the names of all customers who have a loan at the Perryridge branch, but no account at any branch of the bank

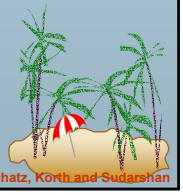
$$\{t \mid \exists s \in \textit{borrower}(t[\textit{customer-name}] = s[\text{customer-name}] \wedge \exists u \in \textit{loan}(u[\textit{branch-name}] = \text{“Perryridge”} \wedge u[\textit{loan-number}] = s[\text{loan-}\textit{number}])) \wedge \textbf{not}\ \exists v \in \textit{depositor}\ (v[\textit{customer-name}] = t[\text{customer-name}])\}$$



Example Queries

- Find the names of all customers having a loan from the Perryridge branch, and the cities they live in

$$\{t \mid \exists s \in \textit{loan}(s[\textit{branch-name}] = \text{“Perryridge”} \wedge \exists u \in \textit{borrower}\ (u[\textit{loan-number}] = s[\textit{loan-number}] \wedge t\ [\textit{customer-name}] = u[\textit{customer-name}]) \wedge \exists\ v \in \textit{customer}\ (u[\textit{customer-name}] = v[\textit{customer-name}] \wedge t[\textit{customer-city}] = v[\textit{customer-city}]))\}$$



Example Queries

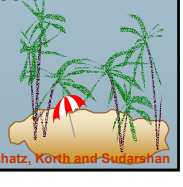
- Find the names of all customers who have an account at all branches located in Brooklyn:

$$\{t \mid \exists\, c \in \text{customer}\ (t[\text{customer.name}] = c[\text{customer-name}]) \wedge$$
$$\forall\, s \in \textit{branch}(s[\textit{branch-city}] = \text{“Brooklyn”} \Rightarrow$$
$$\exists\, u \in \textit{account}\ (\ s[\textit{branch-name}] = u[\text{branch-name}]$$
$$\wedge \exists\, s \in \textit{depositor}\ (\ t[\textit{customer-name}] = s[\text{customer-name}]$$
$$\wedge\ s[\textit{account-number}] = u[\text{account-number}]\ ))\ )\}$$



Safety of Expressions

- It is possible to write tuple calculus expressions that generate infinite relations.
- For example, $\{t \mid \neg\, t \in r\}$ results in an infinite relation if the domain of any attribute of relation r is infinite
- To guard against the problem, we restrict the set of allowable expressions to safe expressions.
- An expression $\{t \mid P(t)\}$ in the tuple relational calculus is *safe* if every component of t appears in one of the relations, tuples, or constants that appear in P
  - NOTE: this is more than just a syntax condition.
    - E.g. $\{ t \mid t[A]=5 \vee \textbf{true} \}$ is not safe --- it defines an infinite set with attribute values that do not appear in any relation or tuples or constants in P.



Domain Relational Calculus

- A nonprocedural query language equivalent in power to the tuple relational calculus
- Each query is an expression of the form:

$$\{ < x_1, x_2, \ldots, x_n > \mid P(x_1, x_2, \ldots, x_n)\}$$

  - $x_1, x_2, \ldots, x_n$ represent domain variables
  - P represents a formula similar to that of the predicate calculus



Example Queries

- Find the *loan-number, branch-name,* and *amount* for loans of over $1200

$$\{< l, b, a > \mid < l, b, a > \in loan \wedge a > 1200\}$$

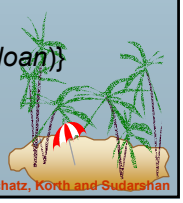
- Find the names of all customers who have a loan of over $1200

$$\{< c > \mid \exists\, l, b, a\, (< c, l > \in borrower \wedge < l, b, a > \in loan \wedge a > 1200)\}$$

- Find the names of all customers who have a loan from the Perryridge branch and the loan amount:

$$\{< c, a > \mid \exists\, l\, (< c, l > \in borrower \wedge \exists b(< l, b, a > \in loan \wedge b = \text{“Perryridge”}))\}$$

or $\{< c, a > \mid \exists\, l\, (< c, l > \in borrower \wedge < l, \text{“Perryridge”}, a > \in loan)\}$



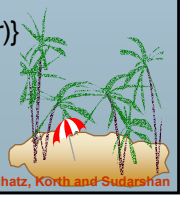
Example Queries

- Find the names of all customers having a loan, an account, or both at the Perryridge branch:

$$\{< c > \mid \exists\, l\, (\{< c, l > \in \textit{borrower} \wedge \exists\, b,a(< l, b, a > \in \textit{loan} \wedge b = \text{“Perryridge”})) \vee \exists\, a(< c, a > \in \textit{depositor} \wedge \exists\, b,n(< a, b, n > \in \textit{account} \wedge b = \text{“Perryridge”}))\}$$

- Find the names of all customers who have an account at all branches located in Brooklyn:

$$\{< c > \mid \exists\, s, n\, (< c, s, n > \in \text{customer}) \wedge \forall\, x,y,z(< x, y, z > \in \textit{branch} \wedge y = \text{“Brooklyn”}) \Rightarrow \exists\, a,b(< x, y, z > \in \textit{account} \wedge < c,a > \in \textit{depositor})\}$$



Safety of Expressions

$$\{ < x_1, x_2, \ldots, x_n > \mid P(x_1, x_2, \ldots, x_n)\}$$

is safe if all of the following hold:

1. All values that appear in tuples of the expression are values from *dom*(P) (that is, the values appear either in P or in a tuple of a relation mentioned in P).
2. For every “there exists” subformula of the form $\exists\, x\, (P_1(x))$, the subformula is true if and only if there is a value of x in $dom(P_1)$ such that $P_1(x)$ is true.
3. For every “for all” subformula of the form $\forall_x\, (P_1\, (x))$, the subformula is true if and only if $P_1(x)$ is true for all values x from $dom\, (P_1)$.



End of Chapter 3

Result of $\sigma_{branch\text{-}name = \text{“Perryridge”}}(loan)$

loan-number	*branch-name*	*amount*
L-15	Perryridge	1500
L-16	Perryridge	1300



©Silberschatz, Korth and Sudarshan

Loan Number and the Amount of the Loan

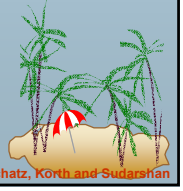
loan-number	*amount*
L-11	900
L-14	1500
L-15	1500
L-16	1300
L-17	1000
L-23	2000
L-93	500



©Silberschatz, Korth and Sudarshan

Names of All Customers Who Have Either a Loan or an Account

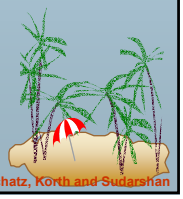
customer-name
Adams
Curry
Hayes
Jackson
Jones
Smith
Williams
Lindsay
Johnson
Turner



©Silberschatz, Korth and Sudarshan

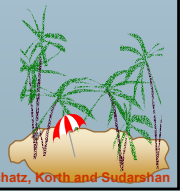
Customers With An Account But No Loan

customer-name
Johnson
Lindsay
Turner



Result of *borrower* × *loan*

customer-name	*borrower. loan-number*	*loan. loan-number*	*branch-name*	*amount*
Adams	L-16	L-11	Round Hill	900
Adams	L-16	L-14	Downtown	1500
Adams	L-16	L-15	Perryridge	1500
Adams	L-16	L-16	Perryridge	1300
Adams	L-16	L-17	Downtown	1000
Adams	L-16	L-23	Redwood	2000
Adams	L-16	L-93	Mianus	500
Curry	L-93	L-11	Round Hill	900
Curry	L-93	L-14	Downtown	1500
Curry	L-93	L-15	Perryridge	1500
Curry	L-93	L-16	Perryridge	1300
Curry	L-93	L-17	Downtown	1000
Curry	L-93	L-23	Redwood	2000
Curry	L-93	L-93	Mianus	500
Hayes	L-15	L-11		900
Hayes	L-15	L-14		1500
Hayes	L-15	L-15		1500
Hayes	L-15	L-16		1300
Hayes	L-15	L-17		1000
Hayes	L-15	L-23		2000
Hayes	L-15	L-93		500
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
Smith	L-23	L-11	Round Hill	900
Smith	L-23	L-14	Downtown	1500
Smith	L-23	L-15	Perryridge	1500
Smith	L-23	L-16	Perryridge	1300
Smith	L-23	L-17	Downtown	1000
Smith	L-23	L-23	Redwood	2000
Smith	L-23	L-93	Mianus	500
Williams	L-17	L-11	Round Hill	900
Williams	L-17	L-14	Downtown	1500
Williams	L-17	L-15	Perryridge	1500
Williams	L-17	L-16	Perryridge	1300
Williams	L-17	L-17	Downtown	1000
Williams	L-17	L-23	Redwood	2000
Williams	L-17	L-93	Mianus	500



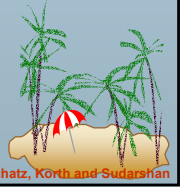
Result of $\sigma_{branch\text{-}name = \text{“Perryridge”}}$ (*borrower* × *loan*)

customer-name	*borrower. loan-number*	*loan. loan-number*	*branch-name*	*amount*
Adams	L-16	L-15	Perryridge	1500
Adams	L-16	L-16	Perryridge	1300
Curry	L-93	L-15	Perryridge	1500
Curry	L-93	L-16	Perryridge	1300
Hayes	L-15	L-15	Perryridge	1500
Hayes	L-15	L-16	Perryridge	1300
Jackson	L-14	L-15	Perryridge	1500
Jackson	L-14	L-16	Perryridge	1300
Jones	L-17	L-15	Perryridge	1500
Jones	L-17	L-16	Perryridge	1300
Smith	L-11	L-15	Perryridge	1500
Smith	L-11	L-16	Perryridge	1300
Smith	L-23	L-15	Perryridge	1500
Smith	L-23	L-16	Perryridge	1300
Williams	L-17	L-15	Perryridge	1500
Williams	L-17	L-16	Perryridge	1300



Result of $\Pi_{customer\text{-}name}$

customer-name
Adams
Hayes



Result of the Subexpression

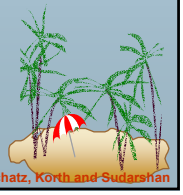
balance
500
400
700
750
350



©Silberschatz, Korth and Sudarshan

Largest Account Balance in the Bank

balance
900



©Silberschatz, Korth and Sudarshan

Customers Who Live on the Same Street and In the Same City as Smith

customer-name
Curry
Smith



©Silberschatz, Korth and Sudarshan

Customers With Both an Account and a Loan at the Bank

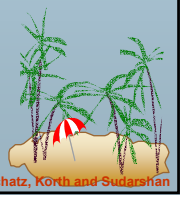
customer-name
Hayes
Jones
Smith



©Silberschatz, Korth and Sudarshan

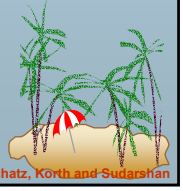
Result of $\Pi_{customer\text{-}name,\ loan\text{-}number,\ amount}$ $(borrower \bowtie loan)$

customer-name	*loan-number*	*amount*
Adams	L-16	1300
Curry	L-93	500
Hayes	L-15	1500
Jackson	L-14	1500
Jones	L-17	1000
Smith	L-23	2000
Smith	L-11	900
Williams	L-17	1000



Result of $\Pi_{branch\text{-}name}(\sigma_{customer\text{-}city = \text{“Harrison”}}(customer \bowtie account \bowtie depositor))$

branch-name
Brighton
Perryridge



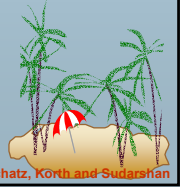
Result of $\Pi_{branch\text{-}name}(\sigma_{branch\text{-}city = \text{“Brooklyn”}}(branch))$

branch-name
Brighton
Downtown



Result of $\Pi_{customer\text{-}name,\ branch\text{-}name}(depositor \bowtie account)$

customer-name	*branch-name*
Hayes	Perryridge
Johnson	Downtown
Johnson	Brighton
Jones	Brighton
Lindsay	Redwood
Smith	Mianus
Turner	Round Hill



The *credit-info* Relation

customer-name	*branch-name*
Hayes	Perryridge
Johnson	Downtown
Johnson	Brighton
Jones	Brighton
Lindsay	Redwood
Smith	Mianus
Turner	Round Hill



©Silberschatz, Korth and Sudarshan

Result of $\Pi_{customer\text{-}name,\ (limit\ -\ credit\text{-}balance)\ as\ credit\text{-}available}(credit\text{-}info)$.

customer-name	*credit-available*
Curry	250
Jones	5300
Smith	1600
Hayes	0



©Silberschatz, Korth and Sudarshan

The *pt-works* Relation

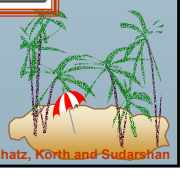
employee-name	*branch-name*	*salary*
Adams	Perryridge	1500
Brown	Perryridge	1300
Gopal	Perryridge	5300
Johnson	Downtown	1500
Loreena	Downtown	1300
Peterson	Downtown	2500
Rao	Austin	1500
Sato	Austin	1600



©Silberschatz, Korth and Sudarshan

The *pt-works* Relation After Grouping

employee-name	*branch-name*	*salary*
Rao	Austin	1500
Sato	Austin	1600
Johnson	Downtown	1500
Loreena	Downtown	1300
Peterson	Downtown	2500
Adams	Perryridge	1500
Brown	Perryridge	1300
Gopal	Perryridge	5300



©Silberschatz, Korth and Sudarshan

Result of *branch-name* $\mathcal{G}$ sum(*salary*) (*pt-works*)

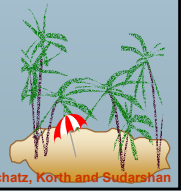
branch-name	*sum of salary*
Austin	3100
Downtown	5300
Perryridge	8100



©Silberschatz, Korth and Sudarshan

Result of *branch-name* $\mathcal{G}$ sum *salary*, max(*salary*) as *max-salary* (*pt-works*)

branch-name	*sum-salary*	*max-salary*
Austin	3100	1600
Downtown	5300	2500
Perryridge	8100	5300

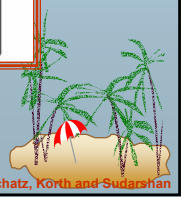


©Silberschatz, Korth and Sudarshan

The *employee* and *ft-works* Relations

employee-name	*street*	*city*
Coyote	Toon	Hollywood
Rabbit	Tunnel	Carrotville
Smith	Revolver	Death Valley
Williams	Seaview	Seattle

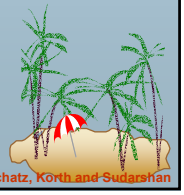
employee-name	*branch-name*	*salary*
Coyote	Mesa	1500
Rabbit	Mesa	1300
Gates	Redmond	5300
Williams	Redmond	1500



©Silberschatz, Korth and Sudarshan

The Result of *employee* ⋈ *ft-works*

employee-name	*street*	*city*	*branch-name*	*salary*
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500



©Silberschatz, Korth and Sudarshan

The Result of *employee* ⋈ *ft-works*

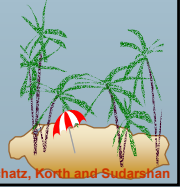
employee-name	*street*	*city*	*branch-name*	*salary*
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500
Smith	Revolver	Death Valley	*null*	*null*



©Silberschatz, Korth and Sudarshan

Result of *employee* ⋉ *ft-works*

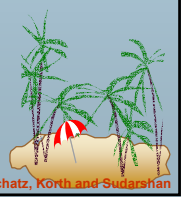
employee-name	*street*	*city*	*branch-name*	*salary*
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500
Gates	*null*	*null*	Redmond	5300



©Silberschatz, Korth and Sudarshan

Result of *employee* ⋈ *ft-works*

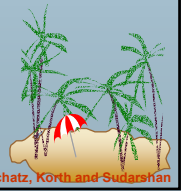
employee-name	*street*	*city*	*branch-name*	*salary*
Coyote	Toon	Hollywood	Mesa	1500
Rabbit	Tunnel	Carrotville	Mesa	1300
Williams	Seaview	Seattle	Redmond	1500
Smith	Revolver	Death Valley	*null*	*null*
Gates	*null*	*null*	Redmond	5300



Tuples Inserted Into *loan* and *borrower*

loan-number	*branch-name*	*amount*
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500
null	*null*	1900

customer-name	*loan-number*
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17
Johnson	*null*

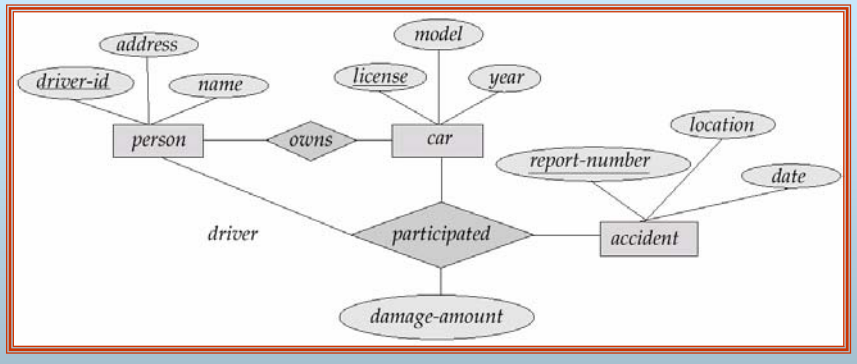


Names of All Customers Who Have a Loan at the Perryridge Branch

customer-name
Adams
Hayes

©Silberschatz, Korth and Sudarshan

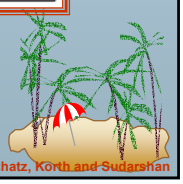
E-R Diagram



©Silberschatz, Korth and Sudarshan

The *branch* Relation

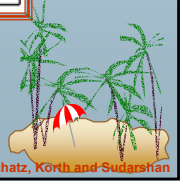
branch-name	*branch-city*	*assets*
Brighton	Brooklyn	7100000
Downtown	Brooklyn	9000000
Mianus	Horseneck	400000
North Town	Rye	3700000
Perryridge	Horseneck	1700000
Pownal	Bennington	300000
Redwood	Palo Alto	2100000
Round Hill	Horseneck	8000000



©Silberschatz, Korth and Sudarshan

The *loan* Relation

loan-number	*branch-name*	*amount*
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500



©Silberschatz, Korth and Sudarshan

The *borrower* Relation

customer-name	*loan-number*
Adams	L-16
Curry	L-93
Hayes	L-15
Jackson	L-14
Jones	L-17
Smith	L-11
Smith	L-23
Williams	L-17

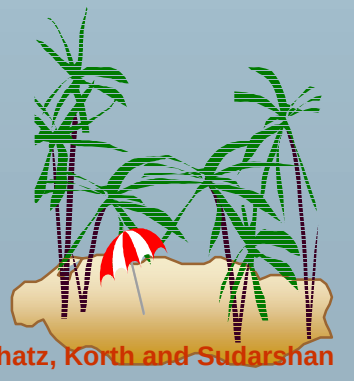


©Silberschatz, Korth and Sudarshan



Chapter 8: Object-Oriented Databases

- Need for Complex Data Types
- The Object-Oriented Data Model
- Object-Oriented Languages
- Persistent Programming Languages
- Persistent C++ Systems



©Silberschatz, Korth and Sudarshan

Need for Complex Data Types

- Traditional database applications in data processing had conceptually simple data types
  - Relatively few data types, first normal form holds
- Complex data types have grown more important in recent years
  - E.g. Addresses can be viewed as a
    - Single string, or
    - Separate attributes for each part, or
    - Composite attributes (which are not in first normal form)
  - E.g. it is often convenient to store multivalued attributes as-is, without creating a separate relation to store the values in first normal form
- Applications
  - computer-aided design, computer-aided software engineering
  - multimedia and image databases, and document/hypertext databases.



©Silberschatz, Korth and Sudarshan

Object-Oriented Data Model

- Loosely speaking, an **object** corresponds to an entity in the E-R model.
- The *object-oriented paradigm* is based on *encapsulating* code and data related to an object into single unit.
- The object-oriented data model is a logical data model (like the E-R model).
- Adaptation of the object-oriented programming paradigm (e.g., Smalltalk, C++) to database systems.



©Silberschatz, Korth and Sudarshan

Object Structure

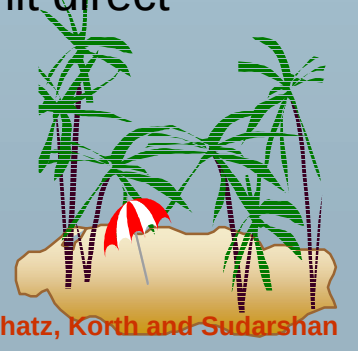
- An object has associated with it:
  - ★ A set of **variables** that contain the data for the object. The value of each variable is itself an object.
  - ★ A set of **messages** to which the object responds; each message may have zero, one, or more *parameters.*
  - ★ A set of **methods**, each of which is a body of code to implement a message; a method returns a value as the *response* to the message
- The physical representation of data is visible only to the implementor of the object
- Messages and responses provide the only external interface to an object.
- The term message does not necessarily imply physical message passing. Messages can be implemented as procedure invocations.



©Silberschatz, Korth and Sudarshan

Messages and Methods

- Methods are programs written in general-purpose language with the following features
  - ★ only variables in the object itself may be referenced directly
  - ★ data in other objects are referenced only by sending *messages*.
- Methods can be read-only or update methods
  - ★ Read-only methods do not change the value of the object
- Strictly speaking, every attribute of an entity must be represented by a variable and two methods, one to read and the other to update the attribute
  - ★ e.g., the attribute *address* is represented by a variable *address* and two messages *get-address* and *set-address*.
  - ★ For convenience, many object-oriented data models permit direct access to variables of other objects.



©Silberschatz, Korth and Sudarshan

Object Classes

- Similar objects are grouped into a **class**; each such object is called an **instance** of its class
- All objects in a class have the same
  - ★ Variables, with the same types
  - ★ message interface
  - ★ methods

  The may differ in the values assigned to variables
- Example: Group objects for people into a *person* class
- Classes are analogous to entity sets in the E-R model



©Silberschatz, Korth and Sudarshan

Class Definition Example

```
class employee {
        /*Variables */
            string    name;
            string    address;
            date      start-date;
            int       salary;
        /* Messages */
            int       annual-salary();
            string    get-name();
            string    get-address();
            int       set-address(string new-address);
            int       employment-length();
};
```

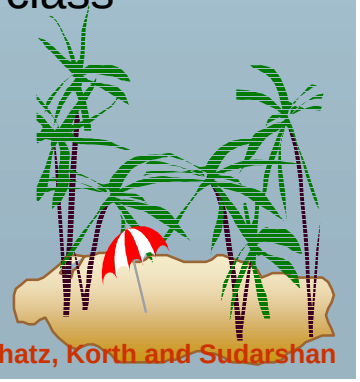
- Methods to read and set the other variables are also needed with strict encapsulation
- Methods are defined separately
  - E.g. **int** *employment-length*() { **return** *today*() – *start-date*;}
    int *set-address*(**string** *new-address*) { *address* = *new-address*;}



©Silberschatz, Korth and Sudarshan

Inheritance

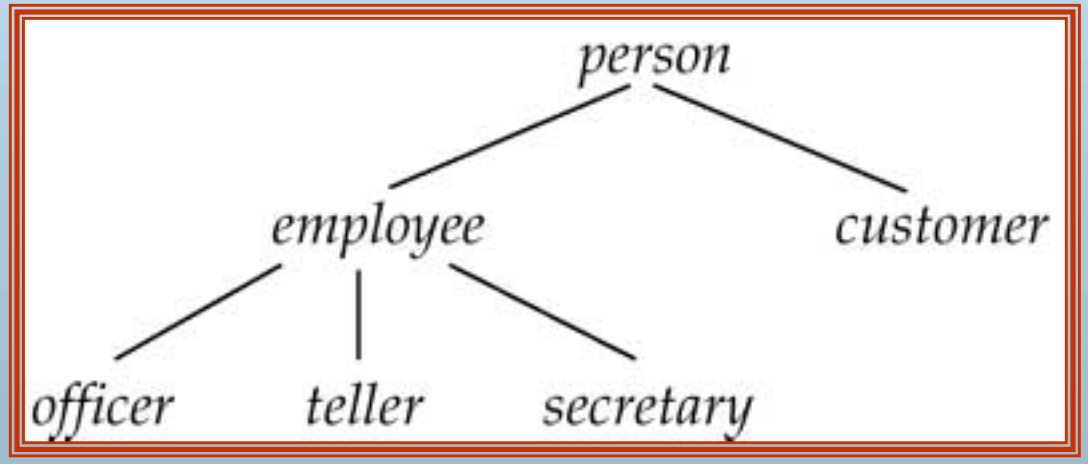
- E.g., class of bank customers is similar to class of bank employees, although there are differences
  - ★ both share some variables and messages, e.g., *name* and *address.*
  - ★ But there are variables and messages specific to each class e.g., *salary* for employees and *credit-rating* for customers.
- Every employee is a person; thus *employee* is a specialization of *person*
- *Similarly, customer* is a specialization of *person.*
- Create classes *person, employee* and *customer*
  - ★ variables/messages applicable to all persons associated with class *person.*
  - ★ variables/messages specific to employees associated with class *employee*; similarly for *customer*



©Silberschatz, Korth and Sudarshan

Inheritance (Cont.)

- Place classes into a specialization/IS-A hierarchy
  - ⋆ variables/messages belonging to class *person* are *inherited* by class *employee* as well as *customer*
- Result is a **class hierarchy**



Note analogy with ISA Hierarchy in the E-R model

©Silberschatz, Korth and Sudarshan

Class Hierarchy Definition

```
class person{
    string  name;
    string  address:
    };
class customer isa person {
    int credit-rating;
    };
class employee isa person {
    date start-date;
    int salary;
    };
class officer isa employee {
    int office-number,
    int expense-account-number,
    };
.
.
.
```



©Silberschatz, Korth and Sudarshan

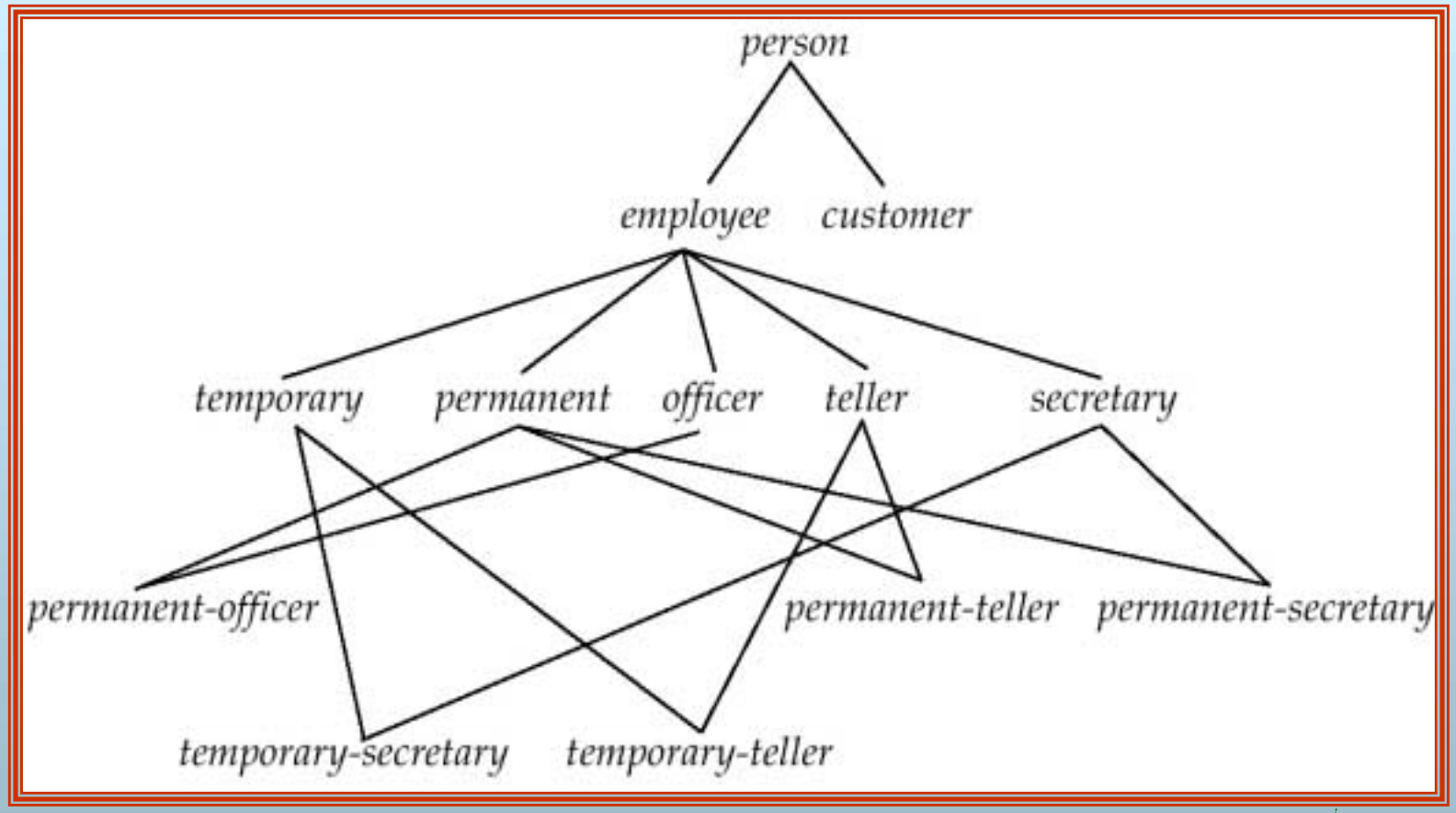
Class Hierarchy Example (Cont.)

- Full variable list for objects in the class *officer:*
  - *office-number, expense-account-number:* defined locally
  - *start-date, salary:* inherited from *employee*
  - *name, address:* inherited from *person*
- Methods inherited similar to variables.
- **Substitutability** — any method of a class, say *person,* can be invoked equally well with any object belonging to any subclass, such as subclass *officer* of *person.*
- **Class extent***:* set of all objects in the class. Two options:
  1. Class extent of *employee* includes all *officer, teller* and *secretary* objects.
  2. Class extent of *employee* includes only employee objects that are not in a subclass such as *officer, teller,* or *secretary*
     - This is the usual choice in OO systems
     - Can access extents of subclasses to find all objects of subtypes of employee



©Silberschatz, Korth and Sudarshan

Example of Multiple Inheritance



Class DAG for banking example.

©Silberschatz, Korth and Sudarshan

Multiple Inheritance

- With multiple inheritance a class may have more than one superclass.
  - The class/subclass relationship is represented by a **directed acyclic graph** (**DAG**)
  - Particularly useful when objects can be classified in more than one way, which are independent of each other
    - E.g. temporary/permanent is independent of Officer/secretary/teller
    - Create a subclass for each combination of subclasses
      - Need not create subclasses for combinations that are not possible in the database being modeled
- A class inherits variables and methods from all its superclasses
- There is potential for ambiguity when a variable/message N with the same name is inherited from two superclasses A and B
  - No problem if the variable/message is defined in a shared superclass
  - Otherwise, do one of the following
    - flag as an error,
    - rename variables (A.N and B.N)
    - choose one.



©Silberschatz, Korth and Sudarshan

More Examples of Multiple Inheritance

- Conceptually, an object can belong to each of several subclasses
  - A *person* can play the roles of *student*, a *teacher* or *footballPlayer,* or any combination of the three
    - E.g., student teaching assistant who also play football
- Can use multiple inheritance to model "roles" of an object
  - That is, allow an object to take on any one or more of a set of types
- But many systems insist an object should have a **most-specific class**
  - That is, there must be one class that an object belongs to which is a subclass of all other classes that the object belongs to
  - Create subclasses such as *student-teacher* and *student-teacher-footballPlayer* for each combination
  - When many combinations are possible, creating subclasses for each combination can become cumbersome



©Silberschatz, Korth and Sudarshan

Object Identity

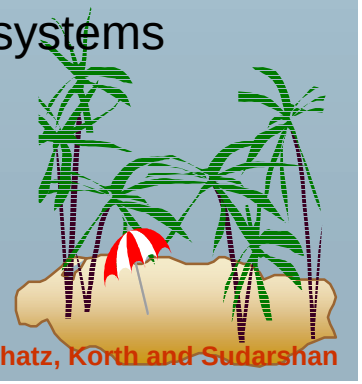
- An object retains its identity even if some or all of the values of variables or definitions of methods change over time.
- Object identity is a stronger notion of identity than in programming languages or data models not based on object orientation.
  - Value – data value; e.g. primary key value used in relational systems.
  - Name – supplied by user; used for variables in procedures.
  - Built-in – identity built into data model or programming language.
    - no user-supplied identifier is required.
    - Is the form of identity used in object-oriented systems.



©Silberschatz, Korth and Sudarshan

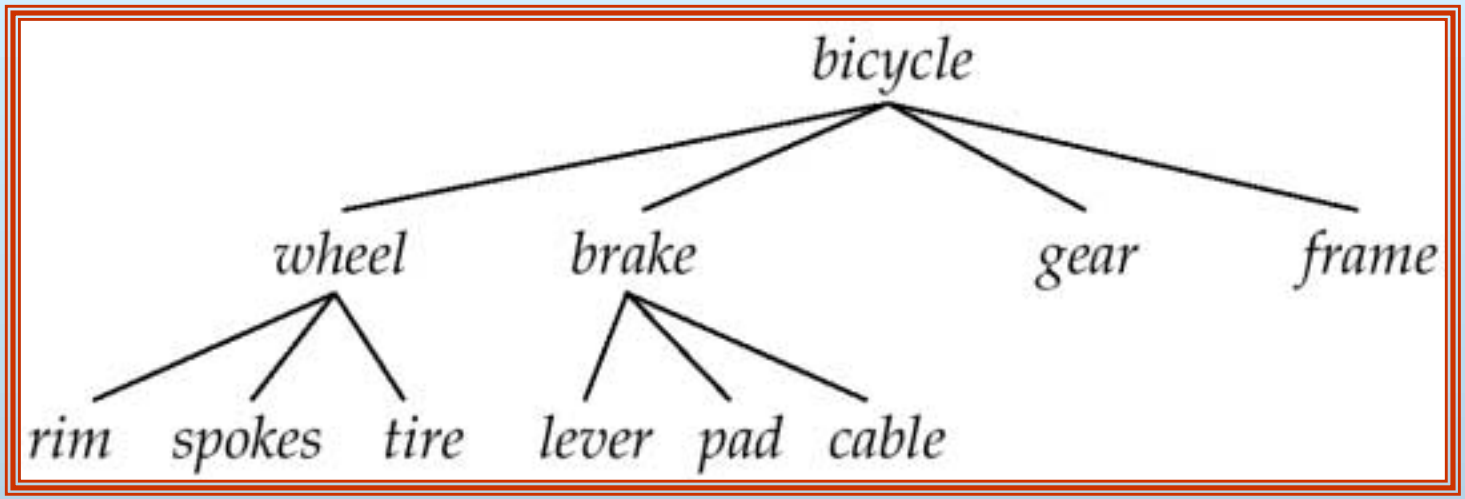
Object Identifiers

- **Object identifiers** used to uniquely identify objects
  - Object identifiers are unique:
    - no two objects have the same identifier
    - each object has only one object identifier
  - E.g., the *spouse* field of a *person* object may be an identifier of another *person* object.
  - can be stored as a field of an object, to refer to another object.
  - Can be
    - system generated (created by database) or
    - external (such as social-security number)
  - System generated identifiers:
    - Are easier to use, but cannot be used across database systems
    - May be redundant if unique identifier already exists



©Silberschatz, Korth and Sudarshan

Object Containment

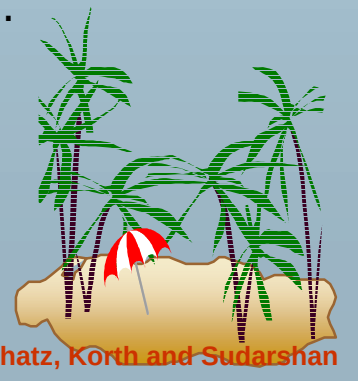


- Each component in a design may contain other components
- Can be modeled as containment of objects. Objects containing; other objects are called **composite** objects.
- Multiple levels of containment create a **containment hierarchy**
  - ★ links interpreted as **is-part-of,** not **is-a.**
- Allows data to be viewed at different granularities by different users.

©Silberschatz, Korth and Sudarshan

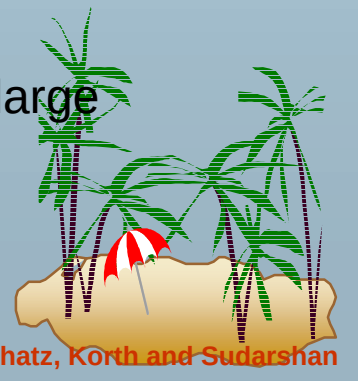
Object-Oriented Languages

- Object-oriented concepts can be used in different ways
  - Object-orientation can be used as a design tool, and be encoded into, for example, a relational database
    - analogous to modeling data with E-R diagram and then converting to a set of relations)
  - The concepts of object orientation can be incorporated into a programming language that is used to manipulate the database.
    - **Object-relational systems** – add complex types and object-orientation to relational language.
    - **Persistent programming languages** – extend object-oriented programming language to deal with databases by adding concepts such as persistence and collections.



Persistent Programming Languages

- Persistent Programming languages allow objects to be created and stored in a database, and used directly from a programming language
  - allow data to be manipulated directly from the programming language
    - No need to go through SQL.
  - No need for explicit format (type) changes
    - format changes are carried out transparently by system
    - Without a persistent programming language, format changes becomes a burden on the programmer
      - More code to be written
      - More chance of bugs
  - allow objects to be manipulated in-memory
    - no need to explicitly load from or store to the database
      - Saved code, and saved overhead of loading/storing large amounts of data



©Silberschatz, Korth and Sudarshan

Persistent Prog. Languages (Cont.)

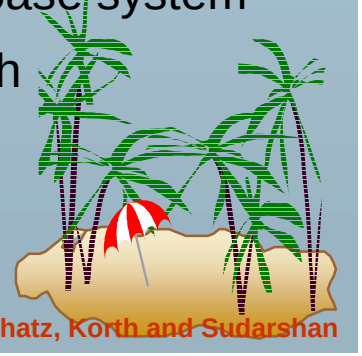
- Drawbacks of persistent programming languages
  - ★ Due to power of most programming languages, it is easy to make programming errors that damage the database.
  - ★ Complexity of languages makes automatic high-level optimization more difficult.
  - ★ Do not support declarative querying as well as relational databases



©Silberschatz, Korth and Sudarshan

Persistence of Objects

- Approaches to make transient objects persistent include establishing
  - ★ Persistence by Class – declare all objects of a class to be persistent; simple but inflexible.
  - ★ Persistence by Creation – extend the syntax for creating objects to specify that that an object is persistent.
  - ★ Persistence by Marking – an object that is to persist beyond program execution is marked as persistent before program termination.
  - ★ Persistence by Reachability - declare (root) persistent objects; objects are persistent if they are referred to (directly or indirectly) from a root object.
    - ➢ Easier for programmer, but more overhead for database system
    - ➢ Similar to garbage collection used e.g. in Java, which also performs reachability tests

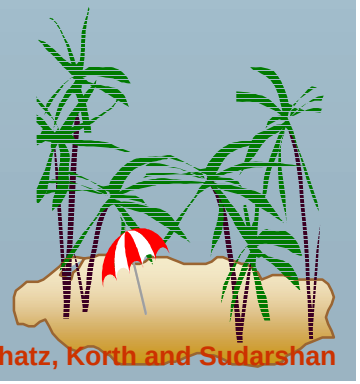


©Silberschatz, Korth and Sudarshan



Object Identity and Pointers

- A persistent object is assigned a persistent object identifier.
- Degrees of permanence of identity:
  - ★ Intraprocedure – identity persists only during the executions of a single procedure
  - ★ Intraprogram – identity persists only during execution of a single program or query.
  - ★ Interprogram – identity persists from one program execution to another, but may change if the storage organization is changed
  - ★ Persistent – identity persists throughout program executions and structural reorganizations of data; required for object-oriented systems.



©Silberschatz, Korth and Sudarshan

Object Identity and Pointers (Cont.)

- In O-O languages such as C++, an object identifier is actually an in-memory pointer.
- Persistent pointer – persists beyond program execution
  - ★ can be thought of as a pointer into the database
    - ➢ E.g. specify file identifier and offset into the file
  - ★ Problems due to database reorganization have to be dealt with by keeping forwarding pointers



©Silberschatz, Korth and Sudarshan

Storage and Access of Persistent Objects

How to find objects in the database:

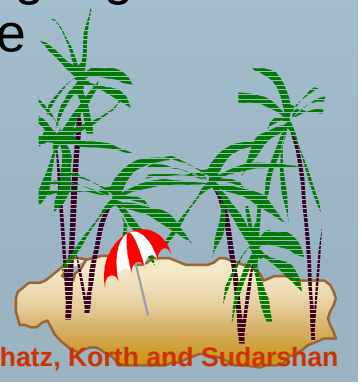
- Name objects (as you would name files)
  - ★ Cannot scale to large number of objects.
  - ★ Typically given only to class extents and other collections of objects, but not objects.
- Expose object identifiers or persistent pointers to the objects
  - ★ Can be stored externally.
  - ★ All objects have object identifiers.
- Store collections of objects, and allow programs to iterate over the collections to find required objects
  - ★ Model collections of objects as **collection types**
  - ★ **Class extent** - the collection of all objects belonging to the class; usually maintained for all classes that can have persistent objects.



©Silberschatz, Korth and Sudarshan

Persistent C++ Systems

- C++ language allows support for persistence to be added without changing the language
  - ⋆ Declare a class called Persistent_Object with attributes and methods to support persistence
  - ⋆ **Overloading** – ability to redefine standard function names and operators (i.e., +, –, the pointer deference operator –>) when applied to new types
  - ⋆ **Template classes** help to build a type-safe type system supporting collections and persistent types.
- Providing persistence without extending the C++ language is
  - ⋆ relatively easy to implement
  - ⋆ but more difficult to use
- Persistent C++ systems that add features to the C++ language have been built, as also systems that avoid changing the language



©Silberschatz, Korth and Sudarshan

ODMG C++ Object Definition Language

- The Object Database Management Group is an industry consortium aimed at standardizing object-oriented databases
  - in particular persistent programming languages
  - Includes standards for C++, Smalltalk and Java
  - ODMG-93
  - ODMG-2.0 and 3.0 (which is 2.0 plus extensions to Java)
    - Our description based on ODMG-2.0
- ODMG C++ standard avoids changes to the C++ language
  - provides functionality via template classes and class libraries



©Silberschatz, Korth and Sudarshan

ODMG Types

- Template class d_Ref<class> used to specify references (persistent pointers)
- Template class d_Set<class> used to define sets of objects.
  - ★ Methods include insert_element(e) and delete_element(e)
- Other collection classes such as d_Bag (set with duplicates allowed), d_List and d_Varray (variable length array) also provided.
- d_ version of many standard types provided, e.g. d_Long and d_string
  - ★ Interpretation of these types is platform independent
  - ★ Dynamically allocated data (e.g. for d_string) allocated in the database, not in main memory



©Silberschatz, Korth and Sudarshan

ODMG C++ ODL: Example

```
class Branch  :  public d_Object {
 ....
}
class Person  :  public d_Object {
   public:
      d_String    name;       // should not use String!
      d_String    address;
};
class Account : public d_Object {
   private:
      d_Long      balance;
   public:
      d_Long      number;
      d_Set <d_Ref<Customer>> owners;
      int         find_balance();
      int         update_balance(int delta);
};
```



©Silberschatz, Korth and Sudarshan

ODMG C++ ODL: Example (Cont.)

```
class Customer  :  public Person {
    public:
       d_Date              member_from;
       d_Long              customer_id;
       d_Ref<Branch> home_branch;
       d_Set <d_Ref<Account>> accounts; };
```



©Silberschatz, Korth and Sudarshan

Implementing Relationships

- Relationships between classes implemented by references
- Special reference types enforces integrity by adding/removing inverse links.
  - ★ Type d_Rel_Ref<Class, InvRef> is a reference to Class, where attribute InvRef of Class is the inverse reference.
  - ★ Similarly, d_Rel_Set<Class, InvRef> is used for a set of references
- Assignment method (=) of class d_Rel_Ref is overloaded
  - ★ Uses type definition to automatically find and update the inverse link
  - ★ Frees programmer from task of updating inverse links
  - ★ Eliminates possibility of inconsistent links
- Similarly, insert_element() and delete_element() methods of d_Rel_Set use type definition to find and update the inverse link automatically



©Silberschatz, Korth and Sudarshan

Implementing Relationships

- E.g.

```
extern const char _owners[ ],   _accounts[ ];
class Account : public d.Object {
        ....
        d_Rel_Set <Customer, _accounts> owners;
}
  // .. Since strings can't be used in templates ...
const char _owners= "owners";
const char _accounts= "accounts";
```



©Silberschatz, Korth and Sudarshan

ODMG C++ Object Manipulation Language

- Uses persistent versions of C++ operators such as new(db)

  d_Ref<Account> account = new(bank_db, “Account”) Account;

  - ★ new allocates the object in the specified database, rather than in memory.
  - ★ The second argument (“Account”) gives typename used in the database.
- Dereference operator -> when applied on a d_Ref<Account> reference loads the referenced object in memory (if not already present) before continuing with usual C++ dereference.
- Constructor for a class – a special method to initialize objects when they are created; called automatically on new call.
- Class extents maintained automatically on object creation and deletion
  - ★ Only for classes for which this feature has been specified
    - ➢ Specification via user interface, not C++
  - ★ Automatic maintenance of class extents not supported in earlier versions of ODMG



©Silberschatz, Korth and Sudarshan

ODMG C++OML: Database and Object Functions

- Class d_Database provides methods to
  - open a database: open(databasename)
  - give names to objects: set_object_name(object, name)
  - look up objects by name: lookup_object(name)
  - rename objects: rename_object(oldname, newname)
  - close a database (close());
- Class d_Object is inherited by all persistent classes.
  - provides methods to allocate and delete objects
  - method mark_modified() must be called *before* an object is updated.
    - Is automatically called when object is created



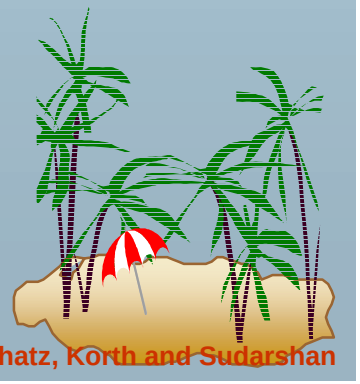
©Silberschatz, Korth and Sudarshan

ODMG C++ OML: Example

```
int create_account_owner(String name, String Address){
    Database bank_db.obj;
    Database * bank_db= & bank_db.obj;
    bank_db =>open("Bank-DB");
    d.Transaction Trans;
    Trans.begin();

    d_Ref<Account> account = new(bank_db) Account;
    d_Ref<Customer> cust = new(bank_db) Customer;
    cust->name - name;
    cust->address = address;
    cust->accounts.insert_element(account);
    ... Code to initialize other fields

    Trans.commit();
}
```



©Silberschatz, Korth and Sudarshan

ODMG C++ OML: Example (Cont.)

- Class extents maintained automatically in the database.
- To access a class extent:
  d_Extent<Customer> customerExtent(bank_db);
- Class d_Extent provides method
  d_Iterator<T> create_iterator()
  to create an iterator on the class extent
- Also provides select(pred) method to return iterator on objects that satisfy selection predicate pred.
- Iterators help step through objects in a collection or class extent.
- Collections (sets, lists etc.) also provide create_iterator() method.



©Silberschatz, Korth and Sudarshan

ODMG C++ OML: Example of Iterators

```
int print_customers() {
    Database bank_db_obj;
    Database * bank_db = &bank_db_obj;
    bank_db->open (“Bank-DB”);
    d_Transaction Trans; Trans.begin ();

    d_Extent<Customer> all_customers(bank_db);
    d_Iterator<d_Ref<Customer>> iter;
    iter = all_customers–>create_iterator();
    d_Ref <Customer> p;
    while{iter.next (p))
        print_cust (p);  // Function assumed to be defined elsewhere
    Trans.commit();
}
```



©Silberschatz, Korth and Sudarshan

ODMG C++ Binding: Other Features

- Declarative query language OQL, looks like SQL
  - ★ Form query as a string, and execute it to get a set of results (actually a bag, since duplicates may be present)

```
d_Set<d_Ref<Account>> result;
d_OQL_Query q1("select a
                from Customer c, c.accounts a
                where c.name='Jones'
                      and a.find_balance() > 100");
d_oql_execute(q1, result);
```

- Provides error handling mechanism based on C++ exceptions, through class d_Error
- Provides API for accessing the schema of a database.



©Silberschatz, Korth and Sudarshan

Making Pointer Persistence Transparent

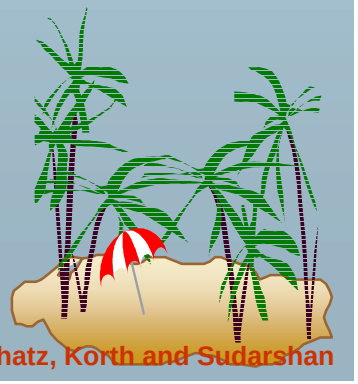
- Drawback of the ODMG C++ approach:
  - Two types of pointers
  - Programmer has to ensure mark_modified() is called, else database can become corrupted
- ObjectStore approach
  - Uses *exactly* the same pointer type for in-memory and database objects
  - Persistence is transparent applications
    - Except when creating objects
  - Same functions can be used on in-memory and persistent objects since pointer types are the same
  - Implemented by a technique called pointer-swizzling which is described in Chapter 11.
  - No need to call mark_modified(), modification detected automatically.



©Silberschatz, Korth and Sudarshan

Persistent Java Systems

- ODMG-3.0 defines extensions to Java for persistence
  - ★ Java does not support templates, so language extensions are required
- Model for persistence: persistence by reachability
  - ★ Matches Java's garbage collection model
  - ★ Garbage collection needed on the database also
  - ★ Only one pointer type for transient and persistent pointers
- Class is made **persistence capable** by running a post-processor on object code generated by the Java compiler
  - ★ Contrast with pre-processor used in C++
  - ★ Post-processor adds mark_modified() automatically
- Defines collection types DSet, DBag, DList, etc.
- Uses Java iterators, no need for new iterator class



©Silberschatz, Korth and Sudarshan

ODMG Java

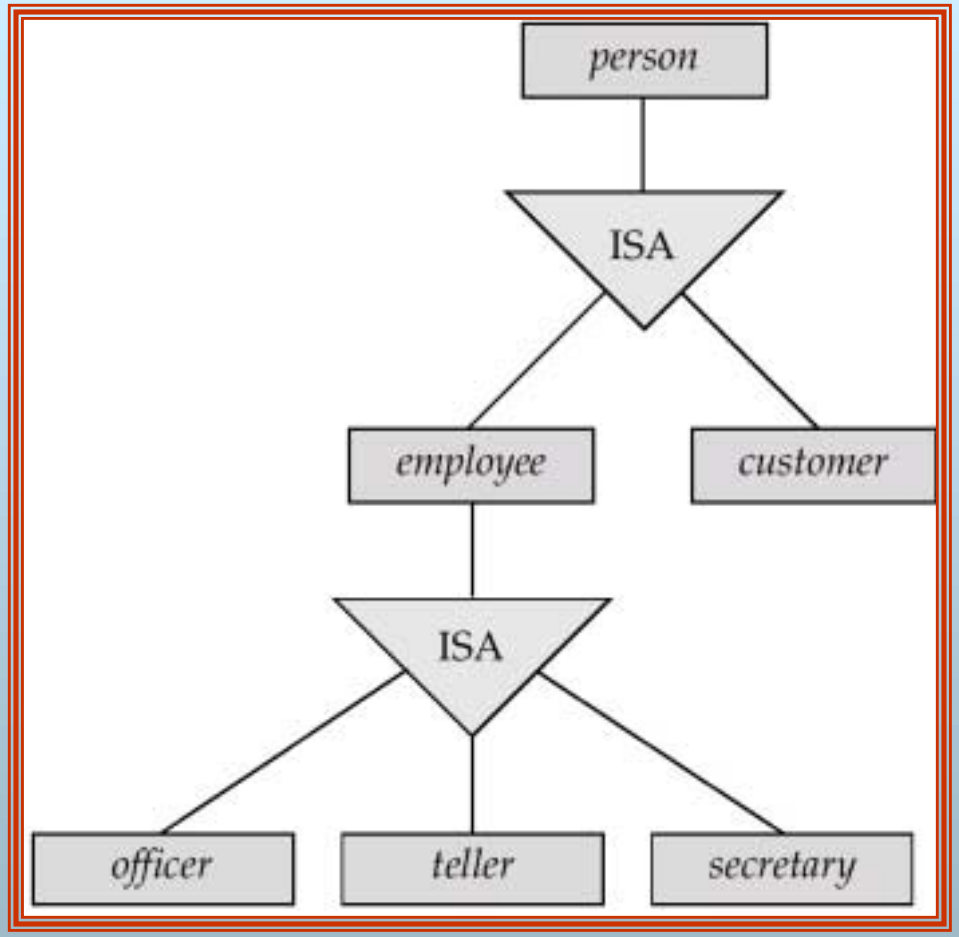
- Transaction must start accessing database from one of the root object (looked up by name)
  - ★ finds other objects by following pointers from the root objects
- Objects referred to from a fetched object are allocated space in memory, but not necessarily fetched
  - ★ Fetching can be done lazily
  - ★ An object with space allocated but not yet fetched is called a hollow object
  - ★ When a hollow object is accessed, its data is fetched from disk.



©Silberschatz, Korth and Sudarshan

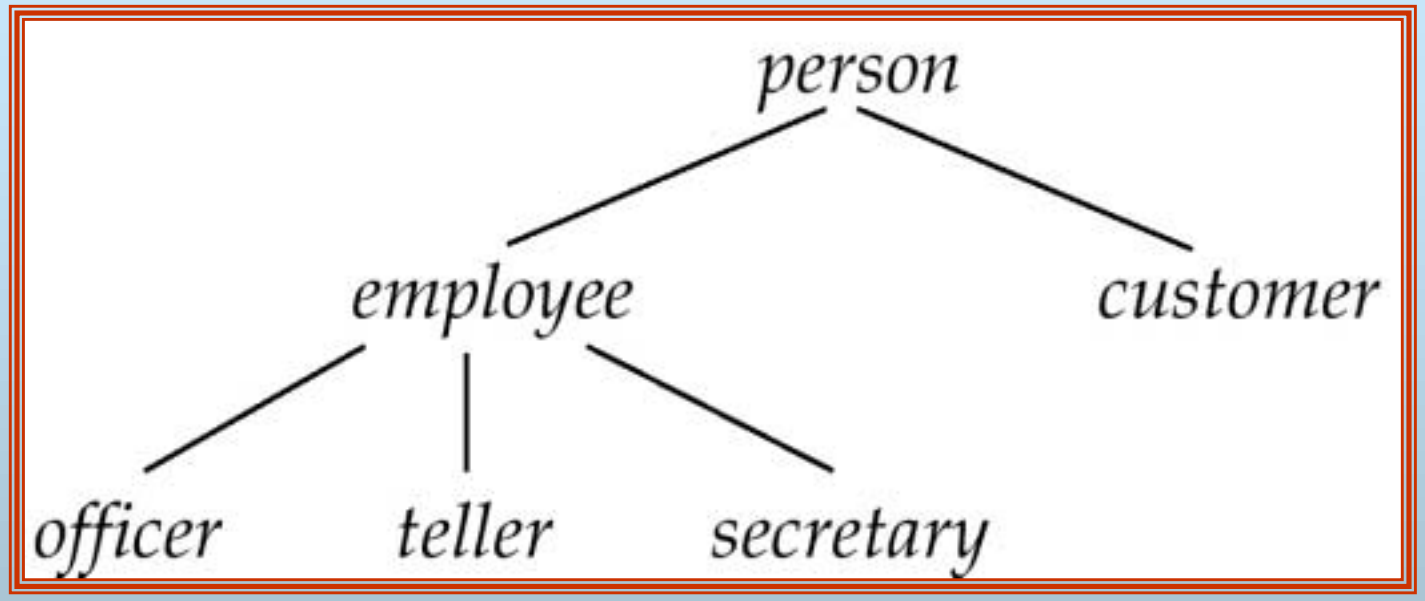
End of Chapter

Specialization Hierarchy for the Bank Example



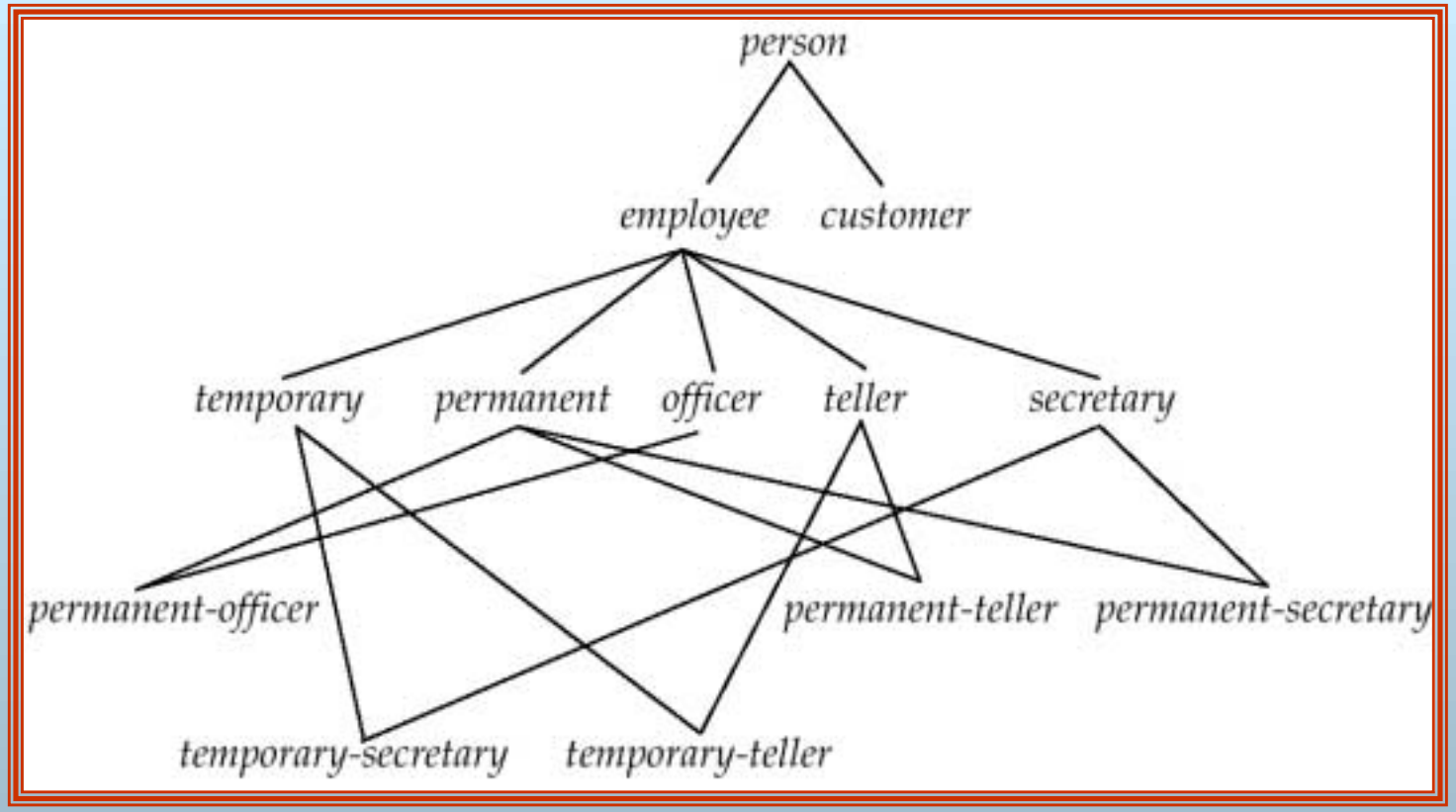
©Silberschatz, Korth and Sudarshan

Class Hierarchy Corresponding to Figure 8.2



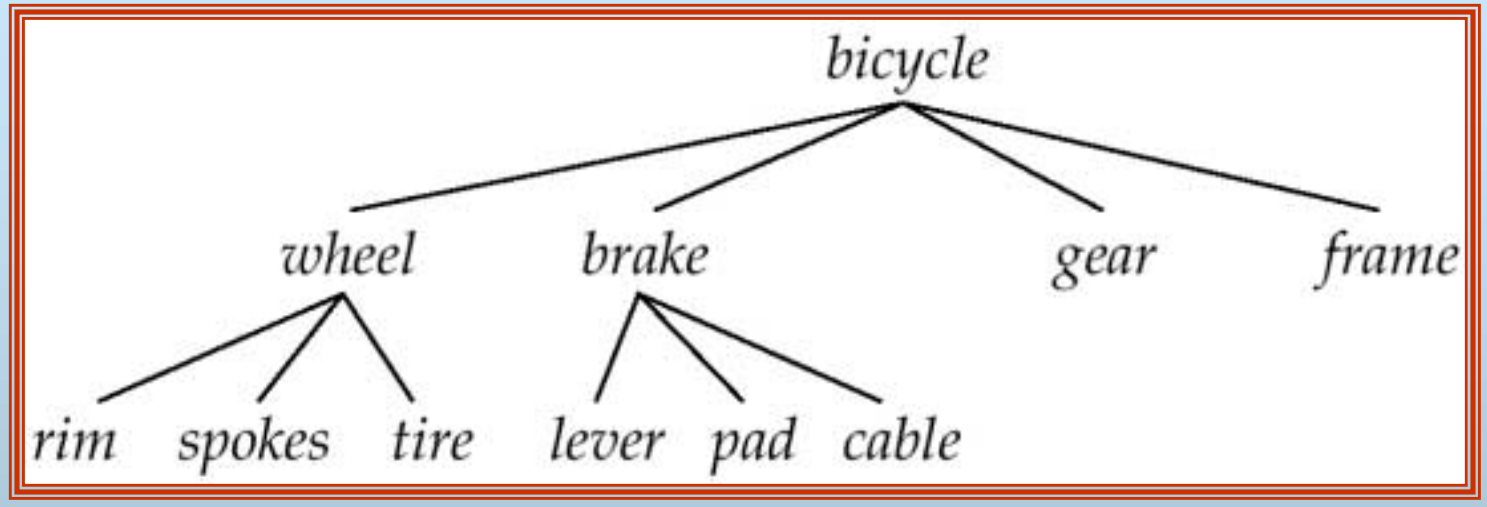
©Silberschatz, Korth and Sudarshan

Class DAG for the Bank Example



©Silberschatz, Korth and Sudarshan

Containment Hierarchy for Bicycle-Design Database



©Silberschatz, Korth and Sudarshan

CACHE : Specifies how many values of a sequence oracle pre-allocates and keeps in memory for faster access.The minimum value for this parameter is two.

NOCHACHE : Specifies that values of a sequence are not pre-allocated.

ORDER : This guarantees that sequence numbers are generated in the order of request.This is only necessary if using parallel server in parallel mode option .In exclusive mode option ,a sequence always generates numbers in order.

NOORDER : This does not guarantee sequence numbers are generated in order of request.This is only necessary if you are using parallel server in parallel mode option. If the ORDER/NOORDER clause is omitted , a sequence takes the NOORDER clause by default.

Example

Create a sequence by the name ADDR_SEQ ,which will generate numbers from 1 uptp 9999 in ascending order with an interval of 1.The sequence must restart from the number 1 after generating number 999.

CREATE SEQUENCE ADDR_SEQ **INCREMENT BY 1 START WITH 1 MINVALUE 1 MAXVALUE 999 CYCLE ;**

Referencing a sequence

Once a sequence is created SQL can be used to view the values held in its cache.To simply view sequence value use a SELECT sentence as described below.

SELECT <SequenceName>.Nextval from DUAL ;

This will display the next value held in the cache on the VDU screen. Everytime nextval references a sequence its output is automatically incremented from the old value to the new value ready for use.

To reference the current value of a sequence:

SELECT <SequenceName>.CurrVal FROM DUAL;

Dropping a Sequence

The DROP SEQUENCE command is used to remove the sequence from the database.

Syntax:

DROP SEQUENCE <SequenceName> ;

2.5 DATA CONTROL LANGUAGE

The data control language commands are related to the security of database. They perform tasks of assigning privilages, so users can access certain objects in the database. This section deals with DCL commands.

1. GRANT Command :

The objects created by one user are not accessible by another user unless the owner of those objects gives such permissions to other users. These permissions can be given by using the **GRANT** statement. One user can grant permission to another user if he is the owner of the object or has the permission to grant access to other users.

The grant statement provides various types of access to database objects such as tables, views and sequences.

Syntax :

GRANT {object privilages}
ON object name
To user name
[with GRANT OPTION]

Object privilages :

Each object privilage that is granted authorizes the grantee to perform some operations on the object. The user can grant all the privilages or grant only specific object privilages.

The list of object privilages is as follows :

Alter - allows the grantee to change the table definition with the ALTER TABLE command.

Delete - allows the grantee to remove the records from the table with the DELETE command.
Index - allows the grantee to create an index on table with the CREATE INDEX command.
Insert - allows the grantee to add records to the table with the INSERT command.
Select - allows the grantee to query the tables with SELECT command.
Update - allows the grantee to modify the records in tables with UPDATE command.
With grant option : It allows the grantee to grant object privilages to other users.

Example 1 : Grant all privilages on student table to user Pradeep.

```
SQL >  GRANT ALL
       ON student
       To Pradeep;
```

Example 2 : Grant select and update privilages on student table to mita

```
SQL>   GRANT SELECT, UPDATE
       ON student
    To Mita;
```

Example 3 : Grant all privilages on student table to user Sachin with grant option.

```
SQL>   GRANT ALL
       ON student
    To Sachin
       WITH GRANT OPTION;
```

2. REVOKE Command :

The REVOKE statement is used to deny the grant given on an object.

Syntax :

```
REVOKE {object privilages}
       ON object name
       FROM user name;
```

The list of object privilages is :

Alter - allows the grantee to change the table definition with the ALTER TABLE command.
Delete - allows the grantee to remove the records from the table with the DELETE command.
Index - allows the grantee to create an index on table with the CREATE INDEX command.
Insert - allows the grantee to add records to the table with the INSERT command.
Select - allows the grantee to query the tables with SELECT command.
Update - allows the grantee to modify the records in tables with UPDATE command.

You cannot use REVOKE command to perform following operations :

1. Revoke the object privilages that you didn't grant to the revokee.
2. Revoke the object privilages granted through the operating system.

Example 1 : Revoke Delete privilege on student table from Pradeep.

```
REVOKE DELETE
ON student
From    Pradeep;
```

Example 2 : Revoke the remaining privilages on student that were granted to Pradeep.

```
Revoke ALL
ON  student
FROM   Pradeep
```

3. COMMIT Command :

Commit command is used to permanently record all changes that the user has made to the database since the last commit command was issued or since the beginning of the database session.

Syntax :

COMMIT;

Implicity COMMIT :

The actions that will force a commit to occur even without your instructing it to are :

quit, exit,
create table or create view
drop table or drop view
grant or revoke
connect or disconnect
alter
audit and non-audit

Using any of these commands is just like using commit. Until you commit, only you can see how your work affects the tables. Anyone else with access to these tables will continue to get the old information.

4. ROLLBACK command :

The ROLLBACK statement does the exact opposite of the commit statement. It ends the transaction but undoes any changes made during the transaction. Rollback is useful for two reasons :

(1) If you have made a mistake, such as deleting the wrong row for a table, you can use rollback to restore the original data. Rollback will take you back to intermediate statement in the current transaction, which means that you do not have to erase the entire transaction.

(2) ROLLBACK is useful if you have started a transaction that you cannot complete. This might occur if you have a logical problem or if there is an SQL statement that does not execute successfully. In such cases rollback allows you to return to the starting point to allow you to take corrective action and perhaps try again.

Syntax : ROLLBACK [WORK] [TO [SAVEPOINT] save point]

where

WORK - is optional and is provided for ANSI compatibility

SAVEPOINT - is optional and is used to rollback a partial transaction, as far as the specified save point.

Savepoint : is a savepoint created during the current transaction.

Using rollback without savepoint clause.

1. Ends the transaction.
2. Undoes all the changes in the current transaction.
3. Erases all savepoints in that transaction
4. Releases the transaction locks.

Using rollback with the to savepoint clause.

1. Rolls back just a portion of the transaction.
2. Retains the savepoint rolled back to, but losses those created after the named savepoint.
3. Releases all tables and row locks that were acquired since the savepoint was taken.

Example :

To rollback entire transaction : ROLLBACK,

To rollback to savepoint sps : ROLLBACK TO SAVEPOINT sps;

Savepoints :

Savepoints mark and save the current point in the current processing of a transaction. Used with the ROLLBACK statement, savepoints can undo part of a transaction.

By default the maximum number of savepoints per transaction is 5. An active savepoint is the one that is specified since the last commit or rollback.

Syntax : SAVEPOINT savepoint :

After a savepoint, is created, you can either continue processing, commit your work rollback the entire transaction, or rollback to the savepoint.

2.6 SELECT QUERY AND CLAUSES

The basic structure of an SQL expression consists of three clauses :

select, from and where,

- The select clause corresponds to the projection operation of the relational algebra.
  It is used to list the attributes desired in the result of a query.
- The from clause corresponds to the cartesian product operation of the relational algebra. It lists the relations to be scanned in the elevation of the expression.
- The where clause corresponds to the selection predicate of the relational algebra.
  It consists of predicate involving attributes of the relations that appear in the from clause.

Simple SQL query i.e. select statement has the form :

select $A_1, A_2, \ldots\ldots, A_n$

from $r_1, r_2, \ldots\ldots, r_m$

where P.

The variables are defined as follows :

$A_1, A_2, \ldots, A_n$ represent the attributes.

$r_1, r_2, \ldots, r_m$ represent the relations from which the attributes are selected.

P - is the predicate.

This query is equivalent to the relational algebra expression

$$\Pi_{A_1 A_2 \ldots A_n} (\sigma_p (r_1 \times r_2 \times \ldots \times r_m))$$

where clause is optional. If the where clause is omitted, the predicate P is true.

Select clause forms the cartesian product of relations named in the *from* clause, performs a relational algebra selection using the *where* clause and then projects the results onto the attributes of the select clause.

The purpose of select statement is to retrieve and display data from one or more database tables It is an extremely powerful statement capable of performing the equivalent relational algebra's Selection, Projection, and Join operations in a single statement. Select is the most frequently used SQL command and has the following general form :

```
SELECT      DISTINCT |ALL]
FROM        Table_Name [alias][,…]
[WHERE      condition]
[GROUP BY   column_List] [HAVING condition]
[ORDER BY   column_List]
```

The sequence of processing in a select statement is :

FROM

WHERE

GROUP BY

HAVING

1. Every SQL statement is terminated by a semicolon.
2. An SQL statement can be entered on one line or split across several lines for clarity.

3. SQL isn't case sensitive. You can mix uppercase and lowercase when referencing SQL keywords (Such as SELECT and INSERT), table names, and column names.

However, case does matter when referencing to the contents of a column.

For Example : If you ask for all customers whose last names begin with 'a' and all customer names are stored in uppercase, you won't receive any rows at all.

4. SELECT Command :

The basic structure of an SQL expression consists of three clauses :

select, from and where

- The select clause corresponds to the projection operation of the relational algebra.
  It is used to list the attributes desired in the result of a query.
- The from clause corresponds to the cartesian product operation of the relational algebra. It lists the relations to be scanned in the elevation of the expression.
- The where clause corresponds to the selection predicate of the relational algebra.
  It consists of predicate involving attributes of the relations that appear in the from clause.

Simple SQL query i.e. select statement has the form :

select $A_1, A_2, \ldots\ldots, A_n$

from $r_1, r_2, \ldots\ldots, r_m$

where P.

The variables are defined as follows :

$A_1, A_2, \ldots, A_n$ represent the attributes.

$r_1, r_2, \ldots, r_m$ represent the relations from which the attributes are selected.

P - is the predicate.

This query is equivalent to the relational algebra expression

$$\Pi_{A_1 A_2 \ldots A_n} (\sigma_p (r_1 \times r_2 \times r_3 \ldots \times r_m))$$

where clause is optional. If the where clause is omitted, the predicate P is true.

Select clause forms the cartesian product of relations named in the *from* clause, performs a relational algebra selection using the *where* clause and then projects the results onto the attributes of the select clause.

A simple select statement :

At a minimum, select statement contains the following two elements.

- The select list, the list of columns to be retrieved.
- The from clause, the tables from which to retrieve the rows.

Example : Consider the student database table.

(1) A simple select statement - a query that retrieves only student_id from the student table is given

```
SQL> select student_id
     from student;
     student_id
     ----------
     S 10231
     S 10232
     S 10233
     S 10234
     S 10235
     S 10236
```

6 rows selected.

(2) To select student_id and students Last name, the select statement is :

```
SQL>  select student_id, First_name
      from student;
```

student_id	First_name
S 10231	Sachin
S 10232	Rahul
S 10233	Ajay
S 10234	Sunil
S 10235	Kapil
S 10236	Anil

6 rows selected.

To select all columns in the table you can use

```
select  *
from table_name;
```

Example :

```
SQL>  select  *
      from student;
```

student_id	Last_name	First_name	B Date	State	City
S 10231	Deshpande	Sachin	12/3/78	Maharashtra	Pune
S 10232	Gandhi	Rahul	9/2/58	Delhi	Delhi
S 10233	Kapur	Ajay	7/12/62	Maharashtra	Bombay
S 10234	Kulkarni	Sunil	6/9/75	Maharashtra	Pune
S 10235	Dev	Kapil	2/3/71	Tamilnadu	Madras
S 10236	Kumar	Anil	5/9/80	Maharashtra	Bombay

The results returned by every SELECT statement constitutes a temporary table. Each received record is a row in this temporary table, and each element of the select list is a column. If a query does not return any record, the temporary - table can be thought of as empty.

Expressions in the select list :

In addition to specifying columns, you also can specify expressions in the select list.

Following arithmetic operators can be used in select list :

Description	Operator
Addition	+
Subtraction	–
Multiplication	*
Division	/

For example, consider the following queries using operators in select list :

```
SQL>Select E_name, Salary * 1000
    from Employee;
```

E_name	Salary * 1000
Sachin	1,00,00,000
Rahul	2,00,00,000
Ajay	1,00,00,000
Anil	1,00,00,000

4 rows selected.

```
SQL>Select Ename, Salary + 10000
    from Employee;
    E_name              Salary + 10000
    Sachin                  20,000
    Rahul                   30,000
    Ajay                20,000
    Anil                30,000
        4 rows selected.
```

Select statement using where clause :

select and *from* clauses provide you with either some columns and all rows or all columns and all rows. But if you want only certain rows, you need to add another clause, the *where* clause.

where clause consists of one or more conditions that must be satisfied before a row is retrieved by the query.

It searches for a condition and narrows your selection of data.

For example, consider select statement with where clause given below :

```
SQL>Select Student_id, First_name
    from Student
    where Student_id = 'S10234';
    Student_id                  First_name
    S10234                      Sunil
    1 row selected
SQL>Select E_name  Salary
    from Employee
    where Salary  > 10000;
    E_name                      Salary
    Rahul                       20000
    Anil                        20000
    2 row selected
```

where uses the logical connectives : **and, or** and **not**.

where clause uses the comparison operators

Description	Operator
Less than	<
Less than or equal to	<=
Greater than	>
Greater than or equal to	>=
Equal to	=
Not equal to	!= or < >

```
SQL>Select E_name, Salary
    from  Employee
    where Salary  >  10000 and E_name  =  Anil
        Ename                               Salary
    Anil                                20000
    1 row selected.
```

5. Views in SQL :

A view in SQL terminology is a single table that is derived from other tables. These other tables could be base tables or previously defined views. A view does not necessarily exist in physical form; it is considered a virtual table in contrast to base tables whose tuples are actually stored in the database. This limits the possible update operations that can be applied to views but does not provide any limitations on querying a view. We can think of view as a way specifying a table that we need not exist physically.

Specification of Views in SQL :

The command to specify a view is CREATE VIEW. 'We give the view a table name, a list of attribute names, and a query to specify the contents of view. If none of the view attributes result from applying functions or arithmetic operations, we do not have to specify attribute names for the view as they will be the same as the names of the attributes of the defining tables.

Example :

Consider the following relation scheme and corresponding relation.

employee_schema (emp_name, street, city)

works_schema (emp_name, comp_name, salary

company_schema (comp_name, city)

emp_name	street	city
Sachin	XYZ	Pune
Rahul	ABC	Bombay
Raj	ABC	Pune
Ajay	XYZ	Bombay
Anil	XYZ	Delhi
Sunil	ABC	Bombay

emp_name	Comp_name	salary
Sachin	TCS	10000
Rahul	MBT	12000
Raj	PCS	13000
Ajay	MBT	14000
Anil	PCS	15000
Sunil	TCS	11000

Comp_name	city
TCS	Delhi
MBT	Bombay
PCS	Pune

```
Create view emp_detail (emp, comp, street, city)
As  select  C.emp_name, C.comp_name, E.street, E.city
    from Employee E.company C
    where E.emp_name = C.emp_name;
```

A view is always up date; if we modify the base tables on which the view is defined, the view automatically reflects these changes. Hence, the view is not realized at the time of view definition but rather at the time we specify a query on the view. It is the responsibility of the DBMS and not the user to make sure that the view is up to date.

If we do not need a view any more, we can use the DROP VIEW command to dispose of it.

```
Drop View emp_detail;
```

Updating of views :

(1) A view with a single defining table is up datable if the view attributes contain the primary key or some other candidate key of the base relation, because this maps each view tuple to a single base tuple.

(2) Views defined on multiple tables using joins are generally not updatable.

(3) Views defined using grouping and aggregate functions are not updatable.

Example :

Consider the view consisting of branch names and names of customers who have either an account or a loan at that branch.

```
SQL>Create view all_customer as
        (select branch_name, customer_name
            from depositor, account
            where depositor·account_number =
        account·account·account_no)
        Union
        (select branch_name, customer_name
            from borrower
            where borrower·loan_number = loan·loan_number);
```

The attribute names of a view can be specified explicitly as follows :

```
SQL>  Create view branch_total_loan (branch_name,
        total_loan) as
      select branch_name, sum (amount)
      from loan
          group by branch_name;
```

6. Indexes in SQL : SQL has statements to create and drop indexes on attributes of base relation. These commands are generally considered to be part of the SQL data definition language (DDL).

An index is a physical access structure that is specified on one or more attributes of the relation. The attributes on which an index is created are termed indexing attributes. An index makes accusing tuples based on conditions that involve its indexing attributes more efficient. This means that in general executing a query will take less time if some attributes involved in the query conditions were indexed than if they were not. This improvement can be dramatic for queries where large relations are involved. In general, if attributes used in selection conditions and in join conditions of a query are indexed, the execution time of the query is greatly improved.

In SQL indexed can be created and dropped dynamically. The create Index command is used to specify an index. Each index is given a name, which is used to drop the index when we do not need it any more.

Example :

```
Create Index  Emp_Index
    ON Employee (Emp_name);
```

In general, the index is arranged in ascending order of the indexing attribute values. If we want the values in descending order we can add the keyword DESC after the attribute name. The default in ASC for ascending. We can also create an index on a combination of attributes.

Example :

```
Create  Index  Emp_Index1
    ON  Employee (Emp_name ASC,
            Comp_name  DESC);
```

There are two additional options on indexes in SQL. The first is to specify the key constraint on the indexing attribute or combination of attributes.

The keyword unique following the CREATE command is used to specify a key. The second option on index creation is to specify whether on index is clustering index. The keyword cluster is used in this case of the end of the create Index command. A base relation can have atmost one clustering index but any number of non_clustering indexes.

To drop an index, we issue the Drop Index command. The reason for dropping indexes is that they are expensive to maintain whenever the base relation is updated and they require additional storage. However, the indexes that specify a key constraint should not be dropped as long as we want the system to continue enforcing that constraint.

Example :

```
Drop Index Emp_Index;
```

7. Sequences

The quickest way to retrieve data from a table is to have a column in the table whose data uniquely identifies a row.By using this column and a specific value in the WHERE condition of a SELECT sentence the oracle engine will be able to identify and retrieve the row the fastest.

To achieve this , a constraint is attached to a specific column in the table that ensures that the column is never left empty and that the data values in the column are unique.Since human beings do data entry,it is quite likely that a duplicate value could be entered ,which violets this constraint and the entire row is rejected.

If the value entered into this column is computer generated it will always fulfill the unique constraint and the row will always be accepted for storage.

Oracle provides an object called a sequence that can generate numeric values. The value generated can have a maximum of 38 digits. A sequence can be defined to:

-Generate numbers in ascending or descending order

-Provide intervals between numbers

-Caching of sequence numbers in memory to speed up their availability

A sequence is an independent object and can be used with any table that requires its output.

Creating Sequences

Always give sequence a name so that it can be referenced later when required.

The minimum information required for generating numbers using a sequence is :

-The starting number

-The maximum number that can be generated by a sequence

- The increment value for generating the next number

This information is provided to oracle at the time of sequence creation

Syntax:

```
CREATE SEQUENCE <SequenceName>
[INCREMENT BY  <IntegerValue>
[START WITH <IntegerValue>
MAXVALUE <IntegerValue> / NOMAXVALUE
MINVALUE <IntegerValue> /NOMINVALUE
CYCLE/NOCYLCLE
CACHE <IntegerValue>/NOCACHE
ORDER/NOORDER ]
```

Keywords and Parameters

INCREMENT BY : -Specifies the interval between sequence numbers. It can be any positive or negative value but not zero.If this clause is omitted ,the default value is 1 .

MINVALUE :- Specifies the sequence minimum value.

NOMINVALUE : Specifies a minimum value of 1 for an ascending sequence and –(10)^26 for a descending sequence.

MAXVALUE: Specifies the maximum value that a sequence can generate.

NOMAXVALUE : Specifies a maximum of 10^27 for an ascending sequence or -1 for a descending sequence. This is the default clause.

START WITH : Speciifes the first sequence number to be generated. The default for an ascending sequence is the sequence minimum value(1) and for a descending sequence, it is the maximum value(-1)

CYCLE: Specifies that the sequence continues to generate repeat values after reaching either its maximum value.

NOCYCLE: Specifies that a sequence cannot generate more values after reaching the maximum value.

CACHE : Specifies how many values of a sequence oracle pre-allocates and keeps in memory for faster access.The minimum value for this parameter is two.

NOCHACHE : Specifies that values of a sequence are not pre-allocated.

ORDER : This guarantees that sequence numbers are generated in the order of request.This is only necessary if using parallel server in parallel mode option .In exclusive mode option ,a sequence always generates numbers in order.

NOORDER : This does not guarantee sequence numbers are generated in order of request.This is only necessary if you are using parallel server in parallel mode option. If the ORDER/NOORDER clause is omitted , a sequence takes the NOORDER clause by default.

Example

Create a sequence by the name ADDR_SEQ ,which will generate numbers from 1 uptp 9999 in ascending order with an interval of 1.The sequence must restart from the number 1 after generating number 999.

CREATE SEQUENCE ADDR_SEQ **INCREMENT BY 1 START WITH 1 MINVALUE 1 MAXVALUE 999 CYCLE ;**

Referencing a sequence

Once a sequence is created SQL can be used to view the values held in its cache.To simply view sequence value use a SELECT sentence as described below.

SELECT <SequenceName>.Nextval from DUAL ;

This will display the next value held in the cache on the VDU screen. Everytime nextval references a sequence its output is automatically incremented from the old value to the new value ready for use.

To reference the current value of a sequence:

SELECT <SequenceName>.CurrVal FROM DUAL;

Dropping a Sequence

The DROP SEQUENCE command is used to remove the sequence from the database.

Syntax:

DROP SEQUENCE <SequenceName> ;

2.5 DATA CONTROL LANGUAGE

The data control language commands are related to the security of database. They perform tasks of assigning privilages, so users can access certain objects in the database. This section deals with DCL commands.

1. GRANT Command :

The objects created by one user are not accessible by another user unless the owner of those objects gives such permissions to other users. These permissions can be given by using the **GRANT** statement. One user can grant permission to another user if he is the owner of the object or has the permission to grant access to other users.

The grant statement provides various types of access to database objects such as tables, views and sequences.

Syntax :

GRANT {object privilages}
ON object name
To user name
[with GRANT OPTION]

Object privilages :

Each object privilage that is granted authorizes the grantee to perform some operations on the object. The user can grant all the privilages or grant only specific object privilages.

The list of object privilages is as follows :

Alter - allows the grantee to change the table definition with the ALTER TABLE command.

CHAPTER 2

DATA MANIPULATION & CONTROL

2.0 Objectives
2.1 Introduction
2.2 Subdivisions of SQL
2.3 Data Definition Language
2.4 Data Manipulation Language Commands
2.5 Data Control Language
2.6 Select Query and Clauses
2.7 Select Statement with Order by Clause
2.8 Group by Clause
2.9 Having Clause
2.10 String Operation
2.11 Distinct Rows
2.12 Rename Operation
2.13 Set Operations
2.14 Aggregate Functions
2.15 Nested Sub Queries
2.16 Embedded SQL
2.17 Dynamic SQL
2.18 Summary
2.19 Check Your Progress - *Answers*
2.20 Questions for Self – Study
2.21 Suggested Readings

2.0 OBJECTIVES

After reading this chapter you will able to

- state SQL, DDL, DML, DCL Statements
- explain Select,group by & having clause
- explain String & set operations
- describe Aggregate Functions
- describe Nested Sub Queries
- describe Embedded & Dynamic SQL

2.1 INTRODUCTION

In this chapter we study the query language : Structured Query Language (SQL) which uses a combination of Relational algebra and Relational calculus.

It is a data sub language used to organize, manage and retrieve data from relational database, which is managed by Relational Database Management System (RDBMS).

Vendors of DBMS like Oracle, IBM, DB2, Sybase, and Ingress use SQL as programming language for their database.

SQL originated with the system R project in 1974 at IBM's San Jose Research Centre.

Original version of SQL was SEQUEL which was an Application Program Interface (API) to the system R project.

The predecessor of SEQUEL was named SQUARE.

SQL-92 is the current standard and is the current version.

The SQL language can be used in two ways :

- Interactively or
- Embedded inside another program.

The SQL is used interactively to directly operate a database and produce the desired results. The user enters SQL command that is immediately executed. Most databases have a tool that allows interactive execution of the SQL language. These include SQL Base's SQL Talk, Oracle's SQL Plus, and Microsoft's SQL server 7 Query Analyzer.

The second way to execute a SQL command is by embedding it in another language such as Cobol, Pascal, BASIC, C, Visual Basic, Java, etc. The result of embedded SQL command is passed to the variables in the host program, which in turn will deal with them. The combination of SQL with a fourth-generation language brings together the best of two worlds and allows creation of user interfaces and database access in one application.

2.2 SUBDIVISIONS OF SQL

Regardless of whether SQL is embedded or used interactively, it can be divided into three groups of commands, depending on their purpose.

- Data Definition Language (DDL).
- Data Manipulation Language (DML).
- Data Control Language (DCL).

Data Definition Language :

Data Definition Language is a part of SQL that is responsible for the creation, updation and deletion of tables. It is responsible for creation of views and indexes also. The list of DDL commands is given below :

CREATE TABLE
ALTER TABLE
DROP TABLE
CREATE VIEW
CREATE INDEX

Data Manipulation Language :

Data manipulation commands manipulate (insert, delete, update and retrieve) data. The DML language includes commands that run queries and changes in data. It includes the following commands :

SELECT
UPDATE
DELETE
INSERT

Data Control Language :

The commands that form data control language are related to the security of the database performing tasks of assigning privileges so users can access certain objects in the database.

The DCL commands are :

GRANT
REVOKE
COMMIT
ROLLBACK

2.3 DATA DEFINITION LANGUAGE

The SQL DDL provides commands for defining relation schemas, deleting relations, creating indices, and modifying relation schemas.

The SQL DDL allows the specification of not only a set of relations but also information about each relation including :

- The schema for each relation.
- The domain of values associated with each attribute.
- The integrity constraints.
- The set of indices to be maintained for each relation.
- The security and authorization information for each relation.
- The physical storage structure of each relation on disk.

Domain/Data Types in SQL :

The SQL - 92 standard supports a variety of built-in domain types, including the following :

(1) Numeric data types include

- Integer numbers of various sizes
  INT or INTEGER
  SMALLINT
- Real numbers of various precision
  REAL
  DOUBLE PRECISION
  FLOAT (n)
- Formatted numbers can be represented by using
  DECIMAL (i, j) or
  DEC (i, j)
  NUMERIC (i, j) or NUMBER (i, j)

where, i - the precision, is the total number of decimal digits

and j - the scale, is the number of digits after the decimal point.

The default for scale is zero and the default for precision is implementation defined.

(2) Character string data types - are either fixed - length or varying - length.

CHAR (n) or CHARACTER (n) - is fixed length character string with user specified length n.

VARCHAR (n) - is a variable length character string, with user - specified maximum length n. The full form of CHARACTER VARYING (n), is equivalent.

(3) Date and Time data types :

There are new data types for date and time in SQL-92.

DATE - It is a calendar date containing year, month and day typically in the form
yyyy : mm : dd

TIME - It is the time of day, in hours, minutes and seconds, typically in the form
HH : MM : SS.

Varying length character strings, date and time were not part of the SQL - 89 standard.

In this section we will study the three Data Definition Language Commands :

CREATE TABLE
ALTER TABLE
DROP TABLE

1. CREATE TABLE Command :

The CREATE TABLE COMMAND is used to specify a new relation by giving it a name and specifying its attributes and constraints.

2.3 DATA DEFINITION LANGUAGE

The SQL DDL provides commands for defining relation schemas, deleting relations, creating indices, and modifying relation schemas.

The SQL DDL allows the specification of not only a set of relations but also information about each relation including :

- The schema for each relation.
- The domain of values associated with each attribute.
- The integrity constraints.
- The set of indices to be maintained for each relation.
- The security and authorization information for each relation.
- The physical storage structure of each relation on disk.

Domain/Data Types in SQL :

The SQL - 92 standard supports a variety of built-in domain types, including the following :

(1) Numeric data types include

- Integer numbers of various sizes
  INT or INTEGER
  SMALLINT
- Real numbers of various precision
  REAL
  DOUBLE PRECISION
  FLOAT (n)
- Formatted numbers can be represented by using
  DECIMAL (i, j) or
  DEC (i, j)
  NUMERIC (i, j) or NUMBER (i, j)

where, i - the precision, is the total number of decimal digits

and j - the scale, is the number of digits after the decimal point.

The default for scale is zero and the default for precision is implementation defined.

(2) Character string data types - are either fixed - length or varying - length.

CHAR (n) or CHARACTER (n) - is fixed length character string with user specified length n.

VARCHAR (n) - is a variable length character string, with user - specified maximum length n. The full form of CHARACTER VARYING (n), is equivalent.

(3) Date and Time data types :

There are new data types for date and time in SQL-92.

DATE - It is a calendar date containing year, month and day typically in the form
yyyy : mm : dd

TIME - It is the time of day, in hours, minutes and seconds, typically in the form
HH : MM : SS.

Varying length character strings, date and time were not part of the SQL - 89 standard.

In this section we will study the three Data Definition Language Commands :

CREATE TABLE
ALTER TABLE
DROP TABLE

1. CREATE TABLE Command :

The CREATE TABLE COMMAND is used to specify a new relation by giving it a name and specifying its attributes and constraints.

The attributes are specified first, and each attribute is given a name, a data type to specify its domain of values and any attribute constraints such as NOT NULL. The key, entity integrity and referential integrity constraints can be specified within the CREATE TABLE statement, after the attributes are declared.

Syntax of create table command :

```
CREATE TABLE table_name (
Column_name 1 data type [NOT NULL],
    :
    :
Column_name n data_type [NOT NULL]);
```

The variables are defined as follows :

If NOT NULL is not specified, the column can have NULL values.

table_name - is the name for the table.

column_name 1 to column_name n - are the valid column names or attributes.

NOT NULL – It specifies that column is mandatory. This feature allows you to prevent data from being entered into table without certain columns having data in them.

Examples of CREATE TABLE Command :

(1) Create Table Employee

```
(E_name          varchar2 (20)      NOT NULL,
B_Date           Date,
Salary           Decimal (10, 12)
Address          Varchar2 (50);
```

(2) Create Table Student

```
(Student_id      Varchar2 (20)      Not Null,
Last_Name        Varchar2 (20)      Not Null,
First_name       Varchar2 (20),
BDate            Date,
State            Varchar2 (20),
City  Varchar2 (20));
```

(3) Create Table Course

```
(Course_id       Varchar2 (5),
Department_id    Varchar2 (20),
Title            Varchar2 (20),
Description      Varchar2 (20));
```

Constraints in CREATE TABLE Command :

CREATE TABLE Command lets you enforce several kinds of constraints on a table : primary key, foreign key and check condition, unique condition.

A constraint clause can constrain a single column or group of columns in a table. There are two ways to specify constraints :

- As part of the column definition i.e. a *column constraint*.
- Or at the end of the create table command i.e. a *table constraint*.

Clauses that constrain several columns are the table constraints.

The Primary Key :

A table's primary key is the set of columns that uniquely identifies each row in the table. CREATE TABLE command specifies the primary key as follows :

```
create table table_name (
Column_name 1 data_type  [not null],
    :
    :
Column_name n data type  [NOT NULL],
[Constraint constraint_name]
```

[Primary key (Column_name A, Column_name B… Column_name X)]);

Variables are defined as follows :
table_name is the name for the table.
column_name 1 through column_name n are the valid column names
data_type is valid datatype
constraint which is optional
constraint_name identifies the primary key
column_name A through column_name X are the table's columns that compose the primary key.

Example :

```
Create table Employee
    (E_name          Varchar2 (20),
    B_Date           Date,
    Salary           Decimal (10, 2),
    Address          Varchar2 (80),
    Constraint       PK_Employee
    Primary key (Ename));
Create table_student
    (Student_id      Varchar2 (20),
    Last_name        Varchar2 (20)      NOT NULL,
    First_name       Varchar2 (20),
    B_Date           Date,
    State            Varchar2 (20),
    City             Varchar2 (20),
    Constraint       PK_Student
    Primary key      Student_id));
Create Table_Course
    (Course_id       Varchar2 (5),
    Department_id    Varchar2 (20),
    Title            Varchar2 (20),
    Description      Varchar2 (20),
    Constraint       PK_Course
    Primary key (Course_id, Department_id));
```

Note : We do not specify NOT NULL constraint for those columns which form the primary key, since those are the mandatory columns by default. Primary keys are subject to several restrictions.

(i) A column that is a part of the primary key cannot be NULL.
(ii) A column that is defined as LONG, or LONG RAW (ORACLE data types) cannot be a part of primary key.
(iii) The maximum number of columns in the primary key is 16.

Foreign Key : A foreign key is a combination of columns with values based on the primary key values from another table. A foreign key constraint also known as a referential integrity constraint, specifies that the values of the foreign key correspond to actual values of primary key in other table.

Create table command specifies the foreign key as follows :

```
Create Table table_name
(Column_name 1          data type [NOT NULL],
```

:

:

Column_name N data type [NOT NULL],

[constraint constraint_name

Foreign key (column_name F1 … Column_name FN) references referenced-table (column_name P1, … column_name P_N)]);

table_name - is the name for the table.
Column_name 1 through column_name N are the valid columns.
constraint_name is the name given to foreign key.
referenced_table - is the name of the table referenced by the foreign key declaration.
column_name F_1 through column_name F_N are the columns that compose the foreign key.
Column_name P_1 through column_name P_N are the columns that compose the primary key in referenced-table.

Examples :

```
Create table_department
    (Department_id          Varchar2 (20),
    Department_name         Varchar2 (20),
    Constraint              PK_Department
    Primary key             (Department_id));
Create table_course
    (Course_id              Varchar2 (20),
    Department_id           Varchar2 (20),
    Title                   Varchar2 (20),
    Description             Varchar2 (20),
    Constraint              PK_course
    Primary key (Course_id, Department_id),
    Constraint              FK - course
Foreign key (Department_id) references Department (Department_id));
```

Thus, primary key of course table is (Course_id, Department_id).

The primary key of Department table is (Department_id).

Foreign key of course table is (Department_id) which references the department table.

When you define a foreign key, the DBMS verifies the following :

(1) A primary key has been defined for table referenced by the foreign key.
(2) The number of columns composing the foreign key matches the number of primary key columns in the referenced table.
(3) The datatype and width of each foreign key columns matches the datatype and width of each primary key column in the referenced table.

Unique Constraint or Candidate key :

A candidate key is a combination of one or more columns, the values of which uniquely identify each row of the table. Create table command specifies the unique constraint as follows :

```
CREATE  TABLE  table_name
    (column_name 1      data_type           [NOT NULL],
        :
        :
    column_name n       data_type           [NOT NULL],
    [constraint         constraint_name
    Unique  (Column_name A,……. Column_nameX)]);
```

Example :

```
Create table student
    (Student_id         Varchar2 (20),
    Last_name           Varchar2 (20),          NOT NULL,
    First_name          Varchar2 (20),          NOT NULL,
    BDate               Date,
    State               Varchar2 (20),
    City                Varchar2 (20),
    Constraint          UK-student
    Unique              (last_name, first_name),
    Constraint          PK-student
    Primary key         (Student_id));
```

A unique constraint is not a substitute for a primary key. Two differences between primary key and unique constraints are :

(1) A table can have only one primary key, but it can have many unique constraints.

(2) When a primary key is defined, the columns that compose the primary key are automatically mandatory. When a unique constraint is declared, the columns that compose the unique constraint are not automatically defined to be mandatory, you must also specify that the column is NOT NULL.

Check Constraint :

Using CHECK constraint SQL can specify the data validation for column during table creation. CHECK clause is a Boolean condition that is either TRUE or FALSE. If the condition evaluates to TRUE, the column value is accepted by database, if the condition evaluates to FALSE, database will return an error code.

The check constraint is declared in CREATE TABLE statement using the syntax :

Column_name datatype [constraint constraint_name] [CHECK (Condition)]

The variables are defined as follows :

Column_name - is the column name

data_type - is the column's data type

constraint_name - is the name given to check constraint condition is the legal SQL

Condition that returns a Boolean value.

Examples :

```
Create table_worker
    (NameVarchar2 (25)                          NOT NULL,
    Age         Number      Constraint CK_worker
    CHECK  (Age  Between  18 AND  65)  );
Create table_instructor
    (Instructor_id      Varchar2 (20),
    Department_id       Varchar2 (20)           NOT NULL,
    Name                Varchar2 (25),
    Position            Varchar2 (25)
    Constraint          CK_instructor
CHECK (Position in ('ASSISTANT PROFESSOR', 'ASSOCIATE PROFESSOR', 'PROFESSOR')),
    Address             Varchar2 (25),
    Constraint          PK_instructor
    Primary  key        (Instructor_id));
```

If the position of the instructor is not one of the three legal values, DBMS will return an error code indicating that a check constraint has been violated.

More than one column can have check constraint.

```
Create table_Patient
```

```
(Patient_id          Varchar2 (25)          Primary key,
    Body_Temp        Number (4, 1)
    Constraint       Patient_BT
    CHECK (Body_Temp >= 60.0 and
    Body_Temp <= 110.0),
    Insurance_StatusChar(1)
    Constraint              Patient_IS
    CHECK  (Insurance-Status in ('Y', 'y', 'N', 'n')));
```

One column can have more than one CHECK constraint.

```
Create table_Loan - application
    (loan_app_no          number (6)        primary key,
    Name                  Varchar2 (20),
    Amount_requestednumber (9, 2)            NOT NULL,
    Amount_approvednumber (9, 2)
    Constraint            Amount_approved_limit
    Check (Amount_approved <= 10,00,000)
    Constraint Amount_Approved_Interval
    Check (Mod (Amount_Approved, 1000) = 0));
```

Establishing a Default value for a column :

By using DEFAULT clause when defining a column, you can establish a default value for that column. This default value is used for a column, whenever, row is inserted into the table without specifying the column in the INSERT statement.

Example :

```
Create table_student
    (Student_id          Varchar2 (20),
    Last_name            Varchar2 (20)          NOT NULL,
    First_name           Varchar2 (20)          NOT NULL,
    B_Date               Date,
    State                Varchar2 (20),
    City                 Varchar2 (20),         DEFAULT  'PUNE'.
    Constraint           PK_student
    Primary key    (Student_id);
```

2. ALTER TABLE Command :

You can modify a table's definition using ALTER TABLE command. This statement changes the structure of a table, not its contents. Using ALTER TABLE command, you can make following changes to the table.

(1) Adding a new column to an existing table.

```
ALTER  TABLE  table_name
ADD (Column_name                          datatype
    :
    :
  Column_name n                           datatype);
```

Example :

```
SQL>    Describe Department;
        Name                    NULL?          Type
        Department_id                  Varachar2 (20)
        Department_name                Varachar2 (20)
SQL>    Alter table Department         ADD (University
        Varchar2 (20),
```

```
        No_of_student  Number (3));
SQL>    Describe  Department;
        Name              Null              Type
        Department_id                       Varachar2 (20)
        Department_Name                     Varachar2 (20)
        University                          Varachar2 (20)
        No_of_student                       Varachar2 (20)
```

(2) Modify an existing column in the existing table.

```
ALTER  TABLE  table_name
MODIFY  (Column_name        datatype  :  constraint,
...     Column_name         datatype  :  constraint,);
```

A column in the table can be modified in following ways -

(i) Changing a column definition from NOT NULL to NULL i.e. from mandatory to optional

Consider a table ex_table.

```
SQL>    describe ex_table;
        Name            NULL?              Type
        Record_no       NOTNULL            Numbers (38)
        Description     Varchar2 (40)
        Current_value   NOT NULL              Number
SQL>    Alter Table ex_table;
        modify (current_value number       Null);
        Table altered
SQL>    Describe ex_table;
        Name            NULL?              Type
        Record_No       NOT NULL           Number (38)
        Description     Varchar2 (40)
        Current_value                      Number
```

(ii) Changing a column definition from NULL to NOT NULL.

If a table is empty, you can define a column to be NOT NULL. However, if table is not empty, you cannot change a column to NOT NULL unless every row in the table has a value for that particular column.

(iii) Increasing and Decreasing a Column's Width :

You can increase a character column's width and can increase the number of digits in a number column at any time.

Example :

```
SQL>    Describe ex_table;
        Name                          NULL ?              Type
        Record_No NOT NULL            Number (38)
        Description Varchar2 (40)
        Current_value                 NOT NULL            Number
SQL>    Alter table ex_table
        modify (Description           Varchar2 (50));
        Table altered
SQL>    Describe ex_table;
        Name                          NULL ?              Type
        Record_No                     NOT NULL            Number (38)
        Description                   Varchar2 (50)
        Current_value                 NOT NULL            Number
```

You can decrease a column's width only if the table is empty or if that column is NULL for every row of table.

(3) Adding a constraint to an existing table :

Any constraint i.e. a primary key, foreign key, unique key or check constraint can be added to an existing table using ALTER TABLE command.

```
ALTER  TABLE  table_name
```

ADD (constraint)

Example :

```
SQL>  Create Table ex_table
      (Record_No     Number (38),
      Description    Varchar2 (40),
      Current_value Number);
      Table created
SQL>  Alter Table ex_table  add
      (Constraint PK_ex_table  primary key (Record-No));
      Table Altered.
```

(4) Dropping the constraints

```
ALTER  TABLE  table_name
DROP  Primary key
Using this you can drop primary key of table.
ALTER  TABLE  Table_name
DROP  constraint  constraint_name
```

Using this you can drop any constraint of the table.

Rules for adding or modifying a column :

Following are the rules for adding column to a table :

(1) You may add a column at any time if NOT NULL is not specified.

(2) You may add a NOT NULL column in three steps :

(i) Add a column without NOT NULL specified,

(ii) Fill every row in that column with data,

(iii) Modify the column to be NOT NULL.

Following are the rules to modify a column.

(1) You can increase a character column's width at any time.

(2) You can increase the number of digits in a NUMBER column at any time.

(3) You can increase or decrease the number of places in a NUMBER column at any time.

If a column is NULL for every row of the table, you can make following changes.

(i) You can change its data type

(ii) You can decrease a character column's width

(iii) You can decrease the number of digits in a NUMBER column.

3. DROP TABLE Command :

Dropping a table means to remove the table's definition from the database. DROP TABLE command is used to drop the table as follows :

```
DROP  TABLE  table_name;
```

Example :

```
(1)SQL >    Drop table_student;
      Table dropped
(2)SQL >    Drop table instructor;
      Table dropped.
```

You drop a table only when you no longer need it.

Note : The truncate command in ORACLE can also be used to remove only the rows or data in the table and not the table definition.

Example :

```
Truncate student
Table truncated
```

Truncating cannot be rolled back.

2.4 DATA MANIPULATION LANGUAGE COMMANDS

The SQL DML includes commands to insert tuples into database, to delete tuples from database and to modify tuples in the database.

ADD (constraint)

Example :

```
SQL>  Create Table ex_table
      (Record_No     Number (38),
      Description    Varchar2 (40),
      Current_value Number);
      Table created
SQL>  Alter Table ex_table  add
      (Constraint PK_ex_table  primary key (Record-No));
      Table Altered.
```

(4) Dropping the constraints

```
ALTER  TABLE  table_name
DROP  Primary key
Using this you can drop primary key of table.
ALTER  TABLE  Table_name
DROP  constraint  constraint_name
```

Using this you can drop any constraint of the table.

Rules for adding or modifying a column :

Following are the rules for adding column to a table :

(1) You may add a column at any time if NOT NULL is not specified.

(2) You may add a NOT NULL column in three steps :

(i) Add a column without NOT NULL specified,

(ii) Fill every row in that column with data,

(iii) Modify the column to be NOT NULL.

Following are the rules to modify a column.

(1) You can increase a character column's width at any time.

(2) You can increase the number of digits in a NUMBER column at any time.

(3) You can increase or decrease the number of places in a NUMBER column at any time.

If a column is NULL for every row of the table, you can make following changes.

(i) You can change its data type

(ii) You can decrease a character column's width

(iii) You can decrease the number of digits in a NUMBER column.

3. DROP TABLE Command :

Dropping a table means to remove the table's definition from the database. DROP TABLE command is used to drop the table as follows :

```
DROP  TABLE  table_name;
```

Example :

```
(1)SQL >    Drop table_student;
      Table dropped
(2)SQL >    Drop table instructor;
      Table dropped.
```

You drop a table only when you no longer need it.

Note : The truncate command in ORACLE can also be used to remove only the rows or data in the table and not the table definition.

Example :

```
Truncate student
Table truncated
```

Truncating cannot be rolled back.

2.4 DATA MANIPULATION LANGUAGE COMMANDS

The SQL DML includes commands to insert tuples into database, to delete tuples from database and to modify tuples in the database.

It includes a query language based on both relational algebra and tuple relational calculus.

In this section we'll study following SQL DML commands.

INSERT
DELETE
UPDATE
SELECT

1. INSERT Command :

The syntax of insert statement is :

```
INSERT INTO table_name
[(column_name [ , column_name] …… [ , column_name])]
VALUES
(column_value [ , column_value] …… [ , column_value]);
```

The variables are defined as follows :

Table_name - is the table in which to insert the row.

column_name - is a column belonging to table.

column_value - is a literal value or an expression whose type matches the corresponding column_name.

The number of columns in the list of column_names must match the number of literal values or expressions that appear in parenthesis after the keyword *values*.

Example :

```
SQL>  Insert into Employee
      (E_name, B_Date, Salary, Address)
      Values
      ('Sachin', '21-MAR-73', 50000.00, 'Mumbai');
      row created
SQL>  Insert into student
      (Student_id, Last_name, First_name)
      Values
      ('SE201', 'Tendulkar', 'Sachin');
      row created
```

If the column names specified in Insert statement are more than values, then it returns an error.

Column and value datatype must match.

According to the syntax of INSERT statement, column list is an optional element. Therefore, if you do not specify the column names to be assigned values, it (DBMS) by default uses all the columns. The column order that DBMS uses is the order in which the columns were specified, when the table was created. However, use of Insert statement without column list is dangerous.

For example,

```
SQL>  Describe ex_class;
      Name                 NULL ?      Type
      Class_building       NOT NULL    Varchar2 (25)
      Class_room           NOT NULL    Varchar2 (25)
      Seating_capacity     Number (38)
SQL>  Insert into ex_class
      Values
      ('250', 'Kothrud Pune', 500);
      1 row created.
```

The row is successfully inserted into the table, because, value and column data types were matching.

But the value 250 is not a correct value for column class_building.

The use of insert without column list may cause following problems.

1. The table definition might change, the number of columns might decrease or increase, and the INSERT fails as a result.
2. The INSERT statement might succeed but the wrong data could be entered in the table.

2. DELETE Command :

The syntax of delete statement is :

```
DELETE FROM table_name
[WHERE condition]
```

The variables are defined as follows :

table_name - is the table to be updated.

condition - is a valid SQL condition.

DELETE Command without WHERE clause will empty the table completely.

Example :

```
SQL> Delete from Student
     Where Student_id = 'SE 201';
     1 row deleted.
SQL> Detete from student
     Where first_name = 'Sachin' and
     Student_id ='SE 202';
     1 row deleted.
```

3. UPDATE Command :

If you want to modify existing data in the database, UPDATE command can be used to do that. With this statement you can update zero or more rows in a table.

The syntax of UPDATE command is :

```
UPDATE table_name
SET column_name : : expression
    [, column_name : : expression]
    [, column_name : : expression]
    [where condition]
```

The variables are defined as follows :

table_name is the table to be updated

column_name is a column in the table being updated.

expression is a valid SQL expression.

condition is a valid SQL condition.

The UPDATE statement references a single table and assigns an expression to at least one column. The WHERE clause is optional; if an UPDATE statement does not contain a WHERE clause, the assignment of a value to a column will be applied to all rows in the table.

Example :

```
SQL> Update Student
     Set
     City = 'Pune',
     State = 'Maharashtra';
SQL> Update Instructor
     Set
     Position = 'Professor'
     where
     Instructor_id = 'P3021';
```

SQL Grammar :

Here, are some grammatical requirements to keep in mind when you are working with SQL.

1. Every SQL statement is terminated by a semicolon.
2. An SQL statement can be entered on one line or split across several lines for clarity.

3. SQL isn't case sensitive. You can mix uppercase and lowercase when referencing SQL keywords (Such as SELECT and INSERT), table names, and column names.

However, case does matter when referencing to the contents of a column.

For Example : If you ask for all customers whose last names begin with 'a' and all customer names are stored in uppercase, you won't receive any rows at all.

4. SELECT Command :

The basic structure of an SQL expression consists of three clauses :

select, from and where

- The select clause corresponds to the projection operation of the relational algebra.
  It is used to list the attributes desired in the result of a query.
- The from clause corresponds to the cartesian product operation of the relational algebra. It lists the relations to be scanned in the elevation of the expression.
- The where clause corresponds to the selection predicate of the relational algebra.
  It consists of predicate involving attributes of the relations that appear in the from clause.

Simple SQL query i.e. select statement has the form :

select $A_1, A_2, \ldots\ldots, A_n$

from $r_1, r_2, \ldots\ldots, r_m$

where P.

The variables are defined as follows :

$A_1, A_2, \ldots, A_n$ represent the attributes.

$r_1, r_2, \ldots, r_m$ represent the relations from which the attributes are selected.

P - is the predicate.

This query is equivalent to the relational algebra expression

$$\Pi_{A_1 A_2 \ldots A_n} (\sigma_p (r_1 \times r_2 \times r_3 \ldots \times r_m))$$

where clause is optional. If the where clause is omitted, the predicate P is true.

Select clause forms the cartesian product of relations named in the *from* clause, performs a relational algebra selection using the *where* clause and then projects the results onto the attributes of the select clause.

A simple select statement :

At a minimum, select statement contains the following two elements.

- The select list, the list of columns to be retrieved.
- The from clause, the tables from which to retrieve the rows.

Example : Consider the student database table.

(1) A simple select statement - a query that retrieves only student_id from the student table is given

```
SQL> select student_id
     from student;
     student_id
     ----------
     S 10231
     S 10232
     S 10233
     S 10234
     S 10235
     S 10236
```

6 rows selected.

(2) To select student_id and students Last name, the select statement is :

```
SQL>  select student_id, First_name
      from student;
```

student_id	First_name
S 10231	Sachin
S 10232	Rahul
S 10233	Ajay
S 10234	Sunil
S 10235	Kapil
S 10236	Anil

6 rows selected.

To select all columns in the table you can use

```
select  *
from table_name;
```

Example :

```
SQL>  select  *
      from student;
```

student_id	Last_name	First_name	B Date	State	City
S 10231	Deshpande	Sachin	12/3/78	Maharashtra	Pune
S 10232	Gandhi	Rahul	9/2/58	Delhi	Delhi
S 10233	Kapur	Ajay	7/12/62	Maharashtra	Bombay
S 10234	Kulkarni	Sunil	6/9/75	Maharashtra	Pune
S 10235	Dev	Kapil	2/3/71	Tamilnadu	Madras
S 10236	Kumar	Anil	5/9/80	Maharashtra	Bombay

The results returned by every SELECT statement constitutes a temporary table. Each received record is a row in this temporary table, and each element of the select list is a column. If a query does not return any record, the temporary - table can be thought of as empty.

Expressions in the select list :

In addition to specifying columns, you also can specify expressions in the select list.

Following arithmetic operators can be used in select list :

Description	Operator
Addition	+
Subtraction	–
Multiplication	*
Division	/

For example, consider the following queries using operators in select list :

```
SQL>Select E_name, Salary * 1000
    from Employee;
```

E_name	Salary * 1000
Sachin	1,00,00,000
Rahul	2,00,00,000
Ajay	1,00,00,000
Anil	1,00,00,000

4 rows selected.

```
SQL>Select Ename, Salary + 10000
    from Employee;
    E_name              Salary + 10000
    Sachin                  20,000
    Rahul                   30,000
    Ajay                20,000
    Anil                30,000
        4 rows selected.
```

Select statement using where clause :

select and *from* clauses provide you with either some columns and all rows or all columns and all rows. But if you want only certain rows, you need to add another clause, the *where* clause.

where clause consists of one or more conditions that must be satisfied before a row is retrieved by the query.

It searches for a condition and narrows your selection of data.

For example, consider select statement with where clause given below :

```
SQL>Select Student_id, First_name
    from Student
    where Student_id = 'S10234';
    Student_id                  First_name
    S10234                      Sunil
    1 row selected
SQL>Select E_name  Salary
    from Employee
    where Salary  > 10000;
    E_name                      Salary
    Rahul                       20000
    Anil                        20000
    2 row selected
```

where uses the logical connectives : **and, or** and **not**.

where clause uses the comparison operators

Description	Operator
Less than	<
Less than or equal to	<=
Greater than	>
Greater than or equal to	>=
Equal to	=
Not equal to	!= or < >

```
SQL>Select E_name, Salary
    from  Employee
    where Salary  >  10000 and E_name  =  Anil
        Ename                               Salary
    Anil                                20000
    1 row selected.
```

5. Views in SQL :

A view in SQL terminology is a single table that is derived from other tables. These other tables could be base tables or previously defined views. A view does not necessarily exist in physical form; it is considered a virtual table in contrast to base tables whose tuples are actually stored in the database. This limits the possible update operations that can be applied to views but does not provide any limitations on querying a view. We can think of view as a way specifying a table that we need not exist physically.

```
(Patient_id          Varchar2 (25)          Primary key,
    Body_Temp        Number (4, 1)
    Constraint       Patient_BT
    CHECK (Body_Temp >= 60.0 and
    Body_Temp <= 110.0),
    Insurance_StatusChar(1)
    Constraint              Patient_IS
    CHECK  (Insurance-Status in ('Y', 'y', 'N', 'n')));
```

One column can have more than one CHECK constraint.

```
Create table_Loan - application
    (loan_app_no          number (6)        primary key,
    Name                  Varchar2 (20),
    Amount_requestednumber (9, 2)            NOT NULL,
    Amount_approvednumber (9, 2)
    Constraint            Amount_approved_limit
    Check (Amount_approved <= 10,00,000)
    Constraint Amount_Approved_Interval
    Check (Mod (Amount_Approved, 1000) = 0));
```

Establishing a Default value for a column :

By using DEFAULT clause when defining a column, you can establish a default value for that column. This default value is used for a column, whenever, row is inserted into the table without specifying the column in the INSERT statement.

Example :

```
Create table_student
    (Student_id          Varchar2 (20),
    Last_name            Varchar2 (20)          NOT NULL,
    First_name           Varchar2 (20)          NOT NULL,
    B_Date               Date,
    State                Varchar2 (20),
    City                 Varchar2 (20),         DEFAULT  'PUNE'.
    Constraint           PK_student
    Primary key    (Student_id);
```

2. ALTER TABLE Command :

You can modify a table's definition using ALTER TABLE command. This statement changes the structure of a table, not its contents. Using ALTER TABLE command, you can make following changes to the table.

(1) Adding a new column to an existing table.

```
ALTER  TABLE  table_name
ADD (Column_name                          datatype
    :
    :
  Column_name n                           datatype);
```

Example :

```
SQL>    Describe Department;
        Name                    NULL?          Type
        Department_id                  Varachar2 (20)
        Department_name                Varachar2 (20)
SQL>    Alter table Department         ADD (University
        Varchar2 (20),
```

```
        No_of_student  Number (3));
SQL>    Describe  Department;
        Name              Null              Type
        Department_id                       Varachar2 (20)
        Department_Name                     Varachar2 (20)
        University                          Varachar2 (20)
        No_of_student                       Varachar2 (20)
```

(2) Modify an existing column in the existing table.

```
ALTER  TABLE  table_name
MODIFY  (Column_name        datatype  :  constraint,
...     Column_name         datatype  :  constraint,);
```

A column in the table can be modified in following ways -

(i) Changing a column definition from NOT NULL to NULL i.e. from mandatory to optional

Consider a table ex_table.

```
SQL>    describe ex_table;
        Name            NULL?              Type
        Record_no       NOTNULL            Numbers (38)
        Description     Varchar2 (40)
        Current_value   NOT NULL              Number
SQL>    Alter Table ex_table;
        modify (current_value number       Null);
        Table altered
SQL>    Describe ex_table;
        Name            NULL?              Type
        Record_No       NOT NULL           Number (38)
        Description     Varchar2 (40)
        Current_value                      Number
```

(ii) Changing a column definition from NULL to NOT NULL.

If a table is empty, you can define a column to be NOT NULL. However, if table is not empty, you cannot change a column to NOT NULL unless every row in the table has a value for that particular column.

(iii) Increasing and Decreasing a Column's Width :

You can increase a character column's width and can increase the number of digits in a number column at any time.

Example :

```
SQL>    Describe ex_table;
        Name                          NULL ?              Type
        Record_No NOT NULL            Number (38)
        Description Varchar2 (40)
        Current_value                 NOT NULL            Number
SQL>    Alter table ex_table
        modify (Description           Varchar2 (50));
        Table altered
SQL>    Describe ex_table;
        Name                          NULL ?              Type
        Record_No                     NOT NULL            Number (38)
        Description                   Varchar2 (50)
        Current_value                 NOT NULL            Number
```

You can decrease a column's width only if the table is empty or if that column is NULL for every row of table.

(3) Adding a constraint to an existing table :

Any constraint i.e. a primary key, foreign key, unique key or check constraint can be added to an existing table using ALTER TABLE command.

```
ALTER  TABLE  table_name
```

ADD (constraint)

Example :

```
SQL>  Create Table ex_table
      (Record_No     Number (38),
      Description    Varchar2 (40),
      Current_value Number);
      Table created
SQL>  Alter Table ex_table  add
      (Constraint PK_ex_table  primary key (Record-No));
      Table Altered.
```

(4) Dropping the constraints

```
ALTER  TABLE  table_name
DROP  Primary key
Using this you can drop primary key of table.
ALTER  TABLE  Table_name
DROP  constraint  constraint_name
```

Using this you can drop any constraint of the table.

Rules for adding or modifying a column :

Following are the rules for adding column to a table :

(1) You may add a column at any time if NOT NULL is not specified.

(2) You may add a NOT NULL column in three steps :

(i) Add a column without NOT NULL specified,

(ii) Fill every row in that column with data,

(iii) Modify the column to be NOT NULL.

Following are the rules to modify a column.

(1) You can increase a character column's width at any time.

(2) You can increase the number of digits in a NUMBER column at any time.

(3) You can increase or decrease the number of places in a NUMBER column at any time.

If a column is NULL for every row of the table, you can make following changes.

(i) You can change its data type

(ii) You can decrease a character column's width

(iii) You can decrease the number of digits in a NUMBER column.

3. DROP TABLE Command :

Dropping a table means to remove the table's definition from the database. DROP TABLE command is used to drop the table as follows :

```
DROP  TABLE  table_name;
```

Example :

```
(1)SQL >    Drop table_student;
      Table dropped
(2)SQL >    Drop table instructor;
      Table dropped.
```

You drop a table only when you no longer need it.

Note : The truncate command in ORACLE can also be used to remove only the rows or data in the table and not the table definition.

Example :

```
Truncate student
Table truncated
```

Truncating cannot be rolled back.

2.4 DATA MANIPULATION LANGUAGE COMMANDS

The SQL DML includes commands to insert tuples into database, to delete tuples from database and to modify tuples in the database.

2

SQL Operators

This document discusses SQL operators used with Oracle Database Lite. Topics include:

Section 2.1, "SQL Operators Overview"

Section 2.2, "Arithmetic Operators"

Section 2.3, "Character Operators"

Section 2.4, "Comparison Operators"

Section 2.5, "Logical Operators"

Section 2.6, "Set Operators"

Section 2.7, "Other Operators"

2.1 SQL Operators Overview

An operator manipulates individual data items and returns a result. The data items are called *operands* or *arguments*. Operators are represented by special characters or by keywords. For example, the multiplication operator is represented by an asterisk (*) and the operator that tests for nulls is represented by the keywords `IS NULL`. There are two general classes of operators: unary and binary. Oracle Database Lite SQL also supports set operators.

2.1.1 Unary Operators

A unary operator uses only one operand. A unary operator typically appears with its operand in the following format.

```
operator operand
```

2.1.2 Binary Operators

A binary operator uses two operands. A binary operator appears with its operands in the following format.

```
operand1 operator operand2
```

2.1.3 Set Operators

Set operators combine sets of rows returned by queries, instead of individual data items. All set operators have equal precedence. Oracle Database Lite supports the following set operators.

- `UNION`

- UNION ALL
- INTERSECT
- MINUS

The levels of precedence among the Oracle Database Lite SQL operators from high to Table 2–1. Operators listed on the same line have the same level of precedence.

Table 2–1 Levels of Precedence of the Oracle Database Lite SQL Operators

Precedence Level	SQL Operator
1	Unary + - arithmetic operators, PRIOR operator
2	* / arithmetic operators
3	Binary + - arithmetic operators, || character operators
4	All comparison operators
5	NOT logical operator
6	AND logical operator
7	OR logical operator

2.1.4 Other Operators

Other operators with special formats accept more than two operands. If an operator receives a null operator, the result is always null. The only operator that does not CONCAT.

2.2 Arithmetic Operators

Arithmetic operators manipulate numeric operands. The '-' operator is also used in date arithmetic. Supported arithm Table 2–2.

Table 2–2 Arithmetic Operators

Operator	Description	Example
+ (unary)	Makes operand positive	`SELECT +3 FROM DUAL;`
- (unary)	Negates operand	`SELECT -4 FROM DUAL;`
/	Division (numbers and dates)	`SELECT SAL / 10 FROM EMP;`
*	Multiplication	`SELECT SAL * 5 FROM EMP;`
+	Addition (numbers and dates)	`SELECT SAL + 200 FROM EMP;`
-	Subtraction (numbers and dates)	`SELECT SAL - 100 FROM EMP;`

2.3 Character Operators

Character operators used in expressions to Table 2–3.

Table 2–3 Character Operators

Operator	Description	Example
||	Concatenates character strings	`SELECT 'The Name of the employee is: ' || ENAME FROM EMP;`

2.3.1 Concatenating Character Strings

With Oracle Database Lite, you can concatenate character strings with the following results.

- Concatenating two character strings results in another character string.
- Oracle Database Lite preserves trailing blanks in character strings by concatenation, regardless of the strings' datatypes.
- Oracle Database Lite provides the `CONCAT` character function as an alternative to the vertical bar operator. For example,

```
SELECT CONCAT (CONCAT (ENAME, ' is a '),job) FROM EMP WHERE SAL > 2000;
```

This returns the following output.

```
CONCAT(CONCAT(ENAME
-----------------------
KING       is a PRESIDENT
BLAKE      is a MANAGER
CLARK      is a MANAGER
JONES      is a MANAGER
FORD       is a ANALYST
SCOTT      is a ANALYST

6 rows selected.
```

- Oracle Database Lite treats zero-length character strings as nulls. When you concatenate a zero-length character string with another operand the result is always the other operand. A null value can only result from the concatenation of two null strings.

2.4 Comparison Operators

Comparison operators used in conditions that compare one expression with another Table 2–4. The result of a comparison can be `TRUE`, `FALSE`, or `UNKNOWN`.

Table 2–4 Comparison Operators

Operator	Description	Example
=	Equality test.	`SELECT ENAME "Employee" FROM EMP WHERE SAL = 1500;`
!=, ^=, <>	Inequality test.	`SELECT ENAME FROM EMP WHERE SAL ^= 5000;`
>	Greater than test.	`SELECT ENAME "Employee", JOB "Title" FROM EMP WHERE SAL > 3000;`
<	Less than test.	`SELECT * FROM PRICE WHERE MINPRICE < 30;`

Table 2–4 (Cont.) Comparison Operators

Operator	Description	Example
>=	Greater than or equal to test.	`SELECT * FROM PRICE WHERE MINPRICE >= 20;`
<=	Less than or equal to test.	`SELECT ENAME FROM EMP WHERE SAL <= 1500;`
IN	"Equivalent to any member of" test. Equivalent to "=ANY".	`SELECT * FROM EMP WHERE ENAME IN ('SMITH', 'WARD');`
ANY/ SOME	Compares a value to each value in a list or returned by a query. Must be preceded by =, !=, >, <, <= or >=. Evaluates to FASLE if the query returns no rows.	`SELECT * FROM DEPT WHERE LOC = SOME ('NEW YORK','DALLAS');`
NOT IN	Equivalent to "!=ANY". Evaluates to FALSE if any member of the set is NULL.	`SELECT * FROM DEPT WHERE LOC NOT IN ('NEW YORK', 'DALLAS');`
ALL	Compares a value with every value in a list or returned by a query. Must be preceded by =, !=, >, <, <= or >= . Evaluates to TRUE if the query returns no rows.	`SELECT * FROM emp WHERE sal >= ALL (1400, 3000);`
[NOT] BETWEEN *x* and *y*	[Not] greater than or equal to *x* and less than or equal to *y*.	`SELECT ENAME, JOB FROM EMP WHERE SAL BETWEEN 3000 AND 5000;`
EXISTS	TRUE if a sub-query returns at least one row.	`SELECT * FROM EMP WHERE EXISTS (SELECT ENAME FROM EMP WHERE MGR IS NULL);`
x [NOT] LIKE *y* [ESCAPE *z*]	TRUE if *x* does [not] match the pattern *y*. Within *y*, the character "%" matches any string of zero or more characters except null. The character "_" matches any single character. Any character following ESCAPE is interpreted literally, useful when *y* contains a percent (%) or underscore (_).	`SELECT * FROM EMP WHERE ENAME LIKE '%E%';`
IS [NOT] NULL	Tests for nulls. This is the only operator that should be used to test for nulls.	`SELECT * FROM EMP WHERE COMM IS NOT NULL AND SAL > 1500;`

2.5 Logical Operators

Logical operators which manipulate the Table 2–5.

Table 2–5 Logical Operators

Operator	Description	Example
NOT	Returns TRUE if the following condition is FALSE. Returns FALSE if it is TRUE. If it is UNKNOWN, it remains UNKNOWN.	SELECT * FROM EMP WHERE NOT (job IS NULL) SELECT * FROM EMP WHERE NOT (sal BETWEEN 1000 AND 2000)
AND	Returns TRUE if both component conditions are TRUE. Returns FALSE if either is FALSE; otherwise returns UNKNOWN.	SELECT * FROM EMP WHERE job='CLERK' AND deptno=10
OR	Returns TRUE if either component condition is TRUE. Returns FALSE if both are FALSE. Otherwise, returns UNKNOWN.	SELECT * FROM emp WHERE job='CLERK' OR deptno=10

2.6 Set Operators

Set operators which combine the results of two
Table 2–6.

Table 2–6 Set Operators

Operator	Description	Example
UNION	Returns all distinct rows selected by either query.	SELECT * FROM (SELECT ENAME FROM EMP WHERE JOB = 'CLERK' UNION SELECT ENAME FROM EMP WHERE JOB = 'ANALYST');
UNION ALL	Returns all rows selected by either query, including all duplicates.	SELECT * FROM (SELECT SAL FROM EMP WHERE JOB = 'CLERK' UNION SELECT SAL FROM EMP WHERE JOB = 'ANALYST');
INTERSECTand INTERSECT ALL	Returns all distinct rows selected by both queries.	SELECT * FROM orders_list1 INTERSECT SELECT * FROM orders_list2

Table 2–6 (Cont.) Set Operators

Operator	Description	Example
MINUS	Returns all distinct rows selected by the first query but not the second.	SELECT * FROM (SELECT SAL FROM EMP WHERE JOB = 'PRESIDENT' MINUS SELECT SAL FROM EMP WHERE JOB = 'MANAGER');

Note: : The syntax for INTERSECT ALL is supported, but it returns the same results as INTERSECT.

2.7 Other Operators

Table 2–7.

Table 2–7 Other Operators

Operator	Description	Example
(+)	Indicates that the preceding column is the outer join column in a join.	SELECT ENAME, DNAME FROM EMP, DEPT WHERE DEPT.DEPTNO = EMP.DEPTNO (+);
PRIOR	Evaluates the following expression for the parent row of the current row in a hierarchical, or tree-structured query. In such a query, you must use this operator in the CONNECT BY clause to define the relationship between the parent and child rows.	SELECT EMPNO, ENAME, MGR FROM EMP CONNECT BY PRIOR EMPNO = MGR;

Chapter 3

QUERY MULTIPLE TABLES

3.0 Objectives
3.1 Introduction
3.2 Joins
3.2.1 Equi-Join.
3.2.2 Non-Equi-Join.
3.2.3 Outer Join versus Inner Join
3.2.4 Joining Table to Itself.
3.3 Procedures and Functions
3.4 Creating a Procedure
3.5 Executing a Procedure
3.6 Deleting a Procedure
3.7 Functions
3.7.1 Aggregate Functions
3.7.2 Date & Time Function
3.7.3 Arithmatic Functions
3.7.4 Character Functions
3.7.5 Conversion Functions
3.7.6 Miscelleneous Functions
3.8 Summary
3.9 Check Your Progress - *Answers*
3.10 Questions for Self – Study
3.11 Suggested Readings

3.0 OBJECTIVES

After reading this chapter you will able to

- explain how to Creating procedure
- explain how to Executing procedure
- explain how to Deleting procedure
- describe Function

3.1 INTRODUCTION

Today you will learn about joins. This information will enable you to gather and manipulate data across several tables. By the end of the day, you will understand and be able to do the following :

- Perform an outer join
- Perform a left join
- Perform a right join
- Perform an equi-join
- Perform a non-equi-join
- Join a table to itself.

3.2 JOINS

One of the most powerful features of SQL is its capability to gather and manipulate data from across several tables. Without this feature you would have to store all the data elements necessary for each application in one table. Without common tables you would need to store the same data in several tables. Imagine having to redesign, rebuild, and repopulate your tables and databases every time your user needed a query with a new piece of information. The JOIN statement of SQL enables you to design smaller, more specific tables that are easier to maintain than larger tables.

Multiple Tables in a Single SELECT Statement

Like Dorothy in The Wizard of Oz, you have had the power to join tables since Day 2, "Introduction to the Query : The SELECT Statement," when you learned about SELECT and FROM. Unlike Dorothy, you do not have to click you heels together three times to perform a join. Use the following two tables, named, cleverly enough, TABLE1 and TABLE2.

INPUT :
SELECT *
FROM TABLE1

OUTPUT :

ROW	REMARKS
=======	=======
row 1	Table 1
row 2	Table 1
row 3	Table 1
row 4	Table 1
row 5	Table 1
row 6	Table 1

INPUT :
SELECT *
FROM TABLE2
OUTPUT :

ROW	REMARKS
=========	========
row 1	table 2
row 2	table 2
row 3	table 2
row 4	table 2
row 5	table 2
row 6	table 2

To join these two tables, type this :

INPUT :
SELECT *
FROM TABLE1, TABLE2
OUTPUT :

ROW	REMARKS	ROW	REMARKS
row 1	Table 1	row 1	table 2
row 1	Table 1	row 2	table 2
row 1	Table 1	row 3	table 2
row 1	Table 1	row 4	table 2
row 1	Table 1	row 5	table 2
row 1	Table 1	row 6	table 2
row 2	Table 1	row 1	table 2
row 2	Table 1	row 2	table 2
row 2	Table 1	row 3	table 2
row 2	Table 1	row 4	table 2
row 2	Table 1	row 5	table 2
row 2	Table 1	row 6	table 2
row 3	Table 1	row 1	table 2
row 3	Table 1	row 2	table 2
row 3	Table 1	row 3	table 2
row 3	Table 1	row 4	table 2
row 3	Table 1	row 5	table 2
row 3	Table 1	row 6	table 2
row 4	Table 1	row 1	table 2
row 4	Table 1	row 2	table 2
row 4	Table 1	row 3	table 2
row 4	Table 1	row 4	table 2
row 4	Table 1	row 5	table 2
row 4	Table 1	row 6	table 2
row 5	Table 1	row 1	table 2
row 5	Table 1	row 2	table 2
row 5	Table 1	row 3	table 2
row 5	Table 1	row 4	table 2
row 5	Table 1	row 5	table 2
row 5	Table 1	row 6	table 2
row 6	Table 1	row 1	table 2
row 6	Table 1	row 2	table 2
row 6	Table 1	row 3	table 2
row 6	Table 1	row 4	table 2
row 6	Table 1	row 5	table 2
row 6	Table 1	row 6	table 2

Thirty-six rows! Where did they come from ? And what kind of join is this ?

A close examination of the result of the first join shows that each row from TABLE1 was added to each row from TABLE2. An extract from this join shows what happened :

OUTPUT :

ROW	REMARKS	ROW	REMARKS
row 1	Table 1	row 1	table 2
row 1	Table 1	row 2	table 2
row 1	Table 1	row 3	table 2
row 1	Table 1	row 4	table 2
row 1	Table 1	row 5	table 2
row 1	Table 1	row 6	table 2

Notice how each row in TABLE2 was combined with row 1 in TABLE1. Congratulations! You have performed your first join. But what kind of join? An inner join? an outer join? or what? Well, actually this type of join is called a cross-join. A cross-join is not normally as useful as the other joins covered today, but this join does illustrate the basic combining property of all joins : Joins bring tables together.

Suppose you sold parts to bike shops for a living. When you designed your database, you built one big table with all the pertinent columns. Every time you had a new requirement, you added a new column or started a new table with all the old data plus the new data required to create a specific query. Eventually, your database would collapse from its own weight-not a pretty sight. An alternative design, based on a relational model, would have you put all related data into one table. Here's how your customer table would look :

INPUT :
SELECT *
FROM CUSTOMER
OUTPUT :

NAME	ADDRESS	STATE	ZIP	PHONE	REMARKS
========	=========	======	====	=======	========
TRUE WHEEL	55O HUSKER	NE	58702	555-4545	NONE
BIKE SPEC	CPT SHRIVE	LA	45678	555-1234	NONE
LE SHOPPE	HOMETOWN	KS	54678	555-1278	NONE
AAA BIKE	10 OLDTOWN	NE	56784	555-3421	JOHN-MGR
JACKS BIKE	24 EGLIN	FL	34567	555-2314	NONE

Finding the Correct Column

When you joined TABLE1 and TABLE2, you used SELECT *, which returned all the columns in both tables. In joining ORDERS to PART, the SELECT statement is a bit more complicated :

SELECT O.ORDEREDON, O.NAME, O.PARTNUM,
P.PARTNUM, P.DESCRIPTION

SQL is smart enough to know that ORDEREDON and NAME exist only in ORDERS and that DESCRIPTION exists only in PART, but what about PARTNUM, which exists in both? If you have a column that has the same name in two tables, you must use an alias in your SELECT clause to specify which column you want to display. A common technique is to assign a single character to each table, as you did in the FROM clause :

FROM ORDERS O, PART P

You use that character with each column name, as you did in the preceding SELECT clause. The SELECT clause could also be written like this :

SELECT ORDEREDON, NAME, O.PARTNUM, P.PARTNUM, DESCRIPTION

But remember, someday you might have to come back and maintain this query. It does not hurt to make it more readable. Now back to the missing statement.

3.2.1 Equi-Joins

An extract from the PART/ORDERS join provides a clue as to what is missing :

30-JUN-1996 TRUE WHEEL 42 54 PEDALS
30-JUN-1996 BIKE SPEC 54 54 PEDALS
30-MAY-1996 BIKE SPEC 10 54 PEDALS

Notice the PARTNUM fields that are common to both tables. What if you wrote the following ?

INPUT :
SELECT O.ORDEREDON, O.NAME, O.PARTNUM,
P.PARTNUM, P.DESCRIPTION

FROM ORDERS O, PART P
WHERE O.PARTNUM = P.PARTNUM
OUTPUT :

ORDEREDON	NAME	PARTNUM	PARTNUM	DESCRIPTION
1-JUN-1996	AAA BIKE	10	10	TANDEM
30-MAY-1996	BIKE SPEC	10	10	TANDEM
2-SEP-1996	TRUE WHEEL	10	10	TANDEM
1-JUN-1996	LE SHOPPE	10	10	TANDEM
30-MAY-1996	BIKE SPEC	23	23	MOUNTAIN BIKE
15-MAY-1996	TRUE WHEEL	23	23	MOUNTAIN BIKE
30-JUN-1996	TRUE WHEEL	42	42	SEATS
1-JUL-1996	AAA BIKE	46	46	TIRES
30-JUN-1996	BIKE SPEC	54	54	PEDALS
1-JUL-1996	AAA BIKE	76	76	ROAD BIKE
17-JAN-1996	BIKE SPEC	76	76	ROAD BIKE
19-MAY-1996	TRUE WHEEL	76	76	ROAD BIKE
11-JUL-1996	JACKS BIKE	76	76	ROAD BIKE
17-JAN-1996	LE SHOPPE	76	76	ROADBIKE

Using the column PARTNUM that exists in both of the preceding tables, you have just combined the information you had stored in the ORDERS table with information from the PART table to show a description of the parts the bike shops have ordered from you. The join that was used is called an equi-join because the goal is to match the values of a column in one table to the corresponding values in the second table.

You can further qualify this query by adding more conditions in the WHERE clause. For example:

INPUT/OUTPUT :
SELECT O.ORDEREDON, O.NAME, O.PARTNUM,
P.PARTNUM, P.DESCRIPTION
FROM ORDERS O, PART P
WHERE O.PARTNUM = P.PARTNUM
AND O.PARTNUM = 76

ORDEREDON	NAME	PARTNUM	PARTNUM	DESCRIPTION
1-JUL-1996	AAA BIKE	76	76	ROAD BIKE
17-JAN-1996	BIKE SPEC	76	76	ROAD BIKE
19-MAY-1996	RUE WHEEL	76	76	ROAD BIKE
11-JUL-1996	JACKS BIKE	76	76	ROAD BIKE
17-JAN-1996	LE SHOPPE	76	76	ROAD BIKE

The number 76 is not very descriptive, and you would not want your sales people to have to memorize a part number. (We have had the misfortune to see many data information systems in the field that require the end user to know some obscure code for something that had a perfectly good name. Please don't write one of those!) Here's another way to write the query :

INPUT/OUTPUT :
SELECT O.ORDEREDON, O.NAME, O.PARTNUM,
P.PARTNUM, P.DESCRIPTION
FROM ORDERS O, PART P

```
WHERE O.PARTNUM = P.PARTNUM
AND P.DESCRIPTION = 'ROAD BIKE'
```

ORDEREDON	NAME	PARTNUM	PARTNUM	DESCRIPTION
1-JUL-1996	AAA BIKE	76	76	ROAD BIKE
17-JAN-1996	BIKE SPEC	76	76	ROAD BIKE
19-MAY-1996	TRUE WHEEL	76	76	ROAD BIKE
11-JUL-1996	JACKS BIKE	76	76	ROAD BIKE
17-JAN-1996	LE SHOPPE	76	76	ROAD BIKE

Along the same line, take a look at two more tables to see how they can be joined. In this example the employee_id column should obviously be unique. You could have employees with the same name, they could work in the same department, and earn the same salary. However, each employee would have his or her own employee_id. To join these two tables, you would use the employee_id column.

EMPLOYEE_TABLE	EMPLOYEE_PAY_TABLE
employee_id	employee_id
last_name	salary
first_name	department
middle_name	supervisor
	marital_status

INPUT :

```
SELECT   E.EMPLOYEE_ID, E.LAST_NAME, EP.SALARY
FROM  EMPLOYEE_TBL E,
         EMPLOYEE_PAY_TBL EP
WHERE   E.EMPLOYEE_ID = EP.EMPLOYEE_ID
 AND      E.LAST_NAME = 'SMITH';
```

OUTPUT :

E.EMPLOYEE_ID	E.LAST_NAME	EP.SALARY
13245	SMITH	35000.00

Back to the original tables. Now you are ready to use all this information about joins to do something really useful: finding out how much money you have made from selling road bikes :

INPUT/OUTPUT :

```
SELECT SUM(O.QUANTITY * P.PRICE) TOTAL
FROM ORDERS O, PART P
WHERE O.PARTNUM = P.PARTNUM
AND P.DESCRIPTION = 'ROAD BIKE'
```

TOTAL
19610.00

With this setup, the sales people can keep the ORDERS table updated, the production department can keep the PART table current, and you can find your bottom line without redesigning your database.

Can you join more than one table? For example, to generate information to send out an invoice, you could type this statement:

INPUT/OUTPUT :
SELECT C.NAME, C.ADDRESS, (O.QUANTITY * P.PRICE) TOTAL
FROM ORDER O, PART P, CUSTOMER C
WHERE O.PARTNUM = P.PARTNUM
AND O.NAME = C.NAME

NAME	ADDRESS	TOTAL
TRUE WHEEL	55O HUSKER	1200.00
BIKE SPEC	CPT SHRIVE	2400.00
LE SHOPPE	HOMETOWN	3600.00
AAA BIKE	10 OLDTOWN	1200.00
TRUE WHEEL	55O HUSKER	2102.70
BIKE SPEC	CPT SHRIVE	2803.60
TRUE WHEEL	55O HUSKER	196.00
AAA BIKE	10 OLDTOWN	213.50
BIKE SPEC	CPT SHRIVE	542.50
TRUE WHEEL	55O HUSKER	1590.00
BIKE SPEC	CPT SHRIVE	5830.00
JACKS BIKE	24 EGLIN	7420.00
LE SHOPPE	HOMETOWN	2650.00
AAA BIKE	10 OLDTOWN	2120.00

You could make the output more readable by writing the statement like this :

INPUT/OUTPUT :
SELECT C.NAME, C.ADDRESS,
O.QUANTITY * P.PRICE TOTAL
FROM ORDERS O, PART P, CUSTOMER C
WHERE O.PARTNUM = P.PARTNUM
AND O.NAME = C.NAME
ORDER BY C.NAME

NAME	ADDRESS	TOTAL
AAA BIKE	10 OLDTOWN	213.50
AAA BIKE	10 OLDTOWN	2120.00
AAA BIKE	10 OLDTOWN	1200.00
BIKE SPEC	CPT SHRIVE	542.50
BIKE SPEC	CPT SHRIVE	2803.60
BIKE SPEC	CPT SHRIVE	5830.00
BIKE SPEC	CPT SHRIVE	2400.00
JACKS BIKE	24 EGLIN	7420.00
LE SHOPPE	HOMETOWN	2650.00
LE SHOPPE	HOMETOWN	3600.00
TRUE WHEEL	55O HUSKER	196.00
TRUE WHEEL	55O HUSKER	2102.70
TRUE WHEEL	55O HUSKER	1590.00
TRUE WHEEL	55O HUSKER	1200.00

You can make the previous query more specific, thus more useful, by adding the DESCRIPTION column as in the following example :

INPUT/OUTPUT :

```
SELECT C.NAME, C.ADDRESS,
O.QUANTITY * P.PRICE TOTAL,
P.DESCRIPTION
FROM ORDERS O, PART P, CUSTOMER C
WHERE O.PARTNUM = P.PARTNUM
AND O.NAME = C.NAME
ORDER BY C.NAME
```

NAME	ADDRESS	TOTAL	DESCRIPTION
==========	==========	=========	==============
AAA BIKE	10 OLDTOWN	213.50	TIRES
AAA BIKE	10 OLDTOWN	2120.00	ROAD BIKE
AAA BIKE	10 OLDTOWN	1200.00	TANDEM
BIKE SPEC	CPT SHRIVE	542.50	PEDALS
BIKE SPEC	CPT SHRIVE	2803.60	MOUNTAIN BIKE
BIKE SPEC	CPT SHRIVE	5830.00	ROAD BIKE
BIKE SPEC	CPT SHRIVE	2400.00	TANDEM
JACKS BIKE	24 EGLIN	7420.00	ROAD BIKE
LE SHOPPE	HOMETOWN	2650.00	ROAD BIKE
LE SHOPPE	HOMETOWN	3600.00	TANDEM
TRUE WHEEL	55O HUSKER	196.00	SEATS
TRUE WHEEL	55O HUSKER	2102.70	MOUNTAIN BIKE
TRUE WHEEL	55O HUSKER	1590.00	ROAD BIKE
TRUE WHEEL	55O HUSKER	1200.00	TANDEM

This information is a result of joining three tables. You can now use this information to create an invoice.

3.2.2 Non-Equi-Joins

Because SQL supports an equi-join, you might assume that SQL also has a non-equi-join. You would be right! Whereas the equi-join uses an = sign in the WHERE statement, the non-equi-join uses everything but an = sign. For example :

INPUT :

```
SELECT O.NAME, O.PARTNUM, P.PARTNUM,
O.QUANTITY * P.PRICE TOTAL
FROM ORDERS O, PART P
WHERE O.PARTNUM > P.PARTNUM
```

OUTPUT :

You can make the previous query more specific, thus more useful, by adding the DESCRIPTION column as in the following example :

INPUT/OUTPUT :

```
SELECT C.NAME, C.ADDRESS,
O.QUANTITY * P.PRICE TOTAL,
P.DESCRIPTION
FROM ORDERS O, PART P, CUSTOMER C
WHERE O.PARTNUM = P.PARTNUM
AND O.NAME = C.NAME
ORDER BY C.NAME
```

NAME	ADDRESS	TOTAL	DESCRIPTION
==========	==========	=========	==============
AAA BIKE	10 OLDTOWN	213.50	TIRES
AAA BIKE	10 OLDTOWN	2120.00	ROAD BIKE
AAA BIKE	10 OLDTOWN	1200.00	TANDEM
BIKE SPEC	CPT SHRIVE	542.50	PEDALS
BIKE SPEC	CPT SHRIVE	2803.60	MOUNTAIN BIKE
BIKE SPEC	CPT SHRIVE	5830.00	ROAD BIKE
BIKE SPEC	CPT SHRIVE	2400.00	TANDEM
JACKS BIKE	24 EGLIN	7420.00	ROAD BIKE
LE SHOPPE	HOMETOWN	2650.00	ROAD BIKE
LE SHOPPE	HOMETOWN	3600.00	TANDEM
TRUE WHEEL	55O HUSKER	196.00	SEATS
TRUE WHEEL	55O HUSKER	2102.70	MOUNTAIN BIKE
TRUE WHEEL	55O HUSKER	1590.00	ROAD BIKE
TRUE WHEEL	55O HUSKER	1200.00	TANDEM

This information is a result of joining three tables. You can now use this information to create an invoice.

3.2.2 Non-Equi-Joins

Because SQL supports an equi-join, you might assume that SQL also has a non-equi-join. You would be right! Whereas the equi-join uses an = sign in the WHERE statement, the non-equi-join uses everything but an = sign. For example :

INPUT :

```
SELECT O.NAME, O.PARTNUM, P.PARTNUM,
O.QUANTITY * P.PRICE TOTAL
FROM ORDERS O, PART P
WHERE O.PARTNUM > P.PARTNUM
```

OUTPUT :

NAME	PARTNUM	PARTNUM	TOTAL
TRUE WHEEL	76	54	162.75
BIKE SPEC	76	54	596.75
LE SHOPPE	76	54	271.25
AAA BIKE	76	54	217.00
JACKS BIKE	76	54	759.50
TRUE WHEEL	76	42	73.50
BIKE SPEC	54	42	245.00
BIKE SPEC	76	42	269.50
LE SHOPPE	76	42	122.50
AAA BIKE	76	42	98.00
AAA BIKE	46	42	343.00
JACKS BIKE	76	42	343.00
TRUE WHEEL	76	46	45.75
BIKE SPEC	54	46	152.50
BIKE SPEC	76	46	167.75
LE SHOPPE	76	46	76.25
AAA BIKE	76	46	61.00
JACKS BIKE	76	46	213.50
TRUE WHEEL	76	23	1051.35
TRUE WHEEL	42	23	2803.60

...

This listing goes on to describe all the rows in the join WHERE O.PARTNUM > P.PARTNUM. In the context of your bicycle shop, this information does not have much meaning, and in the real world the equi-join is far more common than the non-equi-join. However, you may encounter an application in which a non-equi-join produces the perfect result.

3.2.3 Outer Joins versus Inner Joins

Just as the non-equi-join balances the equi-join, an outer join complements the inner join. An inner join is where the rows of the tables are combined with each other, producing a number of new rows equal to the product of the number of rows in each table. Also, the inner join uses these rows to determine the result of the WHERE clause. An outer join groups the two tables in a slightly different way. Using the PART and ORDERS tables from the previous examples, perform the following inner join:

INPUT :

```
SELECT P.PARTNUM, P.DESCRIPTION, P.PRICE,
O.NAME, O.PARTNUM
FROM PART P
JOIN ORDERS O ON ORDERS.PARTNUM = 54
```

OUTPUT :

PARTNUM	DESCRIPTION	PRICE	NAME	PARTNUM
54	PEDALS	54.25	BIKE SPEC	54
42	SEATS	24.50	BIKE SPEC	54
46	TIRES	15.25	BIKE SPEC	54
23	MOUNTAIN BIKE	350.45	BIKE SPEC	54
76	ROAD BIKE	530.00	BIKE SPEC	54
10	TANDEM	1200.00	BIKE SPEC	54

The result is that all the rows in PART are spliced on to specific rows in ORDERS where the column PARTNUM is 54. Here's a RIGHT OUTER JOIN statement :

INPUT/OUTPUT :

```
SELECT P.PARTNUM, P.DESCRIPTION, P.PRICE,
```

```
O.NAME, O.PARTNUM
FROM PART P
RIGHT OUTER JOIN ORDERS O ON ORDERS.PARTNUM = 54
```

PARTNUM	DESCRIPTION	PRICE	NAME	PARTNUM
<null>	<null>	<null>	TRUE WHEEL	23
<null>	<null>	<null>	TRUE WHEEL	76
<null>	<null>	<null>	TRUE WHEEL	10
<null>	<null>	<null>	TRUE WHEEL	42
54	PEDALS	54.25	BIKE SPEC	54
42	SEATS	24.50	BIKE SPEC	54
46	TIRES	15.25	BIKE SPEC	54
23	MOUNTAIN BIKE	350.45	BIKE SPEC	54
76	ROAD BIKE	530.00	BIKE SPEC	54
10	TANDEM	1200.00	BIKE SPEC	54
<null>	<null>	<null>	BIKE SPEC	10
<null>	<null>	<null>	BIKE SPEC	23
<null>	<null>	<null>	BIKE SPEC	76
<null>	<null>	<null>	LESHOPPE	76
<null>	<null>	<null>	LE SHOPPE	10
<null>	<null>	<null>	AAA BIKE	10
<null>	<null>	<null>	AAA BIKE	76
<null>	<null>	<null>	AAA BIKE	46
<null>	<null>	<null>	JACKS BIKE	76

This type of query is new. First you specified a RIGHT OUTER JOIN, which caused SQL to return a full set of the right table, ORDERS, and to place nulls in the fields where ORDERS.PARTNUM <> 54. Following is a LEFT OUTER JOIN statement :

```
INPUT/OUTPUT :
SELECT P.PARTNUM, P.DESCRIPTION,P.PRICE,
O.NAME, O.PARTNUM
FROM PART P
LEFT OUTER JOIN ORDERS O ON ORDERS.PARTNUM = 54
```

PARTNUM	DESCRIPTION	PRICE	NAME	PARTNUM
54	PEDALS	54.25	BIKE SPEC	54
42	SEATS	24.50	BIKE SPEC	54
46	TIRES	15.25	BIKE SPEC	54
23	MOUNTAIN BIKE	350.45	BIKE SPEC	54
76	ROAD BIKE	530.00	BIKE SPEC	54
10	TANDEM	1200.00	BIKE SPEC	54

You get the same six rows as the INNER JOIN. Because you specified LEFT (the LEFT table), PART determined the number of rows you would return. Because PART is smaller than ORDERS, SQL saw no need to pad those other fields with blanks.

Don't worry too much about inner and outer joins. Most SQL products determine the optimum JOIN for your query. In fact, if you are placing your query into a stored procedure (or using it inside a program (both stored procedures and Embedded SQL covered on Day 13, "Advanced SQL Topics"), you should not specify a join type even if your SQL implementation provides the proper syntax. If you do specify a join type, the optimizer chooses your way instead of the optimum way.

Some implementations of SQL use the + sign instead of an OUTER JOIN statement. The + simply means "Show me everything even if something is missing". Here's the syntax :

SYNTAX :

```
SQL> select e.name, e.employee_id, ep.salary,
          ep.marital_status
    from e,ployee_tbl e,
        employee_pay_tbl ep
    where e.employee_id = ep.employee_id(+)
     and e.name like '%MITH';
```

This statement is joining the two tables. The + sign on the ep.employee_id column will return all rows even if they are empty.

3.2.4 Joining a Table to Itself

The syntax of this operation is similar to joining two tables. For example, to join table TABLE1 to itself, type this :

INPUT :

```
SELECT *
FROM TABLE1, TABLE1
```

OUTPUT :

ROW	REMARKS	ROW	REMARKS
row 1	Table 1	row 1	Table 1
row 1	Table 1	row 2	Table 1
row 1	Table 1	row 3	Table 1
row 1	Table 1	row 4	Table 1
row 1	Table 1	row 5	Table 1
row 1	Table 1	row 6	Table 1
row 2	Table 1	row 1	Table 1
row 2	Table 1	row 2	Table 1
row 2	Table 1	row 3	Table 1
row 2	Table 1	row 4	Table 1
row 2	Table 1	row 5	Table 1
row 2	Table 1	row 6	Table 1
row 3	Table 1	row 1	Table 1
row 3	Table 1	row 2	Table 1
row 3	Table 1	row 3	Table 1
row 3	Table 1	row 4	Table 1
row 3	Table 1	row 5	Table 1
row 3	Table 1	row 6	Table 1
row 4	Table 1	row 1	Table 1
row 4	Table 1	row 2	Table 1

...

In its complete form, this join produces the same number of combinations as joining two 6-row tables. This type of join could be useful to check the internal consistency of data. What would happen if someone fell asleep in the production department and entered a new part with a PARTNUM that already existed? That would be bad news for everybody: Invoices would be wrong; your application would probably blow up; and in general you would be in for a very bad time. And the cause of all your problems would be the duplicate PARTNUM in the table on the next page :

INPUT/OUTPUT :

SELECT * FROM PART

PARTNUM	DESCRIPTION	PRICE
54	PEDALS	54.25
42	SEATS	24.50
46	TIRES	15.25
23	MOUNTAIN BIKE	350.45
76	ROAD BIKE	530.00
10	TANDEM	1200.00
76	CLIPPLESS SHOE	65.00

<-NOTE SAME #

You saved your company from this bad situation by checking PART before anyone used it :

INPUT/OUTPUT :

```
SELECT F.PARTNUM, F.DESCRIPTION,
S.PARTNUM,S.DESCRIPTION
FROM PART F, PART S
WHERE F.PARTNUM = S.PARTNUM
AND F.DESCRIPTION <> S.DESCRIPTION
```

PARTNUM	DESCRIPTION	PARTNUM	DESCRIPTION
76	ROAD BIKE	76	CLIPPLESS SHOE
76	CLIPPLESS SHOE	76	ROAD BIKE

3.1,3.2 Check Your Progress

Fill in the blanks

1) ..... can be used for joining of two tables.
2) A..... must return a value.
3) A Procedure have.....

3.3 PROCEDURE AND FUNCTIONS

Procedures are simply a named PL/SQL block, that executes certain task. A procedure is completely portable among platforms in which Oracle is executed.

A function is similar to a procedure. The main difference between the function and procedure is that a function returns a value where procedure does not.

3.3.1 Advantages of using Procedures and Functions

1. **Improved performance :**
   - A block is placed on the database it is parsed at the time it is stored. When it is subsequently executed Oracle already has the block compiled and it is therefore much faster.
   - Reduce the number of calls to the database and decrease network traffic by bundling commands.
2. **Improved maintenance :**
   - Modify routines online without interfering with other users.
   - Modify one routine to affect multiple applications.
   - Modify one routine to eliminate duplicate testing.

Name of Course	**Bachelor of Computer Application (BCA)**
Semester	**IV**
Name of Subject	**Computer Graphics**
Subject Code	**Elective -BCA-404 B**
Marks	**75 Marks**
Lectures	**50 Lectures**

Computer Graphics Objectives:

- To understand the basics of computer graphics, different display devices and applications of computer graphics.
- To learn about algorithmic development of graphics primitives like: point, line, polygon etc.
- To impart knowledge of 2D transformations on graphics objects.
- To familiarize with 2D Viewing and different clipping methods.
- To understand rules for graphics software design.

Computer Graphics Outcomes:

- Knowledge of working of display systems.
- Skill to execute various Scan Conversion algorithms in laboratory so as to draw Graphics primitives.
- Familiarization with 2D graphics.
- Skill to execute various 2D transformations on graphics.
- Use of various graphics packages/functions on graphic.

UNIT - I

1. Introduction to Computer Graphics **14 Lectures**

1.1 Introduction-Definition

1.2 Application areas of Computer Graphics.

1.3 Advantages of computer graphics.

1.4 Graphical user interface.

1.5 Random scan displays, Raster scan displays

1.6 Display devices: Cathode Ray Tubes, Color CRT monitors, Direct View Storage Tube.

1.7 Plotter.

1.8 Joystick, light pen.

UNIT - II

2. Raster Scan Graphics and Transformation: 13 Lectures

2.1 Introduction

2.2 Line, Line Segment

2.3 Line drawing algorithms –

a. Digital Differential Algorithm,

b. Bresenham's line algorithm

2.4 Two dimensional transformation

2.5 Matrix representation.

2.6 Translation.

2.7 Rotation.

2.8 Scaling.

2.9 Reflection

2.10 Shear

UNIT - III

3. Segmented Display Files and Clipping and Windowing: 13 Lectures

3.1 Segment table.

3.2 Functions for segmenting display file

3.3 Posting & unposting segments

3.4 Segment naming scheme

3.5 Default error conditions

3.6 Appending to segments

3.7 Viewing transformation

3.8 2-D clipping

3.9 Simple visibility algorithm

3.10 End point codes

3.11 Midpoint subdivision algorithm

3.12 Polygon clipping algorithm (Sutherland-Hodgman algorithm)

UNIT - IV

4. Geometric Models and Simple Graphics Package: 10 Lectures

4.1 Simple modeling example

4.2 Geometric modeling

4.3 Symbols & instances

4.4 Implementation of Instance transformation

4.5 Ground rules for graphics s/w design

4.6 Function domains

4.7 Graphics primitives

4.8 Windowing function

4.9 Example-a graph plotting program

4.10 Implementation of the functions

References:

1. Principle of Interactive Computer Graphics -Willam Newman&Robert Sproull (TMH)
2. Procedural Elements for Computer Graphics -David F. Rogers(TMH)
3. Computer graphics -A.P.Gogse



This document was created with the Win2PDF “Print to PDF” printer available at

https://www.win2pdf.com

This version of Win2PDF 10 is for evaluation and non-commercial use only.
Visit https://www.win2pdf.com/trial/ for a 30 day trial license.

This page will not be added after purchasing Win2PDF.

https://www.win2pdf.com/purchase/

Introduction to Computer Graphics

The computer is an information processing machine. It is a tool for storing, manipulating and correlating data. There are many ways to communicate the processed information to the user. The computer graphics is one of the most effective and commonly used way to communicate the processed information to the user. It displays the information in the form of graphics objects such as pictures, charts, graphs and diagrams instead of simple text.

Graphics is defined as any sketch or a drawing or a special network that pictorially represents some meaningful information. It is difficult to display an image of any size on the computer screen. This method is simplified by using **Computer graphics**. Computer Graphics is used where a set of image needs to be manipulated or the creation of the image in the form of pixels and is drawn on the computer. Graphics on the computer are produced by using various algorithms and techniques.

In computer graphics pictures or graphics objects are presented as a collection of discrete picture elements called **pixels**. The **pixel** is the smallest addressable screen element. It is the smallest piece of the display screen which we can control. The control is achieved by setting the intensity and Color of the pixel which compose the screen.

Computer Graphics involves technology to access. The Process transforms and presents information in a visual form. The role of computer graphics insensible. In today life, computer graphics has now become a common element in user interfaces, T.V. commercial motion pictures.

Computer Graphics is the creation of pictures with the help of a computer. The end product of the computer graphics is a picture it may be a business graph, drawing, and engineering.

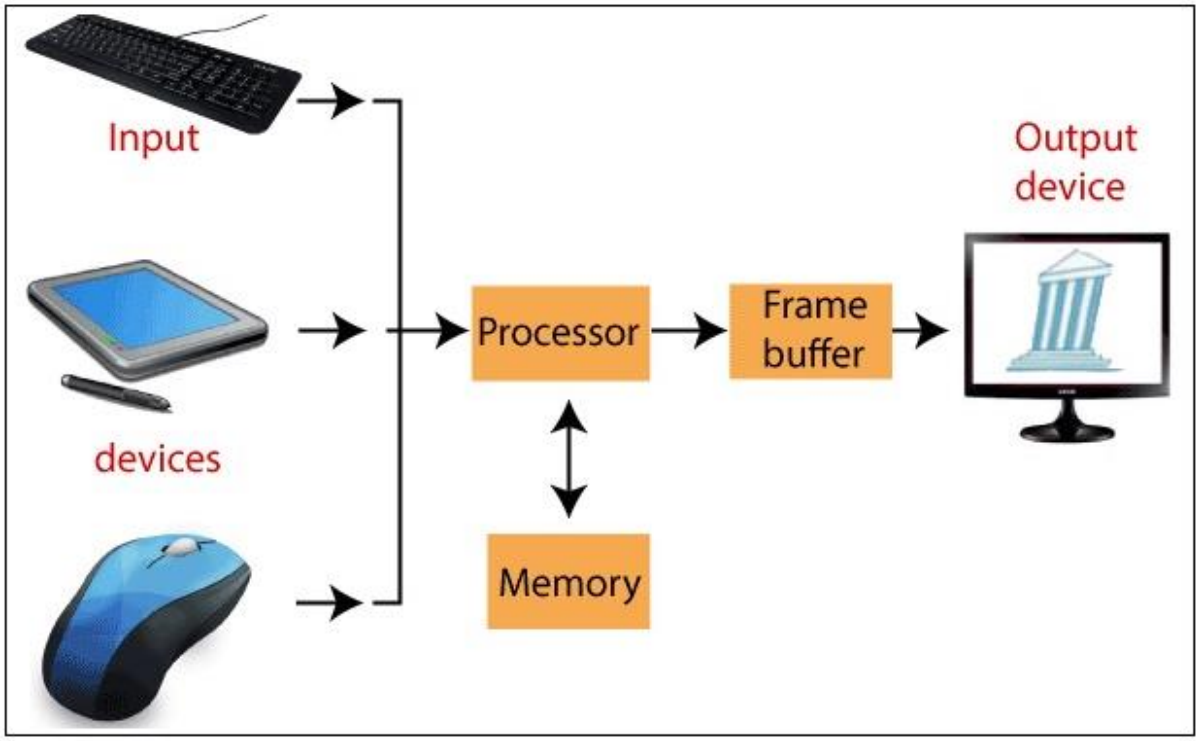
Computer Graphics can be used in UI design, rendering, geometric object, animation and many more.

Computer Graphics refers to several things:

- The manipulation and the representation of the image or the data in a graphical manner.
- Various technologies required for the creation and manipulation.
- Digital synthesis and its manipulation.

Definition of Computer Graphics:

- It is the use of computers to create and manipulate pictures on a display device. It comprises of software techniques to create, store, modify, represents pictures.
- **Computer graphics is an art of drawing pictures on computer screens with the help of programming. It involves computations, creation, and manipulation of data.** In other words, we can say that computer graphics is a rendering tool for the generation and manipulation of images.
- **"Computer graphics is the use of a computer to define, store, manipulate, interrogate, and represent the pictorial output."** An image in computer graphics is made up of a number of pixels.



Types of Computer Graphics:

There are two kinds of computer graphics are–

- Interactive Computer Graphics
- Non-Interactive Computer Graphics

Interactive Computer Graphics

In interactive computer graphics, **users have some controls over the image**, i.e., the user can make any changes to the image produced.

Interactive Computer Graphics involves computer-user two-way communication.

For Example:

- Ping-Pong game.
- Drawing on touch screens.
- Display weather forecast or other moving charts/graphs on the screen.
- Animating pictures or graphics in movies.
- Graphics animation in video games.

Advantages

1. Superior Quality.
2. More accurate outcomes or products.
3. Increased Productivity.
4. Lower cost of development.
5. Increases the ability to understand information and interpret patterns significantly.

Non- Interactive Computer Graphics

Non-interactive computer graphics are also known as passive computer graphics. **It is a type of computer graphics in which the user has no control over the image.** The photo is completely controlled by the instructions of the program, not by the user.

For Example:

- Screen savers.
- Map representation of the data.
- Graphic elements are used in the text, document, and PDF presentation.
- Static images are used in mobile applications and websites.
- Business graphics are used as brochures, business cards, menu of the hotel.

Representation of graphics

We can represent the graphics by following two ways:

1. Raster (Bitmap) Graphics
2. Vector Graphics

Raster Graphics: In raster graphics, the image is presented as a rectangular grid of colored squares. **Raster images are also called bitmap images. Bitmap images are stored as the collection of small individual dots called pixels.**

Bitmap images require high resolution and anti-aliasing for a smooth appearance.

For example– Paint, Photoshop, etc.

Vector Graphics: In vector graphics, the image is represented in the form of continuous geometric objects: line, curve, etc. Vector images are not based on pixel pattern. They use mathematical formulas to draw line and curves. The lines and curves can be combined to create an image.

For Example– PowerPoint, Corel Draw, etc.

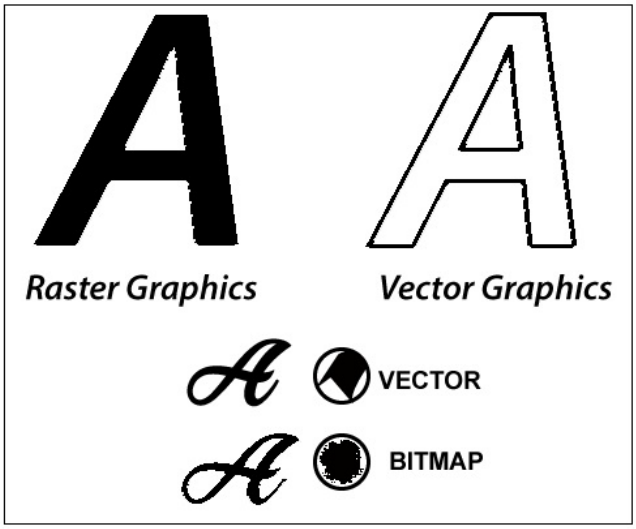


Fig: Raster (Composition of Paths) Vector (Composition of Pixels)

Difference between Raster and Vector Graphics:

Raster Graphics	Vector Graphics
Raster images are the collection of the pixel.	The Vector images are composed of paths.
Scan conversion is required.	Scan Conversion is not required.
Raster Graphics are less costly.	Vector Graphics are more costly compared to raster graphics.
Raster image takes less space to store.	Vector image takes more space.
Raster graphics can draw mathematical curves, polygons, and boundaries.	Vector graphics can only draw continuous and smooth lines.
File Extension: .BMP, .TIF, .JPG etc.	**File Extension:** .SVG, .PDF, .AI etc.

Application areas of computer Graphics

The use of Computer graphics is widely spread. It is used in various areas such as industry, business government organisations education, entertainment and most recently the home. Latest discuss the representative uses of computer graphics in brief.

1. **User Interfaces:**

User-friendliness is one of the main factor underlying the success and popular popularity of any system. It is now well established fact that graphical interfaces provide an attractive and easy interaction between users and computers. The built-in graphics provided with user interfaces use visual control atoms such as buttons menus icons, scroll bar extra which allows user to interact with computer only by mouse click stop typing is necessary only two input text to be stored and manipulated. Computer graphics is used to create search user interfaces.

2. **Computer Art:**

Using computer graphics we can create fine and commercial art which include animation packages, paint packages. These packages provide facilities for designing object shapes and specifying object motion. Cartoon drawing, paintings, logo design can also be done.

3. **Plotting of Graphics and charts:**

In industry business government and educational organisations computer graphics is most commonly used to create 2D and 3D graphics of mathematical physical and economic functions in form of histograms, bars and pie charts. These graphs and charts are very useful for decision making.

4. **Office Automation and desktop Publishing:**

The desktop publishing on personal computers allows the use of graphics for the creation and dissemination of information. Many organisations does the in-house creation and printing of documents poster the desktop publishing allows the user to create documents which contain text, tables, graphs and other forms of drawn or scanned images or pictures. This is one approach towards the office automation.

5. **Computer-aided drafting and Design:**

The Computer-aided drafting uses graphics to design components and system electrical, mechanical, electromechanical and electronic devices such as automobile bodies, structures of buildings, airplane, ships, Very large scale integrated (VLSI) chips, optical systems and computer networks.

6. **Simulation and Animation:**

Use of graphics in simulation makes mathematic models and mechanical systems more realistic and easy to study. The interactive graphics supported by animation software proved their use in production of animated movies and cartoon films.

7. **Process control:**

By the use of computer now it is possible to control various processes in the industry from a remote control room. In such cases, process systems and processing parameters are shown on the computer with graphic symbols and identifications. This makes it easy for operator to monitor and control various processing parameters at a time.

8. **Cartography:**

Computer graphics is also used to represent graphics geographic maps weather maps, oceanographic charts, contour maps, population density maps and so on.

9. **Entertainment:**

Computer graphics finds a major part of its utility in the movie industry and game industry. Used for creating motion pictures, music video, television shows, cartoon animation films. In the game industry where focus and interactivity are the key players, computer graphics helps in providing such features in the efficient way.

10. **Education:**

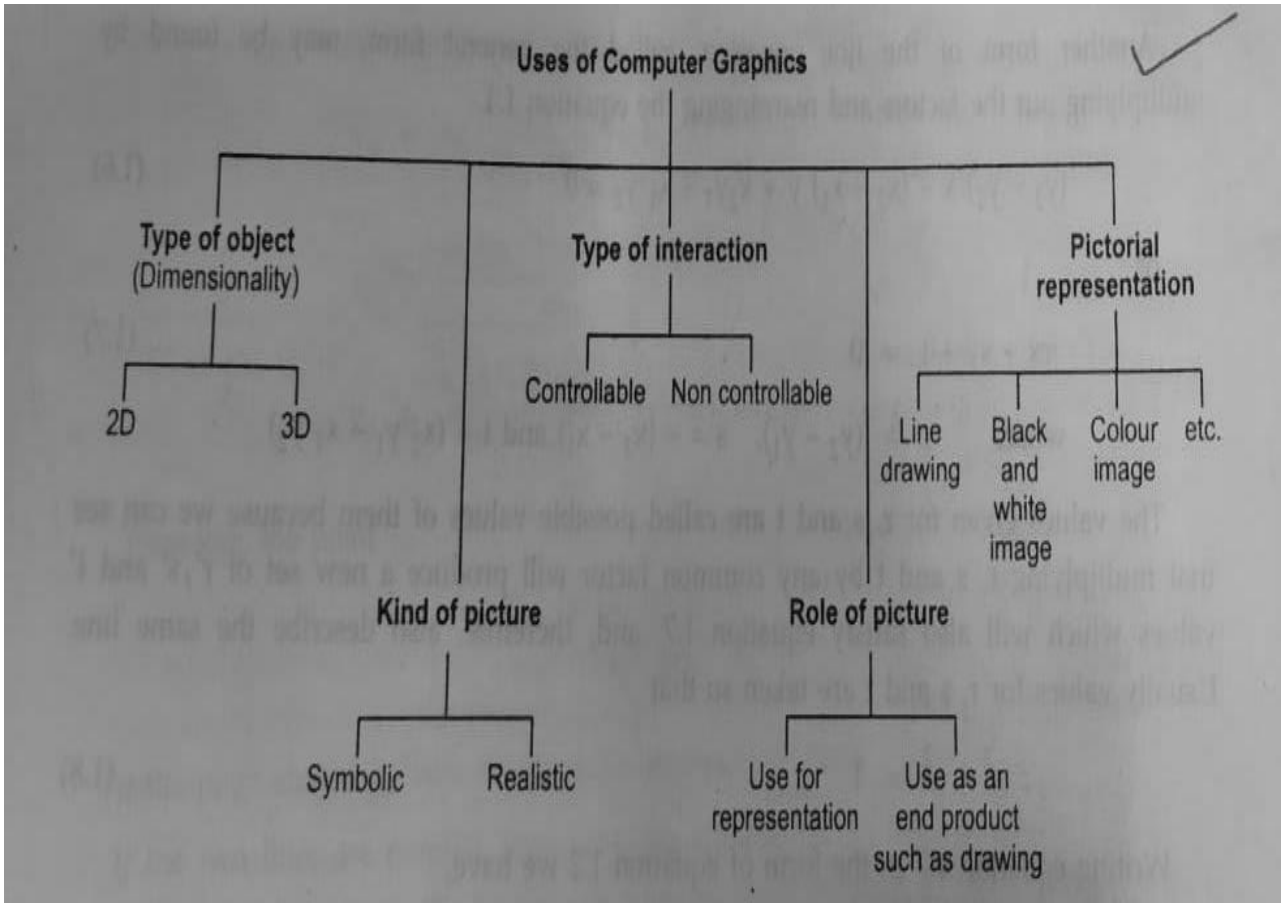
Computer generated models are extremely useful for teaching huge number of concepts and fundamentals in an easy to understand and learn manner. Using computer graphics many educational models can be created through which more interest can be generated among the students regarding the subject.

Advantages of computer Graphics:

1. Today, a high quality graphics display of personal computer provide one of the most natural means of communicating with a computer.
2. It provides tools for producing pictures not only of concrete , real world objects but also of abstract, synthetic objects, such as mathematical surfaces in 4D and of data that have no inherent geometry such as survey results.
3. It has an ability to show moving pictures and thus it is possible to produce animations with computer graphics.
4. It has an ability to show moving pictures and thus it is possible to produce animation with computer graphics.
5. With computer graphics user can also control the animation by adjusting the speed of the portion of the total seen in view, the geometric relationship of the objects in the scene to one another, the amount of detail shown and so on.
6. The computer graphics provides tool called **motion dynamics**. With this tool user can move and tumble objects with respect to a stationary observer or he can make objects stationary and the viewer moving around them. A typical example is walk through made by builder to show flat interior and building surroundings. In many case it is possible also possible to move both objects and viewer.
7. The computer graphics also provides a facility called **update dynamics**. With update dynamics it is possible to change the shape, colour or the other properties of the objects being viewed.
8. With the recent development of **digital signal processing** and **audio synthesis chip** the **interactive graphics can now provide audio feedback along with the graphical feedbacks** to make the simulated environment even more realistic.
9. Computer graphics permits extensive, high bandwidth user computer interaction. It significantly enhances the ability to understand information, to perceive trends and to visualise real or imaginary objects either moving or stationary in a realistic environment. it also makes it possible to get high quality and more resize results and products with a lower analysis and design cost.

Classification of Computer graphics

We have seen various uses of computer graphics. Thesse uses can be classified as shown in following diagram.



Graphical user interface

A GUI (graphical user interface) is a system of interactive visual components for computer software. A GUI displays objects that convey information, and represent actions that can be taken by the user. The objects change Color, size, or visibility when the user interacts with them.

GUI objects include **icons**, **cursors**, and **buttons**. These graphical elements are sometimes enhanced with **sounds**, or **visual effects** like **transparency** and **drop shadows**.

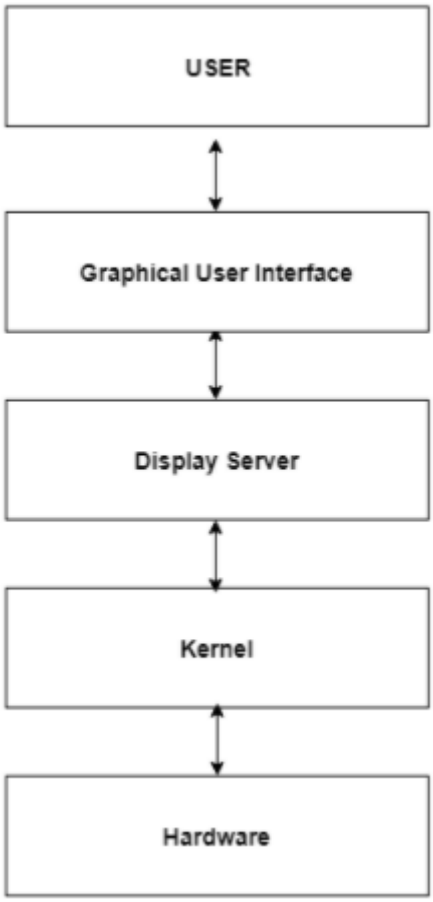
A GUI is considered to be more user-friendly than a text-based command-line interface, such as MS-DOS, or the shell of Unix-like operating systems.

The GUI was first developed at **Xerox PARC** by **Alan Kay**, Douglas Engelbart, and a group of other researchers in 1981. Later, Apple introduced the Lisa computer with a GUI on January 19, 1983.

GUI overview

Below is a picture of the Windows 7 desktop and an example of a GUI.





How does a GUI work?

A GUI uses windows, icons, and menus to carry out commands, such as opening, deleting, and moving files. Although a GUI operating system is primarily navigated using a mouse, a keyboard can also be used via keyboard shortcuts or the arrow keys.

As an example, if you wanted to open a program on a GUI system, you would move the mouse pointer to the program's icon and double-click it.

What are the benefits of GUI?

Unlike a command-line operating system or CUI, like UNIX or MS-DOS, GUI operating systems are easier to learn and use because commands do not need to be memorized. Additionally, users do not need to know any programming languages. Because of their ease of use and more modern appearance, GUI operating systems have come to dominate today's market.

What are examples of a GUI operating system?

- Microsoft Windows
- Apple System 7 and macOS
- Chrome OS
- Linux variants like Ubuntu using a GUI interface.

What are examples of a GUI interface?

- GNOME
- KDE
- Any Microsoft program, including Word, Excel, and Outlook.
- Internet browsers, such as Internet Explorer, Chrome, and Firefox.

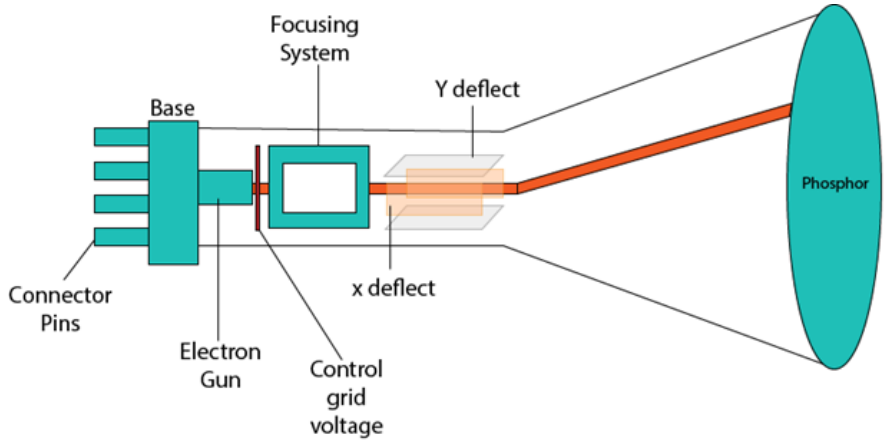
How does the user interact with a GUI?

A pointing device, such as the mouse, is used to interact with nearly all aspects of the GUI. More modern (and mobile) devices also utilize a touch screen. However, as stated in previous sections, it is also possible to navigate a GUI using a keyboard.

Cathode Ray Tube (CRT):

CRT stands for Cathode Ray Tube. CRT is a technology used in traditional computer monitors and televisions. The image on CRT display is created by firing electrons from the back of the tube of phosphorus located towards the front of the screen.

Once the electron heats the phosphorus, they light up, and they are projected on a screen. The color you view on the screen is produced by a blend of red, blue and green light.



Cathode Ray Tube

Components of CRT:

Main Components of CRT are:

Electron Gun: Electron gun consisting of a series of elements, primarily a heating filament (heater) and a cathode. The electron gun creates a source of electrons which are focused into a narrow beam directed at the face of the CRT.

2. **Control Electrode**: It is used to turn the electron beam on and off.

3. **Focusing system**: It is used to create a clear picture by focusing the electrons into a narrow beam.

4. **Deflection Yoke**: It is used to control the direction of the electron beam. It creates an electric or magnetic field which will bend the electron beam as it passes through the area. In a conventional CRT, the yoke is linked to a sweep or scan generator. The deflection yoke which is connected to the sweep generator creates a fluctuating electric or magnetic potential.

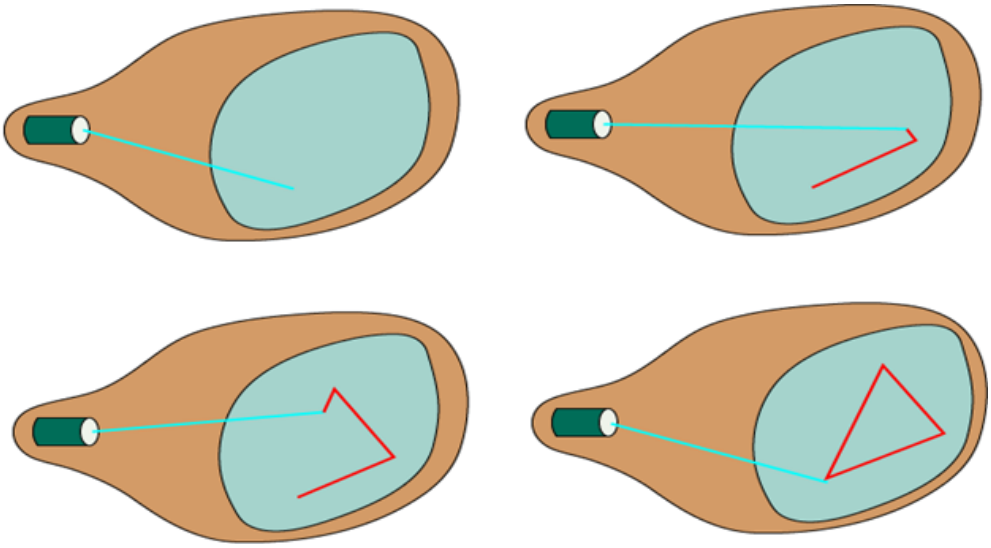
5. **Phosphorus-coated screen**: The inside front surface of every CRT is coated with phosphors. Phosphors glow when a high-energy electron beam hits them. Phosphorescence is

the term used to characterize the light given off by a phosphor after it has been exposed to an electron beam.

Random Scan and Raster Scan Display:

Random Scan Display:

Random Scan System uses an electron beam which operates like a pencil to create a line image on the CRT screen. The picture is constructed out of a sequence of straight-line segments. Each line segment is drawn on the screen by directing the beam to move from one point on the screen to the next, where its x & y coordinates define each point. After drawing the picture. The system cycles back to the first line and design all the lines of the image 30 to 60 time each second. The process is shown in fig:



Random-scan monitors are also known as vector displays or stroke-writing displays or calligraphic displays.

Advantages:

1. A CRT has the electron beam directed only to the parts of the screen where an image is to be drawn.
2. Produce smooth line drawings.
3. High Resolution

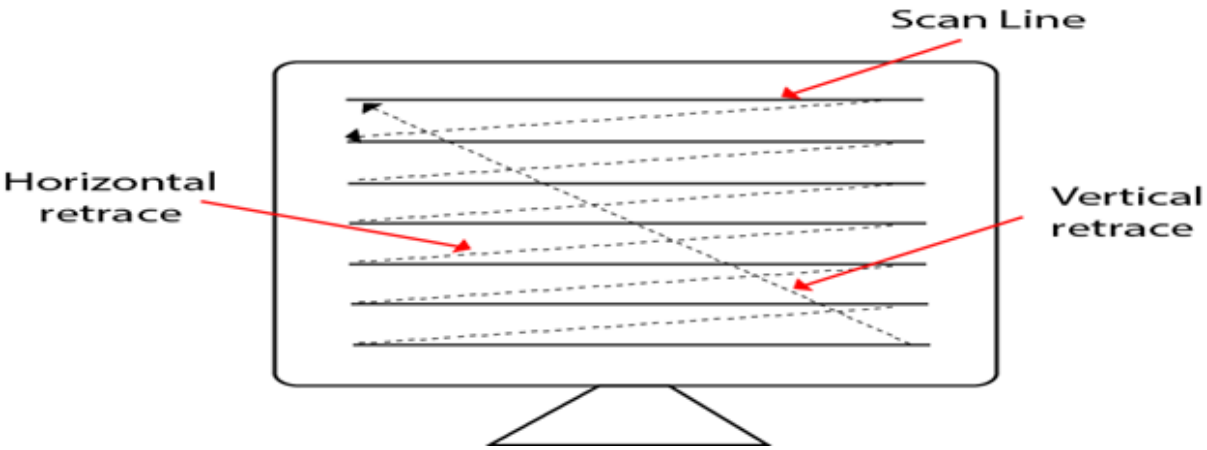
Disadvantages:

1. Random-Scan monitors cannot display realistic shades scenes.

Raster Scan Display:

A Raster Scan Display is based on intensity control of pixels in the form of a rectangular box called Raster on the screen. Information of on and off pixels is stored in refresh buffer or Frame buffer. Televisions in our house are based on Raster Scan Method. The raster scan system can store information of each pixel position, so it is suitable for realistic display of objects. Raster Scan provides a refresh rate of 60 to 80 frames per second.

Frame Buffer is also known as Raster or bit map. In Frame Buffer the positions are called picture elements or pixels. Beam refreshing is of two types. First is horizontal retracing and second is vertical retracing. When the beam starts from the top left corner and reaches the bottom right scale, it will again return to the top left side called at vertical retrace. Then it will again more horizontally from top to bottom call as horizontal retracing shown in fig:



Types of Scanning or travelling of beam in Raster Scan

- **Interlaced Scanning**
- **Non-Interlaced Scanning**

In Interlaced scanning, each horizontal line of the screen is traced from top to bottom. Due to which fading of display of object may occur. This problem can be solved by Non-Interlaced scanning. In this first of all odd numbered lines are traced or visited by an electron beam, then in the next circle, even number of lines are located.

For non-interlaced display refresh rate of 30 frames per second used. But it gives flickers. For interlaced display refresh rate of 60 frames per second is used.

Advantages:

1. Realistic image
2. Million Different colors to be generated
3. Shadow Scenes are possible.

Disadvantages:

1. Low Resolution
2. Expensive

Color CRT Monitors

This was one the earlier CRTs to produce color displays. Coating phosphors of different compounds can produce different colored pictures. But the basic problem of graphics is not to produce a picture of a predetermined color, but to produce color pictures, with the color characteristics chosen at run time.

The basic principle behind colored displays is that combining the 3 basic colors –Red, Blue and Green, can produce every color. By choosing different ratios of these three colors we can produce different colors – millions of them in-fact. We also have basic phosphors, which can produce these basic colors. So, one should have a technology to combine them in different combinations.

There are two popular techniques for producing color displays with a CRT are:

1. Beam Penetration method
2. The Shadow – Mask method.

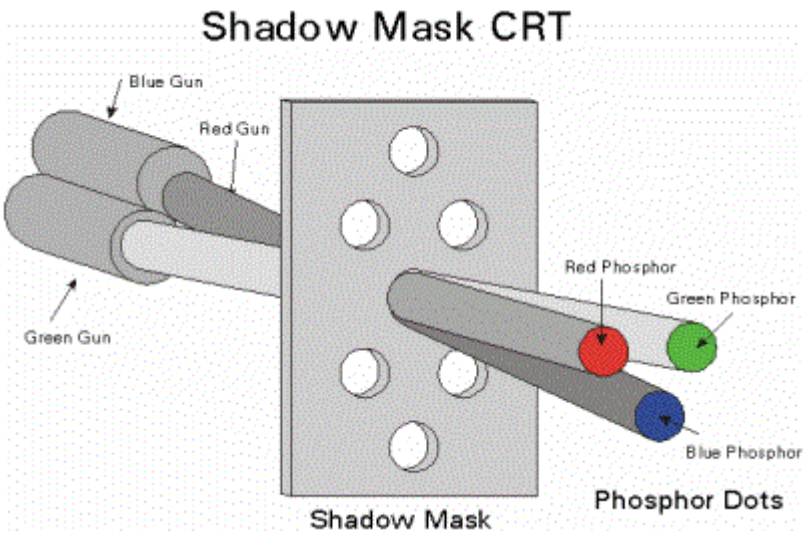
Beam Penetration method

This CRT is similar to the simple CRT, but it makes use of multi coloured phosphorus of number of layers. Each phosphorus layer is responsible for one colour. All other arrangements are similar to simple CRT. It can produce a maximum of 4 to 5 colours

The organization is something like this – The red, green and blue phosphorus are coated in layers – one behind the other. If a low speed beam strikes the CRT, only the red colored phosphorus is activated, a slightly accelerated beam would activate both red and green (because it can penetrate deeper) and a much more activated one would add the blue component also.

But the basic problem is a reliable technology to accelerate the electronic beam to precise levels to get the exact colors – it is easier said than done. However, a limited range of colors can be conveniently produced using the concept.

The Shadow – Mask method



This works, again, on the principle of combining the basic colors – Red, green and Blue – in suitable proportions to get a combination of colors, but it's principle is much more sophisticated and stable.

The shadow mask CRT, instead of using one electron gun, uses 3 different guns placed one by the side of the other to form a triangle or a "Delta" as shown. Each pixel point on the screen is also made up of 3 types of phosphors to produce red, blue and green colors. Just before the phosphor screen is a metal screen, called a "shadow mask".

This plate has holes placed strategically, so that when the beams from the three electron guns are focused on a particular pixel, they get focused on particular color producing pixel only i.e. If for convenience sake we can call the electronic beams as red, blue and green beams (though in practice the colors are produced by the phosphors, and until the beams hit the phosphor dots, they produce no colors), the metal holes focus the red beam onto the red color producing phosphor, blue beam on the blue producing one etc. When focused on to a different pixel, the red beam again focuses on to the red phosphor and so on.

Now, unlike the beam penetration CRTs where the acceleration of the electron beam was being monitored, we now manipulate the intensity of the 3 beams simultaneously. If the red beam is made more intense, we get more of red color in the final combination etc. Since fine-tuning of the beam intensities is comparatively simple, we can get much more combination of colors than the beam penetration case. In fact, one can have a matrix of combinations to produce a wide variety of colors.

The shadow mask CRT, though better than the beam penetration CRT in performance, is not without it's disadvantages. Since three beams are to be focused, the role of the "Shadow mask" becomes critical. If the focusing is not achieved properly, the results tend to be poor. Also, since instead of one pixel point in a monochrome CRT now each pixel is made up of 3 points (for 3 colors), the resolution of the CRT (no. of pixels) for a given screen size reduces.

Another problem is that since the shadow mask blocks a portion of the beams (while focusing them through the holes) their intensities get reduced, thus reducing the overall brightness of the picture. To overcome this effect, the beams will have to be produced at very high intensities to begin with. Also, since the 3 color points, though close to each other, are still not at the same point, the pictures tend to look like 3 colored pictures placed close by, rather than a single picture. Of course, this effect can be reduced by placing the dots as close to one another as possible.

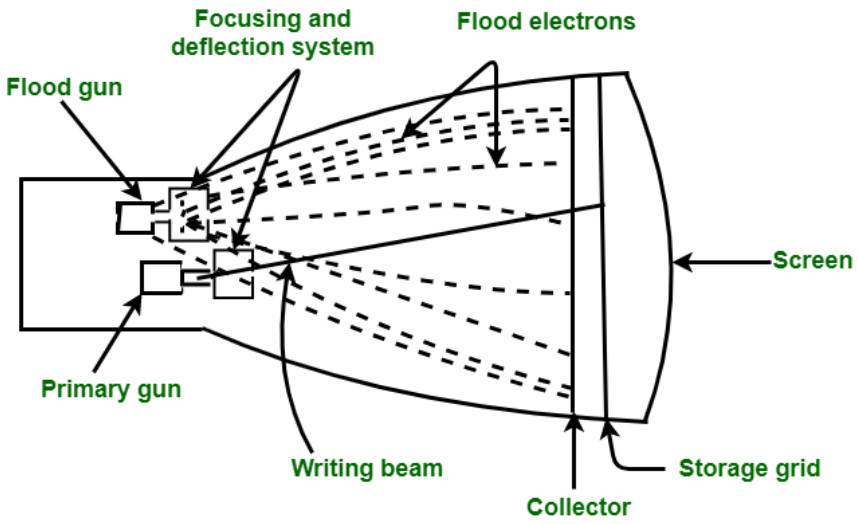
The above displays are called refresh line drawing displays, because the picture vanishes (typically in about 100 Milli seconds) and the pictures have to be continuously refreshed so that the human persistence of vision makes them see as static pictures. They are costly on one hand and also tend to flicker when complex pictures are displayed (Because refreshing because complex).

These problems are partly overcome by devices with inherent storage devices – i.e. they continue to display the pictures, till they are changed or at least for several minutes without the need of being refreshed. We see one such device called the Direct View Storage Tube (DVST) below.

Direct View Storage Tubes:

DVST terminals also use the random scan approach to generate the image on the CRT screen. The term "storage tube" refers to the ability of the screen to retain the image which has been projected against it, thus avoiding the need to rewrite the image constantly.

Direct View Storage Tube (DVST) resembles CRT as it uses electron gun to draw picture and phosphor coated screen to display it. The phosphor used in this is of high persistence. DVST does not use refresh buffer or frame buffer to store picture definition. Picture definition is stored in inside CRT in form positive charged distribution. Because of this reason DVST is knows as Storage Type CRT. In DVST no refreshing is required as result picture drawn on DVST will be seen for several minutes before fading.



Various components of DVST :

1. **Electron guns –**

   Two electron guns are used in DVST : Primary Gun and Flood Gun.

   Primary gun is used to store picture pattern.

   Flood gun is used to maintain picture display on phosphor coated screen.

2. **Phosphor Coated Screen –**

   In DVST the inner surface of CRT is coated with phosphor crystals is of high persistence that emit light when beam of electrons strike them.

3. **Storage Mesh –**

   It is thin and high quality wire that is coated with dielectric and is located just behind phosphor coated screen. Primary gun deposits pattern of positive charge on this grid and it is transferred to phosphor coated screen by continuous flood of electrons produced by flood gun. Thus Storage Mesh stores picture to be displayed in form of positive charge distribution.

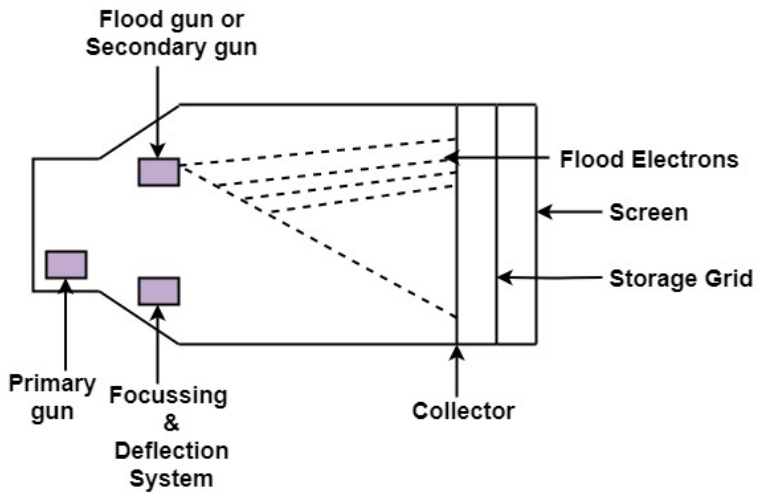
4. **Collector –**

   This grid is located just behind storage mesh and purpose of this negatively charged grid is to smooth out flow of flood electrons.

Working principle of DVST:

In DVST similar with CRT electron gun and phosphor coated method is used. But in this no electron beam is used to directly writing pictures on screen, but instead of this we can used Storage mesh wire grid is used it is just located behind phosphor coated screen. There is also another grid located just behind storage mesh is called Collector and this purpose is to smooth out flow of flood electrons. The flood gun produce large number of electrons, this negatively charged grid reduces speed of these electrons. Then electrons pass through collector at low velocity and attracted by positive charged portions of storage mesh and strike at portions of phosphor coated screen to display picture. Some electrons get repelled by other portions of mesh that are negatively charged.

Since the collector has slowly down electrons, in this way they not able to produce sharpened images. So to reduce this problem, screen itself is maintained at a high positive potential by means of voltage applied to thin aluminium coating between tube face and phosphor.



Direct View Storage Tube

Advantages of DVST :

1. For picture display it does not require refreshing.
2. Display complex pictures at high resolution without any flicker.
3. No use of frame buffer or refresh buffer.
4. Cost is very less.

Disadvantages of DVST :

1. Not used for dynamic graphic such as animation.
2. These systems do not display colors.
3. To erase selected part of an image, entire screen needs to be erased and modified pictures needs to be redrawn.
4. It is not suitable for dynamic graphics applications.

Plotter

A plotter is a computer hardware device much like a printer that is used for printing vector graphics. Instead of toner, plotters use a pen, pencil, marker, or another writing tool to draw multiple, continuous lines onto paper rather than a series of dots like a traditional printer. Though once widely used for computer-aided design, these devices have more or less been phased out by wide-format printers. Plotters produce a hard copy of schematics and other similar applications.

Plotters are a special type of output device. It is suitable for applications:

1. Architectural plan of the building.
2. CAD applications like the design of mechanical components of aircraft.
3. Many engineering applications.



Advantages of Plotters

1. It can produce high-quality output on large sheets.
2. It is used to provide the high precision drawing.
3. It can produce graphics of various sizes.
4. The speed of producing output is high.
5. They can print on a wide variety of flat materials including plywood, aluminum, sheet steel, cardboard, and plastic.
6. Plotters allow the same pattern to be drawn thousands of times without any image degradation.

Disadvantages of Plotters

1. Plotters are quite large compared to a traditional printer.
2. Plotters are also much more expensive than a traditional printer.

Types of Plotters:

Drum Plotter:

It consists of a drum. Paper on which design is made is kept on the drum. The drum can rotate in both directions. Plotters comprised of one or more pen and penholders. The holders are mounted perpendicular to drum surface. The pens are kept in the holder, which can move left to the right as well as right to the left. The graph plotting program controls the movement of pen and drum.



Drum Plotter

Flatbed Plotter:

It is used to draw complex design and graphs, charts. The Flatbed plotter can be kept over the table. The plotter consists of pen and holder. The pen can draw characters of various sizes. There can be one or more pens and pen holding mechanism. Each pen has ink of different color. Different colors help to produce multicolour design of document. The area of plotting is also variable. It can vary A4 to 21'*52'.



Flatbed Plotter

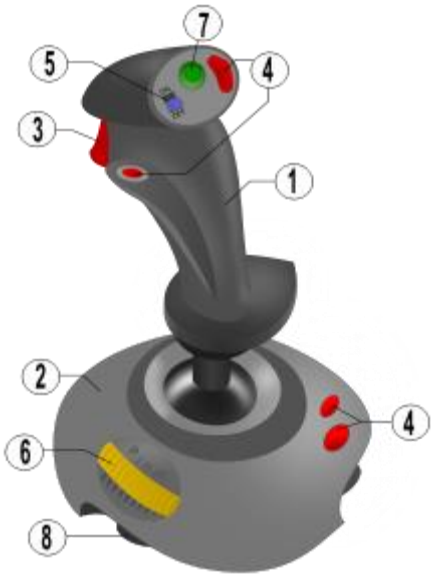
It is used to draw

1. Cars
2. Ships
3. Airplanes
4. Shoe and dress designing
5. Road and highway design

Joystick

A joystick is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it is controlling. A joystick, also known as the control column, is the principal control device in the cockpit of many civilian and military aircraft, either as a centre stick or side-stick. It often has supplementary switches to control various aspects of the aircraft's flight.

Joysticks are often used to control video games, and usually have one or more push-buttons whose state can also be read by the computer. A popular variation of the joystick used on modern video game consoles is the analog stick. Joysticks are also used for controlling machines such as cranes, trucks, underwater unmanned vehicles, wheelchairs, surveillance cameras, and zero turning radius lawn mowers. Miniature finger-operated joysticks have been adopted as input devices for smaller electronic equipment such as mobile phones.



Possible elements of a video game joystick:

1. stick,
2. base,
3. trigger,
4. extra buttons,
5. autofire switch,
6. throttle,
7. hat switch (POV hat),

8. suction cups.

A joystick is an input device that control a character or machine in a computer program, such as a plane in a flight simulator. They look similar to the control device you would find on an arcade game, but nearly always include extra buttons for additional functionality.



The picture shows the Logitech Freedom 2.4, an example of a joystick.

The first joystick was invented at the U.S. Naval Research Laboratory by C. B. Mirick and patented in 1926. It was a two-axis electronic joystick, similar to the joysticks in use today, and was original designed for remotely piloting aircraft.

Computer joystick ports

Today, most computer joysticks connect to the computer using a USB port. Below is a listing of all of the type of ports that have accepted a joystick.

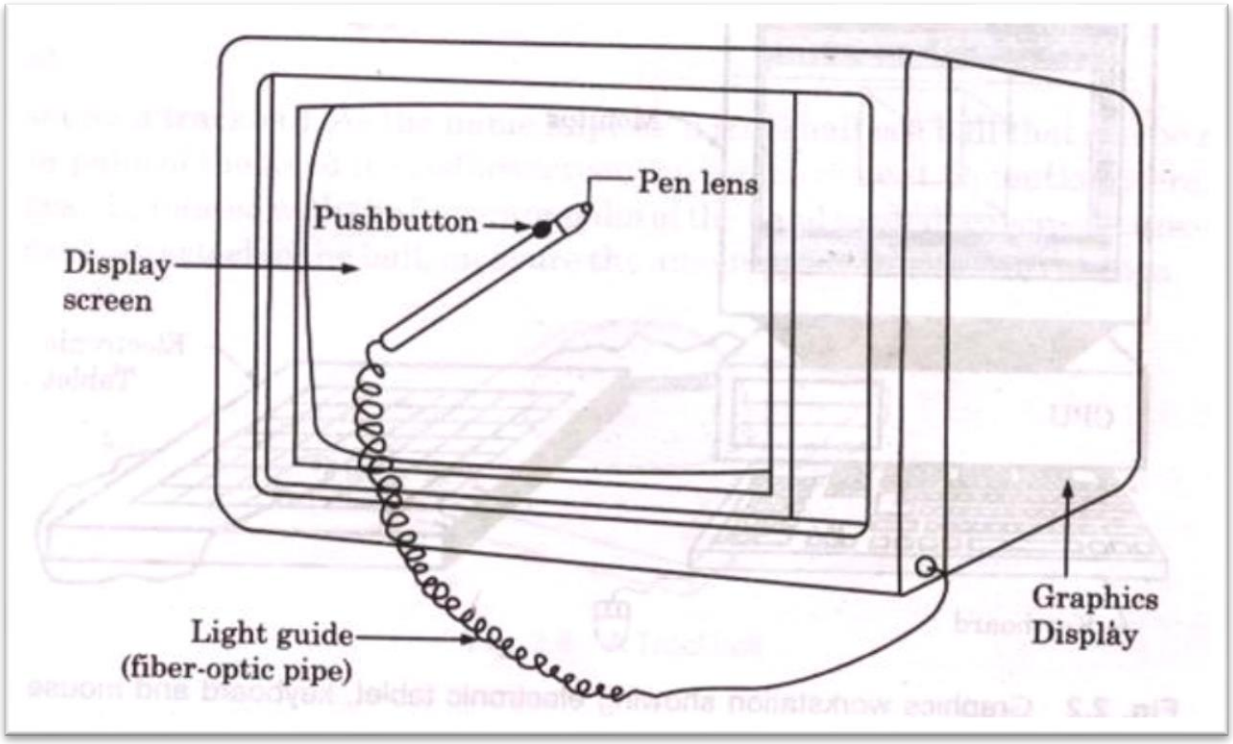
- Bluetooth
- Game port
- Serial Port
- USB

Light Pen

A light pen is a computer input device in the form of a light-sensitive was used in conjunction with a computer's cathode-ray tube (CRT) display.

Light Pen (similar to the pen) is a pointing device which is used to select a displayed menu item or draw pictures on the monitor screen.

It consists of a photocell and an optical system placed in a small tube. When its tip is moved over the monitor screen, and pen button is pressed, its photocell sensing element detects the screen location and sends the corresponding signals to the CPU.



Light Pen Components

It allows the user to point to displayed objects or draw on the screen in a similar way to a touchscreen but with greater positional accuracy.

It is a light sensitive pen-like input device and is used to select objects directly on the computer screen. It is used for making drawing, graphics and for menu selection. Figures and drawings can be made by moving the pen on computer screen.

A light pen can work with any CRT-based display, but its ability to be used with LCDs was unclear (though Toshiba and Hitachi displayed a similar idea at the "Display 2006" show in Japan[1]).



Photo of the Hypertext Editing System (HES) console in use at Brown University, circa October 1969. The photo shows HES on an IBM 2250 Mod 4 display station, including lightpen and programmed function keyboard, channel coupled to Brown's IBM 360 mainframe.

A light pen detects changes in brightness of nearby screen pixels when scanned by cathode-ray tube electron beam and communicates the timing of this event to the computer. Since a CRT scans the entire screen one pixel at a time, the computer can keep track of the expected time of scanning various locations on screen by the beam and infer the pen's position from the latest timestamp.

The first light pen was created around 1955 as part of the Whirlwind project at MIT.

During the 1960s, light pens were common on graphics terminals such as the IBM 2250 and were also available for the IBM 3270 text-only terminal.

Light pen usage was expanded in the early 1980s to music workstations such as the Fairlight CMI and personal computers such as the BBC Micro. IBM PC compatible CGA, HGC and some EGA graphics cards also featured a connector compatible with a light pen, as did early Tandy 1000 computers,[notes 1] the Thomson MO5 computer family, the Atari 8-bit, Commodore 8-bit, some MSX computers[4] and Amstrad PCW home computers. For the MSX computers, Sanyo produced a light pen interface cartridge.[5]

Because the user was required to hold their arm in front of the screen for long periods of time (potentially causing "gorilla arm") or to use a desk that tilts the monitor, the light pen fell out of use as a general-purpose input device.

Uses:

- Light Pens can be used as input coordinate positions by providing necessary arrangements.
- If background color or intensity, a light pen can be used as a locator.
- It is used as a standard pick device with many graphics system.

- It can be used as stroke input devices.
- It can be used as valuators

Pick Devices

Pick devices are used for providing input to the computer by pointing to a location on the computer monitor. The input data is not typed; the data is entered by pointing the pick device directly on the computer screen. Light pen and touch screen are some common pick devices.

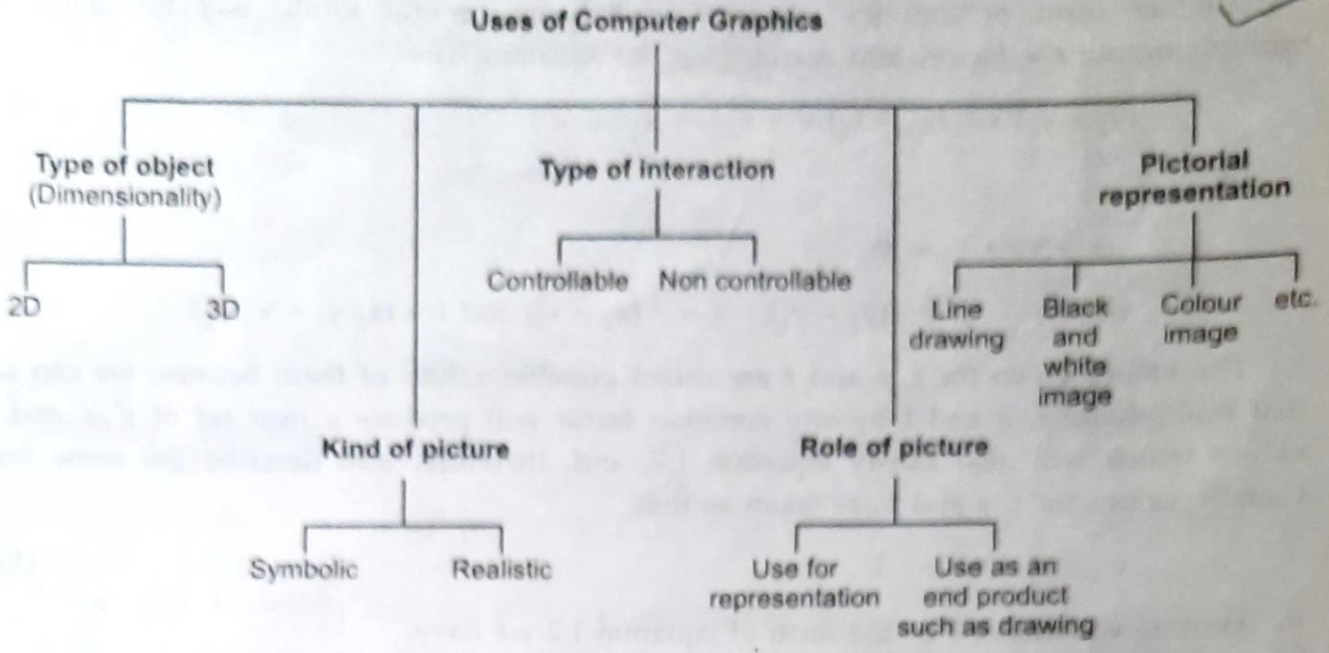


Fig. 1.3

1.2 Lines

Point is the fundamental element of the picture representation. It is nothing but the position in a plane defined as either pairs or triplets of numbers depending on whether the data are two or three dimensional. Thus, (x_1, y_1) or (x_1, y_1, z_1) would represent a point in either two or three dimensional space. Two points would represent a line or edge, and a collection of three or more points a polygon. The representation of curved lines is usually accomplished by approximating them by short straight line segments.

If the two points used to specify a line are (x_1, y_1) and (x_2, y_2), then an equation for the line is given as

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} \quad \ldots (1.1)$$

$$\therefore \quad y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1 \quad \ldots (1.2)$$

or $$y = mx + b \quad \ldots (1.3)$$

where $$m = \frac{y_2 - y_1}{x_2 - x_1} \quad \ldots (1.4)$$

and $$b = y_1 - mx_1 \quad \ldots (1.5)$$

The above equation is called the **slope intercept form** of the line. The **slope** m is the change in height $(y_2 - y_1)$ divided by the change in width $(x_2 - x_1)$ for two points on the line. The **intercept** b is the height at which the line crosses the y-axis.

Another form of the line equation, called the general form, may be found by multiplying out the factors and rearranging the equation 1.1.

$$(y_2 - y_1)\, x - (x_2 - x_1)\, y + x_2 y_1 - x_1\, y_2 = 0 \qquad \ldots(1.6)$$

or

$$rx + sy + t = 0 \qquad \ldots (1.7)$$

where $r = (y_2 - y_1)$, $s = -(x_2 - x_1)$ and $t = (x_2\, y_1 - x_1\, y_2)$

The values given for r, s and t are called possible values of them because we can see that multiplying r, s and t by any common factor will produce a new set of r′, s′ and t′ values which will also satisfy equation 1.7, and, therefore, also describe the same line. Usually values for r, s and t are taken so that

$$r^2 + s^2 = 1 \qquad \ldots (1.8)$$

Writing equation 1.7 in the form of equation 1.2 we have,

$$y = \frac{-rx}{s} - \frac{t}{s}$$

We have, $m = \frac{-r}{s}$ and $b = -\frac{t}{s}$

We can determine whether two lines are crossing or parallel. When two lines cross, they share some point in common. Of course, that point satisfies equations for the two lines. Let us see how to obtain this point.

Consider two lines having slopes m_1 and m_2 and y-intercepts b_1 and b_2 respectively. Then we can write the equations for these two lines as,

$$\text{Line 1 : } y = m_1 x + b_1 \qquad \ldots (1.9)$$

$$\text{Line 2 : } y = m_2 x + b_2 \qquad \ldots (1.10)$$

If point (x_p, y_p) is shared by both lines, then it should satisfy equations 1.9 and 1.10

$$y_p = m_1\, x_p + b_1 \quad \text{and} \qquad \ldots (1.11)$$

$$y_p = m_2\, x_p + b_2 \qquad \ldots (1.12)$$

Equating the above equations gives,

$$m_1\, x_p + b_1 = m_2\, x_p + b_2$$

$$x_p\, (m_1 - m_2) = b_2 - b_1$$

$$x_p = \frac{b_2 - b_1}{m_1 - m_2} \qquad \ldots (1.13)$$

Substituting this into equation for line1, 1.11 (or for line2, 1.12) gives,

$$y_P = m_1\left(\frac{b_2 - b_1}{m_1 - m_2}\right) + b_1$$

$$\therefore \quad y_P = \frac{m_1(b_2 - b_1) + b_1(m_1 - m_2)}{m_1 - m_2}$$

$$\therefore \quad y_P = \frac{b_2 m_1 - b_1 m_2}{m_1 - m_2} \qquad \ldots (1.14)$$

Therefore, the point

$$\left(\frac{b_2 - b_1}{m_1 - m_2}, \frac{b_2 m_1 - b_1 m_2}{m_1 - m_2}\right) \qquad \ldots (1.15)$$

is the point shared by both the lines, i.e. intersection point.

If the two lines are parallel, they have the same slope. In the expression 1.15 for the point of intersection, $m_1 = m_2$. So the denominator becomes zero. So the expression results in a divide by zero. This means, there is no point of intersection for the parallel lines.

If we consider the expression for line given by 1.7, then by the similar mathematical calculations, we will get the intersecting point,

$$\left(\frac{s_1 t_2 - s_2 t_1}{s_2 r_1 - s_1 r_2}, \frac{t_1 r_2 - t_2 r_1}{s_2 r_1 - s_1 r_2}\right) \qquad \ldots (1.16)$$

1.3 Line Segments



Fig. 1.4 A line segment

When we say the line, it extends forever both forward and backward. Any point in a given direction satisfies equation of the line. In graphics, we need to display only pieces of lines. When we consider the piece of line, i.e. only those points which lie between two endpoints, then this is called a line segment. Fig. 1.4 shows an example of a line segment, where P_1 and P_2 are the endpoints.

From the endpoints of a line segment the equation of the line can be obtained. Let the endpoints be P_1 (x_1, y_1) and P_2 (x_2, y_2), and equation of the line, $y = mx + b$. Whether any point lies on the line segment or not, can be determined from the equation of the line and the end-points. If any another point $P_i(x_i, y_i)$, is considered, then it lies on the segment if the following conditions are satisfied.

1. $y_i = mx_i + b$
2. $\min(x_1, x_2) \leq x_i \leq \max(x_1, x_2)$
3. $\min(y_1, y_2) \leq y_i \leq \max(y_1, y_2)$

Note : Here min (x_1, x_2) means the smallest of x_1 and x_2, and max (x_1, x_2) means the largest of x_1 and x_2.

Parametric form

The equations which give the x and y values on the line in terms of a parameter, u are called 'parametric form' of the line equation. This form is useful for constructing line segments. For example, we want line segment between (x_1, y_1) and (x_2, y_2). If the parameter is u and if we want the x coordinate to go uniformly from x_1 to x_2, then the expression for x can be written as,

$$x = x_1 + (x_2 - x_1)u \quad \ldots (1.17)$$

The line segment points correspond to value of the parameter between 0 and 1. So when u is 0, x is x_1. As u approaches to 1, x changes uniformly to x_2. For a line segment, when x changes from x_1 to x_2, at the same time, y must change from y_1 to y_2 uniformly.

i.e. $$y = y_1 + (y_2 - y_1)u \quad \ldots (1.18)$$

The two equations 1.17 and 1.18 together describe a straight line. This can be shown by equating over u. If the parameter value for a point on the line is given, we can easily test if the point lies on the line segment.

Length of a line segment

If the two endpoints of a line segment are known, we can obtain its length. Consider the two endpoints of a line segment as P_1 (x_1, y_1) and P_2 (x_2, y_2) and L is the length of a line segment. Construct a right triangle P_1 P_2 A as shown in Fig. 1.5.

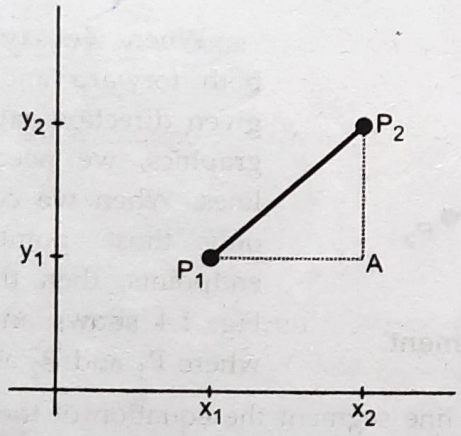


Fig. 1.5 Right triangle P_1P_2A

The Pythagorean theorem states that the square of the length of the hypotenuse is equal to the sum of the squares of the lengths of the two adjacent sides. Here, hypotenuse is P_1 P_2 and two adjacent sides are P_1A and P_2A. The coordinates of P_1, P_2 and A are (x_1, y_1), (x_2, y_2) and (x_2, y_1) respectively. Then the length of the segment P_1P_2, L is given by,

$$L^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \quad \ldots (1.19\ (a))$$

$$\therefore \quad L = [(x_2 - x_1)^2 + (y_2 - y_1)^2]^{1/2} \qquad \ldots(1.19(b))$$

Midpoint of a line segment

To obtain the mid-point of a line segment, take the x coordinate halfway between the x coordinates of the endpoints and y coordinate halfway between the y coordinates of the endpoints. Thus, the mid point is,

$$(x_m, y_m) = \left[\frac{(x_1 + x_2)}{2}, \frac{(y_1 + y_2)}{2}\right] \qquad \ldots (1.20)$$

The mid-point of a line segment can be calculated easily is oftenly useful.

1.4 Vectors

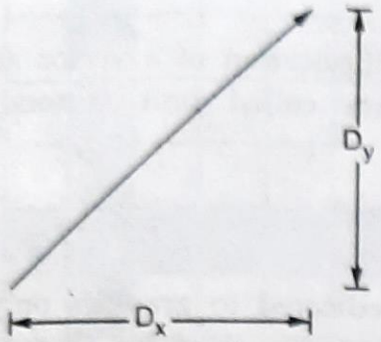


Fig. 1.6 A vector

A vector has a single direction and a length. A vector may be indicated by $[D_x, D_y]$ where D_x denotes the distance on x-axis direction and D_y denote the distance on y-axis direction, as shown in Fig. 1.6.

When we consider a line segment, it has a fixed position in space. But vector does not have a fixed position in space. The vector does not tell us the starting point. It gives how far to move and in which direction.

Vector notation

The vectors may be denoted by a shorthand way. The operations are performed on all coordinates.

For example, the vector form of parametric equations for a line can be written as,

$$V = V_1 + u(V_2 - V_1) \qquad \ldots (1.21)$$

Addition of vectors

The addition of vectors is nothing but the addition of their respective components. If the two vectors are V_1 $[D_{x1}, D_{y1}]$ and V_2 $[D_{x2}, D_{y2}]$, then the addition of V_1 and V_2 is given by,

$$\begin{aligned} V_1 + V_2 &= [D_{x1}, D_{y1}] + [D_{x2}, D_{y2}] \\ &= [D_{x1} + D_{x2}, D_{y1} + D_{y2}] \end{aligned} \qquad \ldots (1.22)$$

We can visualize the vector addition by considering a point A. The first vector V_1 moves from point A to point B, the second vector V_2 moves from point B to point C. Then the addition of V_1 and V_2 is nothing but the vector from point A to point C.

Multiplication of vectors

The vector can be multiplied by a number. The multiplication is performed by multiplying each component of a vector by a number.

Let $V(D_x, D_y]$ by a vector which is to be multiplied by a number n. Then,

$$nV = n [D_x, D_y] = [n D_x, nD_y] \quad \ldots (1.23)$$

The magnitude of the resultant vector is given by,

$$|V| = (D_x^2 + D_y^2)^{1/2} \quad \ldots (1.24)$$

The result of the multiplication changes the magnitude of a vector but prreserves its direction.

Unit vectors

From equation 1.18 and 1.23, we can say that the multiplication of a vector and the reciprocal of its length is equal to 1. Such vectors are called unit vectors. They conveniently capture the direction information.

1.5 Pixels and Frame Buffers

In raster scan displays a special area of memory is dedicated to graphics only. This memory area is called **frame buffer**. It holds the set of intensity values for all the screen points. The stored intensity values are retrieved from frame buffer and displayed on the screen one row (scan line) at a time. Each screen point is referred to as a **pixel** or **pel** (shortened forms of picture element). Each pixel on the screen can be specified by its row and column number. Thus by specifying row and column number we can specify the **pixel position** on the screen.

Intensity range for pixel positions depends on the capability of the raster system. It can be a simple black and white system or colour system. In a simple black and white system, each pixel position is either on or off, so only one bit per pixel is needed to control the intensity of the pixel positions. Additional bits are required when colour and intensity variations can be displayed. Upto 24 bits per pixel are included in high quality display systems, which can require several megabytes of storage space for the frame buffer. On a black and white system with one bit per pixel, the frame buffer is commonly called a **bitmap**. For systems with multiple bits per pixel, the frame buffer is often referred to as a **pixmap**.

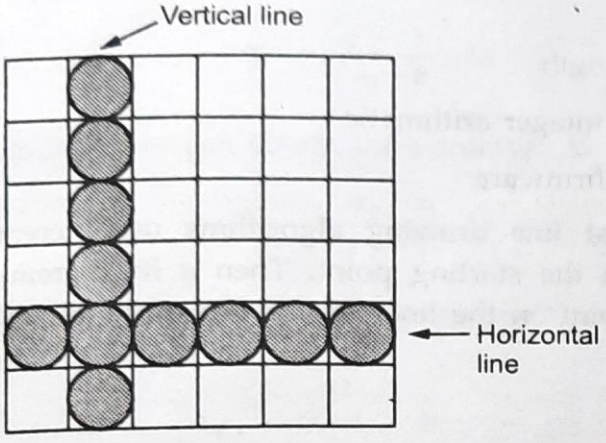
1.6 Vector Generation

The process of 'turning on' the pixels for a line segment is called **vector generation**. In this section, we study the vector generation algorithm, Bresenham's line algorithm and circle drawing algorithms.

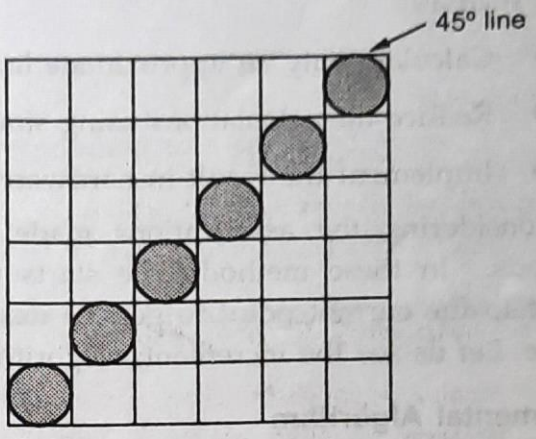
Before discussing specific line drawing algorithms it is useful to note the general requirements for such algorithms. These requirements specify the desired characteristics of line.

- The line should appear as a straight line and it should start and end accurately.
- The line should be displayed with constant brightness along its length independent of its length and orientation.
- The line should be drawn rapidly.

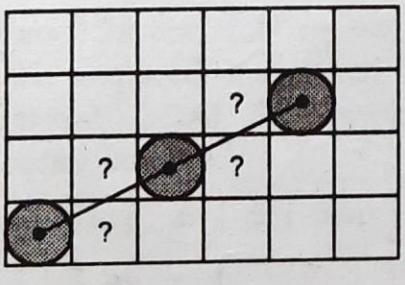
Let us see the different lines drawn in Fig. 1.7.

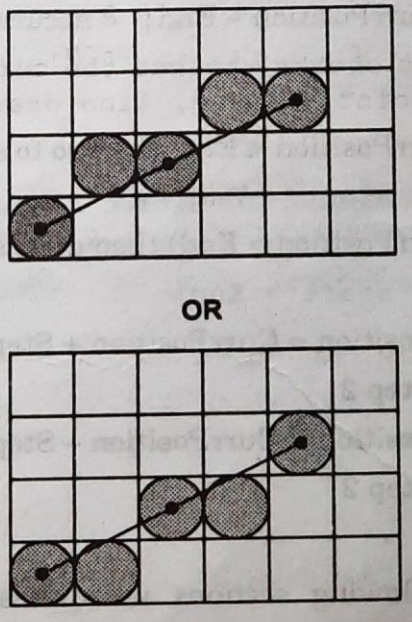


(a) Vertical and Horizontal lines



(b) 45° line





(c) Line with other orientation

Fig. 1.7

As shown in Fig. 1.7 (a), horizontal and vertical lines are straight and have same width. The 45° line is straight but its width is not constant. On the other hand, the line with any other orientation is neither straight nor has same width. Such cases are due to the finite resolution of display and we have to accept approximate pixels in such situations, shown in Fig. 1.7 (c).

The brightness of the line is dependent on the orientation of the line. We can observe that the effective spacing between pixels for the 45° line is greater than for the vertical and horizontal lines. This will make the vertical and horizontal lines appear brighter than the 45° line. Complex calculations are required to provide equal brightness along lines of varying length and orientation. Therefore, to draw line rapidly some compromises are made such as

- Calculate only an approximate line length
- Reduce the calculations using simple integer arithmetic
- Implement the result in hardware or firmware

Considering the assumptions made most line drawing algorithms use incremental methods. In these methods line starts with the starting point. Then a fix increment is added to the current point to get the next point on the line. This is continued till the end of line. Let us see the incremental algorithm.

Incremental Algorithm

1. CurrPosition = Start
   Step = Increment
2. if (| CurrPosition – End | < Accuracy) then go to step 5
   `[ This checks whether the current position is reached upto approximate end point. If yes, line drawing is completed. ]`
   if (CurrPosition < End) then go to step 3
   `[Here start < End]`
   if (CurrPosition > End) then go to step 4
   `[Here start > End]`
3. CurrPosition = CurrPosition + Step
   go to step 2
4. CurrPosition = CurrPosition – Step
   go to step 2
5. Stop.

In the following sections we discuss the line rasterizing algorithms based on the incremental algorithm.

1.6.1 Vector Generation/Digital Differential Analyzer (DDA) Algorithm

The vector generation algorithms which step along the line to determine the pixel which should be turned on are sometimes called **digital differential analyzer (DDA)**.

We know that the slope of a straight line is given as

$$m = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \quad \ldots (1.25)$$

The above differential equation can be used to obtain a rasterized straight line. For any given x interval Δx along a line, we can compute the corresponding y interval Δy from equation 1.25 as

$$\Delta y = \frac{y_2 - y_1}{x_2 - x_1} \Delta x \quad \ldots (1.26)$$

Similarly, we can obtain the x interval Δx corresponding to a specified Δy as

$$\Delta x = \frac{x_2 - x_1}{y_2 - y_1} \Delta y \quad \ldots (1.27)$$

Once the intervals are known the values for next x and next y on the straight line can be obtained as follows

$$x_{i+1} = x_i + \Delta x$$

$$= x_i + \frac{x_2 - x_1}{y_2 - y_1} \Delta y \quad \ldots (1.28)$$

and

$$y_{i+1} = y_i + \Delta y$$

$$= y_i + \frac{y_2 - y_1}{x_2 - x_1} \Delta x \quad \ldots (1.29)$$

The equations 1.28 and 1.29 represent a recursion relation for successive values of x and y along the required line. Such a way of rasterizing a line is called a **digital differential analyzer** (DDA). For simple DDA either Δx or Δy, whichever is larger, is chosen as one raster unit, i.e.

if $|\Delta x| \geq |\Delta y|$ then

else

$$\Delta x = 1$$
$$\Delta y = 1$$

With this simplification, if $\Delta x = 1$ then

we have $y_{i+1} = y_i + \frac{y_2 - y_1}{x_2 - x_1}$ and

$$x_{i+1} = x_i + 1$$

If $\Delta y = 1$ then

we have

$$y_{i+1} = y_i + 1 \text{ and}$$

$$x_{i+1} = x_i + \frac{x_2 - x_1}{y_2 - y_1}$$

Let us see the vector generation/digital differential analyzer (DDA) routine for rasterizing a line

Vector Generation/DDA Line Algorithm

1. Read the line end points (x_1, y_1) and (x_2, y_2) such that they are not equal.
   [if equal then plot that point and exit]
2. $\Delta x = |x_2 - x_1|$ and $\Delta y = |y_2 - y_1|$
3. if $(\Delta x \geq \Delta y)$ then
   length = Δx
   else
   length = Δy
   end if
4. $\Delta x = (x_2 - x_1) / \text{length}$
   $\Delta y = (y_2 - y_1) / \text{length}$
   [This makes either Δx or Δy equal to 1 because length is either $|x_2 - x_1|$ or $|y_2 - y_1|$. Therefore, the incremental value for either x or y is one.]
5. $x = x_1 + 0.5 * \text{Sign}(\Delta x)$
   $y = y_1 + 0.5 * \text{Sign}(\Delta y)$
   [Here, Sign function makes the algorithm work in all quadrant. It returns - 1, 0, 1 depending on whether its argument is < 0, = 0, > 0 respectively. The factor 0.5 makes it possible to round the values in the integer function rather than truncating them.]
6. i = 1 [Begins the loop, in this loop points are plotted]
   While (i ≤ length)
   {
   Plot (Integer (x), Integer (y))
   x = x + Δx
   y = y + Δy
   i = i + 1
   }
7. Stop

Let us see few examples to illustrate this algorithm.

Example 1.1 : *Consider the line from (0, 0) to (4, 6). Use the simple DDA algorithm to rasterize this line*

Solution : Evaluating steps 1 to 5 in the DDA algorithm we have

$x_1 = 0 \qquad y_1 = 0 \qquad x_2 = 4 \qquad y_2 = 6$

$\therefore$ Length $= |y_2 - y_1| = 6$

$\therefore \quad \Delta x = |x_2 - x_1| \,/\, \text{length}$

$= \frac{4}{6}$

and $\quad \Delta y = |y_2 - y_1| \,/\, \text{length}$

$= 6 / 6 = 1$

Initial value for

$x = 0 + 0.5 * \text{Sign}\left(\frac{4}{6}\right) = 0.5$

$y = 0 + 0.5 * \text{Sign}\,(1) = 0.5$

Tabulating the results of each iteration in the step 6 we get,

i	Plot	x	y
		0.5	0.5
1	(0, 0)		
		1.167	1.5
2	(1, 1)		
		1.833	2.5
3	(1, 2)		
		2.5	3.5
4	(2, 3)		
		3.167	4.5
5	(3, 4)		
		3.833	5.5
6	(3, 5)		
		4.5	6.5

Table 1.1

The results are plotted as shown in the Fig. 1.8. It shows that the rasterized line lies to both sides of the actual line, i.e. the algorithm is **orientation dependent**.

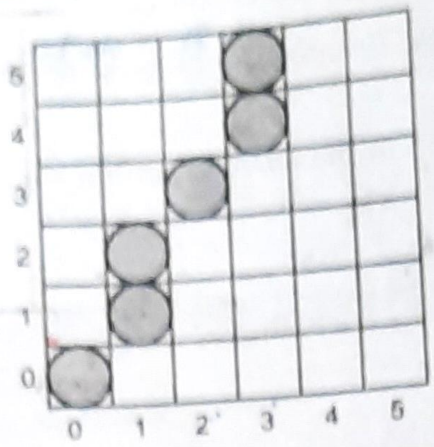


Fig. 1.8 Result for a simple DDA

Example 1.2 : *Consider the line from (0, 0) to (– 6, – 6). Use the simple DDA algorithm to rasterize this line.*

Solution : Evaluating steps 1 to 5 in the DDA algorithm we have

$x_1 = 0$ $y_1 = 0$

$x_2 = -6$ $y_2 = -6$

$\therefore$ Length $= |x_2 - x_1| = |y_2 - y_1| = 6$

$\therefore$ $\Delta x = \Delta y = -1$

Initial values for

$x = 0 + 0.5 * \text{Sign}(-1) = -0.5$

$y = 0 + 0.5 * \text{Sign}(-1) = -0.5$

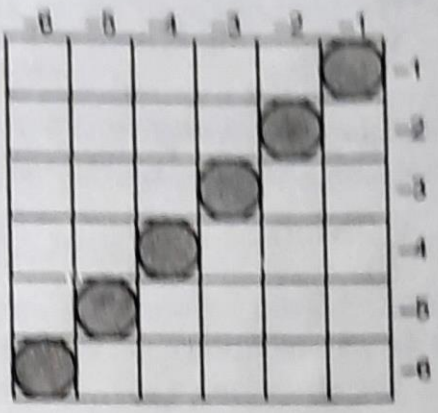
Tabulating the results of each iteration in the step 6 we get,

I	Plot	x	y
		– 0.5	– 0.5
1	(– 1, – 1)		
		– 1.5	– 1.5
2	(– 2, – 2)		
		– 2.5	– 2.5
3	(– 3, – 3)		
		– 3.5	– 3.5
4	(– 4, – 4)		

		-4.5	-4.5
5	(-5, -5)		
		-5.5	-5.5
6	(-6, -6)		
		-6.5	-6.5

Table 1.2

The results are plotted as shown in the Fig. 1.9. It shows that the rasterized line lies on the actual line and it is 45° line.



* -ve pixel values are with reference to pixel at the center of screen

Fig. 1.9 Result for a simple DDA

'C' code for DDA Line Drawing Algorithm

(Softcopy of this program is available at vtubooks.com)

```
#include<stdio.h>
#include<graphics.h>
#include<math.h>
main()
{
float x,y,x1,y1,x2,y2,dx,dy,length;
int i,gd,gm;
clrscr();

/* Read two end points of line
========================== */
printf("Enter the value of x1 :\t");
scanf("%f",&x1);
printf("Enter the value of y1 :\t");
```

```
getch();
closegraph();
}
```

Advantages of DDA Algorithm

1. It is the simplest algorithm and it does not require special skills for implementation.
2. It is a faster method for calculating pixel positions than the direct use of equation $y = mx + b$. It eliminates the multiplication in the equation by making use of raster characteristics, so that appropriate increments are applied in the x or y direction to find the pixel positions along the line path.

Disadvantages of DDA Algorithm

1. Floating point arithmetic in DDA algorithm is still time-consuming.
2. The algorithm is orientation dependent. Hence end point accuracy is poor.

1.6.2 Bresenham's Line Algorithm

Bresenham's line algorithm uses only integer addition and subtraction and multiplication by 2, and we know that the computer can perform the operations of integer addition and subtraction very rapidly. The computer is also time-efficient when performing integer multiplication by powers of 2. Therefore, it is an efficient method for scan-converting straight lines.

The basic principle of Bresenham's line algorithm is to select the optimum raster locations to represent a straight line. To accomplish this the algorithm always increments either x or y by one unit depending on the slope of line. The increment in the other variable is determined by examining the distance between the actual line location and the nearest pixel. This distance is called **decision variable** or the **error**. This is illustrated in the Fig. 1.10.

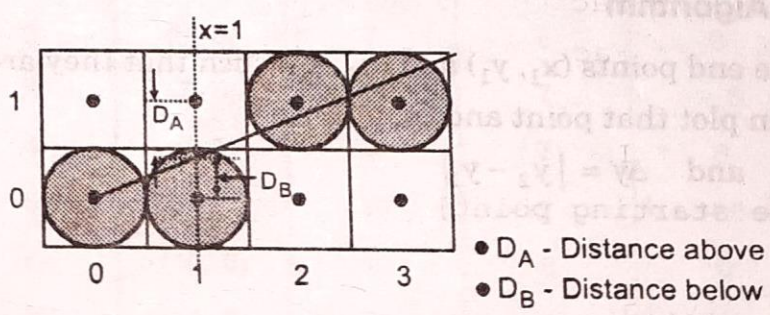


Fig. 1.10

As shown in the Fig. 1.10, the line does not pass through all raster points (pixels). It passes through raster point (0, 0) and subsequently crosses three pixels. It is seen that the intercept of line with the line $x = 1$ is closer to the line $y = 0$, i.e. pixel (1, 0) than to the line $y = 1$ i.e. pixel (1, 1). Hence, in this case, the raster point at (1, 0) better represents the path of the line than that at (1, 1). The intercept of the line with the line $x = 2$ is close to

the line y = 1, i.e. pixel (2, 1) than to the line y = 0, i.e. pixel (2, 0). Hence, the raster point at (2, 1) better represents the path of the line, as shown in the Fig. 1.10.

In mathematical terms error or decision variable is defined as

$$e = D_B - D_A \quad \text{or} \quad D_A - D_B$$

Let us define $e = D_B - D_A$. Now if $e > 0$, then it implies that $D_B > D_A$, i.e., the pixel above the line is closer to the true line. If $D_B < D_A$ (i.e. $e < 0$) then we can say that the pixel below the line is closer to the true line. Thus by checking only the sign of error term it is possible to determine the better pixel to represent the line path.

The error term is initially set as

$$e = 2\Delta y - \Delta x$$

where $\Delta y = y_2 - y_1$, and $\Delta x = x_2 - x_1$

Then according to value of e following actions are taken.

```
while ( e ≥ 0)
    {
        y = y + 1
        e = e – 2 * Δx
    }
x = x + 1
e = e + 2 * Δy
```

When $e \geq 0$, error is initialized with $e = e - 2\Delta x$. This is continued till error is negative. In each iteration y is incremented by 1. When $e < 0$, error is initialized to $e = e + 2\Delta y$. In both the cases x is incremented by 1. Let us see the Bresenham's line drawing algorithm.

Bresenham's Line Algorithm

1. Read the line end points (x_1, y_1) and (x_2, y_2) such that they are not equal.
   [if equal then plot that point and exit]
2. $\Delta x = |x_2 - x_1|$ and $\Delta y = |y_2 - y_1|$
3. [Initialize starting point]
   $x = x_1$
   $y = y_1$
4. $e = 2 * \Delta y - \Delta x$
   [Initialize value of decision variable or error to compensate for nonzero intercepts]
5. i = 1 [Initialize counter]
6. Plot (x, y)
7. while (e ≥ 0)

{

$y = y + 1$

$e = e - 2 * \Delta x$

}

$x = x + 1$

$e = e + 2 * \Delta y$

8. $i = i + 1$
9. if ($i \leq \Delta x$) then go to step 6.
10. Stop

Example 1.3 : *Consider the line from (5, 5) to (13, 9). Use the Bresenham's algorithm to rasterize the line.*

Solution : Evaluating steps 1 through 4 in the Bresenham's algorithm we have,

$$\Delta x = | 13 - 5 | = 8$$

$$\Delta y = | 9 - 5 | = 4$$

$$x = 5$$

$$y = 5$$

$$e = 2 * \Delta y - \Delta x = 2 * 4 - 8$$

$$= 0$$

Tabulating the results of each iteration in the step 5 through 10.

i	Plot	x	y	e
		5	5	0
1	(5, 5)	6	6	- 8
2	(6, 6)	7	6	0
3	(7, 6)	8	7	- 8
4	(8, 7)	9	7	0
5	(9, 7)	10	8	- 8
6	(10, 8)	11	8	0
7	(11, 8)	12	9	- 8
8	(12, 9)	13	9	0
9	(13, 9)	14	10	- 8

Table 1.3

The results are plotted as shown in Fig. 1.11.

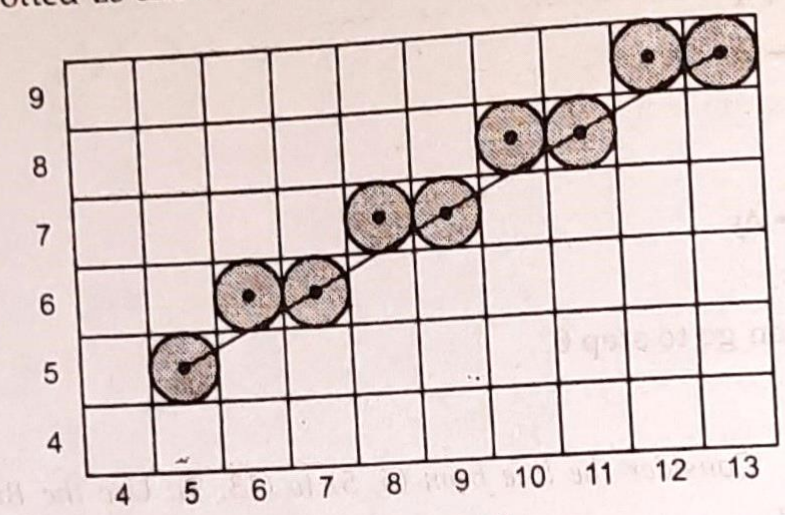


Fig. 1.11

'C' code for Bresenham's Line Drawing Algorithm

(Softcopy of this program is available at vtubooks.com)

```
#include<stdio.h>
#include<graphics.h>
#include<math.h>
main()
{
float x,y,x1,y1,x2,y2,dx,dy,e;
int i,gd,gm;
clrscr();

/* Read two end points of line
---------------------------------- */
printf("Enter the value of x1 :\t");
scanf("%f",&x1);
printf("Enter the value of y1 :\t");
scanf("%f",&y1);
printf("Enter the value of x2 :\t");
scanf("%f",&x2);
printf("Enter the value of y2 :\t");
scanf("%f",&y2);
```

4

2D Transformations

4.1 Introduction

Almost all graphics systems allow the programmer to define picture that include a variety of transformations. For example, the programmer is able to magnify a picture so that detail appears more clearly, or reduce it so that more of the picture is visible. The programmer is also able to rotate the picture so that he can see it in different angles. In this chapter we discuss the 2D transformations.

4.2 Matrices

In computer graphics images are generated from a series of line segments which are represented by the coordinates of their end points. Due to this we can make changes in an image by performing mathematical operations on these coordinates. When we change the image we call image is transformed. Before going to study various image transformations let us review some of the mathematical tools we shall need, namely matrix multiplication.

We know that matrix is a two dimensional array of number. For example,

$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & 3 \\ 0 & 1 & 2 \\ 5 & 4 & 2 \end{bmatrix}$$

are four different matrices. The elements in the matrix are identified by specifying the row and column number. Suppose matrix A is

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Then the element 8 is identified as A (3, 2), i.e. element 8 is in the 3^{rd} row and 2^{nd} column of the matrix A.

Matrix Multiplication

Matrix multiplication is more complex than the simple product of two numbers; it involves simple products and sums of the matrix elements. We can multiply two matrices A and B together only if the number of column of the first matrix A is the same as the number of rows of the second matrix B. If B matrix is

$$B = \begin{bmatrix} 1 & 2 \\ -2 & 0 \\ 3 & 1 \end{bmatrix}$$

then we can multiply matrix A and B because matrix A has 3 colums and matrix B has 3 rows. Unlike multiplication of numbers, the multiplication of matrix is not commutative; that is, while we can multiply A times B, we cannot multiply B times A, because B has 2 columns and A has 3 rows. When we multiply two matrics we get resultant matrices such as it has same number of rows as the first matrix of the two being multiplied and the same number of columns as the second matrix. Therefore, multiplication of A and B gives 3×2 resultant C matrix.

The generalized formula for matrix multiplication is given by

$$C(i,k) = \sum_j A(i,j)\ B(j,k)$$

In our example, the element C(1,1) is found by multiplying each element of the first row of A by the corresponding element of the first column of B and adding these products together.

$$\begin{aligned} C(1,1) &= A(1,1)\ B(1,1) + A(1,2)\ B(2,1) + A(1,3)\ B(3,1) \\ &= (1)\ (1) + (2)\ (-2) + (3)\ (3) \\ &= 6 \end{aligned}$$

Similarly,

$$\begin{aligned} C(1,2) &= A(1,1)\ B(1,2) + A(1,2)\ B(2,2) + A(1,3)\ B(3,2) \\ &= (1)\ (2) + (2)\ (0) + (3)\ (1) \\ &= 5 \end{aligned}$$

Performing similar arithmetic we have

$$C = \begin{bmatrix} 6 & 5 \\ 12 & 14 \\ 18 & 23 \end{bmatrix}$$

Multiplication is associative. This means that if we have several matrices to multiply together, it does not matter which we multiply first. For example, if A, B and C are the three matrices then

$$A\,(BC) = (AB)\,C$$

This is very useful property. It allows us to combine several graphics transformations into a single transformation, thereby making our calculations more efficient.

Identity Matrix

The square matrix which has all the elements equal to 0 except the elements of the main diagonal, which are all 1 is called identity matrix. For example,

$$[1] \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and so on.

The identity matrix is denoted as matrix I and

$$A = A \cdot I$$

4.3 Two Dimensional Transformations

In this section, we describe the general procedures for applying translation, rotation, and scaling parameters to reposition and resize the two dimensional objects.

4.3.1 Translation

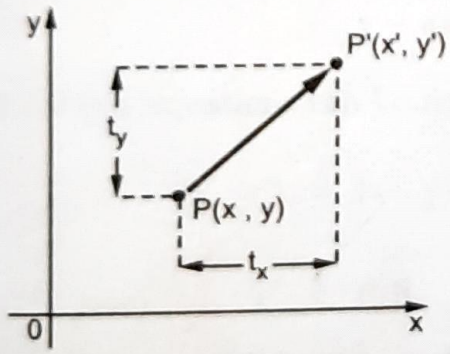


Fig. 4.1

Translation is a process of changing the position of an object in a straight-line path from one co-ordinate location to another. We can translate a two dimensional point by adding translation distances, t_x and t_y, to the original co-ordinate position (x, y) to move the point to a new position (x′, y′), as shown in the Fig. 4.1.

$$x' = x + t_x \qquad \ldots (4.1)$$

$$y' = y + t_y \qquad \ldots (4.2)$$

The translation distance pair (t_x, t_y) is called a **translation vector** or **shift vector**.

It is possible to express the translation equations 4.1 and 4.2 as a single matrix equation by using column vectors to represent coordinate positions and the translation vector :

$$P = \begin{bmatrix} x \\ y \end{bmatrix} \qquad P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \qquad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

This allows us to write the two dimensional translation equations in the matrix form :

$$P' = P + T \qquad \ldots (4.3)$$

Example 4.1 : *Translate a polygon with coordinates A (2, 5), B (7, 10) and C (10, 2) by 3 units in x direction and 4 units in y direction.*

Solution :

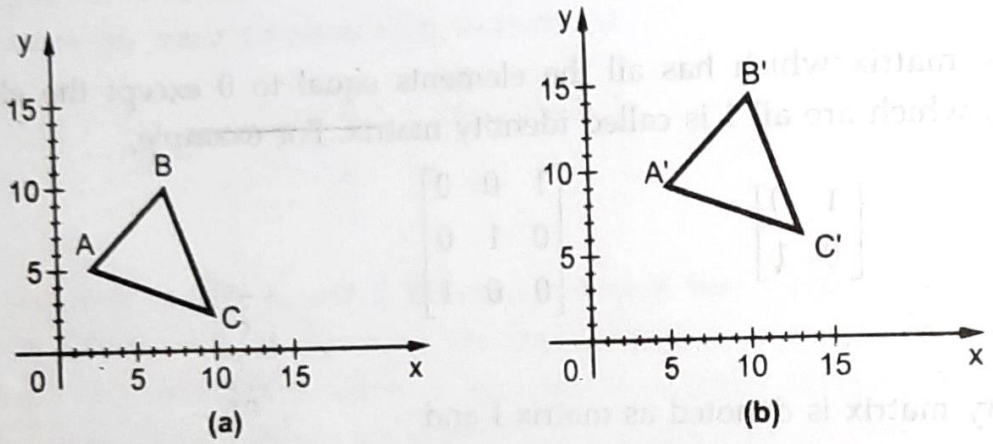


Fig. 4.2 Translation of polygon

$$A' = A + T$$

$$= \begin{bmatrix} 2 \\ 5 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$= \begin{bmatrix} 5 \\ 9 \end{bmatrix}$$

$$B' = B + T$$

$$= \begin{bmatrix} 7 \\ 10 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$= \begin{bmatrix} 10 \\ 14 \end{bmatrix}$$

$$C' = C + T$$

$$= \begin{bmatrix} 10 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$= \begin{bmatrix} 13 \\ 6 \end{bmatrix}$$

4.3.2 Rotation

A two dimensional rotation is applied to an object by repositioning it along a circular path in the xy plane. To generate a rotation, we specify a rotation angle θ and the position of the rotation point about which the object is to be rotated.

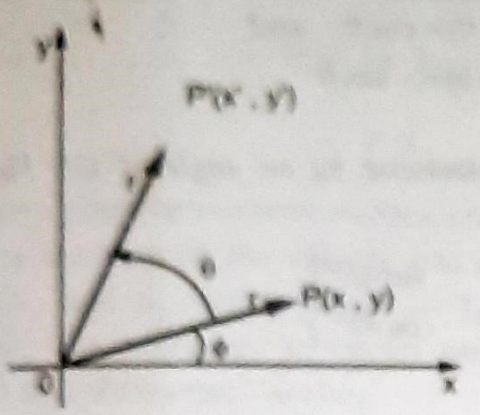


Fig. 4.3

Let us consider the rotation of the object about the origin, as shown in the Fig. 4.3.

Here, r is the constant distance of the point from the origin, angle ϕ is the original angular position of the point from the horizontal, and θ is the rotation angle. Using standard trigonometric equations, we can express the transformed coordinates in terms of angles θ and ϕ as

$$\left.\begin{aligned} x' &= r\cos(\phi+\theta) = r\cos\phi\cos\theta - r\sin\phi\sin\theta \\ y' &= r\sin(\phi+\theta) = r\cos\phi\sin\theta + r\sin\phi\cos\theta \end{aligned}\right\} \quad \ldots(4.4)$$

The original co-ordinates of the point in polar co-ordinates are given as

$$\left.\begin{aligned} x &= r\cos\phi \\ y &= r\sin\phi \end{aligned}\right\} \quad \ldots(4.5)$$

Substituting equations 4.5 into 4.4, we get the transformation equations for rotating a point (x, y) through an angle θ about the origin as :

$$\left.\begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \end{aligned}\right\} \quad \ldots(4.6)$$

The above equations can be represented in the matrix form as given below

$$[x' \;\; y'] = [x \;\; y]\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

$$\therefore \quad P' = P \cdot R \quad \ldots(4.7)$$

where R is rotation matrix and it is given as

$$R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \quad \ldots(4.8)$$

It is important to note that positive values for the rotation angle define counterclockwise rotations about the rotation point and negative values rotate objects in the clockwise sense.

For negative values of θ i.e., for clockwise rotation, the rotation matrix becomes

$$R = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad \because \cos(-\theta) = \cos\theta \text{ and } \sin(-\theta) = -\sin\theta \quad \ldots (4.9)$$

Example 4.2 : *A point (4, 3) is rotated counterclockwise by an angle of 45°. Find the rotation matrix and the resultant point.*

Solution :

$$R = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \cos 45^\circ & \sin 45^\circ \\ -\sin 45^\circ & \cos 45^\circ \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

$$\therefore \quad P' = [4 \;\; 3] \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

$$= \left[4/\sqrt{2} - 3/\sqrt{2} \quad 4/\sqrt{2} + 3/\sqrt{2}\right]$$

$$= \left[1/\sqrt{2} \quad 7/\sqrt{2}\right]$$

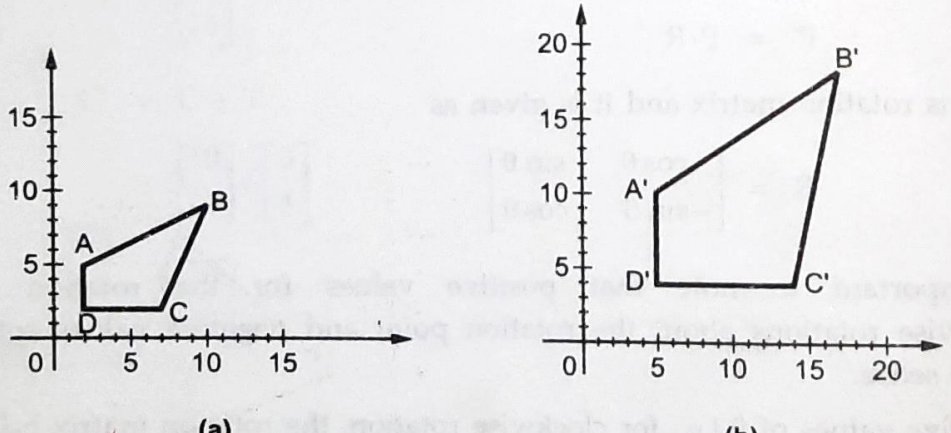
4.3.3 Scaling

A scaling transformation changes the size of an object. This operation can be carried out for polygons by multiplying the co-ordinate values (x, y) of each vertex by scaling factors S_x and S_y to produce the transformed co-ordinates (x′, y′).

$$x' = x \cdot S_x$$

and $$y' = y \cdot S_y \quad \ldots (4.10)$$

Scaling factor S_x scales object in the x direction and scaling factor S_y scales object in the y direction. The equations 4.10 can be written in the matrix form as given below :



(a) (b)

Fig. 4.4

$$[x' \;\; y'] = [x \;\; y] \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

$$= [x \cdot S_x \quad y \cdot S_y] \qquad \ldots (4.11)$$

$$= P \cdot S$$

Any positive numeric values are valid for scaling factors S_x and S_y. Values less than 1 reduce the size of the objects and values greater than 1 produce an enlarged object. For both S_x and S_y values equal to 1, the size of object does not change. To get uniform scaling it is necessary to assign same value for S_x and S_y. Unequal values for S_x and S_y result in a differential scaling.

Example 4.3 : *Scale the polygon with coordinates A (2, 5), B (7, 10) and C (10, 2) by two units in x direction and two units in y direction.*

Solution : Here $S_x = 2$ and $S_y = 2$. Therefore, transformation matrix is given as

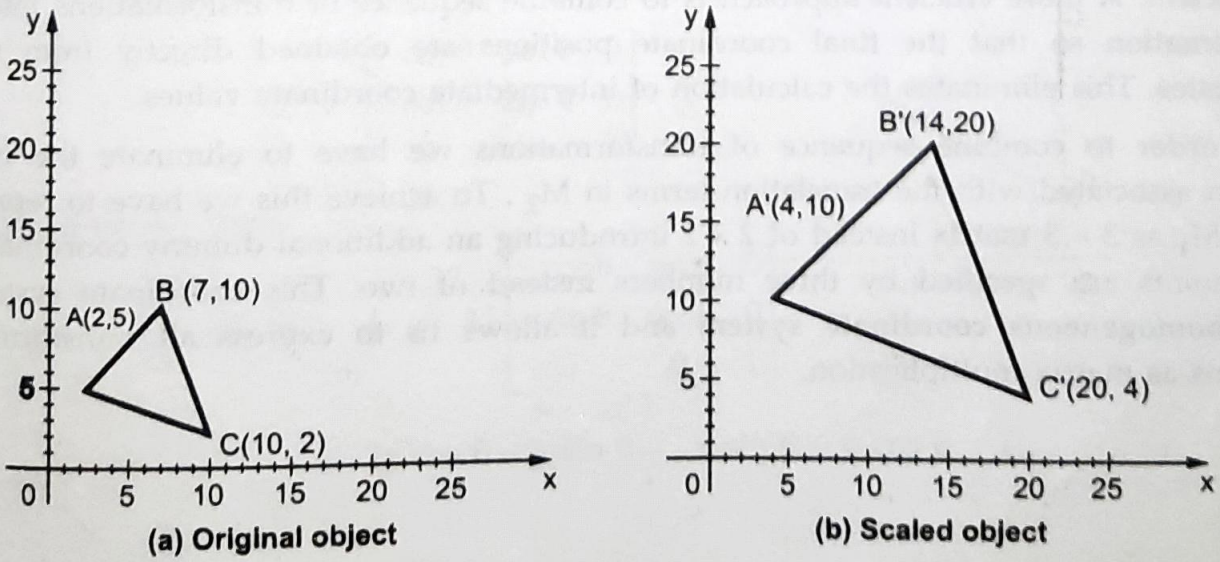
$$S = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

The object matrix is :

$$\begin{matrix} & x & y \\ A & 2 & 5 \\ B & 7 & 10 \\ C & 10 & 2 \end{matrix}$$

$$\therefore \quad \begin{matrix} A' \\ B' \\ C' \end{matrix} \begin{bmatrix} x'_1 & y'_1 \\ x'_2 & y'_2 \\ x'_3 & y'_3 \end{bmatrix} = \begin{bmatrix} 2 & 5 \\ 7 & 10 \\ 10 & 2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 10 \\ 14 & 20 \\ 20 & 4 \end{bmatrix}$$



(a) Original object (b) Scaled object

Fig. 4.5

4.6 Other Transformations

The three basic transformations of scaling, rotating, and translating are the most useful and most common. There are some other transformations which are useful in certain applications. Two such transformations are reflection and shear.

4.6.1 Reflection

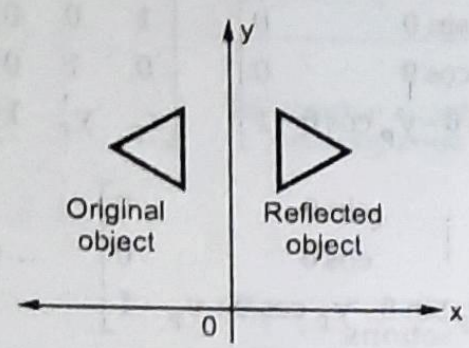
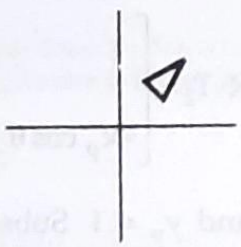
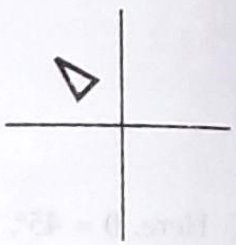
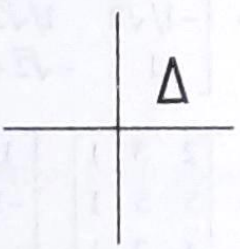
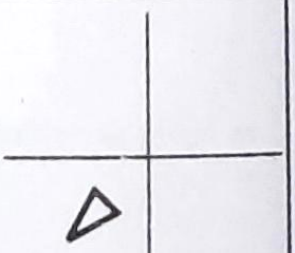


Fig. 4.7 Reflection about y axis

A reflection is a transformation that produces a mirror image of an object relative to an axis of reflection. We can choose an axis of reflection in the xy plane or perpendicular to the xy plane. The table 4.1 gives examples of some common reflections.

Reflection	Transformation matrix	Original image	Reflected image
Reflection about Y-axis	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$		
Reflection about X axis	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$		
Reflection about origin	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$		

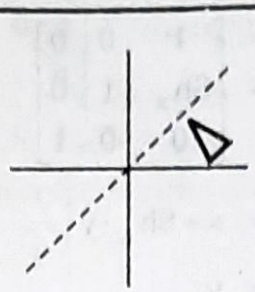
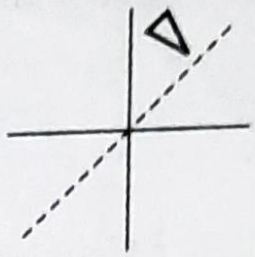
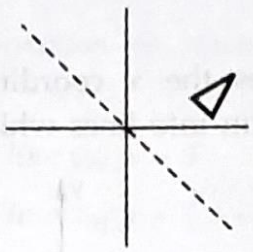
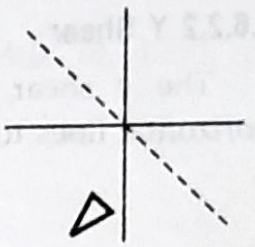
Reflection about line $y = x$	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$		
Reflection about line $y = -x$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$		

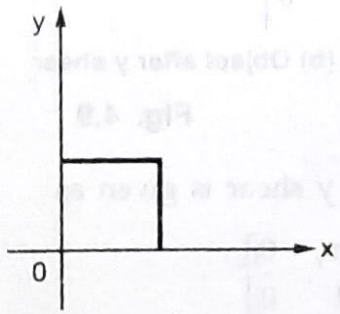
Table 4.1 Common reflections

4.6.2 Shear

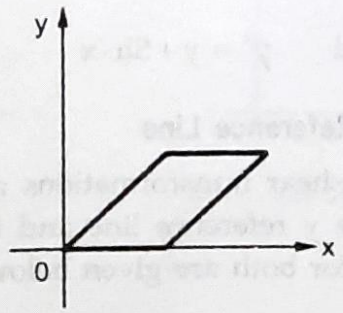
A transformation that slants the shape of an object is called the shear transformation. Two common shearing transformations are used. One shifts x coordinate values and other shifts y coordinate values. However, in both the cases only one coordinate (x or y) changes its coordinates and other preserves its values.

4.6.2.1 X Shear

The x shear preserves the y coordinates, but changes the x values which causes vertical lines to tilt right or left as shown in the Fig. 4.8. The transformation matrix for x shear is given as



(a) Original object



(b) Object after x shear

Fig. 4.8

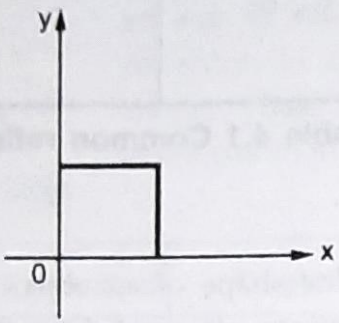
$$X_sh = \begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\therefore$ $x' = x + Sh_x \cdot y$ and

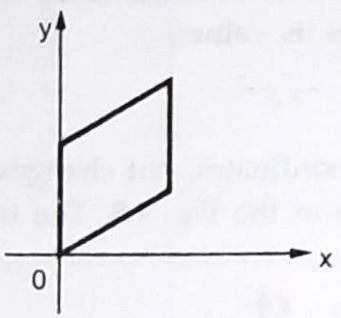
$y' = y$ ... (4.19)

4.6.2.2 Y Shear

The y shear preserves the x coordinates, but changes the y values which causes horizontal lines to transform into lines which slope up or down, as shown in the Fig. 4.9.



(a) Original object



(b) Object after y shear

Fig. 4.9

The transformation matrix for y shear is given as

$$Y_sh = \begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\therefore$ $x' = x$ and $y' = y + Sh \cdot x$ ... (4.20)

4.6.2.3 Shearing Relative to Other Reference Line

We can apply x shear and y shear transformations relative to other reference lines. In x shear transformation we can use y reference line and in y shear we can use x reference line. The transformation matrices for both are given below :

x shear with y reference line : $\begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ -Sh_x \cdot y_{ref} & 0 & 1 \end{bmatrix}$

y shear with x reference line : $\begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & -Sh_y \cdot x_{ref} & 0 \end{bmatrix}$

Example 4.8 : *Apply the shearing transformation to square with A(0, 0), B (1, 0), C (1, 1) and D (0, 1) as given below*

a) Shear parameter value of 0.5 relative to the line $y_{ref} = -1$

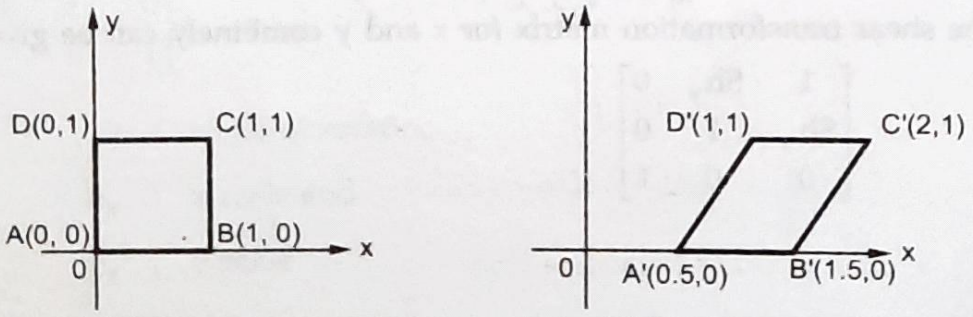
b) Shear parameter value of 0.5 relative to the line $x_{ref} = -1$

Solution : a) Here $Sh_x = 0.5$ and $y_{ref} = -1$

$$\therefore \quad \begin{bmatrix} A' \\ B' \\ C' \\ D' \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ -Sh_x \cdot y_{ref} & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0.5 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 & 0 & 1 \\ 1.5 & 0 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



(a) Original square (b) Sheared square

Fig. 4.10

b) Here $Sh_y = 0.5$ and $x_{ref} = -1$

$$\begin{bmatrix} A' \\ B' \\ C' \\ D' \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \begin{bmatrix} 1 & Sh_y & 0 \\ 0 & 1 & 0 \\ 0 & -Sh_y \cdot x_{ref} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0.5 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & 1 \\ 0 & 1.5 & 1 \end{bmatrix}$$

It is important to note that shearing operations can be expressed as sequence of basic transformations. The sequence of basic transformations involve series of rotation and scaling transformations.

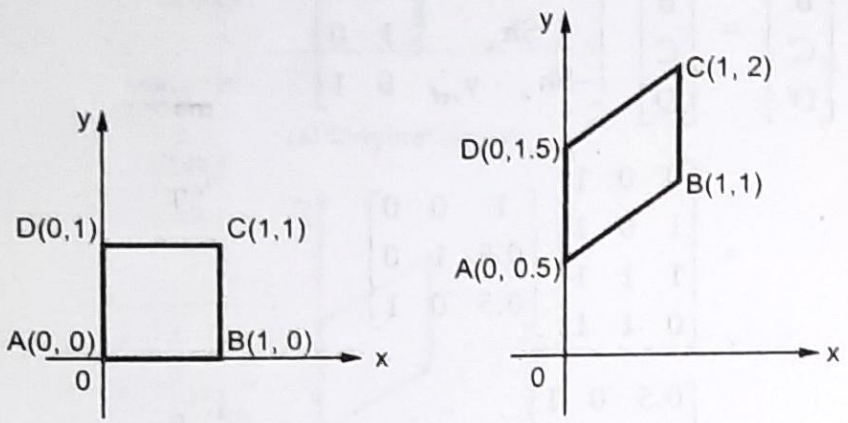


Fig. 4.11

Example 4.9 : *Show how shear transformation may be expressed in terms of rotation and scaling.*

Solution : The shear transformation matrix for x and y combinely can be given as

$$\begin{bmatrix} 1 & Sh_y & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

5 Segments

5.1 Introduction

Uptil now we have considered a single picture on the display. In practice, the image on the display screen is often composed of several pictures or items of information. An image may contain several views of an object and related information. It may also contain close up view of a particular component. For example, we may wish to display an internal plan of a living room. The plan may contain various objects such as sofa-set, T. V., show-case, teapoy, show-pieces, wall hangings and so on. Each object has a set of attributes such as size, colour and its position in the room. We might wish to see all these objects simultaneously or a single object at a time.

To view the entire image or a part of the image with various attributes we need to organize the image information in a particular manner. We know that the image information is stored in the display file. Existing structure of the display file does not satisfy our requirements of viewing the image. Hence the display structure is modified to reflect the subpicture structure. To achieve this display file is divided into **segments**. Each segment corresponds to a component or an object of the overall display and is associated with a set of attributes. Along with the attributes the segment is also associated with the image transformation parameters such as scalings along X and/or Y-direction, rotation and shearing.

Therefore, presence of segment allows :

- Subdivision of the picture.
- Visualization of a particular part of the picture.
- Scaling, rotation and translation of a particular part of the picture.

In this chapter, we will discuss the segmentation of display file along with the creation, renaming, closing and deletion of segments.

5.2 Segment Table

In the previous section, we have seen that the display file is divided into segments and each segment carries information associated with it. To access a particular segment and the information associated with it we must have a unique name assigned to each segment. Along with the name we must have its display file position and its attribute information. The structure used to organize all this information related to segments is called **segment table**.

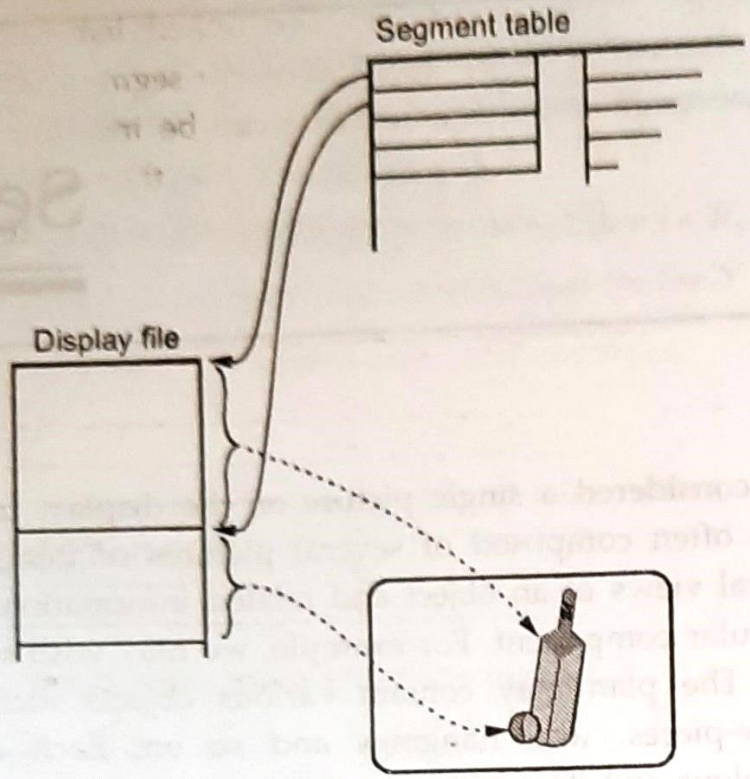


Fig. 5.1 Segment table and display file

It indicates the portions of display file used to construct the picture. The segment table is formed by using arrays. First array holds the display file starting location for that segment, the second array holds the segment size information, while the third indicates the visibility and so on. This is illustrated in Fig. 5.2.

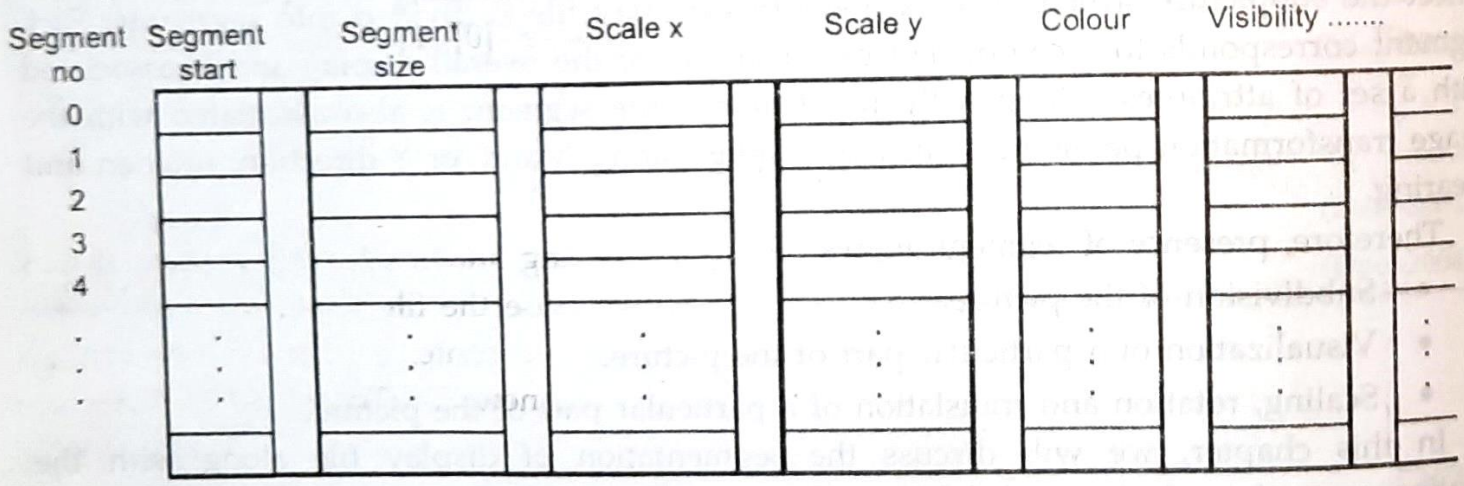


Fig. 5.2 Segment table

Each row in the segment table represents information of one segment including its name, position, size, attributes and the image information parameters. If we wish to make segment 4 visible then this is achieved by setting the corresponding entry in the array to 'ON'. The display file interpreter initially checks the start, size and visible attribute of the segments and it interprets only those segments which are to be made visible.

There are other possible schemes for implementing the segment table, many with the substantial advantages over array scheme. Here, we have chosen array scheme because it allows simple accessing, does not require any new data structure and its updating is straight forward.

An alternative approach is the linked list. The linked list uses the additional field called the link or pointer which gives the location of the segment in the segment table. In case of arrays, maximum number of segments that can be included in the segment table are equal to the length of arrays. But with linked list there is no such limit on the maximum number of segments; the growth of a linked list is dynamic. Fig. 5.3 shows the segment table using linked list.

Segment number	Segment start	Segment size	Scale x	Scale y	Colour	Visibility	Link
1							3
2							4
3							2
4							5
5							Null /

Fig. 5.3 Segment table using linked list

Further more, in case of hidden surface removal (Refer chapter 8 for detailed information) it is necessary to display different segments in specific order. Ordering/sorting of segments in the segment table is quite easy task with linked list structure. The ordering can be achieved by simply adjusting the links. However, in case of arrays we have to move actual segment information in the process of sorting the segments.

In the display file there are few instructions which did not specify any segment. Such instructions are placed in a special "unnamed" segment. Normally index 0 of the array is assigned to unnamed segment. Therefore, segment-size [0] indicates the number of instructions in the unnamed segment.

5.3 Functions for Segmenting the Display File

We are familiar with the common methods of creating and modifying sequential disk files. We open a file before we add data to it, and we close the file when we have added the last data item and the file is complete. To change the contents of a file, we open it again, add the new data to replace the old, and close the new file. To get rid of a file, we delete it.

The very same operations are used for manipulating display file segments. To create a new segment, we create it and then call graphic primitives to add to the segment the lines and text to be displayed, then we close the segment. To modify the contents we rename current segment with the modified segment and to remove segment from the display file, we delete it. Thus we have four basic functions:

- Create segment
- Close segment
- Delete segment
- Rename segment

5.3.1 Segment Creation

In a segment creation process, initially, we have to check whether some other segment is still open. It is not allowed to open two segments at a same time because it is then difficult to assign the drawing instructions to particular segment. Hence, segment must be created or opened when no other segment is currently open. We must give the segment a name so that we can identify it. While doing this it is important to check whether a given segment name is valid or not and whether there already exists a segment with the same name. If so, we have to assign other valid name to the segment. Once a valid segment name is assigned, we have to initialize the items in the segment table under our segment name to indicate that this is a fresh new segment. The first instruction of this segment will be located at the next free storage area in the display file. As we have not entered any instructions into the segment yet, its size is initialized with value zero. The attributes of the segment are initialized as a default attribute values.

Algorithm : Create Segment

1. Check whether any segment is open; if so display error message : "Segment is still open" and go to step 9.
2. Read the name of new segment.
3. Check whether the new segment name is valid; if not display error message : "Not a valid segment name"and go to step 9.
4. Check whether the new segment name is already existing in the same-name list; if so display error message : "Segment name already exists" and go to step 9.
5. Initialize the start of the segment at the next free storage area in the display file.
6. Initialize size of this segment equal to zero.
7. Initialize all attributes of segment to their default values.
8. Indicate that the new segment is now open.
9. Stop.

5.3.2 Closing a Segment

Once a segment is open we can enter the display file instructions in it. The entered commands are then associated with the open segment. After completion of entering all display file instructions, the segment must be closed. To close a segment it is necessary to change the name of the currently open segment. It can be achieved by changing the name of currently open segment as 0. Now the segment with name 0 is open i.e. unnamed segment is open. If there are two unnamed segments in the display file one has to be deleted.

Algorithm : Close Segment

1. Check whether any segment is open; if no, display error message : " No segment open" and go to step 6.
2. Change the name of currently open segment, say 0, i.e., unnamed segment.
3. Delete any unnamed segment instructions which may have been saved, i.e. initialize the unnamed segment with no instructions.

4. Initialize the start of the unnamed segment at the next free storage area in the display file.
5. Initialize size of unnamed segment equal to 0.
6. Stop.

5.3.3 Deleting a Segment

When we want to delete a particular segment from the display file then we must recover the storage space occupied by its instructions and make this space free for some other segment. To do this we must not destroy and re-form the entire display file, but we must delete just one segment, while preserving the rest of the display file. The method to achieve this depends upon the data structure used to represent the display file. Here, we have used arrays to store the display file information. Use of arrays makes the recovery of storage space occupied by the segment very easy and straight forward. However using arrays is not as efficient as some other storage techniques such as link list. In case of arrays the gap left by deleted block is filled by shifting up all segments which are following the deleted segment. This is illustrated in Fig. 5.4.

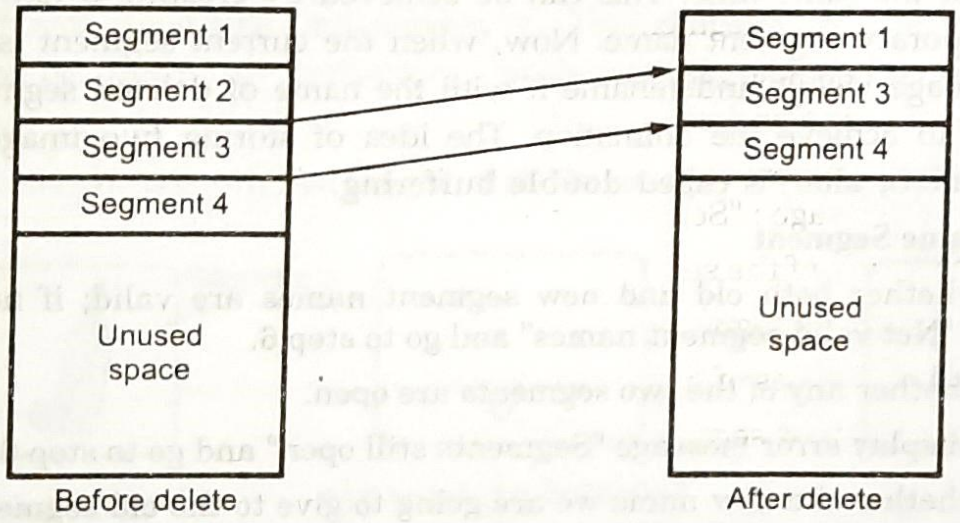


Fig. 5.4 Display file contents before and after deleting segment 2

Algorithm : Delete Segment

1. Read the name of segment which is to be deleted.
2. Check whether that segment name is valid; if not, display error message "Segment not valid" and go to step 8.
3. Check whether the segment is open, if yes, display error message "Can't delete the open segment" and go to step 8.
4. Check whether the size of segment is greater than 0; if no, no processing is required, as segment contains no instructions. Therefore, go to step 8.
5. Shift the display file elements which follow the segment which is to be deleted by its size.
6. Recover the deleted space by resetting the index of the next free instruction.
7. Adjust the starting positions of the shifted segments by subtracting the size of the deleted segment from it.
8. Stop.

When we want to start a completely new picture, we have to delete all of the segments. One way of deleting all segments is to delete them individually. This approach is independent of the data structure used for the display file. A more efficient approach for our array data structure is to simply set the size value of all segments to 0 and initialize the next free space as the first location in the display file.

5.3.4 Renaming a Segment

We can display animated character on the display by presenting sequence of images, each with a slightly different drawing of the character. Let us assume that we have a segment in a display file for animated character. Then to display a new image in the sequence we have to delete the current segment and re-create it with the altered character. The problem in this process is that during the time after the first image is deleted and time before the second image is completely entered, only a partially completed character is displayed on the screen. We can avoid this problem by keeping the next image ready in the display file before deleting the current segment. This means that both segments, the segment which is to be deleted and the segment which is to be replaced with must exist in the display file at the same time. This can be achieved by creating a new invisible image under some temporary segment name. Now, when the current segment is deleted we can make the new image visible and rename it with the name of deleted segment. These steps can be repeated to achieve the animation. The idea of storing two images, one to show and other to create or alter, is called **double buffering**.

Algorithm : Rename Segment

1. Check whether both old and new segment names are valid; if not, display error message "Not valid segment names" and go to step 6.
2. Check whether any of the two segments are open.

   If open, display error message "Segments still open" and go to step 6.
3. Check whether the new name we are going to give to the old segment is not already existing in the display file. If yes, display error message "Segment already exists" and go to step 6.
4. Copy the old segment table entry into the new position.
5. Delete the old segment.
6. Stop.

5.4 More about Segments

We know that whenever we want to display any picture or image on the screen, we have to create segments for various parts of the picture, and we have to set attributes for them. By setting the visible attribute of the segment we can decide whether to display it or not. When visible attribute of a segment is 1 it is included in the active segment list and display file interpreter processes it. Such a segment is called **posted segment**. When visible attribute of a segment is 0 it is not included in the active segment list and display file interpreter skips this segment. Such a segment is called **unposted segment**. Thus visible attribute of a segment decides the posting or unposting of a segment.

Advantages of using segmented display files

1. Segmentation allows to organize display files in subpicture structure.
2. It allows to apply different set of attributes to the different portions of the image.
3. Due to segmentation selective portion of the image can be displayed.
4. Segmentation makes it easier to modify the picture by changing segment attributes or by replacing segments.
5. Segmentation allows application of transformation on selective portion of the image.

5.5 Image Transformation

One of the major advantage of computer graphics is that we can display pictures with certain alterations if required. For example, we can see image from a different angle, we can zoom particular part of the image, we can change the size of the image, we can change the position of particular part of the image. These changes are easy to perform because the graphics image information is stored in different segments and we can apply changes to different segments independently. The changes in the original graphics information can be done by performing mathematical operations which are called **transformations**.

The Fig. 5.5 shows the original image and the images after applying some form of transformation.

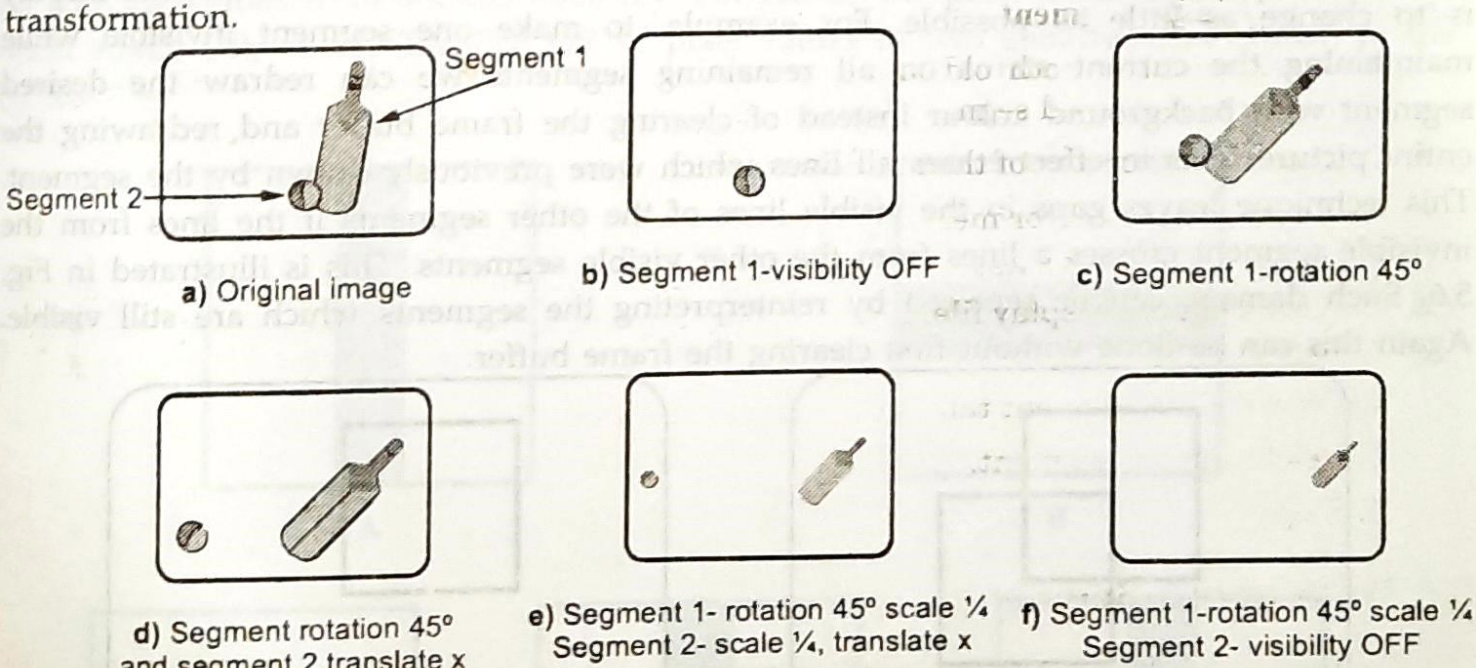


Fig. 5.5

Algorithm : Setting Image Transformation

1. Read the name of segment, say s_name.
2. Check whether the name is valid or not. If not display error message " Not a valid segment " and go to step 8.
3. Read translation factor, say t_x and t_y.

achieved a naming mechanism, we then furnish him with subroutines or functions that effect the desired modifications on the appropriate parts. The display file, provided it resides in addressable memory, possesses the two fundamental attributes we require: we can assign names or labels to sets of instructions, and these instructions are easily modified.

How should we choose our sets of display instructions for naming? One approach is to name each graphic primitive that we add to the display file. There is no reason, however, why these primitive entities should be a particularly appropriate choice; in almost all cases the unit of modification consists of several if not many such entities. So great, indeed, are the potential classes of pictures and of dynamic modifications that we cannot expect a single unit size to suffice. Instead we must let the programmer determine his own units.

In this chapter we shall discuss a simple approach to display-file manipulation. It is by no means the only such scheme, as we shall see in later chapters, but it is a widely accepted approach that is particularly appropriate to the concepts developed in the previous chapter. We shall use the term *segment* for the unit of picture that is named for modification purposes. The segment is a unit of the *display file*; thus the use of segments by the programmer requires him also to recognize the existence of the display file as a stored representation of the displayed image.

The segment is a *logical* unit, not necessarily contiguous either in the display file or on the screen. It is simply a collection of display-file instructions representing graphic primitives that we can manipulate as a single unit. We therefore need functions to perform these manipulations, and a naming scheme so that we can refer to each segment unambiguously.

7-2 FUNCTIONS FOR SEGMENTING THE DISPLAY FILE

In Chapter 6 we raised the notion that primitive graphic functions draw lines and text directly on the screen. We now modify that notion slightly by saying that they deposit display commands in a display-file segment representing the same lines and text. This slight modification to our model of the display process provides the basis for a set of functions for display-file manipulation.

Many readers will be familiar with the common methods of creating and modifying sequential disk files. We *open* a file before we add data to it, and we *close* the file when we have added the last data item and the file is complete. To change the contents of a file, we open it again, add the new data to replace the old, and close the new file. To get rid of a file, we *delete* it.

The very same operations are ideal for manipulating display-file segments. To create a new segment, we open it and then call graphic primitives to add to the segment the lines and text to be displayed; then we close the segment. The same sequence of operations applied to an existing segment will cause that

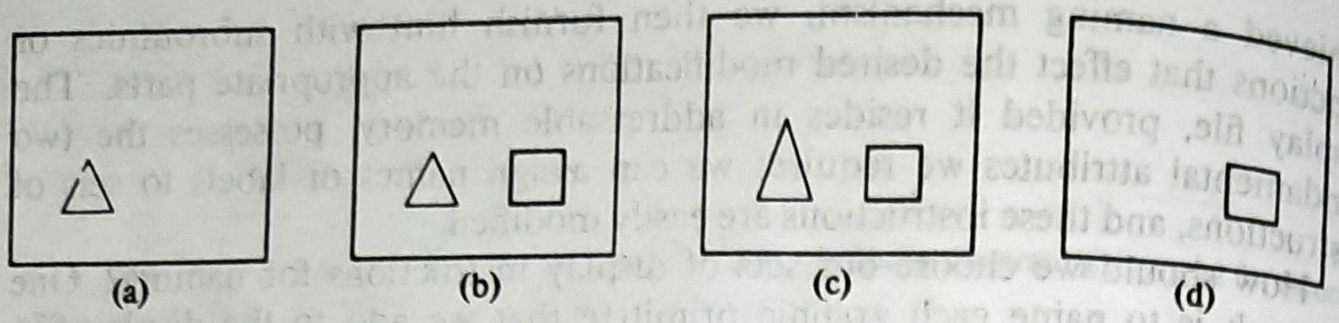


Figure 7-2 Manipulation of triangle and square as separate segments.

segment to be replaced by a new one. To remove a segment from the display file, we delete it. Thus we need only three basic functions:

OpenSegment (*n*)	open a display file segment named *n*
CloseSegment	close the open segment
DeleteSegment (*n*)	remove from the display file the segment named *n*

To illustrate the use of these functions, suppose we wish to display a triangle, as shown in Figure 7-2a. We may use the following statements:

```
InitGraphics;
SetWindow(0, 0, 500, 500);
OpenSegment(t);
MoveTo(100, 100);
LineTo(150, 200); LineTo(200, 100); LineTo(100, 100);
CloseSegment;
```

The display file, if originally empty, now contains the single segment *t* (Figure 7-3a). We can add a square, as shown in Figure 7-2b, as follows:

```
OpenSegment(s);
MoveTo(300, 100);
LineTo(300, 200); LineTo(400, 200); LineTo(400, 100); LineTo(300, 100);
CloseSegment;
```

Now the display file contains the two segments *s* and *t*, as in Figure 7-3b. We can redefine the triangle as follows, to achieve Figure 7-2c:

```
OpenSegment(t);
MoveTo(100, 100);
LineTo(150, 250); LineTo(200, 100); LineTo(100, 100);
CloseSegment;
```

The original segment *t* has been replaced by a new one (Figure 7-3c). Note that we would like the original triangle to remain on the screen until the definition of the new one is complete. We shall encounter this requirement again in Chapter 8 when we discuss double buffering.

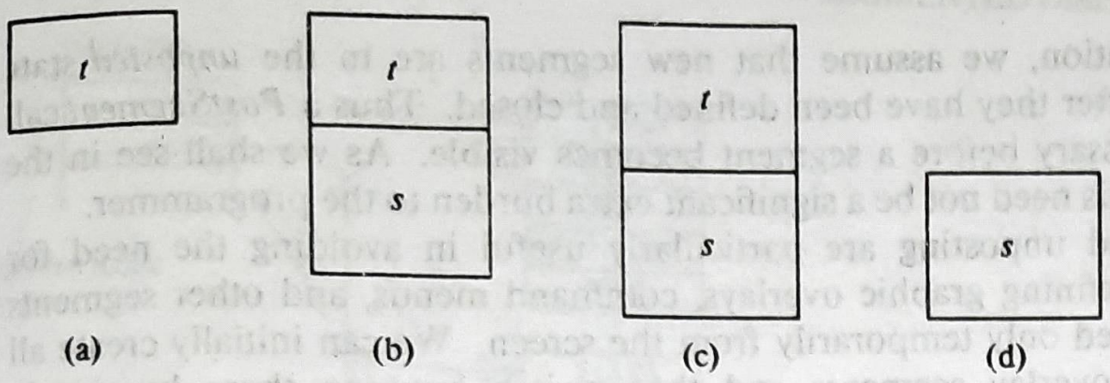


Figure 7-3 Display file segments corresponding to Figure 7-2.

Finally we can erase the triangle (Figure 7-2d):

DeleteSegment (*t*)

The display file now contains only the single segment *s*, as shown in Figure 7-3d.

This is a minimal function set for manipulating segmented display files. The design of the functions shows the influence of the standards of simplicity, completeness, and robustness raised in Chapter 6. Thus there are no more functions than necessary, just enough to do what we want. Two different *OpenSegment* functions could have been provided, one for creating segments and one for replacing existing ones, but this would be capable of misuse — the programmer could try to open a new segment with the same name as an old one.* We could also have required an argument to be passed to *CloseSegment*, to indicate the name of the segment to be closed. This is unnecessary, however, since we can almost always construct display files without opening more than one at a time; thus *CloseSegment* closes the currently open segment and needs no argument.

7-3 POSTING AND UNPOSTING A SEGMENT

We can extend the usefulness of our graphics package considerably by allowing segments to become temporarily invisible. For this we need two additional functions. *Posting* is the action of including a segment in the display refresh cycle; thus posting makes a segment visible. *Unposting* removes the segment from the refresh cycle, so that it is no longer visible. The segment is not destroyed by unposting, and so it can be rendered visible again without redefining it, simply by posting it. The two functions to achieve this effect are

PostSegment (*n*)	add segment *n* to the refresh cycle
UnpostSegment (*n*)	remove segment *n* from the refresh cycle

*File systems generally provide two separate functions here, partly to safeguard against accidentally overwriting a file. Such precautions are not necessary with display-file segments.

By convention, we assume that new segments are in the *unposted* state immediately after they have been defined and closed. Thus a *PostSegment* call is always necessary before a segment becomes visible. As we shall see in the next section, this need not be a significant extra burden to the programmer.

Posting and unposting are particularly useful in avoiding the need for repeatedly redefining graphic overlays, command menus, and other segments that are removed only temporarily from the screen. We can initially create all the menu or overlay segments and then switch between them by simply unposting the old segment and posting the new one.

7-4 SEGMENT NAMING SCHEMES

The most common scheme for naming segments is to provide each segment with a *unique integer name.* This is a departure from normal file-system practice, where text strings are more popular as names. Segment names are used only within the program, however, so they are essentially *internal names* unseen by the user of the program. Integers are not only more compact and more easily manipulated than strings but often more convenient as a means of relating segments of the display file to the items of data they represent.

Suppose, for example, we wish to use a graphical display in an air-traffic control center. We might present the air-traffic controller with a display like that shown in Figure 7-4, in which small symbols represent individual aircraft. From signals received in real time, a computer program maintains a data structure containing the identity, airspeed, heading in degrees, position, and altitude of each aircraft; Figure 7-5 shows a suitable list structure. Each aircraft symbol, together with the text labels showing identity, airspeed, and altitude, may be displayed as a single segment, greatly speeding up the task of modifying the picture. An appropriate name for each segment is then the index, in the main list, of the pointer to the vector defining the aircraft in the data structure. In this way we can easily determine the name of the segment representing any

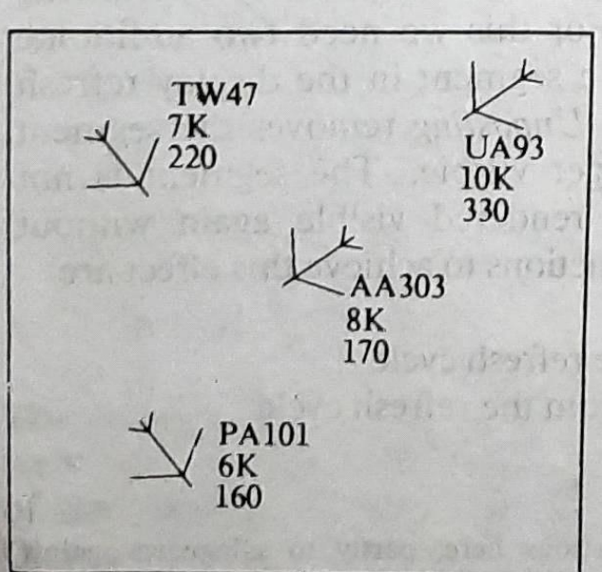


Figure 7-4 An air traffic control display.

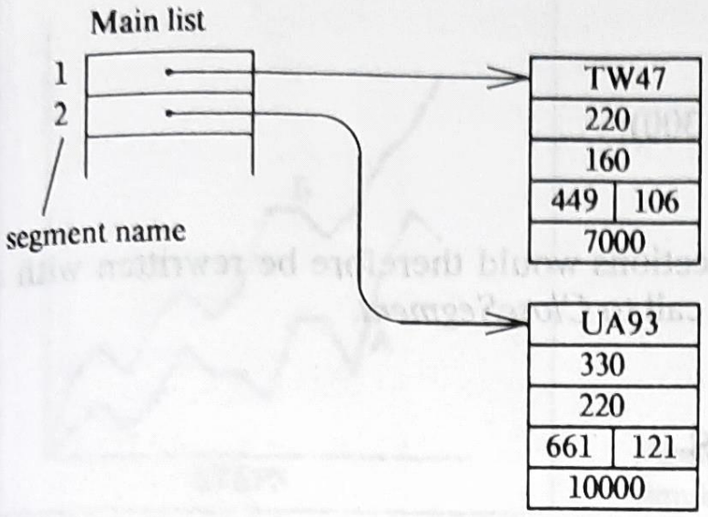


Figure 7-5 List-structure representation of air traffic control information.

particular aircraft in the data structure without using space in the data structure to define it.

7-5 DEFAULT ERROR CONDITIONS

Quite clearly, the programmer can make any one of a number of programming mistakes with the five functions *OpenSegment, CloseSegment, DeleteSegment, PostSegment,* and *UnpostSegment,* by issuing function calls in peculiar orders. For example, he can forget to close a segment, and then try to open another; or he can attempt to delete a nonexistent segment.

There are several solutions to this class of problem. It would obviously be quite simple and satisfactory to generate an error message on each such occasion. Error messages are often superfluous during display-file construction, however. Just the visible evidence on the screen is usually enough to indicate that the display file has been constructed incorrectly. Consequently there is no need to do any more than perform the following default actions:

OpenSegment (*n*)	any currently open segment is closed
CloseSegment	if no segment is currently open, no effect
DeleteSegment (*n*)	any currently open segment is closed; if segment *n* does not exist, no further effect
PostSegment (*n*)	any currently open segment is closed; if segment *n* is nonexistent or already posted, no effect
UnpostSegment (*n*)	any currently open segment is closed; if segment *n* is nonexistent or already unposted, no effect

Not only do these default actions take care of the more common mistakes of omission, but they simplify programming too. It is no longer necessary both to close and to post a segment in order to make it visible; posting alone will suffice. Normal practice, therefore, is to use the following sequence to create a new, visible segment:

```
OpenSegment(s);
MoveTo(200, 200);
LineTo(200, 300); LineTo(300, 300); . . .
PostSegment(s);
```

The examples given in the earlier sections would therefore be rewritten with a call to *PostSegment* in place of each call to *CloseSegment*.

7-6 APPENDING TO SEGMENTS

It is sometimes inconvenient that a display-file segment, once it has been defined, cannot be incrementally modified but must always be completely rebuilt. Consider, for example, a program to display the results of two concurrent simulations in the form of two graphs plotted on the same axes, as shown in Figure 7-6. Every 2 minutes each simulation generates a fresh data point, and the graphs are extended.

One method of writing such a program would be to create a new segment for each addition made to either graph. This would be extremely inefficient, however, since a complete simulation might involve several hundred data points. On the other hand, any alternate approach, such as the use of a single segment for each graph, could involve reconstructing the entire segment after every step.

In situations like this, a function for appending to segments is useful:

AppendToSegment(*n*)	open segment *n* for additions, leaving its present contents intact

Thus to add a line to a graph represented by segment *g* and leave the segment posted, we might write the following statements:

```
AppendToSegment(g);
MoveTo(lastx, lasty);
LineTo(newx, newy);
PostSegment(g);
```

The *AppendToSegment* function, although often useful in writing application programs, has some troublesome side effects. In view of these problems, which are discussed in the next chapter, the *AppendToSegment* function may involve more trouble than it is worth. In a few cases, like the example given, the lack of such a function will be felt by the application programmer; it is usually possible, however, to solve the problem by careful use of segments. Thus in the simulation example we can use a separate segment for each sequence of *n* data points, where *n* is in the range 20 to 50. We use a separate segment to represent each recent addition to either graph and redraw

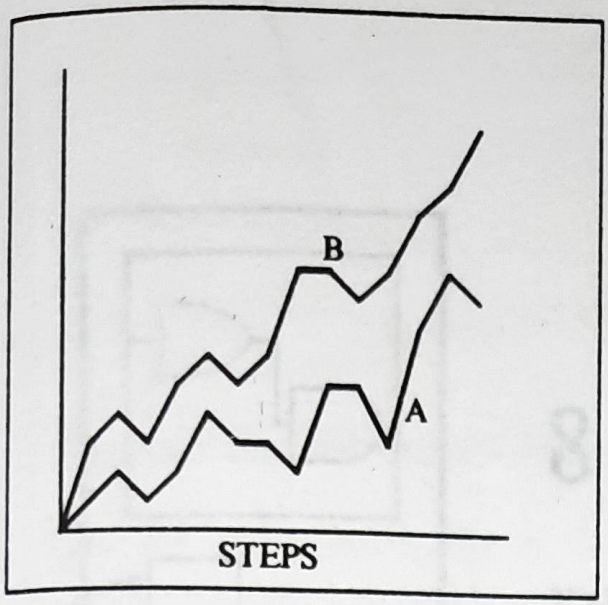


Figure 7-6 Plotting the results of two concurrent simulations.

these additions as a single segment each time the total number of the recent segments reaches *n*. Techniques of this kind are an indication of the complexity that can occasionally be avoided with the provision of an *AppendToSegment* function.

EXERCISES

7-1 What should the default error action be if the application program should call primitive functions without first opening a segment?

7-2 Suppose we wished to be able to manipulate individual display primitives in the same manner as segments: what changes would be necessary to the graphics functions described in Chapters 6 and 7?

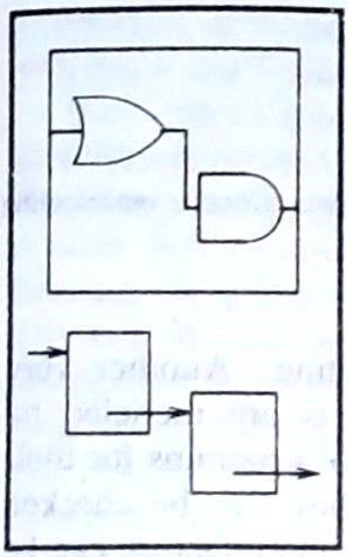
7-3 What would be the advantages and disadvantages of permitting more than one segment to be open at a time?

7-4 Although it is almost universal to use integers as segment names, it is feasible to use other types of data, such as strings and floating-point numbers. Can you think of reasons for using such names?

7-5 Rather than divide the displayed image logically into segments, we could divide it physically, e.g., into small squares. What effect would this have on interactive programming?

7-6 As we have seen, it is possible to provide two functions for opening segments, one to create a new segment, the other to redefine an existing segment. Does such a scheme have advantages? Illustrate your answer with some short program sequences.

7-7 Some graphics packages implement posting and unposting as manipulation of an *attribute* of the segment, i.e. changing its visibility. Can you think of other useful segment attributes?



9

GEOMETRIC MODELS

One of the most popular and effective uses of the digital computer is in the abstract representation of physical systems. The general term we use for this type of application is *modeling*. Computer models can be built of both physical objects and physical processes; some examples include:

Electric circuits. Power companies and authorities often model their power networks to predict loading and performance; designers of electronic circuits use computer models to help them in their design work.

Buildings. Computer models of buildings permit architects to test certain aspects of their designs and to generate plan and perspective views.

National and world economies. It is possible to develop programs that model the economic systems of the world in order to test the repercussions of currency devaluations and other financial strategies.

Weather. Some of the world's largest computers are involved in modeling weather conditions and helping forecast weather changes.

In several of these examples the computer model is used as a basis for *simulation*, i.e., to predict the behavior of the physical system under certain conditions. Simulation is a very important field of computing; a number of programming languages have been designed to help simulation by making it easier to build models, set up conditions, and observe the results. Nevertheless

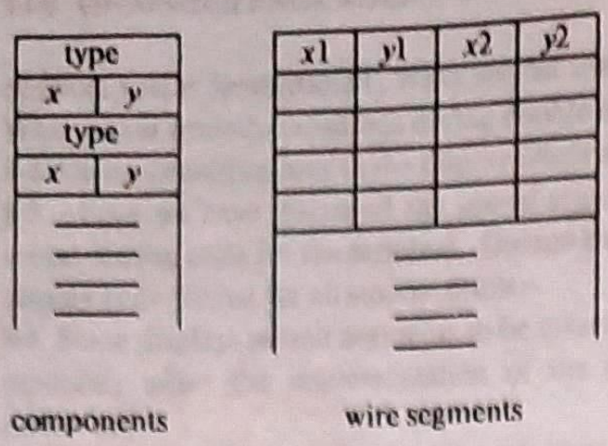


Figure 9-1 A simple data structure representing a circuit model.

simulation represents only one use of computer modeling. Another very important use is in computer-aided design: solid objects are modeled by computer so that their designers can generate machine-tool programs for their manufacture; integrated circuits are modeled so that they can be checked automatically against design rules and the masks for their manufacture can be generated by computer.

Computer graphics has long been associated with modeling. One obvious reason is that computer graphics offers a very convenient way to observe the behavior of computer models. When we simulate weather patterns, for example, we can plot pressure and temperature values on a displayed map, instead of printing them in tabular form; the resulting display is much more easily understood by the user of the model. We can also make use of interactive graphics in design applications, where the construction of the computer model represents a large part of the design process.

This chapter and Chapter 10 discuss ways in which computer graphics can be used in conjunction with modeling. The emphasis of these two chapters is on the generation of images from computer models. The actual techniques for building and manipulating models are not discussed in detail, except in the case of strictly graphical modeling techniques; modeling as a whole is too broad a subject to be covered in a few pages.

9-1 A SIMPLE MODELING EXAMPLE

In this section we shall use a simple example to present a few of the basic concepts of modeling and to discuss ways of displaying modeled information. Other examples of modeling will be used in a similar fashion throughout this chapter and Chapter 10.

Our example concerns modeling a digital electronic circuit. There are several reasons for modeling such circuits in a computer: we may wish to design them, to produce documentary diagrams of them, or to teach students about them. For each application we might construct a different type of model. Since we are not concerned here with the precise purpose of the model, we shall adopt a relatively simple and general one.

Many computer models consist of a mixture of data structures and procedures. In some cases, such as models of solid objects, we find a lot of data and very few procedures. In models of processes, such as weather systems and economics, it is the other way around. Very rarely do we find a model that is entirely in the form of a data structure or entirely procedural.

Our circuit model is no exception to the rule: it consists of a simple data structure supplemented by some procedures. The data structure defines each of the components and connecting wires in the circuit diagram. It does not contain any *relational* information, i.e., information describing relationships between these items, other than identification of the type of each component. This can be seen in the layout of the data structure in Figure 9-1, where each component is defined by an element of data containing its type and coordinate position on the diagram and each line segment of a wire is defined by the coordinates of its two endpoints. Relational information, such as the connections between components, is determined by procedures. Thus we can write a procedure to determine whether a wire and a component are connected by comparing coordinates; we can write another procedure to apply a similar connectivity test to any two wires; then by applying these two procedures iteratively to the contents of the data structure we can determine which components are connected together.

We can apply various processes to this basic model. For example, we can test for excessive *fan-out*, i.e., too many components with inputs connected to the output of a single component. This test will involve first determining all the wires leading from a given component's output connector, then counting how many of these are inputs to other components, and finally comparing this number with the manufacturer's recommended fan-out limit for the component. We can probably think of many other processes that would similarly help designers or students using the model. The most obvious and generally useful of these is a process to display the circuit diagram.

The representation we have chosen for our model makes it very easy to construct the circuit diagram on a display screen. The coordinate system in which component and wire endpoint positions are defined becomes our world

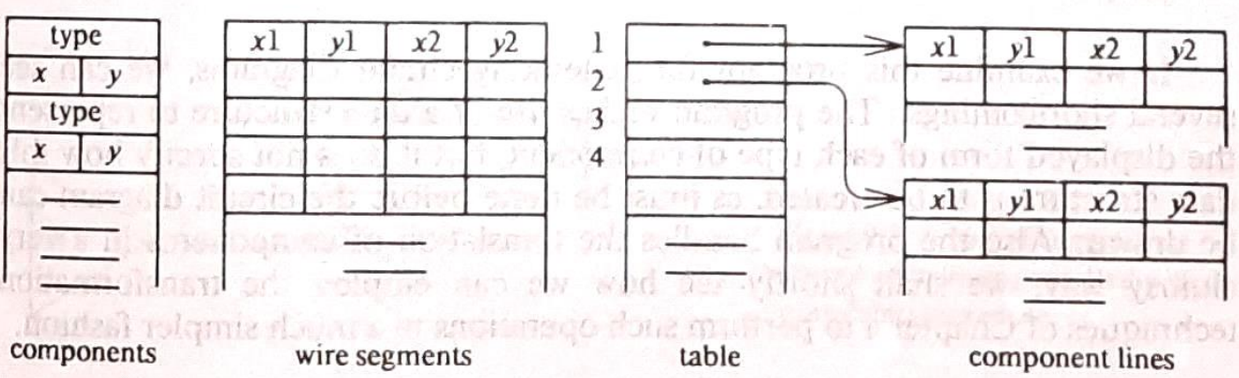


Figure 9-2 Inclusion of component definitions in the model; the component type is used to index into the table containing pointers to the component definitions.

coordinate system. Line segments are then generated using the stored endpoints as parameters. The display of the components is slightly more complicated: we must define how each component is to be displayed on the screen. Figure 9-2 shows these definitions included in the model, by means of sets of lines to represent each type of component, and a table for use in addressing the appropriate set of lines. This approach is in keeping with the use of data structures, rather than procedures, to represent the geometry of connection lines. The following pair of procedures will scan the list of lines and components and display each at its appropriate position:

```
procedure ShowWires;
    var i: integer;
begin
    for i := 1 to nWires do begin
        MoveTo(Wires[i].x1, Wires[i].y1);
        LineTo(Wires[i].x2, Wires[i].y2)
    end;
end;

procedure ShowComponents;
    var i,j: integer; xt,yt: real; c: ↑component;
begin
    for i := 1 to nComponents do begin
        xt := Components[i].x;
        yt := Components[i].y;
        c := ComponentTable[Components[i].comptype];
        with c↑ do begin
            for j := 1 to nLines do begin
                MoveTo(Lines[j].x1 + xt, Lines[j].y1 + yt);
                LineTo(Lines[j].x2 + xt, Lines[j].y2 + yt)
            end
        end
    end
end;
```

If we examine this program for generating circuit diagrams, we can see several shortcomings. The program makes use of a data structure to represent the displayed form of each type of component, but it does not specify how this data structure is to be created, as must be done before the circuit diagram can be drawn. Also the program handles the translation of components in a very clumsy way; we shall shortly see how we can employ the transformation techniques of Chapter 4 to perform such operations in a much simpler fashion.

9-2 GEOMETRIC MODELING

Our digital-circuit model has the property of suggesting a specific geometric form for the circuit diagram. The line segments representing each component and each connection in the circuit are fully defined by the coordinates stored in the data structure. Models which have this property are called *geometric models;* they include a large proportion of the models we construct of physical objects. Nongeometric modeling is usually applied to physical processes, rather than objects; an economic model, for example, would not normally have any specific geometric form.

The conventional approach to geometric modeling is to place the geometry entirely in the data structure, as exemplified by the circuit model. It is not essential to follow this approach. We could, for example, store just the connectivity relationships of the circuit in the data structure and use a procedure to compute the geometry of the connecting lines. Thus if the data structure indicates that the output of component *A* is to be connected to the input of component *B* in Figure 9-3, we could use a procedure to determine the coordinates of a sequence of lines connecting the two components; one possible sequence, shown in the figure, would be generated by the following procedure:

```
procedure LinkPoints(x1, y1, x2, y2: integer);
begin
    MoveTo(x1, y1);
    LineTo((x1 + x2)/2, y1);
    LineTo((x1 + x2)/2, y2);
    LineTo(x2, y2)
end;
```

Taken to an extreme, this approach would allow us to omit all geometric data from the structure, since we could allocate coordinate positions to the components by means of a procedure. The next chapter discusses this kind of procedural approach in greater detail; in this chapter we consider the more conventional type of geometric model, whose geometry resides entirely in the data structure.

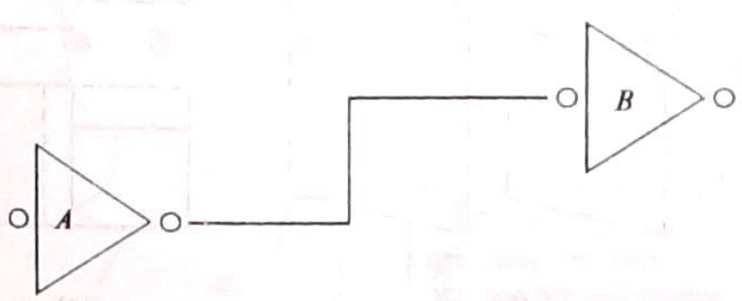


Figure 9-3 A pair of components connected by lines generated with the *LinkPoints* procedure.

Displaying a Geometric Model

A model that defines the geometric form of an object or set of objects does not necessarily define how those objects should be displayed. In Figure 9-4, for example, we see three different displayed representations of the same three-dimensional object. The choice of method we use to display modeled objects depends partly on the purpose of displaying them and partly on the hardware and software available for their display. A great many techniques, some of them to be discussed in Part Five, have been developed to make the display of geometric models more effective. At the same time, certain basic methods exist that apply to the display of nearly all models. These methods are based on the existence of certain common features in nearly all geometric models:

Basic elements of data that typically appear as individual graphic entities on the screen. Examples include the lines and components in the circuit model and the edges or faces of the solid object shown in Figure 9-4. Sometimes the graphical representation of these entities is defined by means of additional data in the model, as in the component definitions of Figure 9-2; sometimes it is left undefined, as in Figure 9-1, in which case the display-generation process must supply it.

Structure imposing certain relationships between the basic data elements. Many different kinds of relationships can be expressed in models, using lists, arrays, sets, rings, and other kinds of structuring mechanisms. Thus we can model collections of similar objects by grouping elements together in lists, we can establish hierarchies by means of tree structures of elements, and we can attach property lists to elements to define their attributes. These are only a few of the wide variety of structuring methods used in computer models. In displaying modeled objects we are concerned principally with the task of traversing such structures in an efficient manner so as to display each of the elements in the structure.

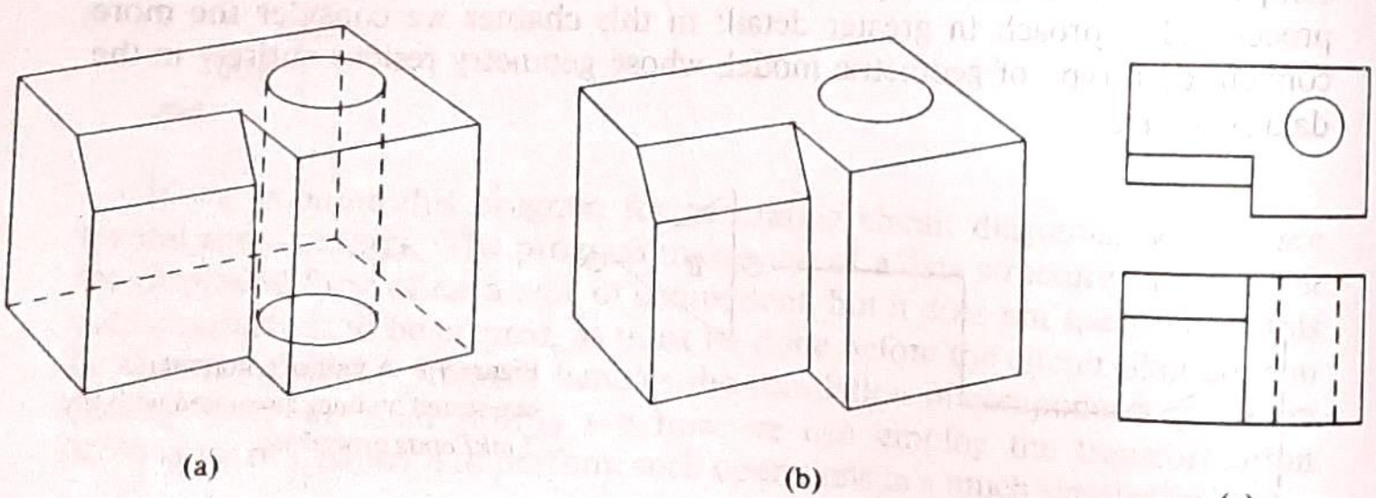


Figure 9-4 Three different views of the same solid object: (a) a perspective view with hidden lines shown dashed; (b) the same view with hidden lines removed; (c) a pair of orthogonal projections.

Transformations that define the position and orientation of each part of the modeled object. In a model that includes many instances of the same component a transformation is applied to each one to place it appropriately in the modeling space. This can be seen in the circuit model of Figure 9-2, which includes translation factors to be applied to each instance of a component. Even when there is no repetitive use of the same component, we often use transformations to position the different parts of the modeled object.

Our earlier example of the digital circuit model exhibits all three of these features, and it also brings to light certain deficiencies in our ability to display such a model. Specifically it shows that while we possess adequate methods for traversing the structure of such models, we need more convenient methods for defining the graphical representation of each element of the model and for applying transformations to them. The next section shows how to add these capabilities to a graphics package.

9-3 SYMBOLS AND INSTANCES

The use of graphical symbols is very common in charts and graphs like the one shown in Figure 9-5. Here we see in the key of the chart a definition of each symbol, and multiple *instances* of each symbol in the chart itself. When we draw charts on a display, we must be able to define symbols so that we can include instances of each symbol in the displayed chart. The same capability is very useful in displaying plans of buildings, electric-circuit diagrams, and drawings of mechanical objects.

The way we define a symbol need not be very different from the way we define other images for display. There is a difference, however, in the choice of coordinate system. When we define a symbol, we do not know where it will be placed in the world coordinate system; indeed the whole point of the symbol is

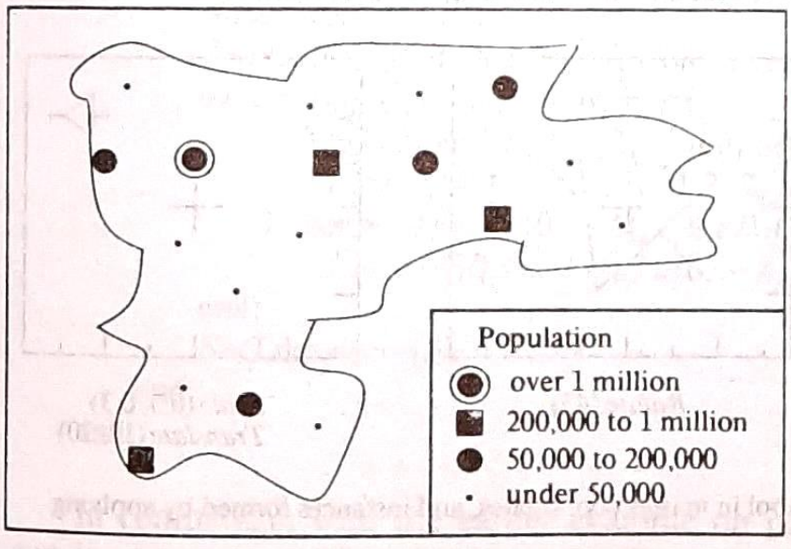


Figure 9-5 Use of symbols in a population map.

that we can replicate it all over the world coordinate space. We must therefore define the symbol in a different coordinate system, called the *master coordinate system*. To display each instance of the symbol, we define an *instance transformation* to be applied to the master-coordinate definition to transform it into world coordinates.

Functions are needed for the specification of instance transformations. We therefore add three new *instance-transformation functions* that allow us to define the scaling, rotation, and translation to be applied to the symbol:

Scale(*sx*, *sy*)	scale the symbol by factors *sx* and *sy* in the *x* and *y* directions relative to the origin
Rotate(θ)	rotate the symbol through $\theta°$ degrees clockwise about the origin
Translate(*tx*, *ty*)	translate the symbol through distances *tx* and *ty* measured in the *x* and *y* directions

The use of this set of three functions follows naturally from the discussion of Chapter 4. Other ways of applying instance transformations are discussed in Chapter 10. By means of calls to one or more of these functions we can orient symbols in a variety of different ways, illustrated by the examples shown in Figure 9-6. As we have seen in Chapter 4, the order in which the functions are called affects the resulting instance transformation. We generally call *Scale* before *Rotate* and call *Translate* last of all. Other orderings can be used, but it is not so easy to predict the effects.

Since the instance transformation produces world coordinates, we must still invoke the viewing transformation on its results. In our simple graphics-package design we included the viewing transformation as a step in the primitive line-drawing function (Figure 9-7a); now we shall extend this function so that it effectively includes an instance transformation (Figure 9-7b). Thus we have augmented the ability of our graphics package to handle symbols by providing three extra functions and by redesigning our primitive graphic functions to incorporate both instance and viewing transformations.

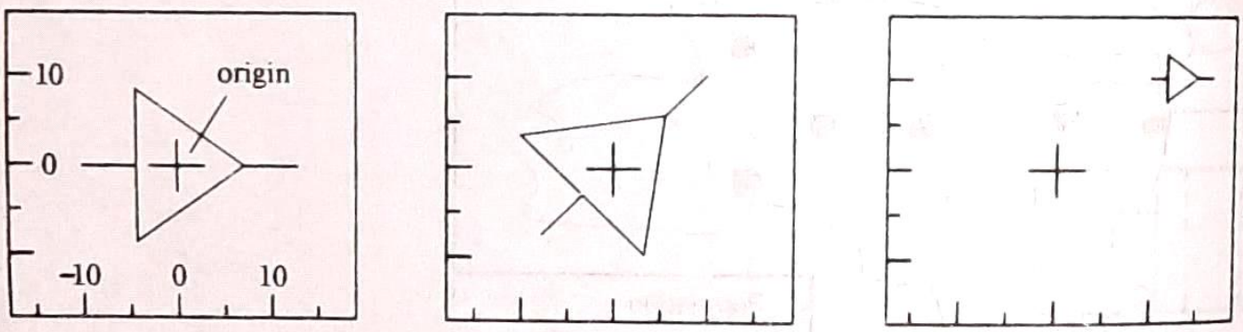


Figure 9-6 The definition of a symbol in master coordinates, and instances formed by applying transformations.

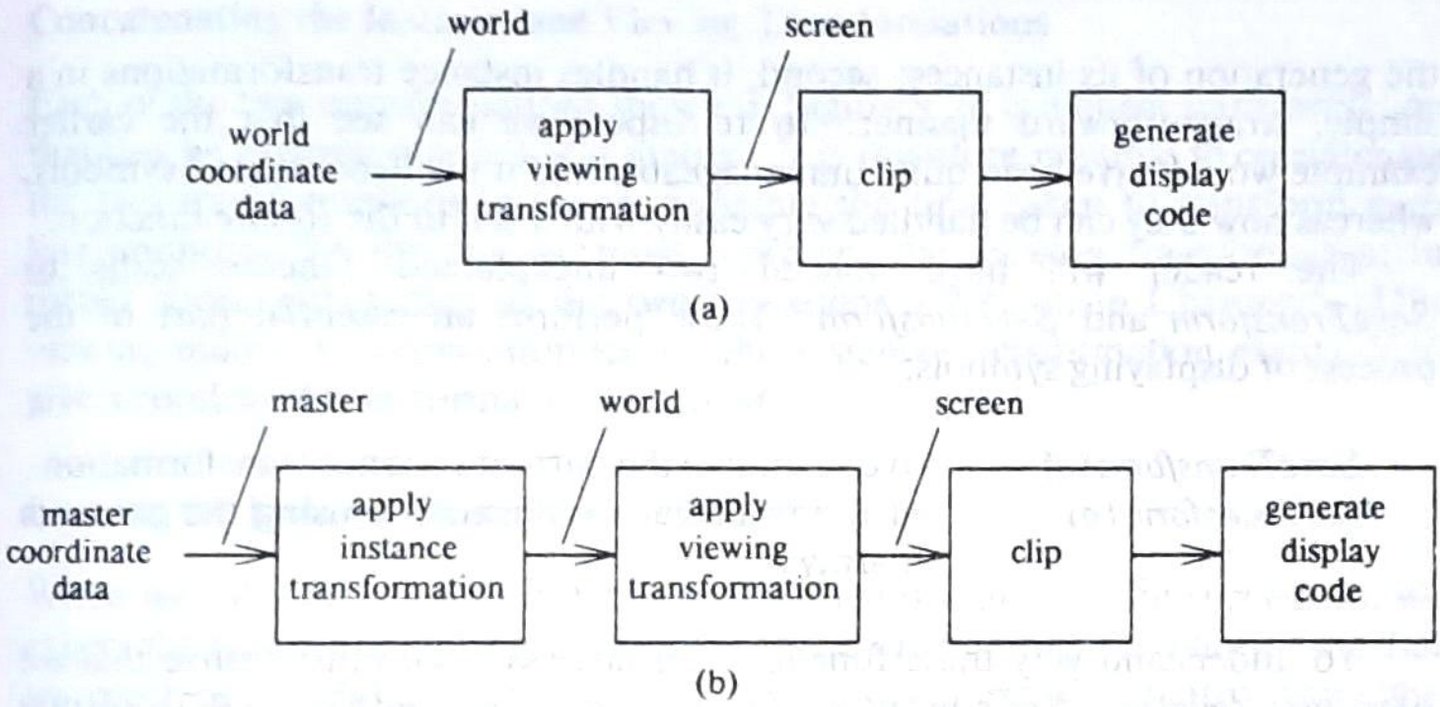


Figure 9-7 Steps in the transformation sequence: (a) without instance transformations; (b) with instance transformations.

How we define symbols and invoke instances of them is shown in the following example. Here we have taken the earlier example of a circuit diagram and, using the data structure of Figure 9-1, have included short sequences of function calls in the *ShowComponents* procedure to define the symbols to have the shapes shown in Figure 9-8. Each such sequence constitutes the master-coordinate definition of the symbol. It is preceded by a call to the *Translate* function that sets the position of the symbol for each of the instances defined in the data structure.

```
procedure ShowComponents;
    var i,j: integer; a: transform;
begin
    for i := 1 to nComponents do begin
        SaveTransform(a);
        Translate(Components[i].x, Components[i].y);
        case Components[i].comptype of
            1: begin MoveTo(0, -2); LineTo(0, 2);
                LineTo(4, 0); LineTo(0, -2) end;
            2: begin MoveTo(0, -4); LineTo(0, 4); LineTo(6, 4);
                LineTo(10, 0); LineTo(6, -4); LineTo(0, -4) end
        end;
        SetTransform(a)
    end
end;
```

In comparison with the earlier example on page 114, this program shows two improvements. First, it takes care of both the definition of each symbol and

the generation of its instances; second, it handles instance transformations in a simple, straightforward manner. In retrospect we can see that the earlier example would have been quite unmanageable had it involved rotated symbols, whereas now they can be handled very easily with a call to the *Rotate* function.

The reader will have noticed two unexplained function calls to *SaveTransform* and *SetTransform*. These perform an essential part of the process of displaying symbols:

SaveTransform(*a*)	save in array *a* the current instance transformation
SetTransform(*a*)	set the instance transformation using the contents of array *a*

To understand why these functions are necessary we must realize that we have now redesigned our primitive functions as shown in Figure 9-7b so that they apply an instance transformation to every coordinate point they receive. At the outset this transformation is set to the identity matrix, and it remains so unless we specify otherwise by calling *Scale*, *Rotate*, or *Translate*. When we encounter a sequence of these function calls, we would like the transformations they specify to be concatenated with the current instance-transformation matrix. Within the loop of our program, however, a call to *Translate* is encountered during each iteration. We do *not* want to concatenate each translation with the previous one; instead we want to discard the previous one and concatenate the new translation with the original instance-transformation matrix. By calling *SetTransform* at the end of the loop we restore the original matrix, saved by an earlier call to *SaveTransform*.

The *SaveTransform* and *SetTransform* functions provide somewhat more power and generality than is needed for our simple example: we could get by with a simple function to reset the instance transformation to the identity function. The ability to save the instance transformation at any moment and to restore it later becomes very useful in the more complex situations described in the next chapter.

9-4 IMPLEMENTATION OF INSTANCE TRANSFORMATIONS

Chapter 4 has shown that the use of 3×3 matrices makes it very easy to concatenate transformations together. We would expect instance-transformation software to be equally straightforward. In conceptual terms it is indeed simple, as we saw in the block diagram of Figure 9-7b. In practice, however, there is need to add refinements to the process which make the implementation of instance transformations slightly more complicated.

Concatenating the Instance and Viewing Transformations

Each of the two transformations shown in Figure 9-7b is a linear transformation that can be expressed as a 3×3 matrix. It is therefore possible to concatenate the two transformations into one, reducing the time taken to transform each line endpont. To do this we must represent the viewing transformation in matrix form, rather than as the two Equations 5-2 given in Chapter 5. This viewing matrix V is premultiplied by the instance-transformation matrix N to give a combined transformation matrix M.

Forming the Instance-Transformation Matrix

When we call two or more instance-transformation functions in succession, we expect the transformations to be applied in the order specified. This means that we must be careful in forming the instance-transformation matrix from the individual matrices representing each function call. Thus if we wish to form a matrix Q from a scaling matrix S, a rotation R, and a translation T, specified in that order, the three component matrices must be concatenated in the same order:

$$Q = S R T \tag{9-1}$$

When we perform this concatenation, we must allow for the possibility that the current instance transformation N_0, which we are about to update, is not the identity matrix. This happens in cases of *multilevel structure* where symbols are formed out of component parts, or are grouped together in *subpictures* (see Figure 9-9). Multilevel structures are discussed more fully in the next chapter; at this point we should understand that their display involves applying a separate instance transformation *at each level.* Thus to display the component rectangle indicated in Figure 9-9 we must first apply the instance transformation of the rectangle relative to the door, then concatenate this with the instance transformation of the door relative to the house, and finally concatenate the instance transformation of the entire house.

The implication of this to the matrix formation process is that after concatenating the individual transformations to form Q, we must use this

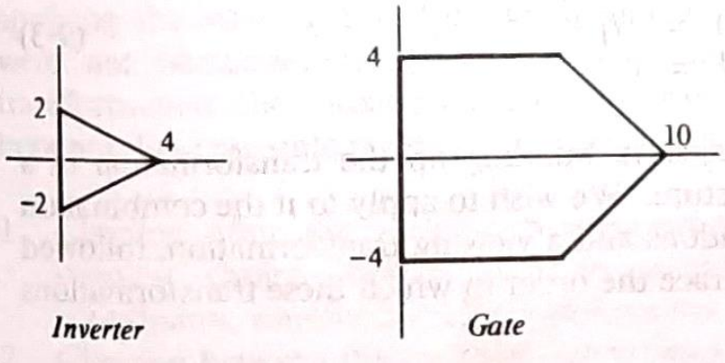


Figure 9-8 Inverter and gate symbols.

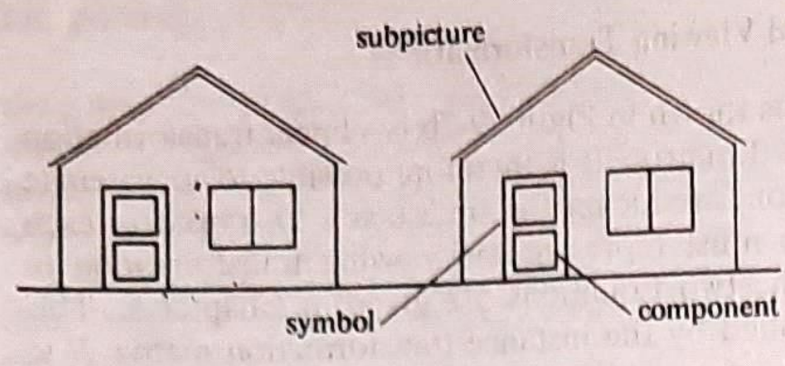


Figure 9-9 Picture with multilevel structure showing subpictures, symbols, and symbol components.

matrix to premultiply the current instance transformation N_0, yielding a new instance transformation N_1. We then multiply N_1 by the viewing transformation V to obtain the combined transformation matrix M:

$$\begin{aligned} Q &= S\,R\,T \\ N_1 &= Q\,N_0 \\ M &= N_1\,V \end{aligned} \tag{9-2}$$

We form the temporary matrix Q by setting it to the identity matrix initially and postmultiplying it by each matrix formed from the parameters passed in function calls (in the above case, by S, R, and T). These matrices are themselves formed from templates based on Equations 4-8 to 4-10 given in Section 4-3.

Each time Q is modified in this way, the combined matrix M representing instance and viewing transformations becomes obsolete. The obvious remedy is to multiply the transformation matrices together after each instance-transformation function call, using Equations 9-2. This is an expensive solution, however, since several function calls are often grouped together, and in these cases we need compute the combined matrix only at the end of the sequence of calls. How can we detect the end of such a sequence?

The solution to this problem lies in modifying our primitive graphic functions to test whether Q has changed since the previous primitive-function call; in cases of change, the matrices are combined and Q is reset to the identity matrix. A simple 1-bit flag makes this check very easy to perform. For consistency the same check must be included in the *SaveTransform* function.

Thus the three steps in updating M are

$$\begin{aligned} N_1 &= Q\,N_0 \\ M &= N_1\,V \\ Q &= I \end{aligned} \tag{9-3}$$

Let us follow through the stages in building up the transformation of a symbol that forms part of a subpicture. We wish to apply to it the combination of two sets of instance transformations and a viewing transformation, followed by a clipping operation. We can trace the order in which these transformations are concatenated:

1. At the outset, the combined transformation matrix M is set equal to the viewing transformation V, and the current-instance transformation N_0 and the temporary matrix Q are set equal to the identity matrix.
2. As we encounter the instance-transformation functions for the subpicture, we postmultiply the temporary matrix Q by each transformation.
3. At the start of the subpicture we encounter a call to *SaveTransform*. Since Q is no longer the identity matrix, we must update the combined matrix M and reset Q to the identity matrix, as in Equation 9-3.
4. Primitives within the subpicture are transformed with M and clipped.
5. We encounter instance-transformation functions to be applied to a symbol called by the subpicture definition. We postmultiply Q by each of these transformations.
6. At the start of the symbol procedure we encounter a graphic primitive function. As in step 3, Q is no longer the identity matrix, so we update the combined matrix M and reset Q to the identity matrix.
7. Lines within the symbol definition are transformed by the combined matrix M before clipping.
8. After displaying the symbol we return to the subpicture definition, where we encounter a call to *SetTransform*. We reset the instance transformation N from the specified array and multiply it by V to compute the combined matrix M. The temporary matrix Q is reset, and we are now ready to proceed with the subpicture definition.

One of the lessons we can learn by following through this example is the importance of the *SaveTransform* and *SetTransform* functions. We can see now how necessary they are to the use of multiple levels of structure; without them we would be unable to restore intermediate values of the instance transformation N, as we must on returning to a subpicture from a symbol procedure call. A less obvious role played by *SaveTransform* is to update the combined transformation M. If this is not done between each set of instance-transformation function calls, two or more sets will be concatenated together in a single temporary matrix Q in the wrong order.

Clipping before and after Transformation

At the end of Chapter 5 we noted that clipping could be performed before applying the viewing transformation, with the advantage that invisible lines were not needlessly transformed. With the introduction of the instance transformation the situation must be reexamined. We now must choose between three possible positions for the clipping stage:

1. Clipping after the viewing transformation (Figure 9-7b). This is the simplest choice, but it involves transforming all lines into screen coordinates, whether they are visible or not.
2. Clipping between the instance and viewing transformations (Figure 9-10a).

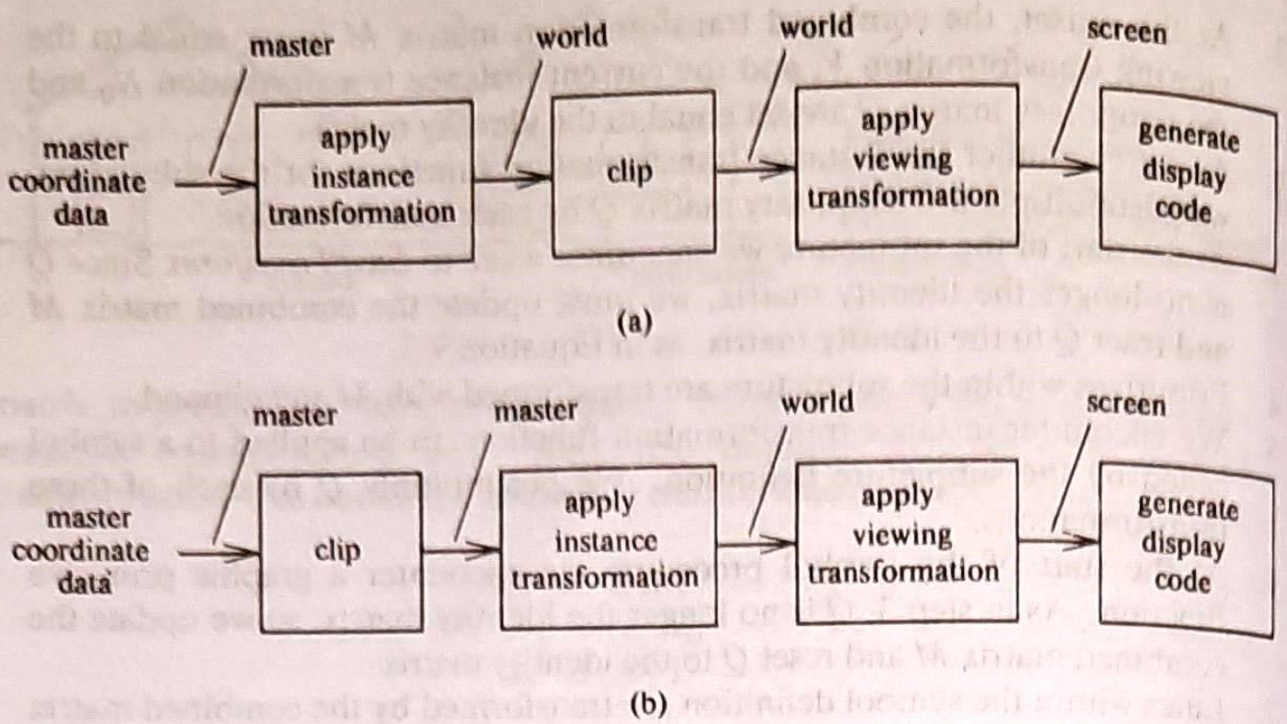


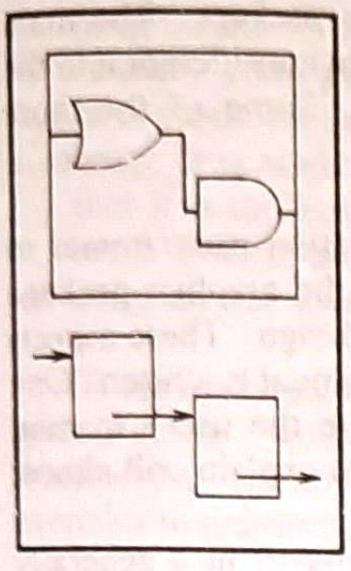
Figure 9-10 (a) Clipping before the viewing transformation; (b) clipping before the instance transformation.

Whereas sequence 1 uses the viewport as a clipping region, this sequence uses the window. It still involves applying the instance transformation to every line but avoids the need to apply the viewing transformation to invisible lines. The overriding disadvantage of this sequence is that it prevents us from concatenating the instance and viewing transformations.

3. Clipping before the instance transformation (Figure 9-10b). This sequence allows us to avoid applying any transformations to invisible lines. It is a difficult process to implement, however, because it implies computing the window's position in master coordinates; which can be done only by applying the inverse of the instance transformation to the window. Worse still, the resulting clipping region will not be aligned with the axes if the instance transformation involves a rotation. Thus we may not be able to use the Cohen-Sutherland clipping algorithm.

It is possible to use sequence 3 in cases where symbols are not rotated. In these cases the inverse of the instance transformation is easily computed, and the clipping region is aligned with the axes. The transformation process must be *adaptive*, switching from sequence 3 to sequence 1 whenever rotated symbols are encountered. The resulting software is relatively complicated.

In most cases, the use of sequence 1 with its final clipping stage provides satisfactory performance. The transformation process is simple, and this reduces the amount of software required and the number of checks that must be invoked during the output process. We can expect to notice degradation in system performance when the amount of invisible information becomes very large. Methods of structuring the information to prevent such a degradation are discussed in the next chapter.



6

A SIMPLE GRAPHICS PACKAGE

A *graphics system* may be defined as any collection of hardware and software designed to make it easier to use graphic input and output in computer programs. The design of graphics systems is a very important aspect of computer graphics. Without such systems, graphics application programs would be extremely difficult to write; only the most expert programmers would be competent to write them, and their rate of software production would be very slow. It is only by constructing graphics systems that we make it possible to exploit the potential uses of computer graphics.

This chapter and the others in Part Two are concerned mainly with the software component of graphics systems. The type of graphics software system we shall consider first is the most basic and essential of all, the *graphics package*. This is a set of subroutines or functions used by an application program to generate pictures on a plotter or display and to handle graphical interaction. We shall consider the graphical display rather than the plotter as our main output device, since we are concerned with graphics packages for interactive use. To start with, we shall employ a single form of output device, the storage-tube display described in Chapter 3, with the command repertoire shown in Figure 3-10. To permit true interactive use we shall extend our discussion during Chapters 7 and 8 to include techniques for output to refresh displays. Eventually we shall consider the design of graphic input software after covering the necessary background material of Chapters 11 and 12.

6-1 GROUND RULES FOR GRAPHICS SOFTWARE DESIGN

This chapter serves mainly to show how the basic concepts presented in Part One can be utilized in the design of a simple graphics package. The main difficulty in constructing such a package is to give adequate consideration to the many issues that affect the usefulness of the package. Some of the more important of these issues are:

Simplicity. Features that are too complex for the application programmer to understand will not be used. It is often difficult for the graphics package designer to detect potentially difficult aspects of his design. These aspects do, however, become more obvious when the user's manual is written. One way to avoid these difficult features is to try to write the user's manual before implementing the system. Anything difficult to explain will almost certainly be difficult to use.

Consistency. A consistent graphics system is one that behaves in a generally predictable manner. Function names, calling sequences, error handling, and coordinate systems all should follow simple and consistent patterns *without exceptions.* The application programmer is then able to build up a conceptual *model* of how the package functions. Again, inconsistencies often show up when the user's manual is written as small-print passages—sentences here and there, often overlooked by the reader, that warn of anomalies in the system.

Completeness. There should be no irritating omissions in the set of functions provided by the system; missing functions will have to be supplied by the application programmer, who may not have the necessary access to the computer's resources to be able to write them. Completeness does not imply comprehensiveness; i.e., the system need not provide every imaginable graphics facility. The system designer must try to design a reasonably small set of functions that can conveniently handle a wide range of applications.

Robustness. Application programmers are capable of extraordinary misuse of graphics systems, either through misunderstanding or through the mischievous enjoyment of trying everything once. The system should accept such treatment with a minimum of complaint. Trivial errors of omission or repetition should be corrected without comment from the system. When the programmer does something seriously wrong, the system should report the error in the most helpful manner possible. Only in extreme circumstances should errors cause termination of execution since this will generally cause the user to lose valuable results.

Performance. Graphics-system performance is often limited by such factors as operating-system response and display characteristics, factors beyond the system designer's control. There is nothing much to be done here by the graphics-system designer except to minimize the effect by omitting highly dynamic graphic functions that demand fast response or expensive

hardware. Apart from this, the system designer should try to maintain consistent performance, so that application programs provide an equally consistent speed of response. It should be possible for the application programmer to achieve this consistency without resorting to special tricks. A really good graphics system offers no special advantages to those who understand the system's internal workings.

Economy. It is always frustrating to write an application program only to find that it is too bulky or too expensive to use. Graphics systems should be small and economical, so that adding graphics to an existing application program can always be considered.

These issues provide us with a number of *ground rules* of graphics software system design. Abiding by these rules is not always easy but is almost always worthwhile. We shall try wherever possible to invoke these rules during the exercises in graphics package design that occur in the next two chapters.

6-2 FUNCTIONAL DOMAINS

In the initial phase of graphics-system design the issue of completeness is more likely than any other to give the designer trouble. All of a sudden he finds himself inundated with conflicting requirements. He starts with a fairly simple design for a storage-tube display; then he decides to extend it to output to a refresh line-drawing display; he adds special input functions for dynamic graphics applications and three-dimensional transformations for displaying molecular models; then since his display equipment contains hardware for scaling but none for rotation, he omits the rotation function from the system. So he proceeds, never sure when he has reached a satisfactory resting place.

Anyone who designs a graphics package in this manner is ignoring two very important points. First, graphics package design must not be unduly influenced by hardware features or by highly specialized application requirements. Second, the functions in a graphics package can be separated into sets, each set concerned with a particular kind of task; it is then possible to address the issue of completeness, one set at a time. There are a number of such function sets to be considered in designing a system; in our first graphics package we shall provide the following two:

1. *Graphic primitives.* These are used to display straight lines, text strings, circular arcs, and other simple graphical items.
2. *Windowing functions.* These allow the programmer to choose his coordinate system for picture definition and to define the boundary of the visible portion of the picture.

Other function sets that we encounter in later chapters include *segmenting functions*, to permit easier modification of the picture; *two-dimensional*

transformation functions, for rotation, translation, and scaling (see Chapter 4); and *input functions*, that allow the user to give commands with the help of a keyboard or graphical input device. In addition to the functions in these sets a number of miscellaneous utility functions for such things as initialization and inquiry always find their way into any graphics package.

The initial phase of system design, then, involves choosing a certain number of these sets to form the *functional domain* of the system. We then attempt to achieve a comprehensive collection of functions within each set while simultaneously attending to the other system-design ground rules.

The reasons behind our initial choice of two function sets are quite simple. In the first place, no system can do without graphic primitives; otherwise no lines can be drawn on the screen. Second, it is essential to provide windowing functions so that the user need not be concerned with details of the screen coordinate system. These are indeed the only two function sets essential to all graphics systems.

6-3 GRAPHIC PRIMITIVES

Graphic primitives are the functions that we use to specify the actual lines and characters that make up the picture. These functions must be chosen carefully to provide the application programmer with convenient methods of describing pictures and at the same time to take advantage of the capabilities of the display hardware. It is not sufficient, for example, to provide only a single function to display a dot at a specified coordinate position: this would be laborious to use and would not exploit the line-drawing capabilities of most displays. We shall use a set of three primitive functions in our graphics package:

MoveTo(x,y)*	set current beam position (x,y) for the next line or text string
LineTo(x,y)	draw a straight line from the current beam position to the point (x,y) and reset the starting point to (x,y)
DrawText(s)	display the string s with its lower left corner at the current beam position; then reset the current beam position to the string's lower right corner

This is a much smaller set of graphic primitives than one normally finds in graphics packages. We shall purposely keep it small to simplify the following discussion. Normally the function set would be augmented with functions to display dots, to draw dashed and dotted lines, to specify lines by their length

*The function names used in this and following chapters adhere to modern programming language practice in using upper- and lowercase characters and in not restricting name length. Different names would be required in a graphics package for use with FORTRAN, for example.

rather than by their endpoint, to draw circular arcs, and so forth. The inclusion of too many primitive functions is a source of complexity and expense, however, and it is preferable to err in the direction of too few functions. Most of the graphic entities omitted can be drawn with combinations of the above set of three functions.

6-4 WINDOWING FUNCTIONS

It is convenient if window and viewport can be defined separately with a pair of functions:

SetWindow(*wxmin, wymin, wxmax, wymax*)
: define window position to extend from (*wxmin, wymin*) at the lower left to (*wxmax, wymax*) at the upper right

SetViewport(*vxmin, vymin, vxmax, vymax*)
: define viewport correspondingly

A single function could of course do the work of both of these functions, but it would then be more difficult to change the window independently of the viewport.

6-5 MISCELLANEOUS FUNCTIONS

To complete this simple graphics package, we need three utility functions:

ClearScreen
: to clear the screen

InitGraphics
: to be called once at the start of execution of every graphics program, to clear the screen, initialize the world-to-screen transformation, etc.

Inquire(*v*)
: to determine essential terminal characteristics; returns with the following information in vector *v*:
  maximum screen *x* in *v.screenxmax*
  maximum screen *y* in *v.screenymax*
  width of text characters in *v.textwidth*
  height of text characters in *v.textheight*,
  etc.

The use of the package is illustrated in the following example program.

6-6 EXAMPLE: A GRAPH-PLOTTING PROGRAM

A general-purpose graph-plotting program is an example of the sort of simple but useful application program we can write with our graphics package. The program makes good use of windowing to perform automatic scaling of axes.

The program is written as a procedure, *GraphPlot*, called with five arguments:

1. The number *n* of data points on the graph

```
procedure GraphPlot(n: integer; var vx, vy: array [1..100] of integer; var sx, sy: string);
    var xmax, ymax, xinterval, yinterval, i: integer; v: InquiryResponse; s: string;
begin
    InitGraphics; Inquire(v);
    xmax := vx[1]; ymax := vy[1];
    for i := 2 to n do begin { Find maximum data values }
        if vx[i] > xmax then xmax := vx[i];
        if vy[i] > ymax then ymax := vy[i]
    end;
    xinterval := RoundUpInterval(xmax/10); xmax := xinterval * 10;
    yinterval := RoundUpInterval(ymax/10); ymax := yinterval * 10;
    SetWindow(0, 0, xmax, ymax); { Region for graph }
    SetViewport(6*v.textwidth, 4*v.textheight,v.screenxmax,v.screenymax);
    MoveTo(0, ymax); LineTo(0, 0); LineTo(xmax, 0); { Draw axes }
    MoveTo(vx[1], vy[1]);
    for i := 2 to n do LineTo(vx[i], vy[i]); { Draw graph }
    SetWindow(0, 0, xmax, 4 * v.textheight); { Region for horizontal scale }
    SetViewport(6 * v.textwidth, 0, v.screenxmax, 4 * v.textheight);
    MoveTo(xinterval, 0); DrawText(sx);
    for i := 1 to 9 do begin
        MoveTo(xinterval * i, 4 * v.textheight);
        LineTo(xinterval * i, 3 * v.textheight);
        MoveTo(xinterval * i - 2 * v.textwidth, v.textheight);
        IntegerToString(xinterval * i, s); DrawText(s)
    end;
    SetWindow(0, 0, 6 * v.textwidth, ymax); { Region for vertical scale }
    SetViewport(0, 4 * v.textheight, 6 * v.textwidth, v.screenymax);
    MoveTo(0, 9.5 * yinterval); DrawText(sy);
    for i := 1 to 9 do begin
        MoveTo(6 * v.textwidth, yinterval * i);
        LineTo(5 * v.textwidth, yinterval * i);
        MoveTo(0, yinterval * i);
        IntegerToString(yinterval * i, s); DrawText(s)
    end;
end;
```

Figure 6-1 Graph-plotting program written in PASCAL.

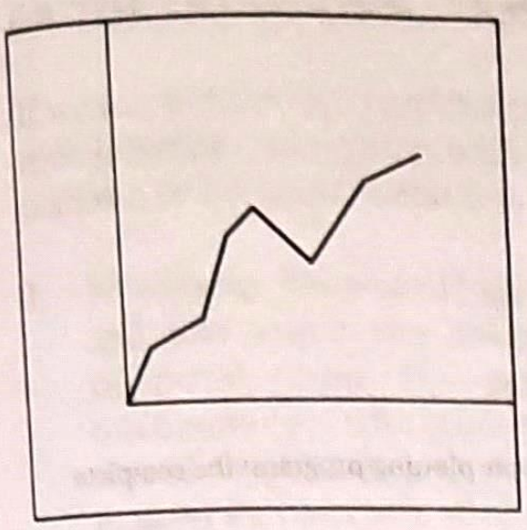


Figure 6-2 Graph-plotting program: plotting the curve and axes.

2. A pointer to a vector *vx* containing *n* coordinate values for the *independent* variable
3. A pointer to a similar vector *vy* for the *dependent* variable
4. A string pointer *sx* for use in labeling the horizontal axis
5. A string pointer *sy* for the vertical axis

In this example we shall for simplicity assume that the origin (0,0) of the graph's coordinate system will always lie at the lower left corner of the graph plot. This allows us to make particularly effective use of windowing in labeling the axes.

The program is shown written in PASCAL in Figure 6-1. After calls to *InitGraphics* and *Inquire*, the program determines the maximum values in the two vectors *vx* and *vy*: these are stored in *xmax* and *ymax* and are used in a call to *SetWindow*. The viewport is set so as to leave a space at the left of the screen, 6 times the width of a displayed character and another at the bottom 4 times a character's height. These spaces are later used for axis labeling. The axes are then displayed, followed by the graph itself, plotted as a sequence of connected straight lines. The appearance of the display now resembles Figure 6-2.

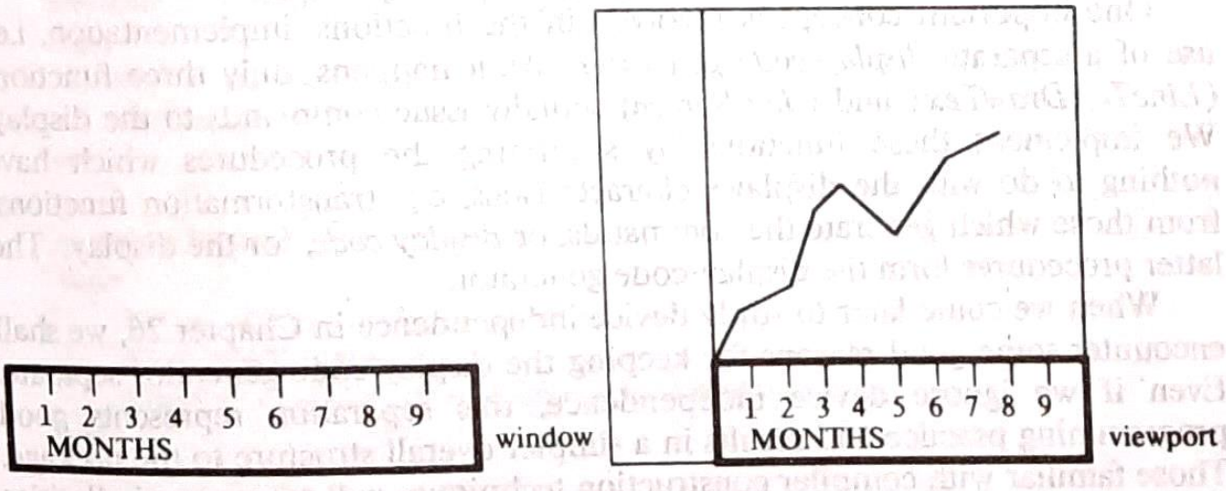


Figure 6-3 Graph-plotting program: labeling the axes using separate viewports.

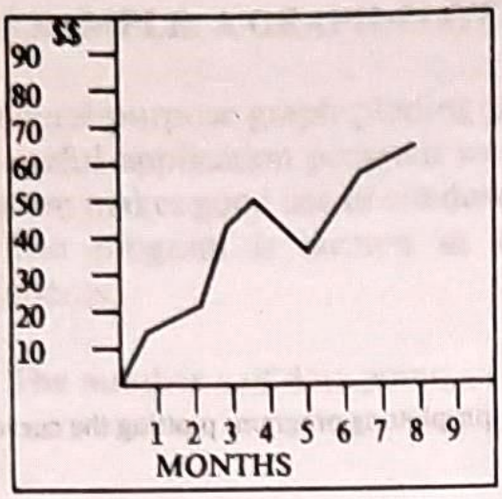


Figure 6-4 Graph-plotting program: the complete display.

To label the horizontal axis we set window and viewport to the same widths as before, but we give them identical heights equal to four character heights (see Figure 6-3). Thus scaling in the horizontal direction is the same as for the graph and its axes, whereas in the vertical direction it is unity. This permits us to position text labels vertically independent of the graph's scaling. The labeling interval is determined as a function of the range of the independent variable, to give ten intervals across the graph. It is then easy to display the divisions, their labels, and the labeling string supplied. The vertical axis is labeled in a similar manner, with the labeling string at the top. Figure 6-4 shows the completed graph display.

6-7 IMPLEMENTATION OF THE FUNCTIONS

It is worthwhile to complete our first exercise in graphics system design by studying in some detail the internal design of each of the functions the system provides. Fortunately none of these functions is particularly complicated, thanks to the overall simplicity of the graphics package itself.

One important concept is involved in the functions' implementation, i.e., use of a separate *display-code generator.* As it happens, only three functions (*LineTo, DrawText,* and *ClearScreen*) actually issue commands to the display. We implement these functions by separating the procedures which have nothing to do with the display's characteristics, e.g. transformation functions, from those which generate the commands, or *display code*, for the display. The latter procedures form the display-code generator.

When we come later to study device independence in Chapter 26, we shall encounter some good reasons for keeping the display-code generator separate. Even if we ignore device independence, this separation represents good programming practice and results in a simpler overall structure to the package. Those familiar with compiler construction techniques will recognize similarities between this approach and the use of a separate code generator in a compiler.

6 Windowing and Clipping

6.1 Introduction

Typically, a graphics package allows us to specify which part of a defined picture is to be displayed and where that part is to be displayed on the display device. Furthermore, the package also provides the use of the scaling, translation and rotation techniques described in the previous chapter to generate a variety of different views of a single picture. We can generate different view of a picture by applying the appropriate scaling and translation. While doing this, we have to identify the visible part of the picture for inclusion in the display image. This selection process is not straight forward. Certain lines may lie partly inside the visible portion of the picture and partly outside. These lines cannot be omitted entirely from the display image because the image would become inaccurate. This is illustrated in Fig. 6.1. The process of selecting and viewing the picture with different views is called **windowing**, and a process which divides each element of the picture into its visible and invisible portions, allowing the invisible portion to be discarded is called **clipping**.

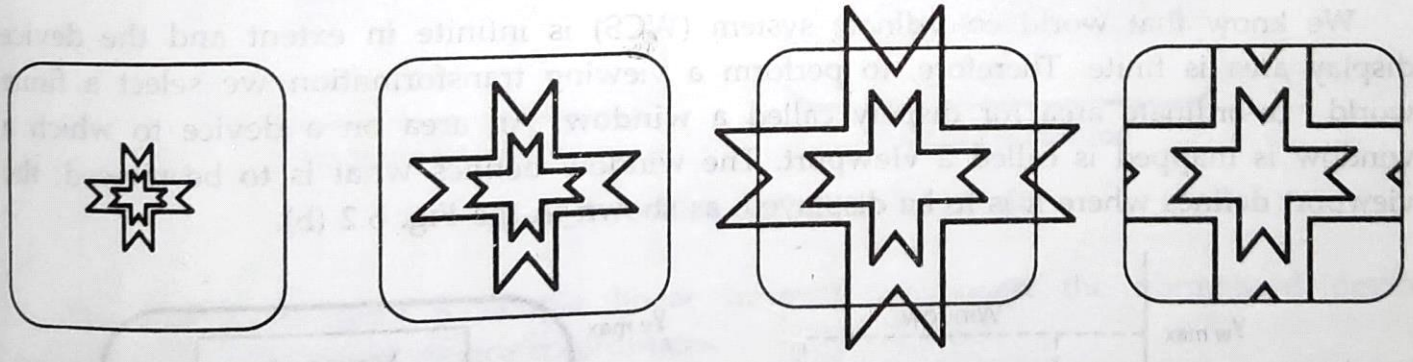


Fig. 6.1

In this chapter we are going to discuss the concepts involved in windowing and various clipping algorithms.

6.2 Viewing Transformation

We know that the picture is stored in the computer memory using any convenient cartesian co-ordinate system, referred to as **world co-ordinate system** (WCS). However, when picture is displayed on the display device it is measured in **physical device co-ordinate system** (PDCS) corresponding to the display device. Therefore, displaying an image of a picture involves mapping the co-ordinates of the points and lines that form the picture into the appropriate physical device co-ordinate where the image is to be displayed. This mapping of co-ordinates is achieved with the use of co-ordinate transformation known as **viewing transformation**. Sometimes the two dimensional viewing transformation is simply referred to as the **window to view port transformation** or the **windowing transformation**.

The viewing transformation which maps picture co-ordinates in the WCS to display co-ordinates in PDCS is performed by the following transformations.

- Converting world co-ordinates to viewing coordinates.
- Normalizing viewing coordinates.
- Converting normalized viewing coordinates to device co-ordinates

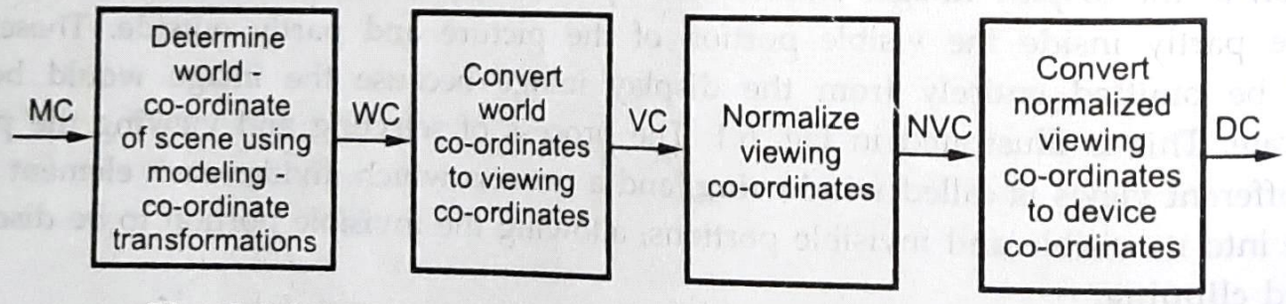


Fig. 6.2 (a) Two-dimensional viewing transformation pipeline

We know that world co-ordinate system (WCS) is infinite in extent and the device display area is finite. Therefore, to perform a viewing transformation we select a finite world co-ordinate area for display called a **window**. An area on a device to which a window is mapped is called a **viewport**. The window defines what is to be viewed; the viewport defines where it is to be displayed, as shown in the Fig. 6.2 (b).

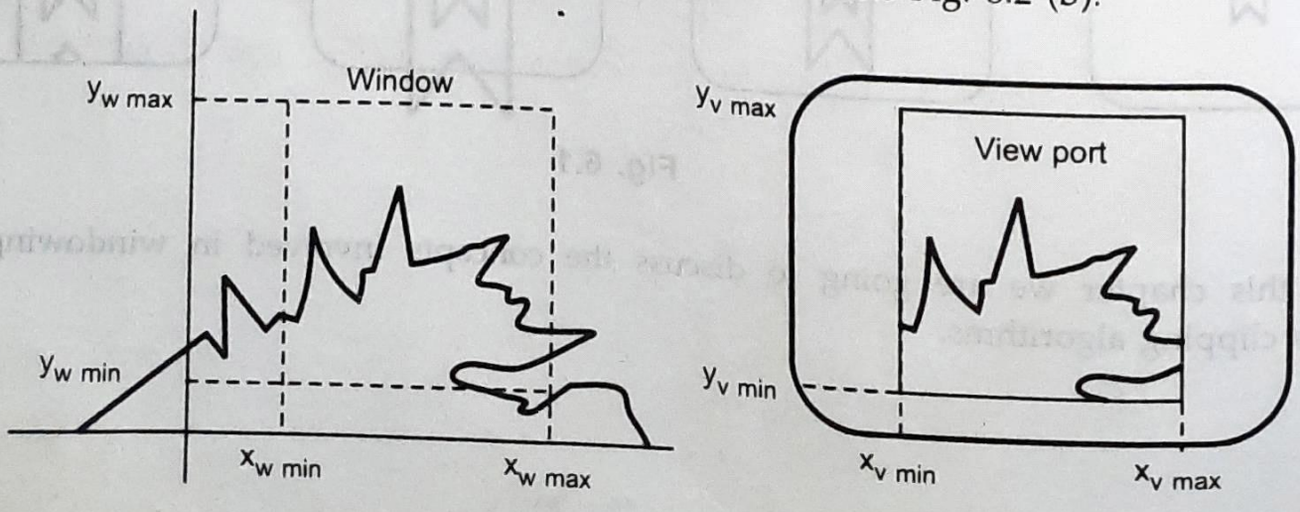
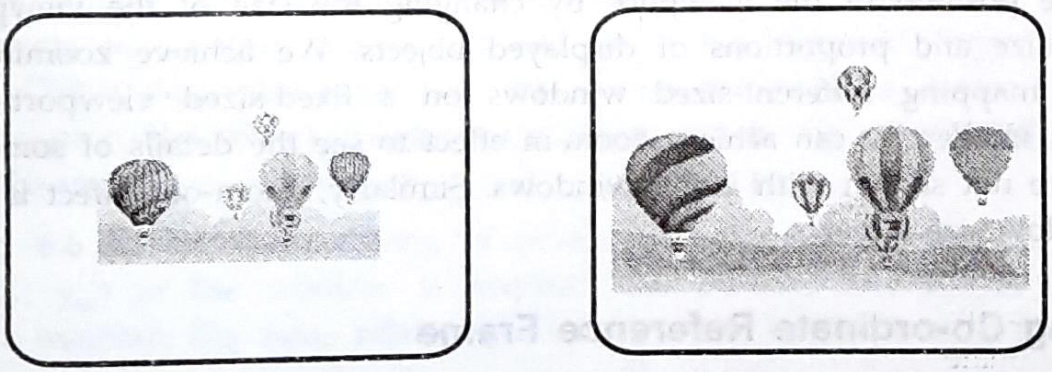


Fig. 6.2 (b) Window and viewport

The Fig. 6.2 (b) shows the two dimensional viewing pipeline. The window defined in world co-ordinate is first transformed into the viewport co-ordinate. For this, viewing coordinate reference frame is used. It provides a method for setting up arbitrary orientations for rectangular windows. Once the viewing reference frame is established, we can transform, descriptions in world co-ordinates to viewing coordinates. We then define a viewport in normalized coordinates.

Normalized Coordinates

We know that, different display devices may have different screen sizes as measured in pixels. Size of the screen in pixels increases as resolution of the screen increases. When picture is defined in the pixel values then it is displayed large in size on the low resolution screen while small in size on the high resolution screen as shown in the Fig. 6.3. To avoid this and to make our programs to be device independent, we have to define the picture co-ordinates in some units other than pixels and use the interpreter to convert these co-ordinates to appropriate pixel values for the particular display device. The device independent units are called the **normalized device co-ordinates**. In these units, the screen measures 1 unit wide and 1 unit length as shown in the Fig. 6.4. The lower left corner of the screen is the origin, and the upper-right corner is the point (1, 1). The point(0.5, 0.5) is the center of the screen no matter what the physical dimensions or resolution of the actual display device may be.



(a) More resolution (b) Less resolution

Fig. 6.3 Picture definition in pixels

The interpreter uses a simple linear formula to convert the normalized device co-ordinates to the actual device co-ordinates.

$$x = x_n \times X_W \quad \ldots (1)$$

$$y = y_n \times Y_H \quad \ldots (2)$$

where

x : Actual device x co-ordinate.

y : Actual device y co-ordinate.

x_n : Normalized x co-ordinate.
y_n : Normalized y co-ordinate.
X_W : Width of actual screen in pixels.
Y_H : Height of actual screen in pixels.

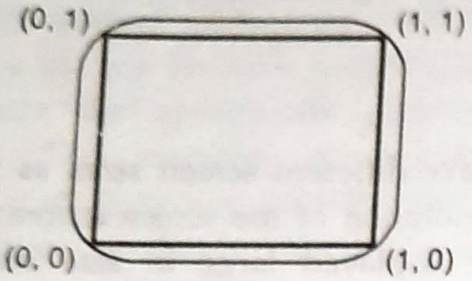


Fig. 6.4 Picture definition in normalized device co-ordinates

The transformation which map the viewing coordinate to normalized device coordinate is called **normalization transformation**. It involves scaling of x and y, thus it is also referred to as **scaling transformation**.

Device Coordinates

Finally, all parts of the picture that lie outside the viewport are clipped, and the contents of the viewport are transferred to device coordinates.

More about Viewport and Window

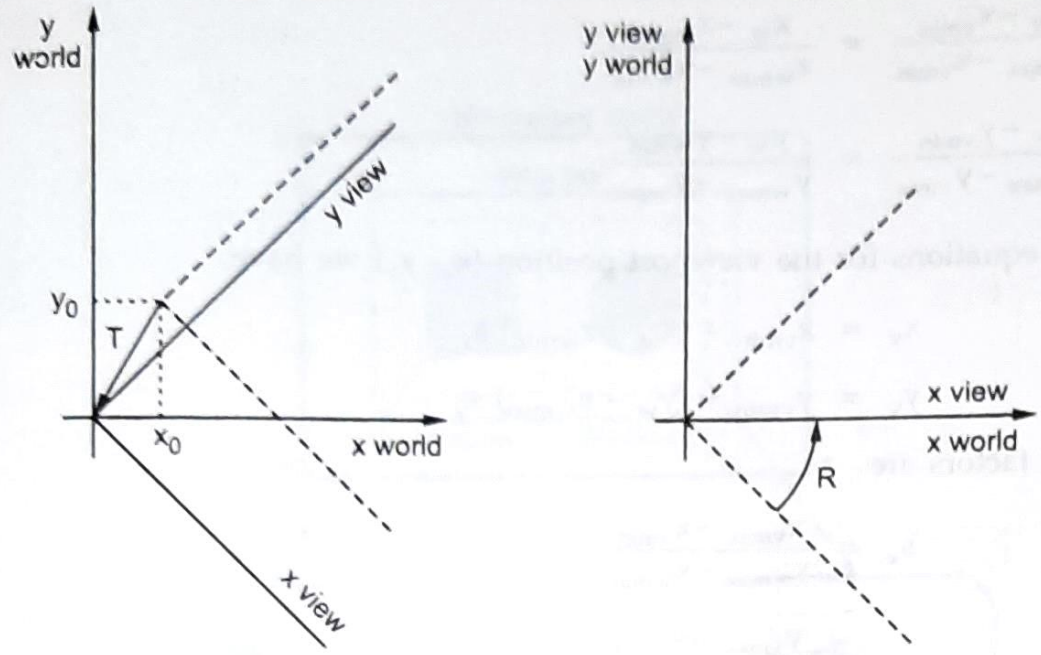
We can see objects at different positions on the display area of an output device by changing the position of the viewport. By changing the size of the viewport, we can change the size and proportions of displayed objects. We achieve zooming effects by successively mapping different-sized windows on a fixed-sized viewport. By making window size smaller we can achieve zoom-in effect to see the details of some part of the scene that are not shown with larger windows. Similarly, zoom-out effect is achieved by making window size larger.

6.3 Viewing Co-ordinate Reference Frame

In section 5.9.2, we have seen how to perform transformation between two Cartesian frames of reference. Similar steps can be applied to obtain the matrix for converting world coordinate positions to viewing co-ordinates. Here, we have to translate the viewing origin to the world origin and then align the two coordinate reference frames by rotation. This is illustrated in Fig. 6.5. As shown in the Fig. 6.5, the composite transformation (translation and rotation) converts world coordinates to viewing coordinates. The matrix used for this composite transformation is given by

$$M_{WC,VC} = T.R$$

where T is the translation matrix that takes the viewing origin point P_0 to the world origin, and R is the rotation matrix that aligns the axes of the two reference frames.



a) Translation of the viewing origin to the world origin.

b) Rotation to align the axes of the two systems.

Fig. 6.5

6.4 Window to Viewport Co-ordinate Transformation

Once the window coordinates are transferred to viewing coordinates we choose the window extents in viewing coordinates and select the viewport limits in normalized coordinates. Object descriptions are then transferred to normalize device coordinates. In this transformation, the relative placement of the object in the normalized coordinates is same as in the viewing coordinates.

The Fig. 6.6 shows the mapping of object from window to viewport. A point at position (x_w, y_w) in the window is mapped into position (x_v, y_v) in the associated viewport. To maintain the same relative placement in the viewport as in the window, we require that

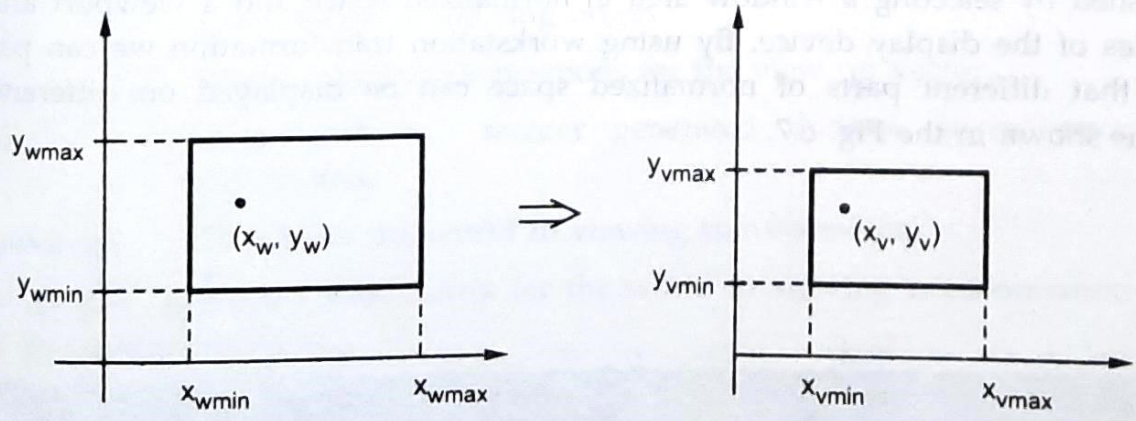
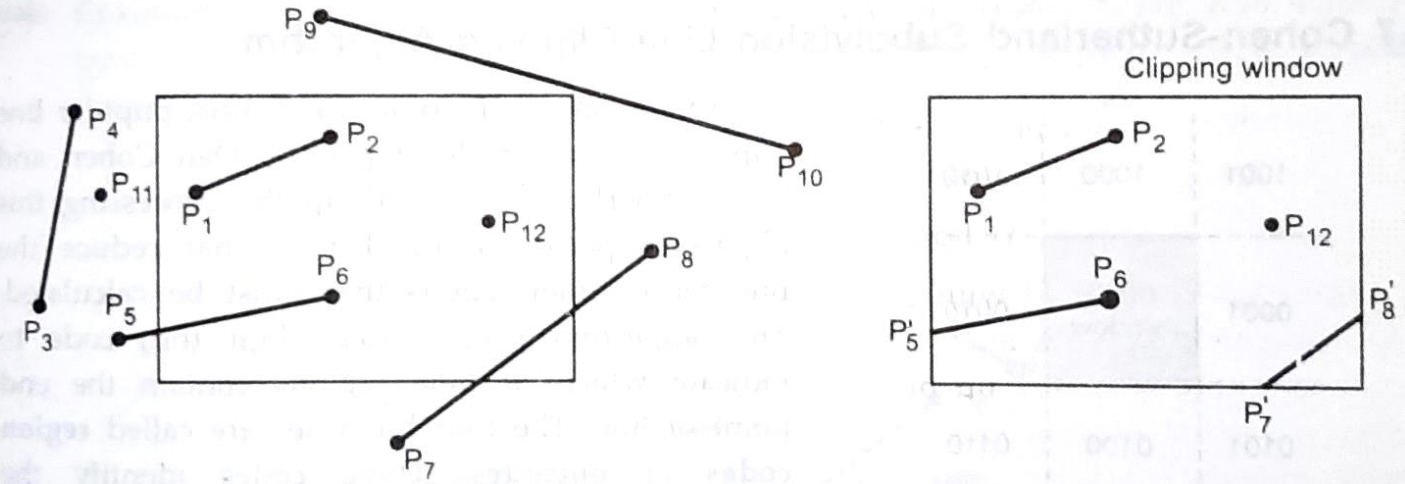


Fig. 6.6 Window to viewport mapping

6.6 2D Clipping

The procedure that identifies the portions of a picture that are either inside or outside of a specified region of space is referred to as clipping. The region against which an object is to be clipped is called a **clip window** or **clipping window.** It usually is in a rectangular shape, as shown in the Fig. 6.8.



(a) Before clipping (b) After clipping

Fig. 6.8

The clipping algorithm determines which points, lines or portions of lines lie within the clipping window. These points, lines or portions of lines are retained for display. All others are discarded.

6.6.1 Point Clipping

The points are said to be interior to the clipping window if

$$x_{w\,min} \leq x \leq x_{w\,max} \quad \text{and}$$

$$y_{w\,min} \leq y \leq y_{w\,max}$$

The equal sign indicates that points on the window boundary are included within the window.

6.6.2 Line Clipping

The lines are said to be interior to the clipping window and hence visible if both end points are interior to the window, e.g., line P_1 P_2 in Fig. 6.8. However, if both end points of a line are exterior to the window, the line is not necessarily completely exterior to the window, e.g. line P_7 P_8 in Fig. 6.8. If both end points of a line are completely to the right of, completely to the left of, completely above, or completely below the window, then the line is completely exterior to the window and hence invisible. For example, line P_3 P_4 in Fig. 6.8.

The lines which across one or more clipping boundaries require calculation of multiple intersection points to decide the visible portion of them. To minimize the intersection calculations and to increase the efficiency of the clipping algorithm, initially, completely visible and invisible lines are identified and then the intersection points are calculated for remaining lines. There are many line clipping algorithms. Let us discuss a few of them.

6.7 Cohen-Sutherland Subdivision Line Clipping Algorithm

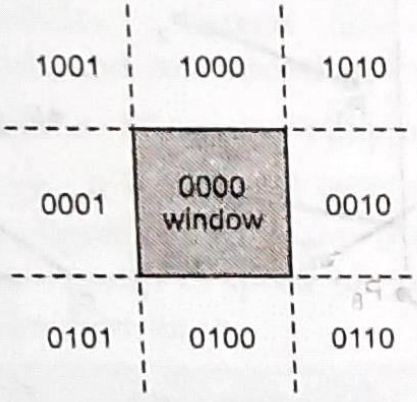


Fig. 6.9 Four-bit codes for nine regions

This is one of the oldest and most popular line clipping algorithm developed by Dan Cohen and Ivan Sutherland. To speed up the processing this algorithm performs initial tests that reduce the number of intersections that must be calculated. This algorithm uses a four digit (bit) code to indicate which of nine regions contain the end point of line. The four bit codes are called **region codes** or **outcodes**. These codes identify the location of the point relative to the boundaries of the clipping rectangle as shown in the Fig. 6.9.

Each bit position in the region code is used to indicate one of the four relative co-ordinate positions of the point with respect to the clipping window : to the left, right, top or bottom. The rightmost bit is the first bit and the bits are set to 1 based on the following scheme :

Set Bit 1 – if the end point is to the **left** of the window

Set Bit 2 – if the end point is to the **right** of the window

Set Bit 3 – if the end point is **below** the window

Set Bit 4 – if the end point is **above** the window

Otherwise, the bit is set to zero.

Once we have established region codes for all the line endpoints, we can determine which lines are completely inside the clipping window and which are clearly outside. Any lines that are completely inside the window boundaries have a region code of 0000 for both endpoints and we trivially accept these lines. Any lines that have a 1 in the same bit position in the region codes for each endpoint are completely outside the clipping

rectangle, and we trivially reject these lines. A method used to test lines for total clipping is equivalent to the logical **AND** operator. If the result of the logical AND operation with two end point codes is not 0000, the line is completely outside the clipping region. The lines that cannot be identified as completely inside or completely outside a clipping window by these tests are checked for intersection with the window boundaries.

Example 6.1 : *Consider the clipping window and the lines shown in Fig. 6.10. Find the region codes for each end point and identify whether the line is completely visible, partially visible or completely invisible.*

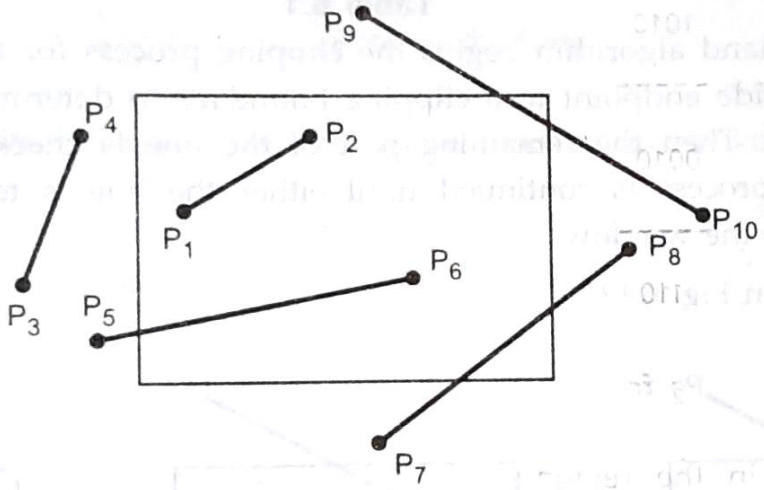


Fig. 6.10

Solution : The Fig. 6.11 shows the clipping window and lines with region codes. These codes are tabulated and end point codes are logically ANDed to identify the visibility of the line in table 6.1.

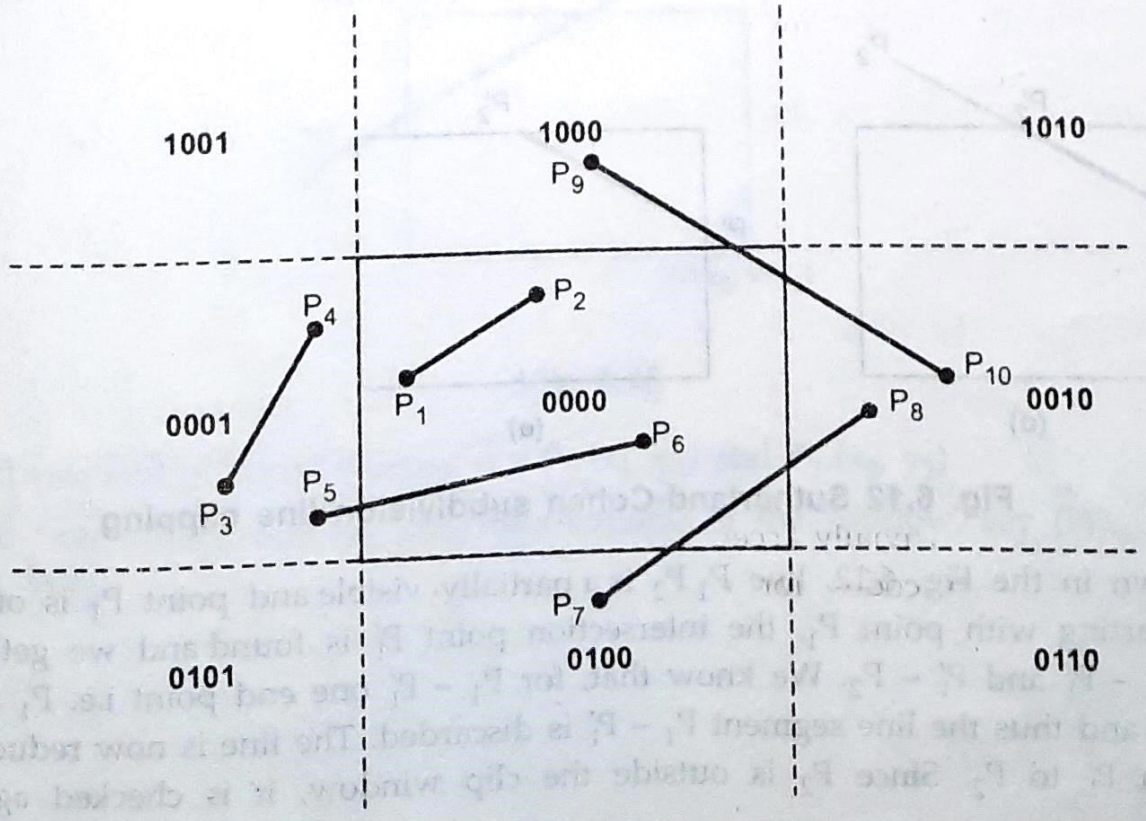


Fig. 6.11

Line	End Point Codes		Logical ANDing	Result
P_1 P_2	0000	0000	0000	Completely visible
P_3 P_4	0001	0001	0001	Completely invisible
P_5 P_6	0001	0000	0000	Partially visible
P_7 P_8	0100	0010	0000	Partially visible
P_9 P_{10}	1000	0010	0000	Partially visible

Table 6.1

The Cohen-Sutherland algorithm begins the clipping process for a partially visible line by comparing an outside endpoint to a clipping boundary to determine how much of the line can be discarded. Then the remaining part of the line is checked against the other boundaries, and the process is continued until either the line is totally discarded or a section is found inside the window.

This is illustrated in Fig. 6.12.

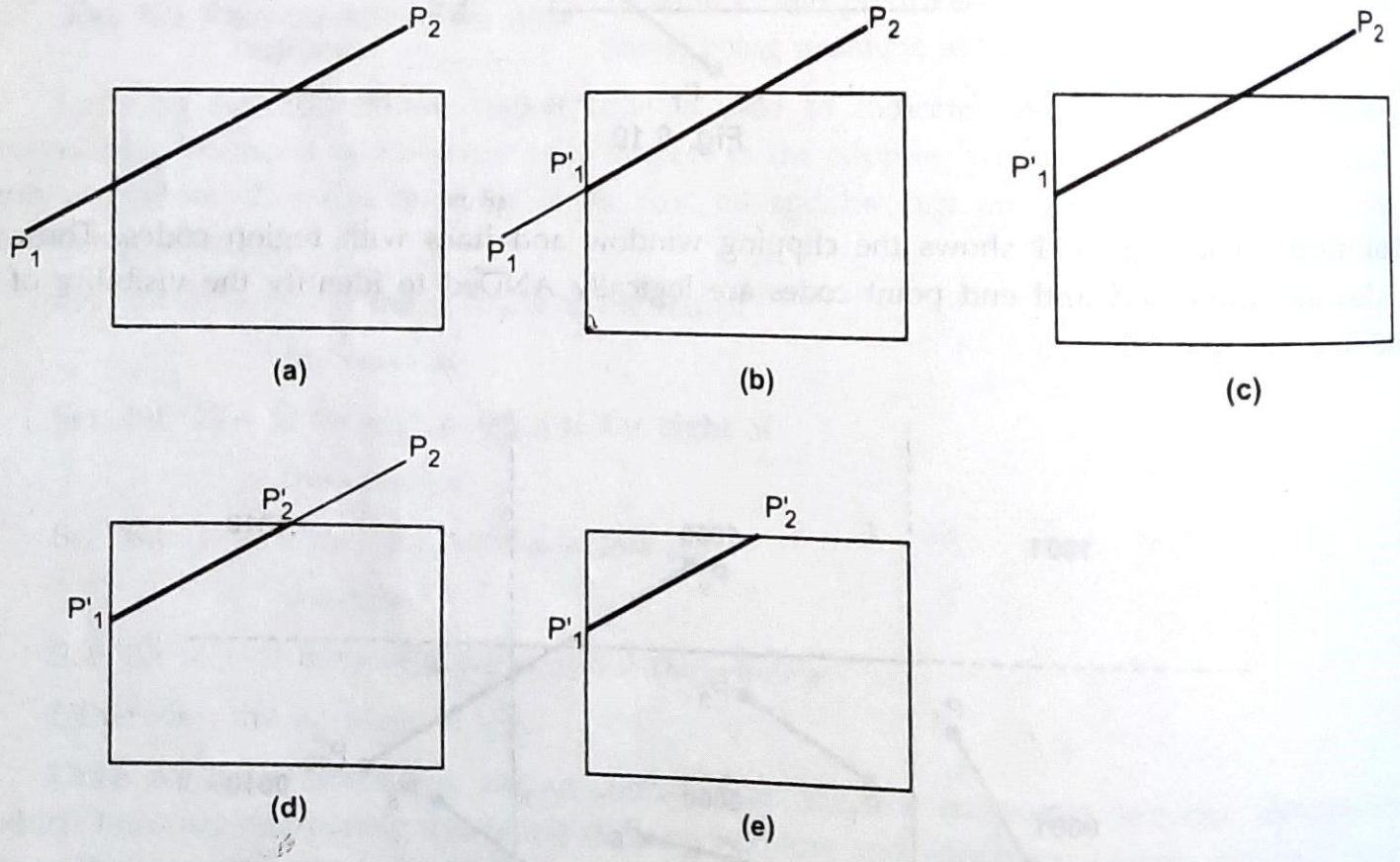


Fig. 6.12 Sutherland-Cohen subdivision line clipping

As shown in the Fig. 6.12, line $P_1 P_2$ is a partially visible and point P_1 is outside the window. Starting with point P_1, the intersection point P_1' is found and we get two line segments $P_1 - P_1'$ and $P_1' - P_2$. We know that, for $P_1 - P_1'$ one end point i.e. P_1 is outside the window and thus the line segment $P_1 - P_1'$ is discarded. The line is now reduced to the section from P_1' to P_2. Since P_2 is outside the clip window, it is checked against the

boundaries and intersection point P_2' is found. Again the line segment is divided into two segments giving $P_1' - P_2'$ and $P_2' - P_2$. We know that, for $P_2' - P_2$ one end point i.e. P_2 is outside the window and thus the line segment $P_2' - P_2$ is discarded. The remaining line segment $P_1' - P_2'$ is completely inside the clipping window and hence made visible.

The intersection points with a clipping boundary can be calculated using the slope-intercept form of the line equation. The equation for line passing through points $P_1 (x_1, y_1)$ and $P_2 (x_2, y_2)$ is

$$y = m(x - x_1) + y_1 \quad \text{or} \quad y = m(x - x_2) + y_2 \qquad \ldots (1)$$

where $$m = \frac{y_2 - y_1}{x_2 - x_1}$$ (slope of the line)

Therefore, the intersections with the clipping boundaries of the window are given as :

Left : $x_L, y = m(x_L - x_1) + y_1 \quad ; \quad m \neq \infty$

Right : $x_R, y = m(x_R - x_1) + y_1 \quad ; \quad m \neq \infty$

Top : $y_T, x = x_1 + \left(\frac{1}{m}\right)(y_T - y_1) \quad ; \quad m \neq 0$

Bottom : $y_B, x = x_1 + \left(\frac{1}{m}\right)(y_B - y_1) \quad ; \quad m \neq 0$

Sutherland and Cohen subdivision line clipping algorithm :

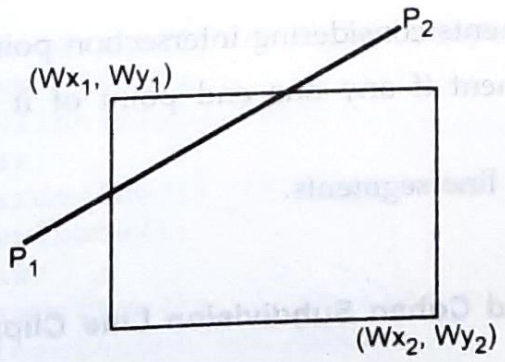


Fig. 6.13

1. Read two end points of the line say $P_1 (x_1, y_1)$ and $P_2 (x_2, y_2)$.
2. Read two corners (left-top and right-bottom) of the window, say (Wx_1, Wy_1 and Wx_2, Wy_2).

3. Assign the region codes for two endpoints P_1 and P_2 using following steps :

   Initialize code with bits 0000

   Set Bit 1 – if $(x < Wx_1)$

   Set Bit 2 – if $(x > Wx_2)$

   Set Bit 3 – if $(y < Wy_2)$

   Set Bit 4 – if $(y > Wy_1)$

4. Check for visibility of line P_1 P_2

   a) If region codes for both endpoints P_1 and P_2 are zero then the line is completely visible. Hence draw the line and go to step 9.

   b) If region codes for endpoints are not zero and the logical ANDing of them is also nonzero then the line is completely invisible, so reject the line and go to step 9.

   c) If region codes for two endpoints do not satisfy the conditions in (4a and 4b)the line is partially visible.

5. Determine the intersecting edge of the clipping window by inspecting the region codes of two endpoints.

   a) If region codes for both the end points are non-zero, find intersection points P_1' and P_2' with boundary edges of clipping window with respect to point P_1 and point P_2, respectively

   b) If region code for any one end point is non zero then find intersection point P_1' or P_2' with the boundary edge of the clipping window with respect to it.

6. Divide the line segments considering intersection points.

7. Reject the line segment if any one end point of it appears outsides the clipping window.

8. Draw the remaining line segments.

9. Stop.

'C' code for Sutherland and Cohen Subdivision Line Clipping Algorithm

(Softcopy of this program is available at vtubooks.com)

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<dos.h>
#include<math.h>
#include<graphics.h>
/* Defining structure for end point of line */
typedef struct co-ordinate
```

BCA Third Year Proposed Syllabus
w.e.f. 2021-2022

BCA III Year SEM - V			
Course category	**Subject Code**	**Subjects**	**Credits**
Core Course	BCA-501	System Analysis and Design (SAAD)	04
Core Course	BCA-502	Web Development and PHP Programming	04
Core Course	BCA-503	Mobile Application Development	04
Chose any one from the below Elective courses			
Elective Subject	BCA-504-A	Computer Network	04
	BCA-504-B	Cloud Computing	04
Chose any one Open Elective courses			
Open Elective	BCA-505 A	University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental courses OR	04
	BCA-505-B	Linux Operating System	04
Lab / Practical	BCA-506	Lab- PHP Programming	02
	BCA-507	Lab- Mobile Application Development	02
		Total Credit	**24**
BCA III Year SEM - VI			
	Subject Code	**Subjects**	**Credits**
Core Course	BCA-601	Software Engineering	04
Core Course	BCA-602	Python	04
Core Course	BCA-603	Project Development activity and Seminar	04
Chose any one from the below Elective courses			
Elective Subject	BCA-604-A	Windows Programming	04
	BCA-604-B	Digital Image Processing	04
Chose any one Open Elective courses			
Open Elective	BCA-605 A	University recognized MOOC (NPTEL / SWAYAM / others) OR Intra / Inter Departmental courses OR	04
	BCA-605-B	Cyber Security	
Lab / Practical	BCA-606	Lab- Python	02
	BCA-607	Lab- Windows Programming and DIP	02
		Total Credit	**24**

Name of Course	**Bachelor of Computer Application (BCA)**
Semester	**V**
Name of Subject	**System Analysis and Design(SAAD)**
Subject Code	**BCA- 501**
Marks	**75 Marks**
Lectures	**50 Lectures**

Objectives:

Develop and evaluate system requirements. Work effectively in a team environment as well as explain the need for and value of a formalized step-by-step approach to the analysis, design, and implementation of computer information systems. Use tools and techniques for process and data modeling.

Outcomes:

Upon successful completion of this course, you will be able to gather data to analyze and specify the requirements of a system. Design system components and environments & build general and detailed models that assist programmers in implementing a system. It also design a database for storing data, a user interface for data input and output, and controls to protect the system and its data.

Unit - I

1. Introduction of System Concept & System Development Life Cycle **10**

- 1.1 System Concept: Definition, Characteristics
- 1.2 Elements of system, Physical and abstract system
- 1.3 Open and closed system, man-made information systems
- 1.4 System Development Life Cycle: Various phases of system development
- 1.5 Considerations for system planning and control for system success
- 1.6 Role of system analyst

Unit - II

2. System Planning, Feasibility study & Cost-Benefit Analysis **10**

- 2.1 Basis for planning in system analysis: Dimensions of Planning
- 2.2 Initial Investigation: Determining user's requirements and analysis
- 2.3 fact finding process and techniques
- 2.4 Feasibility study: Technical
- 2.5 Operational & Economic Feasibilities
- 2.6 Cost/Benefit Analysis

Unit - III

3. Tools of structured Analysis **15**

- 3.1 Data Flow diagram
- 3.2 Data dictionary
- 3.3 IPO charts
- 3.4 HIPO charts
- 3.5 Gantt charts
- 3.6 Pseudo codes
- 3.7 Flow charts
- 3.8 Decision tree
- 3.9 Decision tables

Unit - IV

4. Form and database design **15**

4.1 Input/ Output and Form Design
4.2 Introduction to files and database design
4.3 File structures and organization
4.4 Objectives of database design
4.5 Logical and physical view of data

References:

1. Igor Hawryszkiewyez, “Introduction to System Analysis and Design”, 4th edition, Prentice-Hall
2. Jeffrey L. Whitten, and Lonnie D. Bentey, “Systems analysis and Design Methods”, 4th edition, Tata McGraw-Hill
3. Mark Lejk, and David Deeks, “An Introduction to System Analysis Techniques”, Prentice Hall.
4. Don Yeates, Maura Shields and David Helmy, “System Analysis and Design”, Longman group limited, 1994

System Analysis and Design - Quick Guide

System Analysis and Design - Overview

Systems development is systematic process which includes phases such as planning, analysis, design, deployment, and maintenance. Here, in this tutorial, we will primarily focus on −

- Systems analysis
- Systems design

Systems Analysis

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

Analysis specifies **what the system should do**.

Systems Design

It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently.

System Design focuses on **how to accomplish the objective of the system**.

System Analysis and Design (SAD) mainly focuses on −

- Systems
- Processes
- Technology

What is a System?

The word System is derived from Greek word Systema, which means an organized relationship between any set of components to achieve some common cause or objective.

A system is "an orderly grouping of interdependent components linked together according to a plan to achieve a specific goal."

Constraints of a System

A system must have three basic constraints −

- A system must have some **structure and behavior** which is designed to achieve a predefined objective.
- **Interconnectivity** and **interdependence** must exist among the system components.
- The **objectives of the organization** have a **higher priority** than the objectives of its subsystems.

For example, traffic management system, payroll system, automatic library system, human resources information system.

Properties of a System

A system has the following properties −

Organization

Organization implies structure and order. It is the arrangement of components that helps to achieve predetermined objectives.

Interaction

It is defined by the manner in which the components operate with each other.

For example, in an organization, purchasing department must interact with production department and payroll with personnel department.

Interdependence

Interdependence means how the components of a system depend on one another. For proper functioning, the components are coordinated and linked together according to a specified plan. The output of one subsystem is the required by other subsystem as input.

Integration

Integration is concerned with how a system components are connected together. It means that the parts of the system work together within the system even if each part performs a unique function.

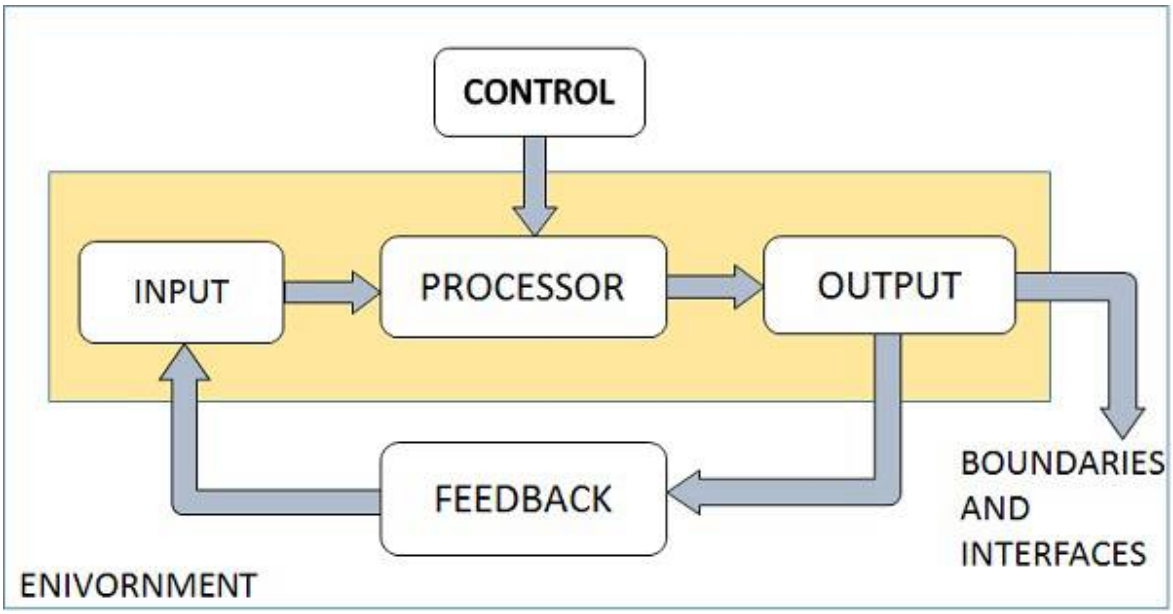
Central Objective

The objective of system must be central. It may be real or stated. It is not uncommon for an organization to state an objective and operate to achieve another.

The users must know the main objective of a computer application early in the analysis for a successful design and conversion.

Elements of a System

The following diagram shows the elements of a system −



Outputs and Inputs

- The main aim of a system is to produce an output which is useful for its user.
- Inputs are the information that enters into the system for processing.
- Output is the outcome of processing.

Processor(s)

- The processor is the element of a system that involves the actual transformation of input into output.
- It is the operational component of a system. Processors may modify the input either totally or partially, depending on the output specification.
- As the output specifications change, so does the processing. In some cases, input is also modified to enable the processor for handling the transformation.

Control

- The control element guides the system.
- It is the decision–making subsystem that controls the pattern of activities governing input, processing, and output.
- The behavior of a computer System is controlled by the Operating System and software. In order to keep system in balance, what and how much input is needed is determined by Output Specifications.

Feedback

- Feedback provides the control in a dynamic system.
- Positive feedback is routine in nature that encourages the performance of the system.
- Negative feedback is informational in nature that provides the controller with information for action.

Environment

- The environment is the “supersystem” within which an organization operates.
- It is the source of external elements that strike on the system.
- It determines how a system must function. For example, vendors and competitors of organization’s environment, may provide constraints that affect the actual performance of the business.

Boundaries and Interface

- A system should be defined by its boundaries. Boundaries are the limits that identify its components, processes, and interrelationship when it interfaces with another system.
- Each system has boundaries that determine its sphere of influence and control.
- The knowledge of the boundaries of a given system is crucial in determining the nature of its interface with other systems for successful design.

Types of Systems

The systems can be divided into the following types −

Physical or Abstract Systems

- Physical systems are tangible entities. We can touch and feel them.
- Physical System may be static or dynamic in nature. For example, desks and chairs are the physical parts of computer center which are static. A programmed computer is a dynamic system in which programs, data, and applications can change according to the user's needs.
- Abstract systems are non-physical entities or conceptual that may be formulas, representation or model of a real system.

Open or Closed Systems

- An open system must interact with its environment. It receives inputs from and delivers outputs to the outside of the system. For example, an information system which must adapt to the changing environmental conditions.
- A closed system does not interact with its environment. It is isolated from environmental influences. A completely closed system is rare in reality.

Adaptive and Non Adaptive System

- Adaptive System responds to the change in the environment in a way to improve their performance and to survive. For example, human beings, animals.
- Non Adaptive System is the system which does not respond to the environment. For example, machines.

Permanent or Temporary System

- Permanent System persists for long time. For example, business policies.
- Temporary System is made for specified time and after that they are demolished. For example, A DJ system is set up for a program and it is dissembled after the program.

Natural and Manufactured System

- Natural systems are created by the nature. For example, Solar system, seasonal system.
- Manufactured System is the man-made system. For example, Rockets, dams, trains.

Deterministic or Probabilistic System

- Deterministic system operates in a predictable manner and the interaction between system components is known with certainty. For example, two molecules of hydrogen and one molecule of oxygen makes water.
- Probabilistic System shows uncertain behavior. The exact output is not known. For example, Weather forecasting, mail delivery.

Social, Human-Machine, Machine System

- Social System is made up of people. For example, social clubs, societies.
- In Human-Machine System, both human and machines are involved to perform a particular task. For example, Computer programming.
- Machine System is where human interference is neglected. All the tasks are performed by the machine. For example, an autonomous robot.

Man–Made Information Systems

- It is an interconnected set of information resources to manage data for particular organization, under Direct Management Control (DMC).
- This system includes hardware, software, communication, data, and application for producing information according to the need of an organization.

  Man-made information systems are divided into three types −
- **Formal Information System** − It is based on the flow of information in the form of memos, instructions, etc., from top level to lower levels of management.
- **Informal Information System** − This is employee based system which solves the day to day work related problems.
- **Computer Based System** − This system is directly dependent on the computer for managing business applications. For example, automatic library system, railway reservation system, banking system, etc.

Systems Models

Schematic Models

- A schematic model is a 2-D chart that shows system elements and their linkages.
- Different arrows are used to show information flow, material flow, and information feedback.

Flow System Models

- A flow system model shows the orderly flow of the material, energy, and information that hold the system together.
- Program Evaluation and Review Technique (PERT), for example, is used to abstract a real world system in model form.

Static System Models

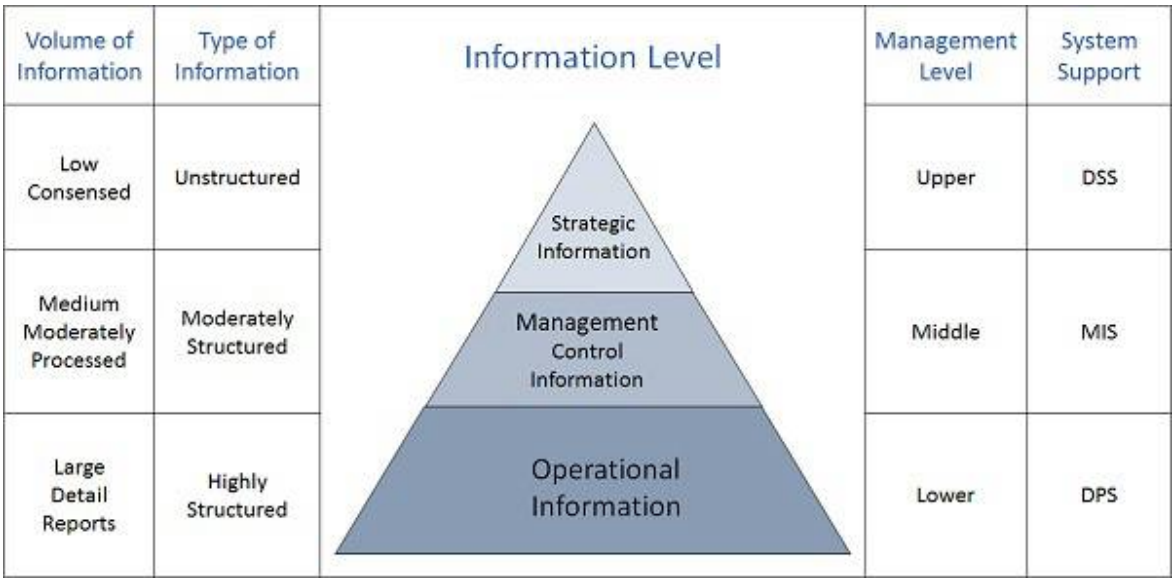
- They represent one pair of relationships such as *activity–time* or *cost–quantity*.
- The Gantt chart, for example, gives a static picture of an activity-time relationship.

Dynamic System Models

- Business organizations are dynamic systems. A dynamic model approximates the type of organization or application that analysts deal with.
- It shows an ongoing, constantly changing status of the system. It consists of −
  - Inputs that enter the system
  - The processor through which transformation takes place
  - The program(s) required for processing
  - The output(s) that result from processing.

Categories of Information

There are three categories of information related to managerial levels and the decision managers make.



Strategic Information

- This information is required by topmost management for long range planning policies for next few years. For example, trends in revenues, financial investment, and human resources, and population growth.
- This type of information is achieved with the aid of Decision Support System (DSS).

Managerial Information

- This type of Information is required by middle management for short and intermediate range planning which is in terms of months. For example, sales analysis, cash flow projection, and annual financial statements.
- It is achieved with the aid of Management Information Systems (MIS).

Operational information

- This type of information is required by low management for daily and short term planning to enforce day-to-day operational activities. For example, keeping employee attendance records, overdue purchase orders, and current stocks available.
- It is achieved with the aid of Data Processing Systems (DPS).

System Development Life Cycle

An effective System Development Life Cycle (SDLC) should result in a high quality system that meets customer expectations, reaches completion within time and cost evaluations, and works effectively and efficiently in the current and planned Information Technology infrastructure.

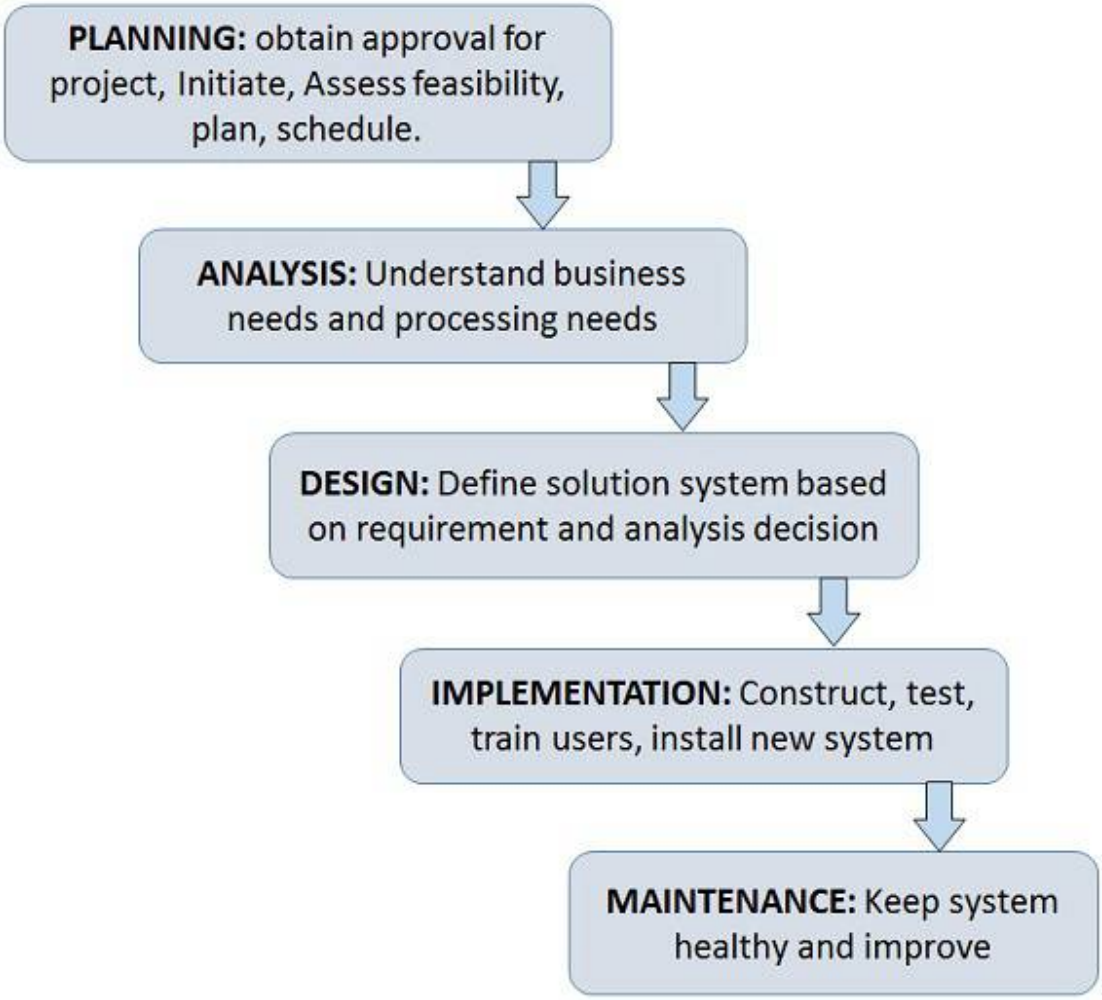
System Development Life Cycle (SDLC) is a conceptual model which includes policies and procedures for developing or altering systems throughout their life cycles.

SDLC is used by analysts to develop an information system. SDLC includes the following activities −

- requirements
- design
- implementation
- testing
- deployment
- operations
- maintenance

Phases of SDLC

Systems Development Life Cycle is a systematic approach which explicitly breaks down the work into phases that are required to implement either new or modified Information System.



Feasibility Study or Planning

- Define the problem and scope of existing system.
- Overview the new system and determine its objectives.
- Confirm project feasibility and produce the project Schedule.
- During this phase, threats, constraints, integration and security of system are also considered.
- A feasibility report for the entire project is created at the end of this phase.

Analysis and Specification

- Gather, analyze, and validate the information.
- Define the requirements and prototypes for new system.
- Evaluate the alternatives and prioritize the requirements.
- Examine the information needs of end-user and enhances the system goal.
- A Software Requirement Specification (SRS) document, which specifies the software, hardware, functional, and network requirements of the system is prepared at the end of this phase.

System Design

- Includes the design of application, network, databases, user interfaces, and system interfaces.
- Transform the SRS document into logical structure, which contains detailed and complete set of specifications that can be implemented in a programming language.
- Create a contingency, training, maintenance, and operation plan.
- Review the proposed design. Ensure that the final design must meet the requirements stated in SRS document.
- Finally, prepare a design document which will be used during next phases.

Implementation

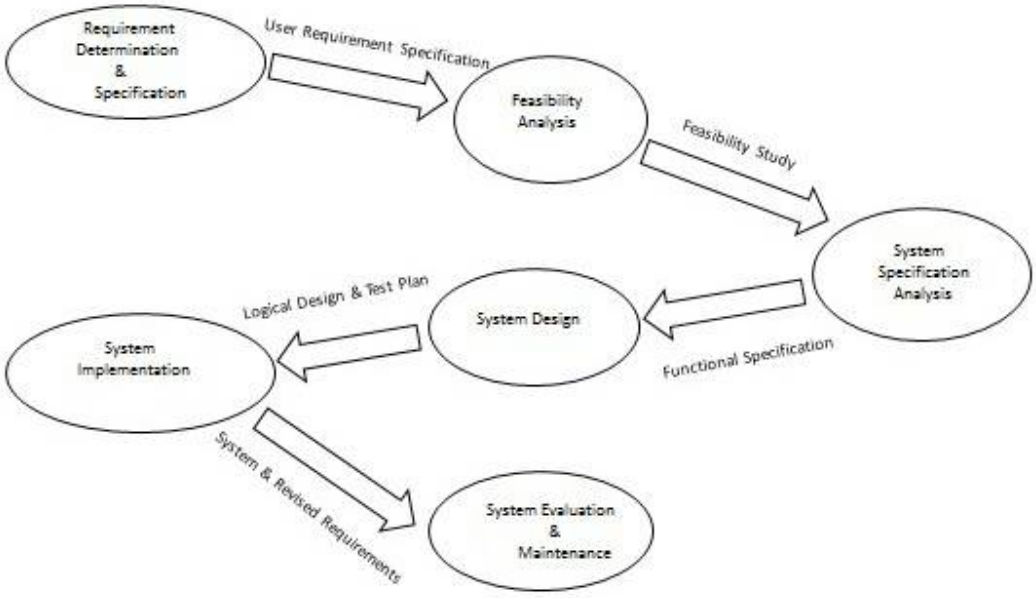
- Implement the design into source code through coding.
- Combine all the modules together into training environment that detects errors and defects.
- A test report which contains errors is prepared through test plan that includes test related tasks such as test case generation, testing criteria, and resource allocation for testing.
- Integrate the information system into its environment and install the new system.

Maintenance/Support

- Include all the activities such as phone support or physical on-site support for users that is required once the system is installing.
- Implement the changes that software might undergo over a period of time, or implement any new requirements after the software is deployed at the customer location.
- It also includes handling the residual errors and resolve any issues that may exist in the system even after the testing phase.
- Maintenance and support may be needed for a longer time for large systems and for a short time for smaller systems.

Life Cycle of System Analysis and Design

The following diagram shows the complete life cycle of the system during analysis and design phase.



Role of System Analyst

The system analyst is a person who is thoroughly aware of the system and guides the system development project by giving proper directions. He is an expert having technical and interpersonal skills to carry out development tasks required at each phase.

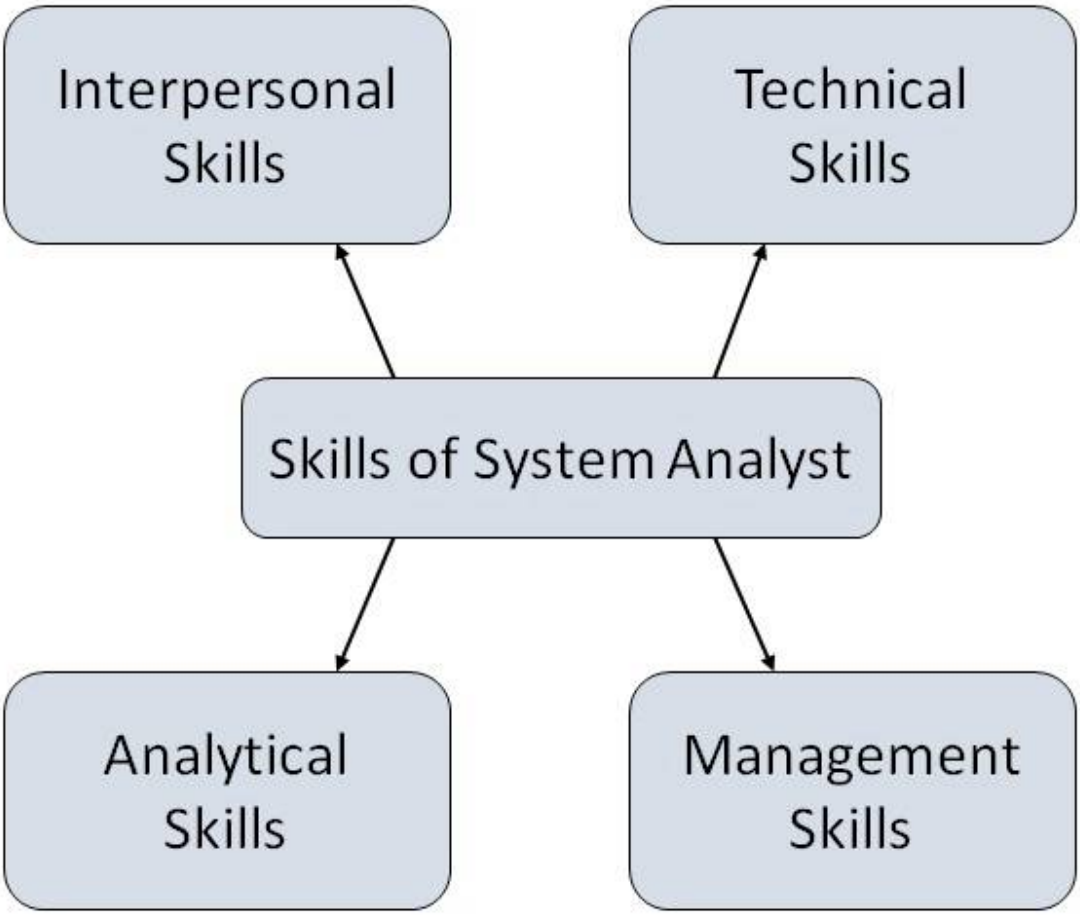
He pursues to match the objectives of information system with the organization goal.

Main Roles

- Defining and understanding the requirement of user through various Fact finding techniques.
- Prioritizing the requirements by obtaining user consensus.
- Gathering the facts or information and acquires the opinions of users.
- Maintains analysis and evaluation to arrive at appropriate system which is more user friendly.
- Suggests many flexible alternative solutions, pick the best solution, and quantify cost and benefits.
- Draw certain specifications which are easily understood by users and programmer in precise and detailed form.
- Implemented the logical design of system which must be modular.
- Plan the periodicity for evaluation after it has been used for some time, and modify the system as needed.

Attributes of a Systems Analyst

The following figure shows the attributes a systems analyst should possess −



Interpersonal Skills

- Interface with users and programmer.
- Facilitate groups and lead smaller teams.
- Managing expectations.
- Good understanding, communication, selling and teaching abilities.
- Motivator having the confidence to solve queries.

Analytical Skills

- System study and organizational knowledge
- Problem identification, problem analysis, and problem solving
- Sound commonsense
- Ability to access trade-off
- Curiosity to learn about new organization

Management Skills

- Understand users jargon and practices.
- Resource & project management.
- Change & risk management.
- Understand the management functions thoroughly.

Technical Skills

- Knowledge of computers and software.
- Keep abreast of modern development.
- Know of system design tools.
- Breadth knowledge about new technologies.

System Analysis & Design - System Planning

What is Requirements Determination?

A requirement is a vital feature of a new system which may include processing or capturing of data, controlling the activities of business, producing information and supporting the management.

Requirements determination involves studying the existing system and gathering details to find out what are the requirements, how it works, and where improvements should be made.

Major Activities in requirement Determination

Requirements Anticipation

- It predicts the characteristics of system based on previous experience which include certain problems or features and requirements for a new system.
- It can lead to analysis of areas that would otherwise go unnoticed by inexperienced analyst. But if shortcuts are taken and bias is introduced in conducting the investigation, then requirement Anticipation can be half-baked.

Requirements Investigation

- It is studying the current system and documenting its features for further analysis.
- It is at the heart of system analysis where analyst documenting and describing system features using fact-finding techniques, prototyping, and computer assisted tools.

Requirements Specifications

- It includes the analysis of data which determine the requirement specification, description of features for new system, and specifying what information requirements will be provided.
- It includes analysis of factual data, identification of essential requirements, and selection of Requirement-fulfillment strategies.

Information Gathering Techniques

The main aim of fact finding techniques is to determine the information requirements of an organization used by analysts to prepare a precise SRS understood by user.

Ideal SRS Document should −

Unit - II

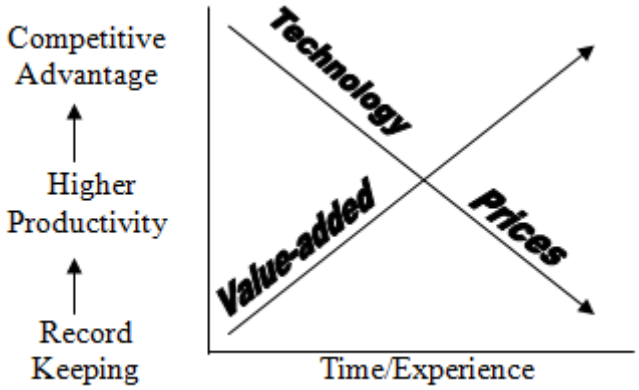
System Planning, Feasibility study & Cost-Benefit Analysis

Dimensions of Planning:

The following conditions dictate today's business strategies:

- High interest rates make it more important that business realizes a good return on investment
- Inflation puts pressure on profit when it occurs.
- The growing trend toward guaranteed employment suggests that costs are becoming fixed and the commitment to business expansion may not be easily changed.
- Resource shortages impede expansion.
- Regulatory constraints slow entry into the market.
- Increased productivity paves the way for expansion.

Information systems embedded in an organization provide users with the opportunity to add value to products and business operations at lower costs as shown in figure. Therefore they must be carefully planned.



What is Requirements Determination?

A requirement is a vital feature of a new system which may include processing or capturing of data, controlling the activities of business, producing information and supporting the management.

Requirements determination involves studying the existing system and gathering details to find out what are the requirements, how it works, and where improvements should be made.

Major Activities in requirement Determination

Requirements Anticipation

- It predicts the characteristics of system based on previous experience which include certain problems or features and requirements for a new system.

- It can lead to analysis of areas that would otherwise go unnoticed by inexperienced analyst. But if shortcuts are taken and bias is introduced in conducting the investigation, then requirement Anticipation can be half-baked.

Requirements Investigation

- It is studying the current system and documenting its features for further analysis.
- It is at the heart of system analysis where analyst documenting and describing system features using fact-finding techniques, prototyping, and computer assisted tools.

Requirements Specifications

- It includes the analysis of data which determine the requirement specification, description of features for new system, and specifying what information requirements will be provided.
- It includes analysis of factual data, identification of essential requirements, and selection of Requirement-fulfillment strategies.

Information Gathering Techniques

The main aim of fact finding techniques is to determine the information requirements of an organization used by analysts to prepare a precise SRS understood by user.

Ideal SRS Document should −

- be complete, Unambiguous, and Jargon-free.
- specify operational, tactical, and strategic information requirements.
- solve possible disputes between users and analyst.
- use graphical aids which simplify understanding and design.

There are various information gathering techniques −

Interviewing

Systems analyst collects information from individuals or groups by interviewing. The analyst can be formal, legalistic, play politics, or be informal; as the success of an interview depends on the skill of analyst as interviewer.

It can be done in two ways −

- **Unstructured Interview** − The system analyst conducts question-answer session to acquire basic information of the system.
- **Structured Interview** − It has standard questions which user need to respond in either close (objective) or open (descriptive) format.

Advantages of Interviewing

- This method is frequently the best source of gathering qualitative information.
- It is useful for them, who do not communicate effectively in writing or who may not have the time to complete questionnaire.
- Information can easily be validated and cross checked immediately.
- It can handle the complex subjects.
- It is easy to discover key problem by seeking opinions.
- It bridges the gaps in the areas of misunderstandings and minimizes future problems.

Questionnaires

This method is used by analyst to gather information about various issues of system from large number of persons.

There are two types of questionnaires −

- **Open-ended Questionnaires** − It consists of questions that can be easily and correctly interpreted. They can explore a problem and lead to a specific direction of answer.

- **Closed-ended Questionnaires** − It consists of questions that are used when the systems analyst effectively lists all possible responses, which are mutually exclusive.

Advantages of questionnaires

- It is very effective in surveying interests, attitudes, feelings, and beliefs of users which are not co-located.
- It is useful in situation to know what proportion of a given group approves or disapproves of a particular feature of the proposed system.
- It is useful to determine the overall opinion before giving any specific direction to the system project.
- It is more reliable and provides high confidentiality of honest responses.
- It is appropriate for electing factual information and for statistical data collection which can be emailed and sent by post.

Review of Records, Procedures, and Forms

Review of existing records, procedures, and forms helps to seek insight into a system which describes the current system capabilities, its operations, or activities.

Advantages

- It helps user to gain some knowledge about the organization or operations by themselves before they impose upon others.
- It helps in documenting current operations within short span of time as the procedure manuals and forms describe the format and functions of present system.
- It can provide a clear understanding about the transactions that are handled in the organization, identifying input for processing, and evaluating performance.
- It can help an analyst to understand the system in terms of the operations that must be supported.
- It describes the problem, its affected parts, and the proposed solution.

Observation

This is a method of gathering information by noticing and observing the people, events, and objects. The analyst visits the organization to observe the working of current system and understands the requirements of the system.

Advantages

- It is a direct method for gleaning information.
- It is useful in situation where authenticity of data collected is in question or when complexity of certain aspects of system prevents clear explanation by end-users.
- It produces more accurate and reliable data.
- It produces all the aspect of documentation that are incomplete and outdated.

Joint Application Development (JAD)

It is a new technique developed by IBM which brings owners, users, analysts, designers, and builders to define and design the system using organized and intensive workshops. JAD trained analyst act as facilitator for workshop who has some specialized skills.

Advantages of JAD

- It saves time and cost by replacing months of traditional interviews and follow-up meetings.
- It is useful in organizational culture which supports joint problem solving.
- Fosters formal relationships among multiple levels of employees.
- It can lead to development of design creatively.
- It Allows rapid development and improves ownership of information system.

Secondary Research or Background Reading

This method is widely used for information gathering by accessing the gleaned information. It includes any previously gathered information used by the marketer from any internal or external source.

Advantages

- It is more openly accessed with the availability of internet.
- It provides valuable information with low cost and time.
- It act as forerunner to primary research and aligns the focus of primary research.
- It is used by the researcher to conclude if the research is worth it as it is available with procedures used and issues in collecting them.

Feasibility Study

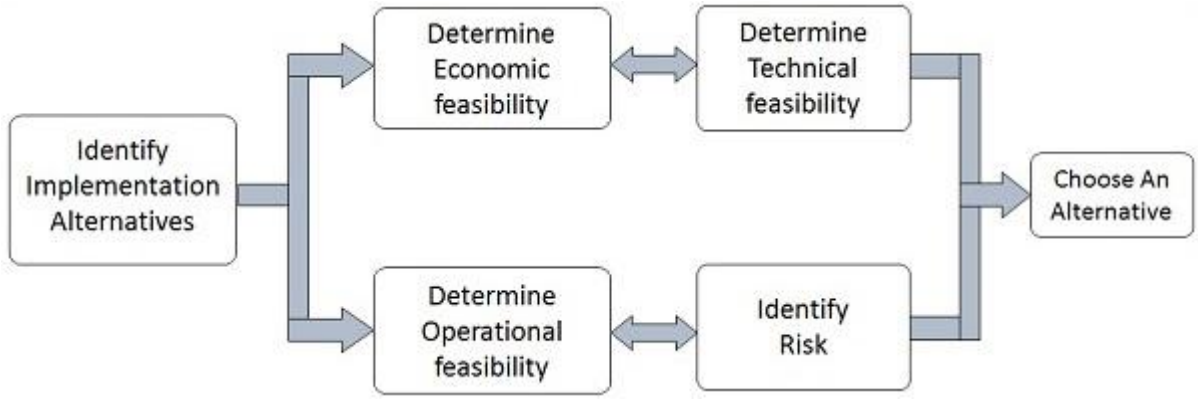
Feasibility Study can be considered as preliminary investigation that helps the management to take decision about whether study of system should be feasible for development or not.

- It identifies the possibility of improving an existing system, developing a new system, and produce refined estimates for further development of system.
- It is used to obtain the outline of the problem and decide whether feasible or appropriate solution exists or not.
- The main objective of a feasibility study is to acquire problem scope instead of solving the problem.
- The output of a feasibility study is a formal system proposal act as decision document which includes the complete nature and scope of the proposed system.

Steps Involved in Feasibility Analysis

The following steps are to be followed while performing feasibility analysis −

- Form a project team and appoint a project leader.
- Develop system flowcharts.
- Identify the deficiencies of current system and set goals.
- Enumerate the alternative solution or potential candidate system to meet goals.
- Determine the feasibility of each alternative such as technical feasibility, operational feasibility, etc.
- Weight the performance and cost effectiveness of each candidate system.
- Rank the other alternatives and select the best candidate system.
- Prepare a system proposal of final project directive to management for approval.



Types of Feasibilities

Economic Feasibility

- It is evaluating the effectiveness of candidate system by using cost/benefit analysis method.

- It demonstrates the net benefit from the candidate system in terms of benefits and costs to the organization.
- The main aim of Economic Feasibility Analysis (EFS) is to estimate the economic requirements of candidate system before investments funds are committed to proposal.
- It prefers the alternative which will maximize the net worth of organization by earliest and highest return of funds along with lowest level of risk involved in developing the candidate system.

Technical Feasibility

- It investigates the technical feasibility of each implementation alternative.
- It analyzes and determines whether the solution can be supported by existing technology or not.
- The analyst determines whether current technical resources be upgraded or added it that fulfill the new requirements.
- It ensures that the candidate system provides appropriate responses to what extent it can support the technical enhancement.

Operational Feasibility

- It determines whether the system is operating effectively once it is developed and implemented.
- It ensures that the management should support the proposed system and its working feasible in the current organizational environment.
- It analyzes whether the users will be affected and they accept the modified or new business methods that affect the possible system benefits.
- It also ensures that the computer resources and network architecture of candidate system are workable.

Behavioral Feasibility

- It evaluates and estimates the user attitude or behavior towards the development of new system.
- It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employee's job status on new ways of conducting business.

Schedule Feasibility

- It ensures that the project should be completed within given time constraint or schedule.
- It also verifies and validates whether the deadlines of project are reasonable or not.

Cost/Benefit Analysis

Cost Benefit analysis is thing that everyone must do so as to think of a powerful or an efficient system. But while thinking out on cost and benefit analysis, we also need to find out factors that really affect benefits and costs of system. In developing cost estimates for a system, we need to consider some of cost elements. Some elements among them are hardware, personnel, facility, operating and supply cost. The following are the cost factors :

1. **Hardware cost –**
   Hardware cost includes actual purchase and peripherals (external devices) that are connected to computer. For example, printer, disk drive etc. Actually, finding actual cost of hardware is generally more difficult especially, when system is shared by various users so as to compared to a system which dedicated stand alone . In some case, best way is to treat it as operating cost.
2. **Personnel costs –**
   Personnel costs includes EDP staff salaries and benefits as well as pay for those who are

involved in process of development of system. Cost occurred during development of system which are one time costs and are also called development cost. Once system is installed, cost of operating and maintaining system becomes recurring cost that one has to pay very frequently based on requirement.

3. **Facility cost –**
   Facility cost is amount of money that is spent in preparation of a site that is physical where application or computer will be in operation. This includes wiring, flooring, lighting and air conditioning. These costs are treated as one- time costs and are included into overall cost estimate of candidate system.
4. **Operating costs –**
   These includes all costs associated with day-to-day(everyday) operation of system and amount depends on number of shifts, nature of applications. There are various ways of covering operating costs. One approach is to treat operating costs as an overhead. Another approach is to charge money from each authorized user for amount of processing they require from system. Amount charged is based on computer time or time they spend on system, staff time ad volume of output produced.
5. **Supply costs –**
   Supply cost are variable costs that increase with increased use of paper, disks and like. They should be estimated and included in overall cost of system.

A system is also expected to provide health benefits. First task is to identify each benefit and then assign some value to it for purpose of cost/ benefit analysis. Benefits may be tangible and intangible, direct or indirect.

Two major benefits are improving performance and minimizing cost of processing of system. The performance category emphasizes improvement in accuracy of or access to information and easier access to system by authorized users. Minimizing costs through an efficient system – error control or reduction of staff- is a benefit that should be measured and included in cost/benefit analysis. The determination of costs and benefit entails following steps :

1. Identify the costs and benefits pertaining to given project.
2. Categorize the various costs and benefits for analysis.
3. Select a method of evaluation.
4. Interpret the results of the analysis.
5. Take action.

UNIT-III

Tools of Structured Analysis

IPO: (INPUT-PROCESS-OUTPUT)

INPUT:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well designed input forms and screens have following properties −

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding −
    - What are the inputs needed for the system?
    - How end users respond to different elements of forms and screens.

PROCESSOR :

The processor is the element of a system that involves the actual transformation of input into output. It is the operational component of a system. Processor may modify the input totally or partially, depending on the specification of the output. This means that as the output specification change so does the processing. In some cases Input is also modified to enable the processor to handle the transformation.

OUTPUT:

There a number of output medias that provide the computer based output.

1. Printers (Line. Matrix and Daisy Wheel)

2. CRT Screen Display (Monitor, LCD screen (lap top)

3. Plotters (Graph, Drum)

Plotters are used to print large charts, graphs & maps. It is another important output device to get permanent output. These are normally found in constructions companies.

4. Audio Response: Use for instructions.

5. Projectors:

Analyst can select any one or more output devices before selection the analyst should considered the print format & layouts in any case the output preparation is very critical & requires more skills & abilities to meet user requirements. There are some standards for printed output that should be followed.

1. Give each output a specific name or title.
2. Provide a sample of the output layout, print areas & locations of each field.
3. Describe whether each output field is to include zeros, spaces between fields and alphabetic or any other data.
4. Specify the procedure for providing the accuracy of output data.
5. Give page no to each page for easy reference. I know as pagination.

HIPO

HIPO stands for Hierarchical Input Process Output. HIPO diagram assesses the system and facilitates documentation. It organizes the software system's modules into a hierarchy. HIPO diagrams can be used to get a high-level picture of the system's functions. The functions are further divided into sub-functions, using HIPO diagrams demonstrating the system functions.

HIPO diagrams are excellent for documentation. By presenting the system's structure in a graphical format, designers and managers may quickly comprehend it.

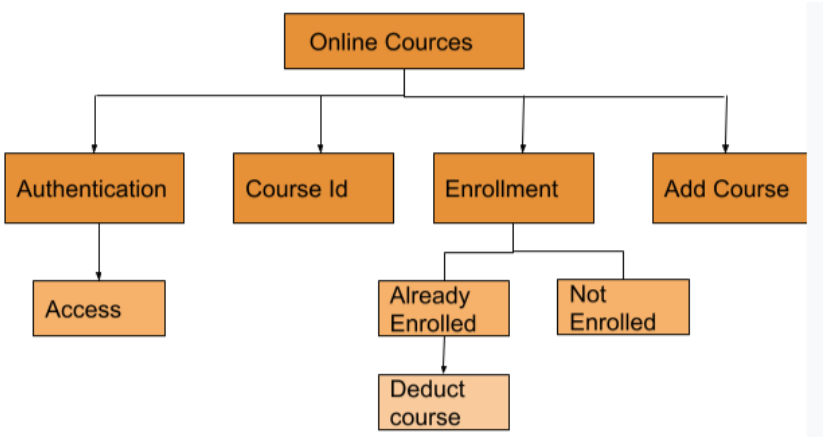
Advantages of HIPO

1. Offer more detail than other types of documentation.
2. Each module's input and output are properly labeled.
3. Can be used to quickly determine what modules call the module under investigation.

Disadvantages of HIPO

1. The biggest downside of HIPO is that the documentation for a programme can become quite large: each module has its page, regardless of its actual size.
2. Structure charts and pseudocode are more extensively utilised than HIPO charts.
3. HIPO does not provide any data flow or control flow information.

Example of HIPO Diagram



Gantt chart-Generalized Activity Normalization Time Table (GANTT) chart

Gantt chart is a type of a bar chart that is used for illustrating project schedules. Gantt charts can be used in any projects that involve effort, resources, milestones and deliveries.

At present, Gantt charts have become the popular choice of project managers in every field.

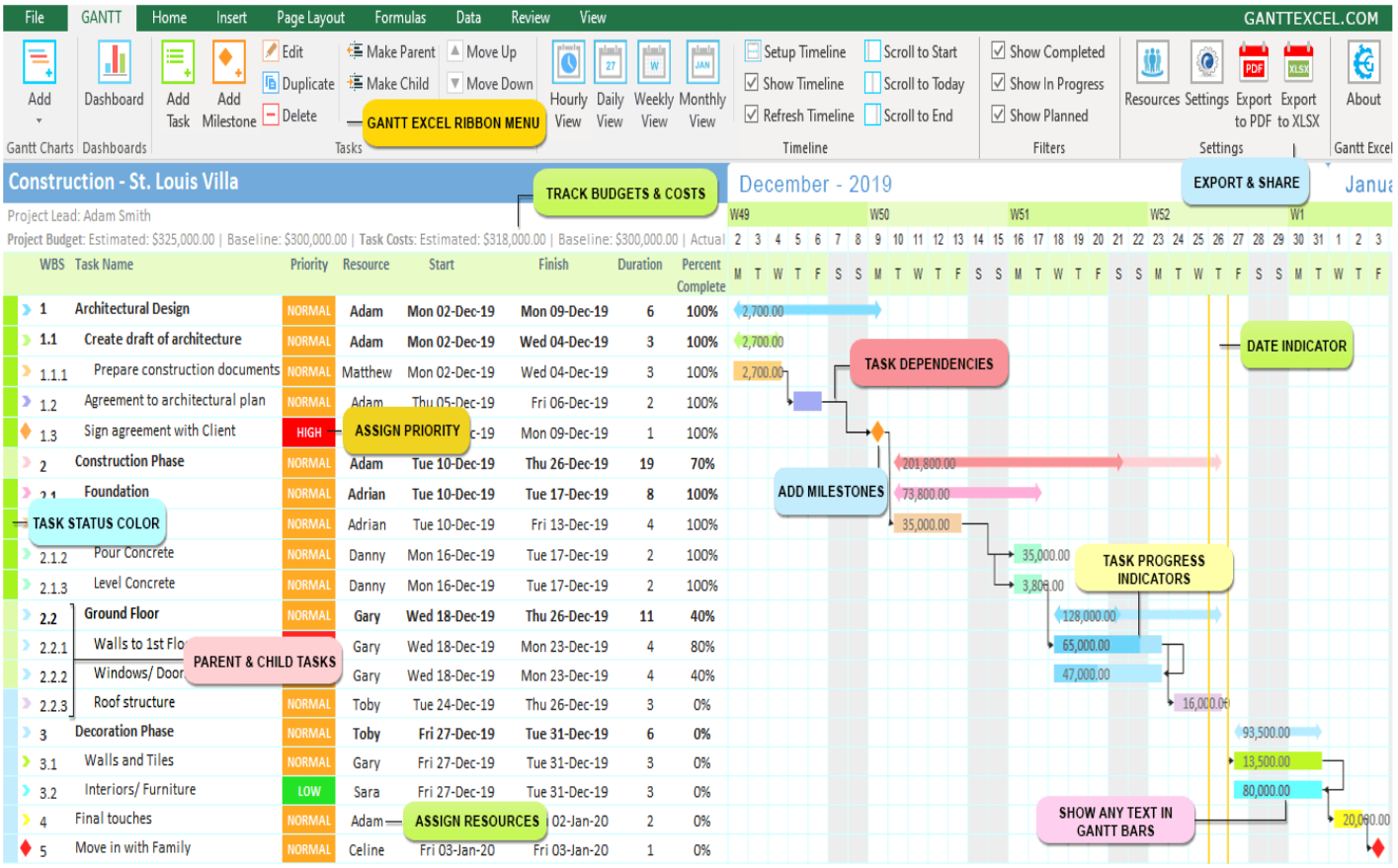
Gantt charts allow project managers to track the progress of the entire project. Through Gantt charts, the project manager can keep a track of the individual tasks as well as of the overall project progression.

In addition to tracking the progression of the tasks, Gantt charts can also be used for tracking the utilization of the resources in the project. These resources can be human resources as well as materials used.

Gantt chart was invented by a mechanical engineer named Henry Gantt in 1910. Since the invention, Gantt chart has come a long way. By today, it takes different forms from simple paper based charts to sophisticated software packages.

Gantt chart represents following things:

- All the tasks are listed at leftmost column.
- The horizontal bars indicate or represent required time by corresponding particular task.
- When occurring of multiple horizontal bars takes place at same time on calendar, then that means concurrency can be applied for performing particular tasks.
- The diamonds indicate milestones.



Gantt charts are one of the many project management tools. They present in one chart all the tasks in a project. They show the order in which they tasks should be done and the time needed to complete them. This is valuable information to have when managing projects for the following reasons:

- **Progress monitoring.** Project managers can see if individual tasks are completed on time and adjust the project schedule. The charts also show which goals were met on schedule, helping managers gauge employee productivity.
- **Project planning.** Managers can set deadlines, milestones and schedules for various project components.
- **Resource management.** Project planners can coordinate resource allocation with the project schedule. Managers can see the amount of time each process takes and designate resources accordingly.

Pseudo code

Pseudo code is a term which is often used in programming and algorithm based fields. It is a methodology that allows the programmer to represent the implementation of an algorithm. Simply, we can say that it's the cooked up representation of an algorithm. Often at times,

algorithms are represented with the help of pseudo codes as they can be interpreted by programmers no matter what their programming background or knowledge is. Pseudo code, as the name suggests, is a false code or a representation of code which can be understood by even a layman with some school level programming knowledge.

Algorithm**:** It's an organized logical sequence of the actions or the approach towards a particular problem. A programmer implements an algorithm to solve a problem. Algorithms are expressed using natural verbal but somewhat technical annotations.

Pseudo code: It's simply an implementation of an algorithm in the form of annotations and informative text written in plain English. It has no syntax like any of the programming language and thus can't be compiled or interpreted by the computer.

Advantages of Pseudocode

- Improves the readability of any approach. It's one of the best approaches to start implementation of an algorithm.
- Acts as a bridge between the program and the algorithm or flowchart. Also works as a rough documentation, so the program of one developer can be understood easily when a pseudo code is written out. In industries, the approach of documentation is essential. And that's where a pseudo-code proves vital.
- The main goal of a pseudo code is to explain what exactly each line of a program should do, hence making the code construction phase easier for the programmer.

Example:

```
This program will allow the user to check
   the number whether it's even or odd.
```

Tools of Structured Analysis

Chapter 3

Structured Analysis

- It is a set of techniques and graphical tools that allow the analyst to develop a new kind of system that is understandable to the user

Why do we use these tools?

- Use graphics whenever possible to help communicate better with the user.
- Differentiate between logical and physical system
- Build a logical system model to familiarize the user with system characteristics and interrelationships before implementation

Data Flow Diagram

- It was first developed by Larry Constantine as a way of expressing system requirements in a graphical form.
- It is also known as Bubble Chart

DFD Symbols

- **Square** defines a source or destination of data.

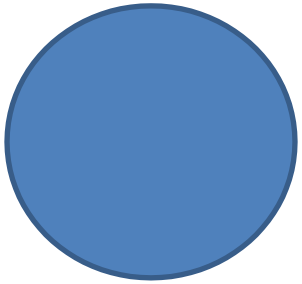


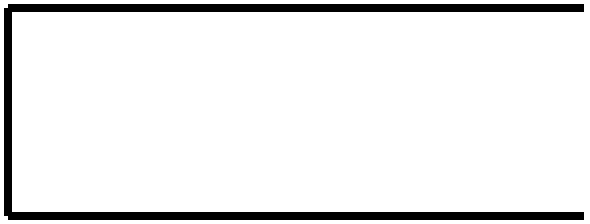
- **Arrow** identifies data flow, means the data in motion. It is a pipeline through which information flows.



Continued…

- **Circle or a bubble** represents a process that transforms incoming data flow into outgoing data





- **Open rectangle** is a data store, or data at rest, or a temporary repository of data

Constructing a DFD

- Processes should be named and numbered for easy reference
- The direction of flow is from top to bottom and from left to right
- Data flow from the source to destination, although they may flow back to a source
- When a process is exploded into lower level details, they are numbered
- The names of data stores, sources, and destinations are written in capital letters. Process and data flow names have the first letter of each word capitalized

Data Dictionary

- A structured place to keep details of the contents of data flows, processes, and data store.
- It is a structured repository of data about data.
- It is a set of definitions of all DFD elements

Advantages of Data Dictionary

- Documentation- it is a valuable reference in any organization.
- It improves analyst/user communication by establishing consistent definitions of various elements, terms and procedures
- It is important step in building a database

Items to be defined in Data Dictionary

- Data Elements- smallest unit of data that provides for no further decomposition.

  For example: date consists of day, month and year

- Data Structure- a group of data elements handled as a unit.

  For example: phone is a data structure consisting of four data elements: area-code-exchange-number-extension.

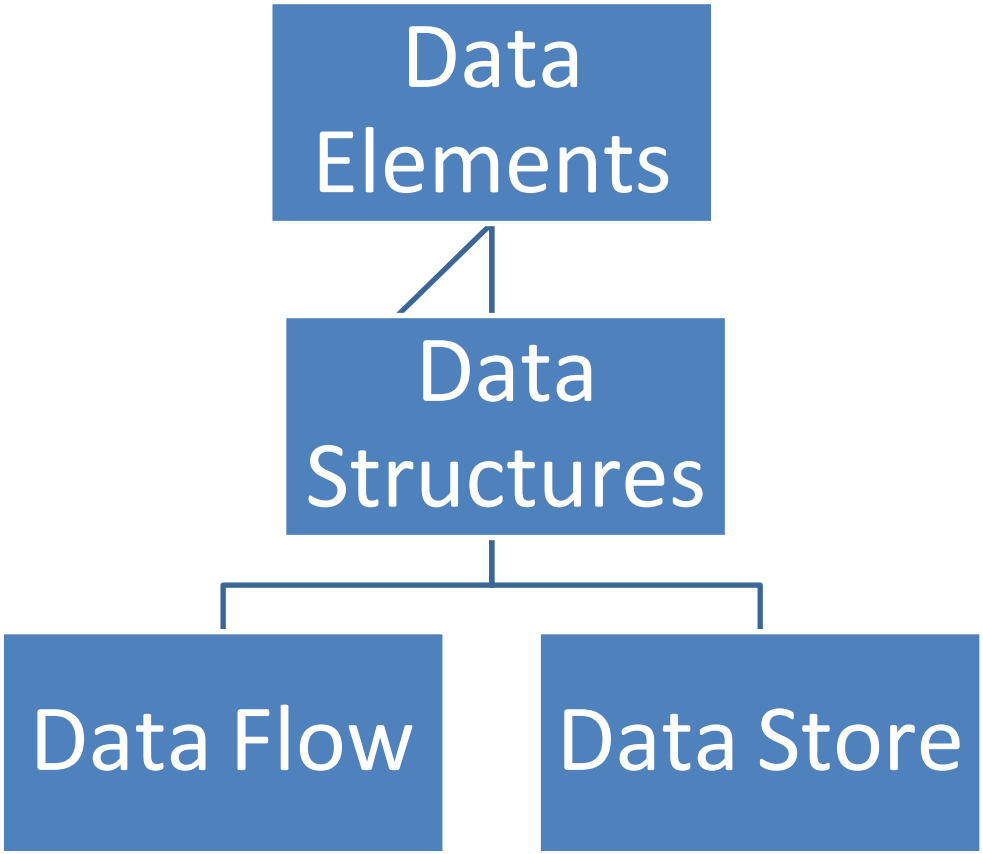
- Data Flows and Data Stores- data flows are data structures in motion, whereas data stores are data structures at rest. A data store is a location where data structures are temporarily located.

Data Dictionary

Smallest Unit of Data

Group of Data Elements

Groups of Data Structures



Data Elements

- For e.g.

  Author Name:

  First

  Middle

  Last

  Alias

  The Description of Data Element should include:

  1. Name
  2. Description &
  3. An Alias (Synonym)

Data Elements

- Whether or not Data Element has the following:
  - A Different Name:
    - For e.g. A Purchase Order may exist as Pur. Order, Purchase Ord., or P.O. We will record all these in Data Dictionary under Definition of Purchase Order.
  - Usage Characteristics
    - Range of Values or Frequency of use or both.
    - 2 types:
      - Value within Range: For e.g. Payroll between 1000 and 10000 = Continuous Value.
      - Specific Value: For e.g. Depts. In a Firm coded 100 (Accounting), 110 (HR), 111 (Operations) etc.

Data Elements

- Control Information
  - Such as Source, Date of Origin, Users, or Access Authorization.

    For e.g. Looking for Properties of Word Doc.
- Physical Location
  - In terms of Record of File or Database.

    For e.g. Where Storage is done C Drive, D Drive, CD ROM etc.

Data Structures

- It is the Group of Elements .

  For e.g.

  Data Structures: Book Details

  Data Elements: Author Name (M)

  Title of the Book (M)

  ISBN (Optional)

  Publisher Name (M)

  Quantity Ordered (M)

  Some Element are Mandatory whereas others are Optional

Data Flows and Data Stores

- Data Flows = Data Structures in Motion
- Data Stores = Data Structures at Rest

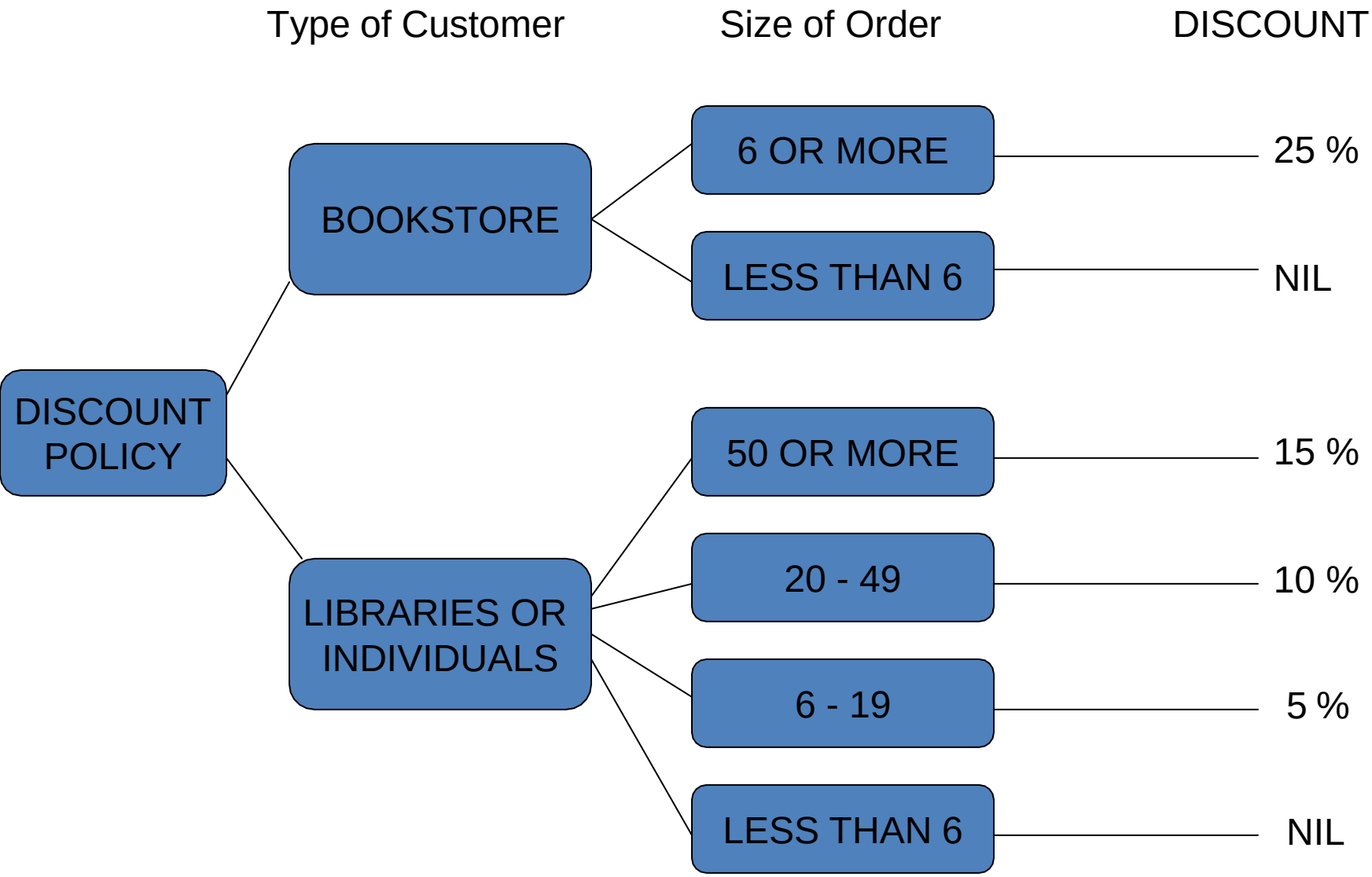
For e.g.

Data Flow/Store	Comments
Book Details	From ABC Book Store
Edition	4th
Quantity	10 Copies
Customer Details	

Decision Tree

- Once the data elements are defined in the data dictionary, we begin to focus on the processes.
- **For example:**

  Bookstores get a trade discount of 25%; for orders from libraries and individuals, 5% allowed on orders of 6-19 copies per book title; 10% on orders for 20-49 copies per book title; 15% on orders for 50 copies or more per book title


Type of Customer
Size of Order
DISCOUNT
DISCOUNT POLICY
BOOKSTORE
LIBRARIES OR INDIVIDUALS
6 OR MORE
LESS THAN 6
50 OR MORE
20 - 49
6 - 19
LESS THAN 6
25 %
NIL
15 %
10 %
5 %
NIL

Structured English

- Structures English is like structured programming, it uses logical construction and sentences designed to carry out instructions
- Designs are made through IF, THEN, ELSE, and SO statements

An Example

IF order is from Bookstore

and-IF order is for 6 copies or more per book title

THEN: Discount is 25%

ELSE (order is for fewer than 6 copies per book title)

SO: no discount is allowed

ELSE (order is from libraries or individuals)

Continued…

ELSE (order is from libraries or individuals)

SO-IF order is for 50 copies or more per book title

Discount is 15%

ELSE IF order is for 20 to 49 copies per book title

Discount is 10%

ELSE IF order is for 6 to 19 copies per book title

Discount is 5%

ELSE (order is for less than 6 copies per book order)

SO: no discount is allowed

Decision Tables

- It is a table of possibilities foe defining a problem and the actions to be taken
- It is a single representation of the relationships between conditions and actions
- It consists of two parts: stub and entry
- The stub part is divided into an upper quadrant called the condition stub and a lower quadrant called the action stub
- The entry part is also divided into an upper quadrant, called the condition entry and a lower quadrant called the action entry

Condition Stub		Condition Entry					
		1	2	3	4	5	6
	Customer is Bookstore	Y	Y	N	N	N	N
IF	Order size 6 copies or more ?	Y	N	N	N	N	N
(Condition)	Customer Librarian or Individual			Y	Y	Y	Y
	Order-size 50 copies or more ?			Y	N	N	N
	Order-size 20-49 copies ?				Y	N	N
	Order-size 6-19 copies ?					Y	N
Then	Allow 25% Discount	X					
(action)	Allow 15% Discount			X			
	Allow 10% Discount				X		
	Allow 5% Discount					X	
	No Discount allowed		X				X
	Action Stub	**Action Entry**					

Unit - IV

Form and Database design

Input Design

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well designed input forms and screens have following properties −

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding −
    - What are the inputs needed for the system?
    - How end users respond to different elements of forms and screens.

Objectives for Input Design

The objectives of input design are −

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

Data Input Methods

It is important to design appropriate data input methods to prevent errors while entering data. These methods depend on whether the data is entered by customers in forms manually and later entered by data entry operators, or data is directly entered by users on the PCs.

A system should prevent user from making mistakes by −

- Clear form design by leaving enough space for writing legibly.
- Clear instructions to fill form.
- Clear form design.
- Reducing key strokes.
- Immediate error feedback.

Some of the popular data input methods are −

- Batch input method (Offline data input method)
- Online data input method
- Computer readable forms
- Interactive data input

Input Integrity Controls

Input integrity controls include a number of methods to eliminate common input errors by end-users. They also include checks on the value of individual fields; both for format and the completeness of all inputs.

Audit trails for data entry and other system operations are created using transaction logs which gives a record of all changes introduced in the database to provide security and means of recovery in case of any failure.

Output Design

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design

The objectives of input design are −

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end users requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

Let us now go through various types of outputs −

External Outputs

Manufacturers create and design external outputs for printers. External outputs enable the system to leave the trigger actions on the part of their recipients or confirm actions to their recipients.

Some of the external outputs are designed as turnaround outputs, which are implemented as a form and re-enter the system as an input.

Internal outputs

Internal outputs are present inside the system, and used by end-users and managers. They support the management in decision making and reporting.

There are three types of reports produced by management information −

- **Detailed Reports** − They contain present information which has almost no filtering or restriction generated to assist management planning and control.
- **Summary Reports** − They contain trends and potential problems which are categorized and summarized that are generated for managers who do not want details.
- **Exception Reports** − They contain exceptions, filtered data to some condition or standard before presenting it to the manager, as information.

Output Integrity Controls

Output integrity controls include routing codes to identify the receiving system, and verification messages to confirm successful receipt of messages that are handled by network protocol.

Printed or screen-format reports should include a date/time for report printing and the data. Multipage reports contain report title or description, and pagination. Pre-printed forms usually include a version number and effective date.

Forms Design

Both forms and reports are the product of input and output design and are business document consisting of specified data. The main difference is that forms provide fields for data input but reports are purely used for reading. For example, order forms, employment and credit application, etc.

- During form designing, the designers should know −
  - who will use them
  - where would they be delivered
  - the purpose of the form or report
- During form design, automated design tools enhance the developer's ability to prototype forms and reports and present them to end users for evaluation.

Objectives of Good Form Design

A good form design is necessary to ensure the following −

- To keep the screen simple by giving proper sequence, information, and clear captions.
- To meet the intended purpose by using appropriate forms.
- To ensure the completion of form with accuracy.
- To keep the forms attractive by using icons, inverse video, or blinking cursors etc.
- To facilitate navigation.

Introduction to files and database design:

Database Management System (DBMS) is a collection of interrelated data. It consists of a set of programs to access the data. DBMS contains information about a particular enterprise. It provides an environment that is both convenient and efficient to use. Some of the Database Applications are as following:

• Banking: all transactions

• Airlines: reservations, schedules

• Universities: Registration, grade

• Manufacturing: production, inventory, order, supply chain.

• Sales: customers, products and purchases

File System vs. Database DBMS

(Database Management System) and File System are two ways that could be used to manage, store, retrieve and manipulate data. A File System is a collection of raw data files stored in the hard-drive whereas DBMS is a bundle of applications that is dedicated for managing data stored in databases. It is the integrated system used for managing digital databases, which allows the storage of database content, creation/ maintenance of data, search and other functionalities. Both systems can be used to allow the user to work with data in a similar way. A File System is one of the earliest ways of managing data. But due the shortcomings present in using a File System to store electronic data, Database Management Systems came in to use sometime later, as they provide mechanisms to solve those problems. But it should be noted that, even in a DBMS, data are eventually (physically) stored in some sort of files.

File System :

As mentioned above, in a typical File System electronic data are directly stored in a set of files. If only one table is stored in a file, they are called flat files. They contain values at each row separated with a special delimiter like commas. In order to query some random data, first it is required to parse each row and load it to an array at run time. But for this file should be read sequentially (because, there is no control mechanism in files), therefore it is quite inefficient and time consuming. The burden of locating the necessary file, going through the records (line by line), checking for the existence of a certain data, remembering what files/records to edit is on the user. The user either has to perform each task manually or has to write a script that does them automatically with the help of the file management capabilities of the operating system. Because of these reasons, File Systems are easily vulnerable to serious issues like inconsistency, inability for concurrency, data isolation, threats on integrity and lack of security.

DBMS :

DBMS, sometimes just called a database manager, is a collection of computer programs that is dedicated for the management (i.e. organization, storage and retrieval) of all databases that are installed in a system (i.e. hard drive or network). There are different types of Database Management Systems existing in the world, and some of them are designed for the proper management of databases configured for specific purposes. Most popular commercial Database Management Systems are Oracle, DB2 and Microsoft Access. All these products provide means of allocation of different levels of privileges for different users, making it possible for a DBMS to be controlled centrally by a single administrator or to be allocated to several different people. There are four important elements in any Database Management System.

File structures and organization:

It describes how records are stored within a file.

There are four file organization methods −

- **Serial** − Records are stored in chronological order (in order as they are input or occur). **Examples** − Recording of telephone charges, ATM transactions, Telephone queues.
- **Sequential** − Records are stored in order based on a key field which contains a value that uniquely identifies a record. **Examples** − Phone directories.
- **Direct (relative)** − Each record is stored based on a physical address or location on the device. Address is calculated from the value stored in the record's key field. Randomizing routine or hashing algorithm does the conversion.
- **Indexed** − Records can be processed both sequentially and non-sequentially using indexes.

Comparision

	Serial	Sequential	Direct	Index
Type of Access	Batch	Batch	Online	Batch or Online
Data Organization	Serial	Sequentially by key value	No particular order	Sequentially and by index
Flexibility in handling inquiries	No	No	Yes	Yes
Availability of up to date Data	No	No	Yes	Yes
Speed Retrieval	Slow	Slow	Very Fast	Fast
Activity	High	High	Low	High
Volatility	Low	Low	High	High
Example	ATM Transition Queue	Payroll process script billing operation	Online reservation and banking transaction	Customer ordering and billing

Types of Files used in an Organization System

Following are the types of files used in an organization system −

- **Master file** − It contains the current information for a system. For example, customer file, student file, telephone directory.
- **Table file** − It is a type of master file that changes infrequently and stored in a tabular format. For example, storing Zipcode.
- **Transaction file** − It contains the day-to-day information generated from business activities. It is used to update or process the master file. For example, Addresses of the employees.
- **Temporary file** − It is created and used whenever needed by a system.
- **Mirror file** − They are the exact duplicates of other files. Help minimize the risk of downtime in cases when the original becomes unusable. They must be modified each time the original file is changed.
- **Log files** − They contain copies of master and transaction records in order to chronicle any changes that are made to the master file. It facilitates auditing and provides mechanism for recovery in case of system failure.
- **Archive files** − Backup files that contain historical versions of other files.

Objectives of database design:

Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed database are easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.

The main objectives of database design in DBMS are to produce logical and physical designs models of the proposed database system.

The logical model concentrates on the data requirements and the data to be stored independent of physical considerations. It does not concern itself with how the data will be stored or where it will be stored physically.

The physical data design model involves translating the logical DB design of the database onto physical media using hardware resources and software systems such as database management systems (DBMS).

The following are the objectives of a good database design −

Avoid Redundant Data

The table in the database should be constructed following standards and with utmost dedication. It should have different fields and minimize redundant data. The table should always have a Primary Key that would be a unique id.

Faultless Information

The database should follow the standards and conventions and provide meaningful information useful to the organization.

Data Integrity

Integrity assists in guaranteeing that the values are valid and faultless. Data Integrity is set to tables, relationships, etc.

Modify

The database developed should be worked upon with the conventions and standards, so that it can be easily modified whenever the need arise.

Logical and physical view of data:

The process of designing a database carries various conceptual approaches that are needed to be kept in mind. An ideal and well-structured database design must be able to:

1. Save disk space by eliminating redundant data.
2. Maintains data integrity and accuracy.
3. Provides data access in useful ways.
4. Comparing Logical and Physical data models.

Logical View

A logical data model generally describes the data in as many details as possible, without having to be concerned about the physical implementations in the database. Features of logical data model might include:

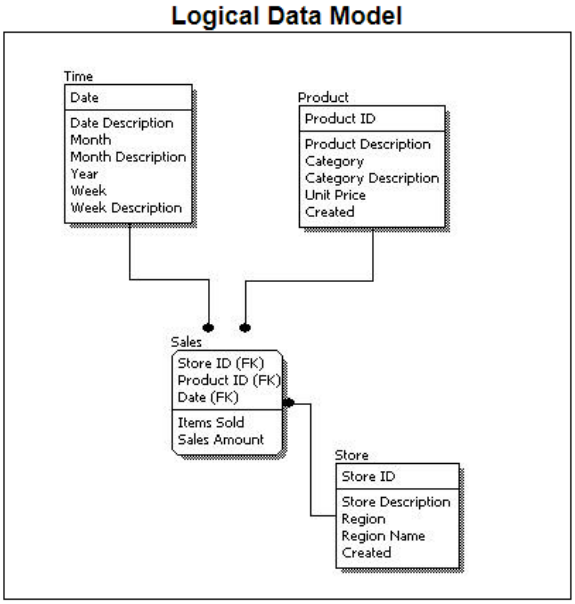
1. All the entities and relationships amongst them.
2. Each entity has well-specified attributes.
3. The primary key for each entity is specified.
4. Foreign keys which are used to identify a relationship between different entities are specified.
5. Normalization occurs at this level.

A logical model can be designed using the following approach:

1. Specify all the entities with primary keys.
2. Specify concurrent relationships between different entities.
3. Figure out each entity attributes
4. Resolve many-to-many relationships.
5. Carry out the process of normalization.

Also, one important factor after following the above approach is to critically examine the design based on requirement gathering. If the above steps are strictly followed, there are chances of creating a highly efficient database design that follows the native approach.

To understand these points, see the image below to get a clear picture.



If we compare the logical data model as shown in the figure above with some sample data in the diagram, we can come up with facts that in a conceptual data model there are no presence of a primary key whereas a logical data model has primary keys for all of its attributes. Also, logical data model the cover relationship between different entities and carries room for foreign keys to establish relationships among them.

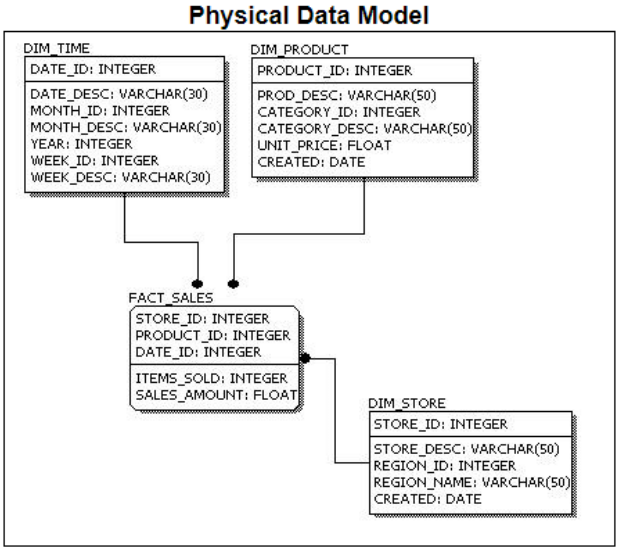
Physical

A Physical data mode generally represents how the approach or concept of designing the database. The main purpose of the physical data model is to show all the **structures** of the table including the **column name, column data type, constraints, keys(primary and foreign)**, and the relationship among tables. The following are the features of a physical data model:

1. Specifies all the columns and tables.
2. Specifies foreign keys that usually define the relationship between tables.
3. Based on user requirements, de-normalization might occur.
4. Since the physical consideration is taken into account so there will straightforward reasons for difference than a logical model.
5. Physical models might be different for different RDBMS. For example, the data type column may be different in MySQL and SQL Server.

While designing a physical data model, the following points should be taken into consideration:

1. Convert the entities into tables.
2. Convert the defined relationships into foreign keys.
3. Convert the data attributes into columns.
4. Modify the data model constraints based on physical requirements.



Comparing this physical data model with the logical with the previous logical model, we might conclude the differences that in a physical database entity names are considered table names and attributes are considered column names. Also, the data type of each column is defined in the physical model depending on the actual database used.