THE COPPERBELT UNIVERSITY

SCHOOL OF GRADUATE STUDIES

COURSE MODULE FOR DISTANCE STUDENTS

GBS 789: MANAGEMENT INFORMATION SYSTEMS

Lecturer: Dr N Chaamwe

chimz@cbu.ac.zm

Introduction

This course, Management Information systems consist of three parts namely; Fundamental of Information System, Database Systems and Database Implementation Techniques. The first module deals with Systems approach to Problem Solving and Developing IS Solutions; the second explores Databases, The database architecture and Designing the Database. The third module discusses Database Management, Database recovery, Data Security, transaction processing and concurrent control.

Course Objectives

The broad objectives of this course are to introduce students to the fundamentals of managing information systems, including topics such as electronic commerce, electronic business and the internet. The course encourages students to apply what they learn from the Course to the real-world management decision making environment. The course offers opportunities to students to be involved in course activities (e.g. seminar style, debates) in such a way as to promote their interpersonal communication and project management skills, and create awareness of the ethics / social implications / professional practice of managing information systems.

Course Outline: Topics

1: Basic Concepts and Elements of Information Systems (IS)

2: Hardware and Software Elements of Information Systems

3: The Accounts Information

4: Enterprise Resource Planning (ERP) – Industry Perspective

5: Knowledge Management and Decision Support Systems

6: Systems Development Life Cycle (SDLC)

7: Managing the Information Systems Data (Information) Assets

Databases, The database architecture and Designing the Database

8: Information Systems Security and Control

Assessment

Ca 40%

- 10% Assignment
- 30% Tests

Exam 60%
Handouts
go to **www.Lechaamwe.weebly.com**

- Lecture Notes – Postgraduate - GAF710 & GBS 789

Assignment.

The Government of Zambia through the Ministry of Education, Science, Vocational Training and Early Education (MOESTVEE) introduced a new ambitious Curriculum from early child education to secondary education. The new curriculum includes ICT as a subject. Discuss the challenges that hinder sustainable implementation of Information Communication Technology (ICT) as a subject in public schools in rural Zambia.

Report Format

Title

Abstract

1. **Introduction**
2. **Literature review**
3. **Discussion**
4. **Conclusion/Recommendations**

References

(avoid citing Wikipedia)

Contents

Introduction 2

Course Objectives 2

Course Outline: Topics 2

Assessment 3

Assignments. 3

1. Part 1. Fundamentals of Information Systems: 6

1.1 An Introduction to Information Systems 6

1.1.1 Information Concepts 6

1.1.2 A System 8

1.1.3 Information System (IS) 9

1.2 An Overview of Management Information Systems 12

1.2.1 Management information system (MIS) 12

1.2.2 System Applications in the Organization 14

1.2.3 Systems From a Functional Perspective 15

1.2.4 Examples of specific systems 17

1.2.5 Enterprise Systems 18

1.3 Organizations and Information Systems 20

1.3.1 What is the Organization? 20

1.3.2 The Changing Role of Information Systems in Organizations 20

1.4 Systems Development Lifecycle 22

1.4.1 Introduction 22

1.4.2 Seven Phases in the SDLC 23

1.4.3 Why Systems Fail 29

1.4.4 Prototyping 30

2. Part 2. Database systems 31

2.1 An Introduction to Database Systems. 31

2.1.1 File based systems (Pre-database systems) 32
2.1.2 Database Architecture 33
2.1.3 Advantages of Database systems 37
2.1.4 Disadvantages of Database systems 37
2.1.5 Lifecycle of Database system development 38
2.2 Introduction to Relational Databases 40
2.2.1 Relational Model Terminologies 40
2.2.2 Properties of Relations 41
2.2.3 Relational Database Integrity Constraints 41
2.3 Approaches for data model development 42
2.4 Entity-Relationship Modeling 43
2.5 Logical Design 47
3. Part 3. System implementation techniques 48
3.1 Transaction Processing 48
3.1.1 Transactions Management 49
3.2 Database Recovery 51
3.2.1 Recovery Facilities 51
3.2.2 Recovery Techniques 51
3.3 Database Security 52
3.3.1 Database Security measures 53
3.4 Concurrency Control 54
3.4.1 Locking Protocols 56
3.4.2 Problems with Locking 57
3.4.3 Two-Phase Locking 57

1. Part 1. Fundamentals of Information Systems:

- This part explores the Systems approach to Problem Solving and Developing IS Solutions.

LECTURE 1

- An Introduction to Information Systems
- Information Concepts
- System Concepts
- System Performance and Standards
- What is an Information System?
- Computer-Based Information Systems

1.1 An Introduction to Information Systems

- Information systems are now used in most professions
  - Sales representatives
  - Managers
  - Financial planners
- They are indispensable for achieving career goals.

1.1.1 Information Concepts

- Data

Collection of Raw facts and figures, Distinct pieces of information, usually formatted in a special way.

Eg. Banda, 1989, Lusaka

- Types of Data

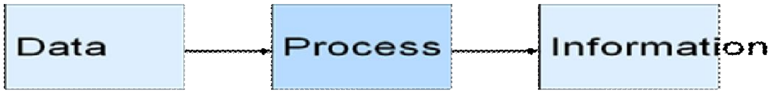
Data	Represented by
Alphanumeric data	Numbers, letters, and other characters
Image data	Graphic images or pictures
Audio data	Sound, noise, tones
Video data	Moving images or pictures

- Information

  A collection of facts organized in such a way that they have additional value beyond the value of the facts themselves

  Eg: Banda was born in Lusaka in 1989

- Data → Information



  - Process

    A set of logically related tasks performed to achieve a defined outcome

    Process can be

    - (noun) An executing program. The term is used loosely as a synonym of task.
    - (verb) To perform some useful operations on data.

- Knowledge

  An awareness and understanding of a set of information and how that information can be made useful to support a specific task

  - Knowledge base

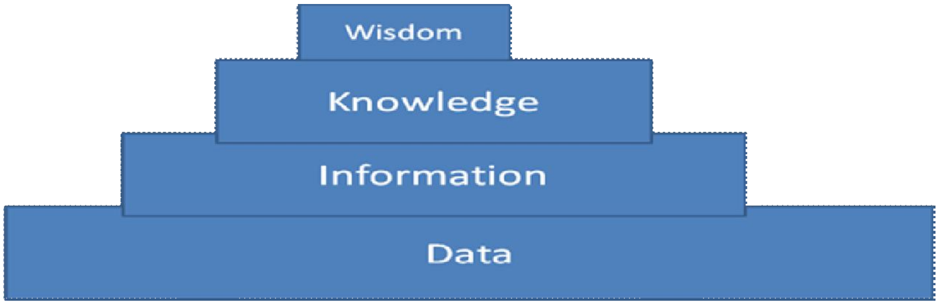
    The collection of data, rules, procedures, and relationships that must be followed to achieve value or the proper outcome

- Wisdom

  Acquired Knowledge used for the betterment of Mankind

  The fear of the Lord is the beginning of Wisdom:

Data – Wisdom Hierarchy



- Characteristics of Valuable Information

Accurate, complete, economical, flexible, reliable, relevant, simple, timely, verifiable, accessible, secure

1.1.2 A System

A set of elements or components that interact to accomplish goals or

A combination of components working together to achieve set objectives

- System Elements

  Inputs, Processing mechanisms, Outputs and Goal

- System Example1

System	Elements			Goal
	Inputs	Processing elements	Outputs	
Movie	Actors, director, staff, sets, equipment	Filming, editing, special effects, distribution	Finished film delivered to movie studio	Entertaining movie, film awards, profits

- System Example2

System	Elements			Goal
	Inputs	Processing elements	Outputs	
University	Students, staff, equipment	Lectures, tutorials, Exams	Graduates	Provide human resorce

- System Performance and Standards
  - Efficiency

  A measure of what is produced divided by what is consumed

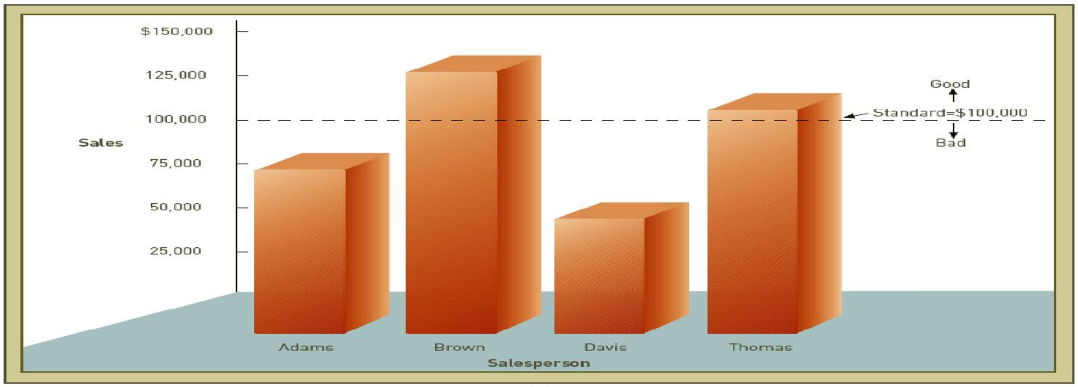
  - Effectiveness

  A measure of the extent to which a system achieves its goals

  - System standard

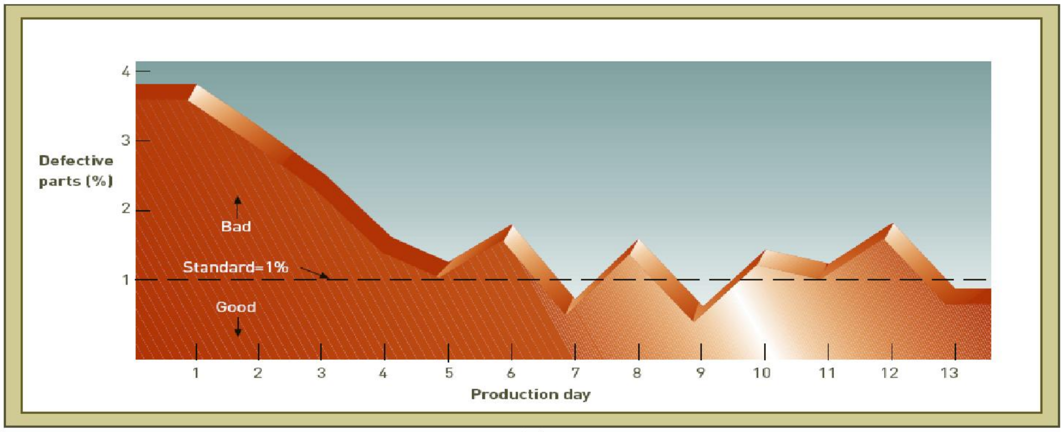
  A specific objective of the system

System standard Example 1



(a)

System standard Example 2



(b)

- System Variables and Parameters
    - System variable

A quantity or item that can be controlled by the decision maker

E.g. the price a company charges for a product

- System parameter

A value or quantity that cannot be controlled by the decision maker

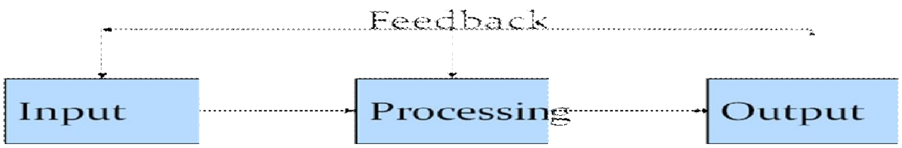
E.g., cost of a raw material

1.1.3 Information System (IS)

- Introduction
    - Definition

A set of interrelated elements or components that collect (input), manipulate (process), and disseminate (output) data and information and provide a feedback mechanism to meet an objective

- Schematic model of an information system



- ○ Input

The activity of gathering and capturing data. Whatever goes into the computer

- ○ Processing

Converting or transforming data into useful outputs

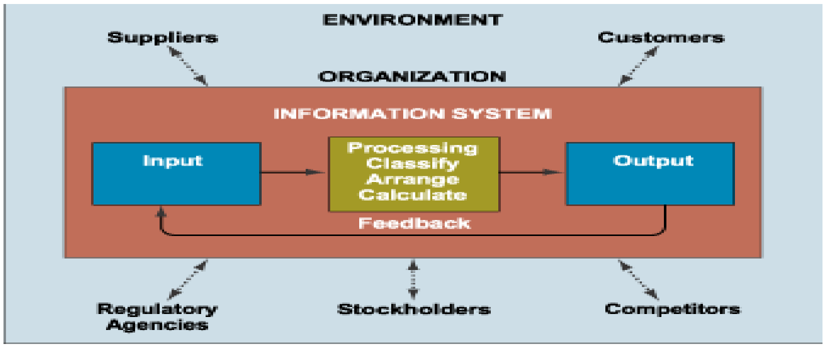
- ○ Output

Useful information, usually in the form of documents and/or reports. Anything that comes out of a computer

- ○ Feedback

Output that is used to make changes to input or processing activities.

- ➢ The Environment of an information system
    - ○ An information system contains information about an organization and its surrounding environment.
    - ○ Environmental factors such as customers, suppliers, competitors, stockholders, and regulatory agencies interact with the organization and its information systems



- ➢ Types of Information Systems
    - ○ Formal Information systems

They rest on accepted and fixed definitions of data and procedures for collecting, storing, processing, disseminating, and using these data.

- ○ Informal information systems (such as office gossip networks)

rely, by contrast, on unstated rules of behavior. There is no agreement on what is information, or on how it will be stored and processed.

    - Manual vs. Computerized Systems

      An information system can be:

      Manual or Computerized

      Manual systems are still widely used

      E.g., some investment analysts manual draw charts and trend lines to assist them in making investment decisions

- Computer-based Information Systems

  A CBIS is composed of…

  Hardware, Software, Databases, Telecommunications, People, Procedures

  Together they are…

  Configured to collect, manipulate, store, and process data into information.

  Another term for CBIS is Technology Infrastructure

  Consists of the shared information system (IS) resources that form the foundation of the information system

    - Hardware

      Computer equipment used to perform input, processing, and output activities

      The objects that you can actually touch, like disks, disk drives, display screens, keyboards, printers, boards, and chips.

    - Software

      Computer programs that govern/determine/control the operation of the computer.

      Computer instructions or data

    - Database

      An organized collection of facts and information.

      A collection of information organized in such a way that a computer program can quickly select desired pieces of data

    - Telecommunications

      The electronic transmission of signals for communications; enables organizations to link computer systems into effective networks.

      Refers to all types of data transmission, from voice to video

        - Networks

      Used to connect computers and computer equipment in a building, around the country, across the world, to enable electronic communications

      A group of two or more computer systems linked together

There are many types of computer networks, including:

- **local-area networks** (LANs) : The computers are geographically close together (that is, in the same building).
- **wide-area networks** (WANs) : The computers are farther apart and are connected by telephone lines or radio waves.
- Internet and Intranet

The world’s largest telecommunications network. A network of networks

A global network connecting millions of computers

- Intranet

  A network that uses Internet technology within an organization

  A network belonging to an organization

- People and Procedures

The most important element in most computer-based information systems

Includes people who manage, run, program, and maintain the system

E.g., IT professionals (you!)

- Procedures

Includes the strategies, policies, methods, and rules for using the CBIS

LECTURE 2

- An Overview of Management Information Systems
- System Applications in the Organization
- Systems From a Functional Perspective
- Examples of specific systems
- Enterprise Systems
- Benefits and Challenges of Enterprise Systems

1.2 An Overview of Management Information Systems

1.2.1 Management information system (MIS)

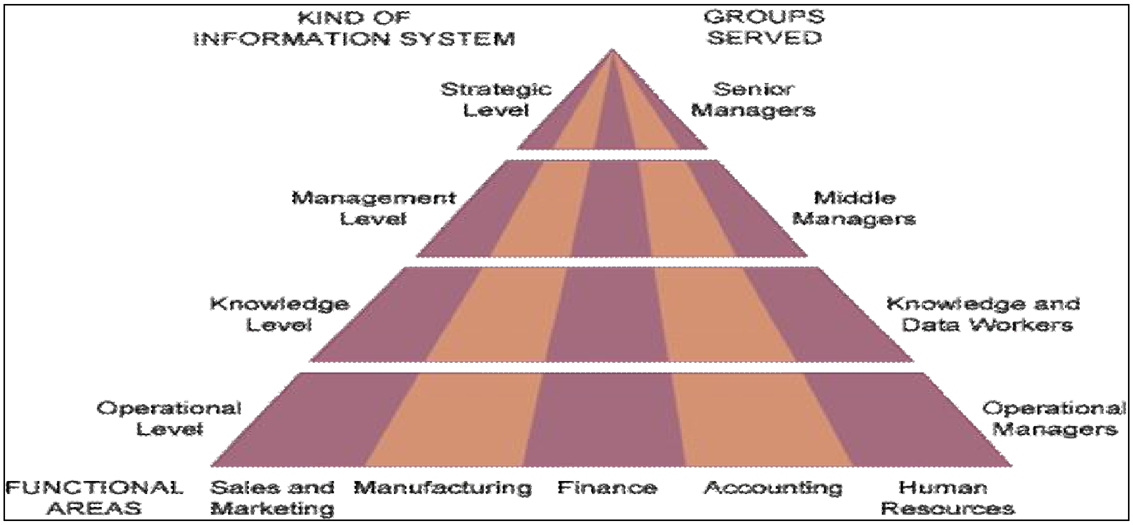
- Integrated collection of people, procedures, databases, and devices
- Provides managers and decision makers with information to help achieve organizational goals
- Can give the organization a competitive advantage

    - Providing the right information to the right people in the right format and at the right time
    - Provides managers with information that supports effective decision making and provides feedback on daily operations

- **Inputs to MIS**
- Internal data sources
    - TPSs and ERP systems and related databases
    - Data warehouses and data marts
    - Specific functional areas throughout the firm
- External data sources
    - Customers, suppliers, competitors, and stakeholders whose data is not already captured by the TPS
    - Internet
    - Extranets
- **Outputs of MIS**
- **Scheduled report**
  - Produced periodically: daily, weekly, or monthly
- **Key-indicator report**
  - At beginning of workday, for quick, corrective action
  - Summary of previous day's critical activities: sales, inventory...
- **Demand report**
  - Developed to give certain info at someone's request
      - e.g. FedEx' site allows customers to track packages
- **Exception report**
  - Automatically produced when a situation is unusual or requires management action
- **Drill-down reports** (e.g. company to dept to salesrep)
  - Provide increasingly detailed data about a situation

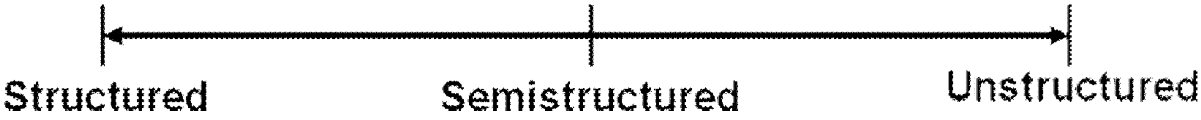
1.2.2 System Applications in the Organization

- Because there are different interests, specialties, and levels in an organization, there are different kinds of systems.
- No single system can provide all of the information an organization needs.
- Systems are built to serve these different organizational interests.
- Four main types of information system serve different organizational levels:
- **Operational-level systems**
    - Support operational managers by keeping track of the elementary activities and transactions of the organization, such as sales, receipts, cash deposits, payroll, credit decisions, and the flow of materials in a factory.
- **knowledge-level systems**
    - Support the organization's knowledge and data workers.
    - The purpose of knowledge-level systems is to help the business firm integrate new knowledge into the business and to help the organization control the flow of paperwork.
- **Management-level systems**
    - Serve the information systems that support the monitoring, controlling, decision-making, and administrative activities of middle managers.
- **Strategic-level systems**
    - help senior management tackle and address strategic issues and long-term trend, both in the firm and in the external environment.
    - Information systems also serve the major business functions, such as sales and marketing, manufacturing, finance, accounting, and human resources.



1.2.3 Systems From a Functional Perspective

- Information systems can be classified by the specific organizational function they serve as well as by organizational level.
- **Transaction Processing Systems**
    - Transaction Processing Systems (TPS) are the basic business systems that serve the operational level of the organization.
    - A transaction processing system is a computerized system that performs and records the daily routine transactions necessary to conduct the business.
- **Knowledge Work and Office Systems**
    - Knowledge Work System (KWS) and office systems serve the information needs at the knowledge level of the organization.
        - **Knowledge workers** are people who hold formal university degree and who are often members of a recognized profession, such as engineers, doctors, lawyers, and scientists.
        - **Data workers** typically have less formal, advanced educational degrees and tend to process rather than create information.
- **Management Information Systems**
    - The term management information systems (MIS) also designates a specific category of information systems serving management-level functions.
    - Management Information Systems (MIS) refer to information systems at the management level of an organization that serve the functions of planning, controlling, and decision making by providing routine summary and exception reports.
- **Decision-Support Systems**
    - Organized collection of people, procedures, software, databases, and devices used to help make decisions that solve problems
    - DSS help managers make decisions that are unique, rapidly changing, not easily specified in advance.
    - Decision-Support Systems (DSS) also serve the management level of the organization.
    - Focus of a DSS is on decision-making effectiveness regarding unstructured or semistructured business problems.
    - **Problems**



- Structured
    - Routine problems with known solution and info
        - e.g. Which customers did not pay?
- Semistructured
    - Less routine problems
    - Involves subjective judgment; info may not be available
        - e.g. Which products should we sell?
- Unstructured
    - Requires human intuition for decisions; info is missing
        - Decrease cold weather inventories due to global warming?

- **Group Support Systems**
    - Consists of most elements in a DSS, plus software to provide effective support in group decision making
    - Also called group decision support system **(GDSS)** or computerized collaborative work system
    - **GSS Software**
        - Often called groupware or workgroup software
        - Helps with joint work group scheduling, communication, and management
        - Examples
            - Windows Meeting Space (in Vista)
                - A built-in social collaboration technology for small groups
                - Replaced NetMeeting in Windows XP
                - Not in Windows 7
            - MS SharePoint Workspace 2010
                - An ecollaboration/virtual teams software program

- MS Office Live Meeting (with panoramic *RoundTable*)
- Lotus Domino (previously Notes) from IBM
- Lotus Domino (previously Notes) from IBM
- Virtual Office from Groove Networks
- Lync (Microsoft)
- Examples of groupware products available on the web
  - Cisco WebEx
  - Genesys Meeting Center
  - GoToMeeting Corporate

- **Executive Support Systems**
  - Senior managers use executive support systems (ESS) to make decisions.
  - ESS serves the strategic level of the organization.
  - ESS are design to incorporate data about external events such as new tax laws or competitors, but they also draw summarized information from internal MIS and DSS.

1.2.4 Examples of specific systems

- **Sales and Marketing Systems**
  - The sale and marketing function is responsible for selling the organization's products or services.
  - Marketing is concerned with identifying the customers for the firm's products or services, determine what they need or want, planning and developing products and services to meet their needs, and advertising and promoting these products and services.

- **Manufacturing and Production Systems**
  - The manufacturing and production function is responsible for actually producing the firm's goods and services.
  - Manufacturing and production activities deal with the planning, development, and maintenance of production facilities; the establishment of production goals; the acquisition, storage, and availability of production materials; and the scheduling of equipment, facilities, materials, and labour required to fashion finished products.

- **Finance and Accounting Systems**
  - The finance function is responsible for managing the firm’s financial assets, such as cash, stocks, bonds, and other investments, in order to maximize the return on these financial assets.
  - The finance function is also in charge of managing the capitalization of the firm.
  - In order to determine whether the firm is getting the best return on its investments, the finance function must obtain a considerable amount of information from sources external to the firm.
- **Human Resources Systems**
  - The human resource function is responsible for attracting, developing, and maintaining the firm’s workforce.
  - Human resources information systems support activities such as identifying potential employees, maintaining complete records on existing employees, and creating programs to develop employees’ talents and skills.
  - Strategic-level human resources system identify the employee requirements (skills, educational level, types of positions, number of positions, and cost) for meeting the firm’s long term business plans.

1.2.5 Enterprise Systems

- A large organization typically has many different kinds of information systems that support different functions, organizational levels, and business processes.
- Many organizations are also building enterprise systems, also known as enterprise resource planning (ERP) systems, to provide firm wide integration.
- Enterprise systems can integrate the key business processes of an entire firm into a single software system that allows information to flow seamlessly throughout the organization.
- **Benefits of Enterprise Systems**
  - Enterprise systems promise to greatly change four dimensions of business:
  - Firm Structure and Organization: Companies can use enterprise systems to support organizational structures that were not previously possible or to create a more disciplined organizational culture.
  - Management: In addition to automating many essential business transactions, such as taking orders, paying suppliers, or changing employee benefit status, enterprise systems can also improve management reporting and decision making.

  - Technology: Enterprise systems promise to create a single, integrated repository that gathers data on all the key business processes.
  - Business: By integrating discrete business processes such as sales, production, finance, and logistics, the entire organization can efficiently respond to customer requests for products or information, forecast new products, and build and deliver them as demand requires.

- **The Challenge of Enterprise Systems**
  - Daunting Implementation:
    - Enterprise systems bring dramatic changes to business.
    - They require not only deep-seated technological changes but also fundamental changes in the way the business operates.
  - High Up-front Cost and Future Benefits
    - The costs of enterprise systems are large, up-front, highly visible, and often politically changed.
    - Although the costs to build the system are obvious, the benefits often cannot be precisely quantified at the beginning of an enterprise project.
  - Inflexibility
    - Enterprise system software tends to be complex and difficult to master, with a worldwide shortage in people with the expertise to install and maintain it.
    - The software is deeply intertwined with corporate business.
  - Realizing Strategic Value
    - Companies may also fail to achieve strategic benefits from enterprise systems if integrating business processes using the generic models provided by standard ERP software prevents the firm from using unique business processes that had been sources of advantage over competitors.

LECTURE 3

- Organizations and Information Systems
- What is the Organization?
- The Changing Role of Information Systems in Organizations
- Implications for the Design and Understanding of Information Systems

1.3 Organizations and Information Systems

- The interaction between information technology and organizations is very complex and is influenced by a great many mediating factors,
- including the organization's structure, standard operating procedures, politics, culture, surrounding environment, and management decision
- Other factors mediating the relationship are bureaucracy, business processes, and pure chance.
- There is no singular effect of computers in all organizations. Instead, different organizations in different circumstances experience different effects from the same technology

1.3.1 What is the Organization?

- An organization is a stable, formal, social structure that takes resources from the environment and processes them to produce outputs.
- Capital and labour are primary production factors provided by the environment.
- The organization (the firm) transforms these inputs into products and services.
- Organizations are also social structures, because they are a collection of social elements (rights, privileges, obligations, and responsibilities, norms, people etc).
- A more realistic behavioural definition of an organization is that, it is a collection of rights, privileges, obligations, and responsibilities that are delicately balanced over a period of time through conflict and conflict resolution.

1.3.2 The Changing Role of Information Systems in Organizations

- Information Technology Infrastructure and Information Technology Services
- How Information Systems Affect Organizations
- The Internet and Organization

- **Information Technology Infrastructure and Information Technology Services**
    - One way that organization can influence how information technology will be used is through decisions about the technical and organizational configuration of systems.
    - Today's new IT infrastructure is designed to make information flow across the enterprise and includes link to customers, vendors, and public infrastructures, including the Internet.
    - Each organization determines how its infrastructure will be configured.

- The formal organization unit or function responsible for technology services is called the information systems department. It is responsible for maintaining the hardware, software, data storage, and networks that comprise the firm' IT infrastructure.

➢ **How Information Systems Affect Organizations**

- Economic Theories

    - Microeconomic model of the firm. It that views information technology as a factor of production that can be freely substantiated for capital and labor.

    - Information technology also helps firms contract in size, because it can reduce transaction costs—the costs incurred when a firm buys on the marketplace what it cannot make itself.

    - According to transaction cost theory, firms and individuals seek to economize on transaction costs, much as they do on production cost. Using markets is expensive because of costs such as locating and communicating with distant suppliers monitoring contract compliance, buying insurance, obtaining information on products and so fort.

- Behavioral Theories

    - Behavioral researchers have theorized that information technology could change the hierarchy of decision making in organizations by lowering the costs of information acquisition and broadening the distribution of information.

    - Information technology could bring information directly from operating units to senior managers, thereby eliminating middle managers and their clerical support workers.

    - Alternatively, information technology could distribute information directly to lower- level workers, who could then make their own decisions based on their own knowledge and information without any management intervention.

➢ **The Internet and Organization**

- Business are rapidly rebuilding some of their key business processes based on Internet Technology and making this technology a key component of their IT infrastructures.

- If prior networking is any guide, one result will be simpler businesses, fewer employees, and much flatter organizations than in the past.

Implications for the Design and Understanding of Information Systems

- Factors to consider when planning a new system are these:
  - The environment in which the organization must function.
  - The structure of the organization: hierarchy, specialization, standard operating procedures.
  - The organization’s culture and politics.
  - The type of organization.
  - The nature and style of leadership.
  - The extent of top management’s support and understanding.
  - The principal interest groups affected by the system.
  - The kinds of tasks, decisions, and business processes that the information system is designed to assist.
  - The history of the organization: past investments in information technology, existing skills, important programs, and human resources.

LECTURE 3

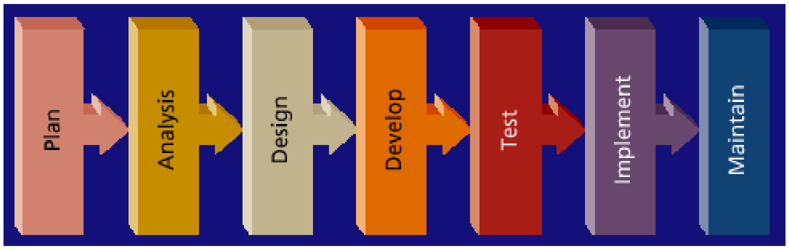
- Systems development Lifecycle
- Seven Phases in the SDLC
- Why Systems Fail
- Prototyping

1.4 Systems Development Lifecycle

1.4.1 Introduction

- Why do businesses build information systems?
- How does a business know when it is time to replace the old information system with a new one ?
- Developing of an information system may fail.
- There are many factors that must be considered and come together in order to develop a successful information system.

- The SDLC = Systems Development Life Cycle is one approach to reduce the risk of failure.
- ***Systems development life cycle (SDLC) -*** a structured step-by-step approach for developing information systems.



1.4.2 Seven Phases in the SDLC

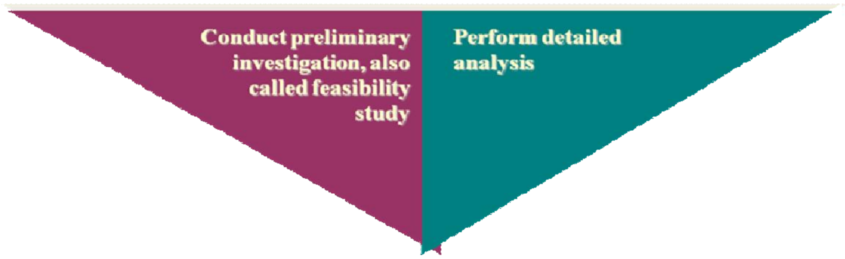
- There are literally hundreds of activities associated with each phase of the SDLC
- Such activities include:
  - Determining budgets
  - Gathering systems requirements
  - Documentation
  - Modeling
- *Phase 1: Plan*
  - involves determining a solid plan for developing your information system.
  - The three of the most important activities involved during the planning phase:
    - Identify and select the system for development
    - Set project scope
    - Develop project plan
  - Identify and select the system for development
    - Key question: “What systems are required to support the strategic goals of your organization?”
    - Companies cannot develop all proposed IS, so they look into the critical success factors.
    - **Critical success factor (CSF)** - a factor simply critical to your organization’s success.

        - Once the system to be developed in defined, then the project needs to be scoped**.**
    - ***Set project scope*** - clearly defines the high-level system requirements.
        - It is a birds-eye-view of the project.
        - ***Project scope document*** - a written definition of the project scope and is usually no longer than a paragraph.
    - Developing a project plan is the final activity of the planning phase.
        - ***Project plan*** - defines the what, when, and who questions of system development including all activities to be performed, the individuals, or resources, who will perform the activities, and the time required to complete each activity.
        - ***The project plan should include:***
            - ***Project milestones*** - represent key dates for which you need a certain group of activities performed.
            - ***Project manager -*** an individual who is an expert in project planning and management, defines and develops the project plan and tracks the plan to ensure all key project milestones are completed on time.

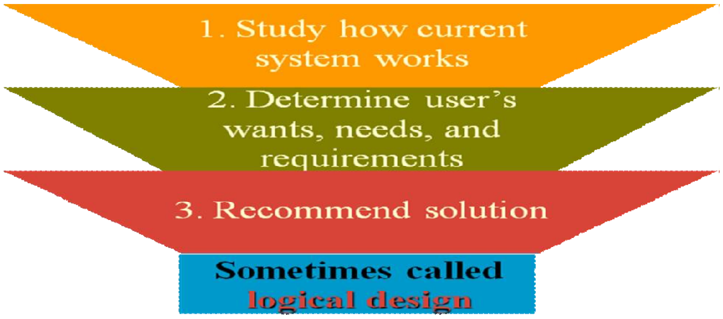
- Phase 2: Analysis

The analysis phase involves

- The end users
- The IT specialists Working together To understand and Document the business requirements for the system.


Conduct preliminary investigation, also called feasibility study
Perform detailed analysis

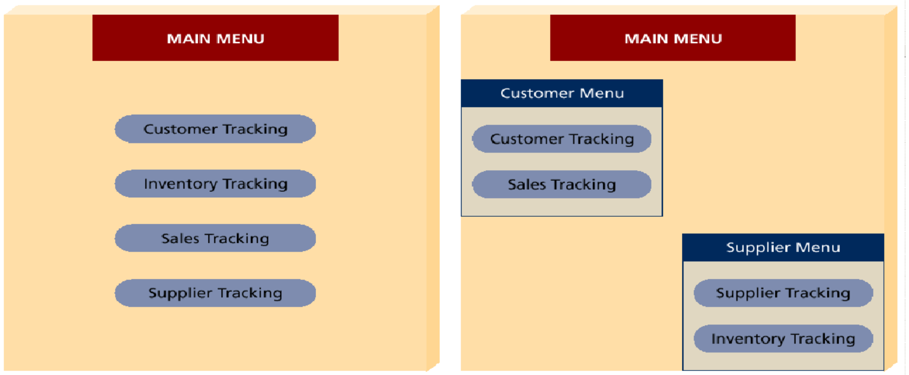
- What is the preliminary investigation?
  - Determine exact nature of problem or improvement and whether it is worth pursuing
  - Findings are presented in feasibility report, also known as a feasibility study
- What is detailed analysis?


1. Study how current system works
2. Determine user's wants, needs, and requirements
3. Recommend solution
Sometimes called logical design

- A useful way to gather system requirements is:
  - Joint application development (JAD) - knowledge workers and IT specialists meet, sometimes for several days, to define or review the business requirements for the system.
  - Requirements definition document – prioritizes the business requirements and places them in a formal comprehensive document.
  - Sign-off - the knowledge workers' actual signatures indicating they approve all of the business requirements.

- *Phase 3: Design*
  - The primary goal of the design phase
    - Build the technical architecture required to support the system.
    - This includes
      - Design of the technical architecture
      - Design system models.
    - Technical architecture –
      - Defines the hardware, software, and telecommunications equipment required to run the system.
      - The final architecture must meet your needs in terms of

        - Time, Cost, Technical feasibility, Flexibility
    - It is important to ensure that the final architecture meet
        - Current system needs
        - Future system needs
- ***Modeling*** - the activity of drawing a graphical representation of a design.
    - You model everything you build including
        - Reports, Programs, databases, etc.
    - There are many different types of modeling activities performed during the design stage
        - Graphical user interface (GUI)
        - The interface to an information system.

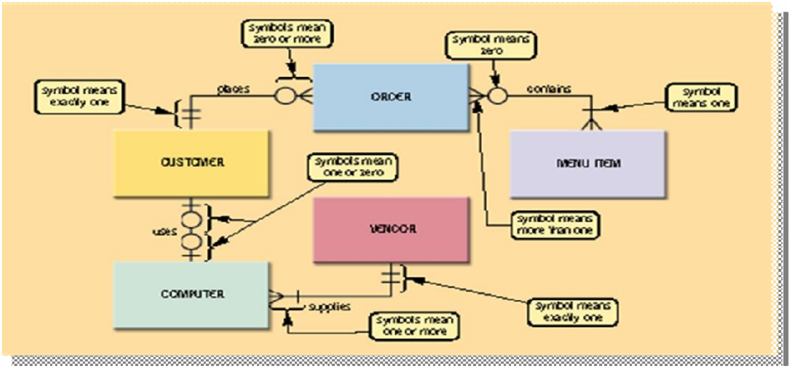


GUI Design: Alphabetical order of buttons

GUI Design: Categorical order of buttons

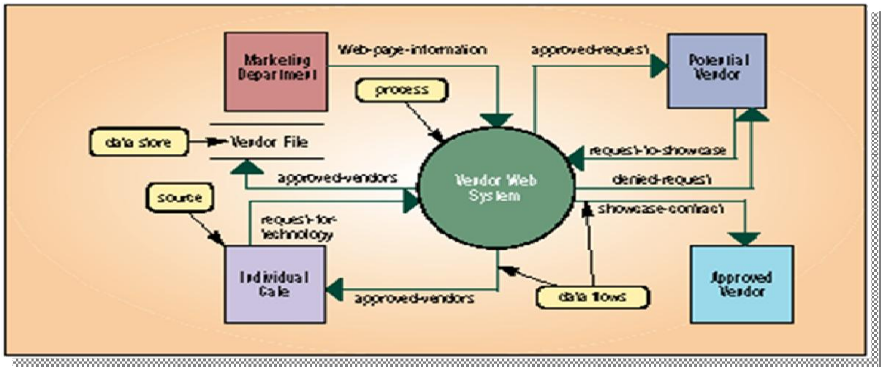
- Entity relationship diagram

Tool that graphically shows connections between entities in system

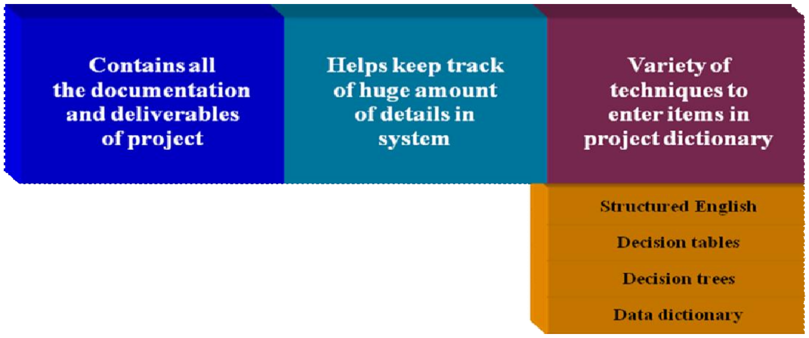


- Data flow diagram

Tool that graphically shows flow of data in system



- Project dictionary



Contains all the documentation and deliverables of project

Helps keep track of huge amount of details in system

Variety of techniques to enter items in project dictionary

Structured English

Decision tables

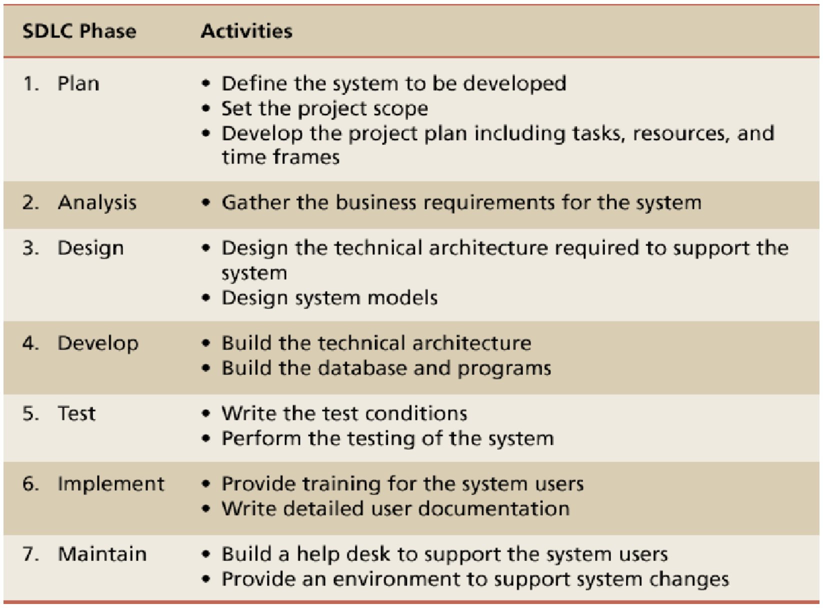
Decision trees

Data dictionary

- Phase 4: Development
  - Take all of your detailed design documents from the design phase and transform them into an actual system.
  - Activities during this phase include
    - Coding programs
    - Creating databases
    - Deploying the telecommunications equipment
    - Installing hardware and software
  - In other words
    - Build the technical architecture.
    - Build the database and programs.

- *Phase 5: Test*
  - Verifies that the system works and meets all of the business requirements defined in the analysis phase.
  - ***Test conditions*** - the detailed steps the system must perform along with the expected results of each step.
  - Testing is critical
    - Must have test plans
    - Write the test conditions.
    - Perform the testing of the system.
- *Phase 6: Implement*
  - Bringing the system into life by placing it in the organization
  - During this phase
    - You distribute the system to all of the workers and they begin using the system to perform their everyday jobs.
  - During this phase
    - ***you create User documentation***
    - Highlights how to use the system.
  - You perform training to the employees to use the system
    - ***Online training*** - runs over the Internet or off a CD-ROM.
    - ***Workshop training -*** is held in a classroom environment and lead by an instructor.
- *Phase 7: Maintain*
  - It is the final phase of the system development effort
  - Monitor and support the new system to ensure it continues to meet the business goals.
  - Once the system is in place you need to provide support
    - Build a help desk to support the system users.
    - Provide an environment to support system changes.

Summary of the SDLC phase



SDLC Phase	Activities
1. Plan	• Define the system to be developed • Set the project scope • Develop the project plan including tasks, resources, and time frames
2. Analysis	• Gather the business requirements for the system
3. Design	• Design the technical architecture required to support the system • Design system models
4. Develop	• Build the technical architecture • Build the database and programs
5. Test	• Write the test conditions • Perform the testing of the system
6. Implement	• Provide training for the system users • Write detailed user documentation
7. Maintain	• Build a help desk to support the system users • Provide an environment to support system changes

1.4.3 Why Systems Fail

- Only 20% of systems built today are successful, 80% of systems development fail.
- Five primary reasons why systems fail include:
  - Unclear or missing requirements
  - Skipping SDLC phases
  - Failure to manage project scope
  - Failure to manage project plan
  - Changing technology
- Unclear or Missing Requirements
  - The business requirements drive the entire system.
  - If they are not accurate or complete there is no way the system will be successful.
  - Gathering inaccurate requirements?
    - ✗System must not allow students to add classes
    - ✓System must allow students to add classes
  - Missing Requirement?
    - Forgetting to include into the system the calculation of a student grade point average

- *Skipping SDLC Phases*
  - The first thing individuals tend to do when a project falls behind schedule is to start skipping phases in the SDLC.
  - Skipping any of the phases is sure to lead to system failure.
- *Failure To Manage Project Scope*
  - The project manager must track the status of each activity and adjust the project plan if a activity is added or taking longer than expected.
  - What could happen in a project to this effect is
    - ***Scope creep*** - occurs when the scope of the project increases.
    - ***Feature creep*** - occurs when developers add extra features that were not part of the initial requirements.
- Failure To Manage Project Plan
  - Managing the project plan is one of the biggest challenges during systems development
  - The project plan is the road map you follow during the development of the system.
  - Developing the initial project plan is the easy.
  - Managing, revising, and updating the project plan is the hard part.
- *Changing Technology*
  - Technology changes so fast that it's almost impossible to deliver an information system without feeling the pain of changing technology.

1.4.4 Prototyping

- ***Prototyping*** - the process of building a model that demonstrates the features of a proposed product, service, or system.
- ***Prototype -*** a model of a proposed product, service, or system.
- Prototyping can be used to perform a variety of functions
  - Gathering requirements
  - Helping determine requirements
  - Proof-of-concept prototype - used to prove the technical feasibility of a proposed system.
  - Selling prototype - used to convince people of the worth of a proposed system.

- *The Prototyping Process*
    - The prototyping process involves four steps:
        - Identify basic requirements
        - Develop initial prototype
        - Knowledge worker review
        - Revise and enhance the prototype
- *The Advantages of Prototyping*
    - *T*he advantages of prototyping include
        - Encourages active knowledge worker participation.
        - Helps resolve discrepancies among knowledge workers.
        - Gives knowledge workers a feel for the final system.
        - Helps determine technical feasibility.
        - Helps sell the idea of a proposed system.
- *The Disadvantages of Prototyping*
    - The disadvantages of prototyping may include
        - Leads people to believe the final system will follow shortly.
        - Gives no indication of performance under operational conditions.
        - Leads the project team to forgo proper testing and documentation.
        -

2. Part 2. Database systems

This Part will discuss the Databases, The database architecture and Designing the Database.

LECTURE 1

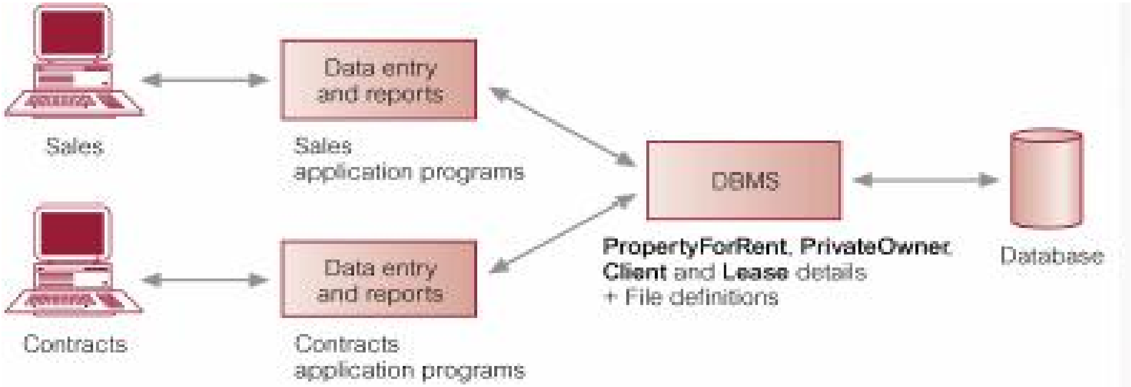
- An Introduction to Database Systems
- File based systems
- Database Architecture
- Advantages and Disadvantages of Database systems
- Database system development life cycle

2.1 An Introduction to Database Systems.

- Database (DB)

- Shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization.
- System catalog (metadata) provides description of data to enable program–data independence.
- Logically related data comprises entities, attributes, and relationships of an organization's information

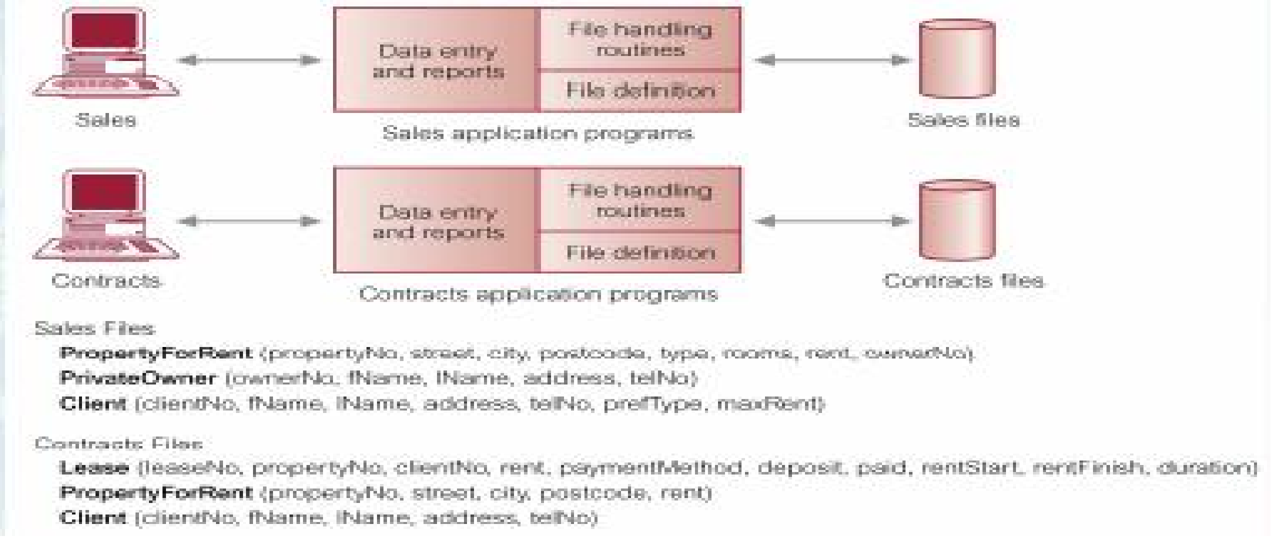
- Database Management System(DBMS)
  - A software system that enables users to define, create, and maintain the database and which provides controlled access to this database.
- Database system (DBS)
  - The collection of the database, the DBMS and application programs

○ 
Data entry and reports
Sales
Sales application programs
DBMS
Database
Data entry and reports
Contracts
Contracts application programs
PropertyForRent, PrivateOwner, Client and Lease details + File definitions

- Purchases using your credit card
- Booking a holiday at the travel agents
- Using the local library
- Taking out insurance
- Using the Internet
- Studying at university

2.1.1 File based systems (Pre-database systems)

- File-based Systems (definition)
  - Collection of application programs that perform services for the end users (e.g. reports).
  - Each program defines and manages its own data.



Sales Files
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files
Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)
PropertyForRent (propertyNo, street, city, postcode, rent)
Client (clientNo, fName, lName, address, telNo)

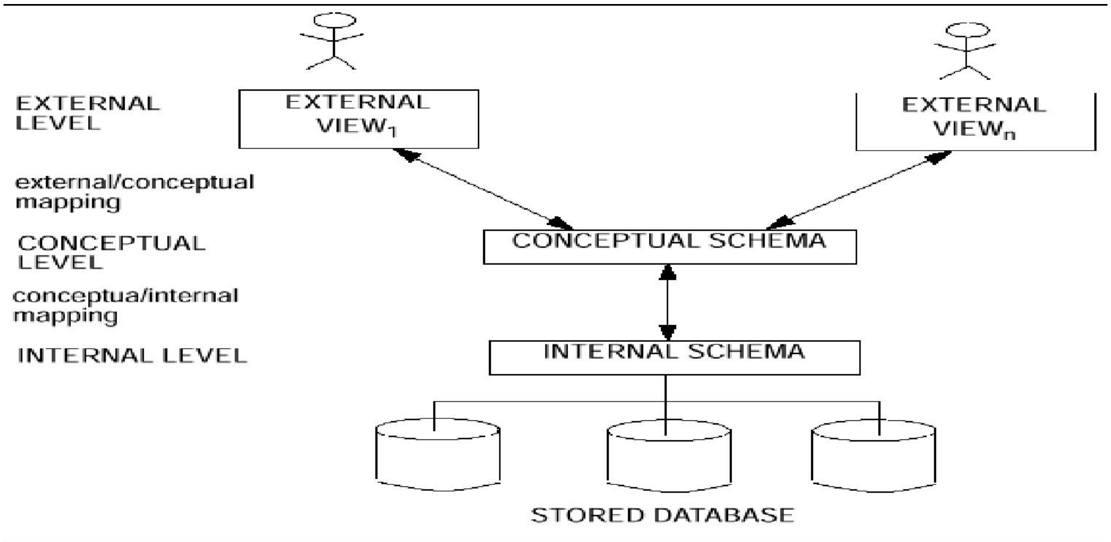
- Limitations of File-based Systems
    - Separation and isolation of data
        - Each program maintains its own set of data.
        - Users of one program may be unaware of potentially useful data held by other programs.
    - Duplication of data
        - Same data is held by different programs.
        - Wasted space and potentially different values and/or different formats for the same item
    - Data dependence
        - File structure is defined in the program code.
    - Incompatible file formats
        - Programs are written in different languages, and so cannot easily access each others files.
    - Fixed Queries/Proliferation of application programs
        - Programs are written to satisfy particular functions.
        - Any new requirement needs a new program.
- Database approach Arose because:
    - Definition of data was embedded in application programs, rather than being stored separately and independently.
    - No control over access and manipulation of data beyond that imposed by application programs.
    - Result
        - the database and Database Management System (DBMS).

2.1.2 Database Architecture

- The proposed and widely accepted database architecture is the three level architecture also known as the ANSI – SPARC architecture.
- ANSI-SPARC architecture is named after the committee that proposed it, the American National Standard Institute, Standards Planning And Requirements Committee.

- Objectives of the ANSI-SPARC architecture
  - All users should be able to access same data.
  - A user's view is immune to changes made in other views.
  - Users should not need to know physical database storage details.
  - DBA should be able to change database storage structures without affecting the users' views.
  - Internal structure of database should be unaffected by changes to physical aspects of storage.
  - DBA should be able to change conceptual structure of database without affecting all

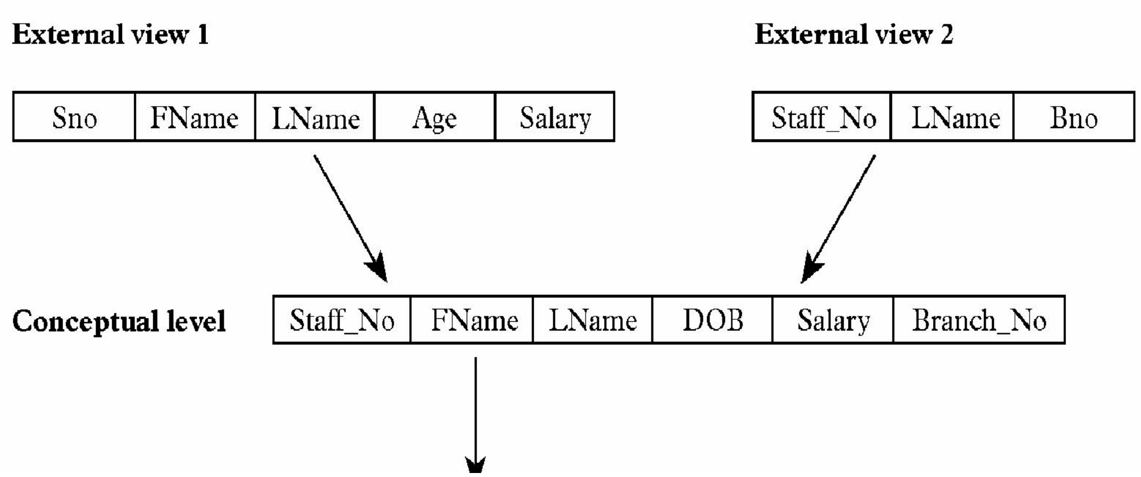
The ANSI- SPARC architecture

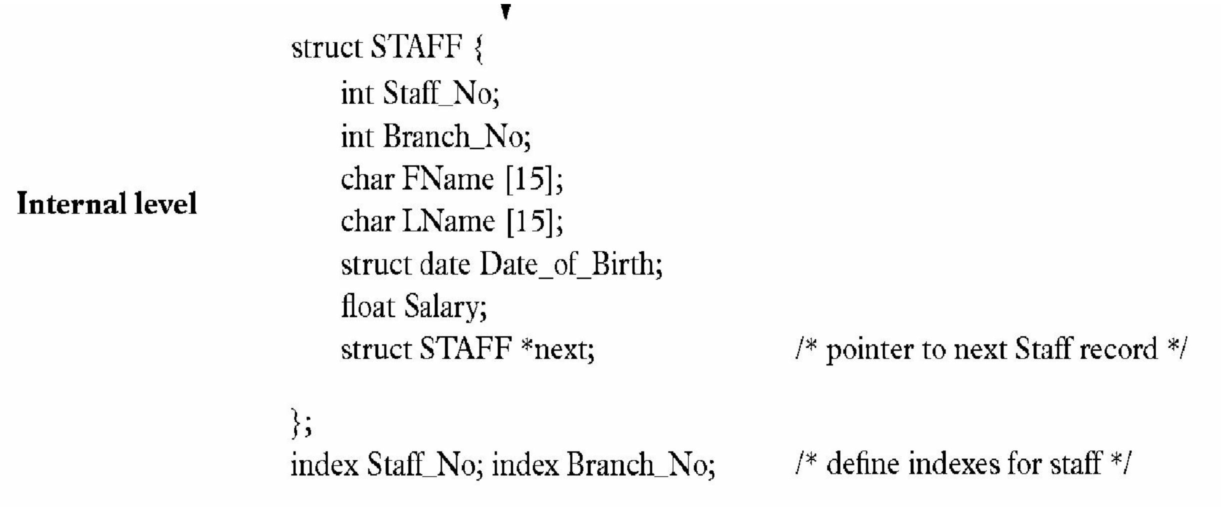


- **External Level**
  - The external level (also known as the user logical level) is the one closest to the users.
  - It is the level concerned with the way the data is seen by individual users.
  - The external schema is used to describe the external level.
  - It contains a description of a portion of the database that is of concern to the specific user.
  - The external view is described in terms of external records, which may be different form the actual stored records.
  - The view may have a different representation of the same data

- Conceptual Level
  - The level represents the community view of the database as seen by the database administrator.
  - It is also known as the community logical level or even sometimes just the logical level.
  - The conceptual schema is used to describe what data is stored in the database and the relationships among the data.
  - The description includes the structure and constraints for the whole database
  - the conceptual schema defines the logical structure of all data in the database.
  - The conceptual schema is defined by a Data Definition Language (DDL).
  - There is only one conceptual schema for the database.
- Internal Level
  - It is the description of the implementation of the conceptual schema by means of physical storage structures.
  - It summarizes how the data are stored on secondary storage devices such as disks and tapes.
  - However, it is still one level above the actual physical storage, which is usually managed by the operating system.
  - The internal level does not deal with the physical records but it deals with the internal records (stored records).
  - Details of how the address space is mapped to physical storage are highly system-specific (e.g. a block or a page)

Example architecture





Internal level

```
struct STAFF {
    int Staff_No;
    int Branch_No;
    char FName [15];
    char LName [15];
    struct date Date_of_Birth;
    float Salary;
    struct STAFF *next;              /* pointer to next Staff record */

};
index Staff_No; index Branch_No;    /* define indexes for staff */
```

  - Conceptual schema changes (e.g.addition/removal of entities).
  - Should not require changes to external schema or rewrites of application programs.

- Physical Data Independence
  - Refers to immunity of conceptual schema to changes in the internal schema.
  - Internal schema changes (e.g. using different file organizations, storage structures/devices).
  - Should not require change to conceptual or external schemas.
- Data Independence
  - Describes the immunity of the upper levels from the changes in the lower levels
- Mappings
  - The process of transforming requests and results between levels are called mappings.
  - The three-level architecture involves certain mappings— one conceptual/internal mapping and several external/conceptual mappings.
  - conceptual/internal mapping
    - defines the correspondence between the conceptual view and the stored databases
    - If a change is made to the storage definition, then the conceptual/internal mapping must be changed accordingly, so that the conceptual schema can remain invariant

- external/conceptual mappings
    - defines the correspondence between the conceptual view and the stored databases
    - If a change is made to the storage definition, then the conceptual/internal mapping must be changed accordingly, so that the conceptual schema can remain invariant

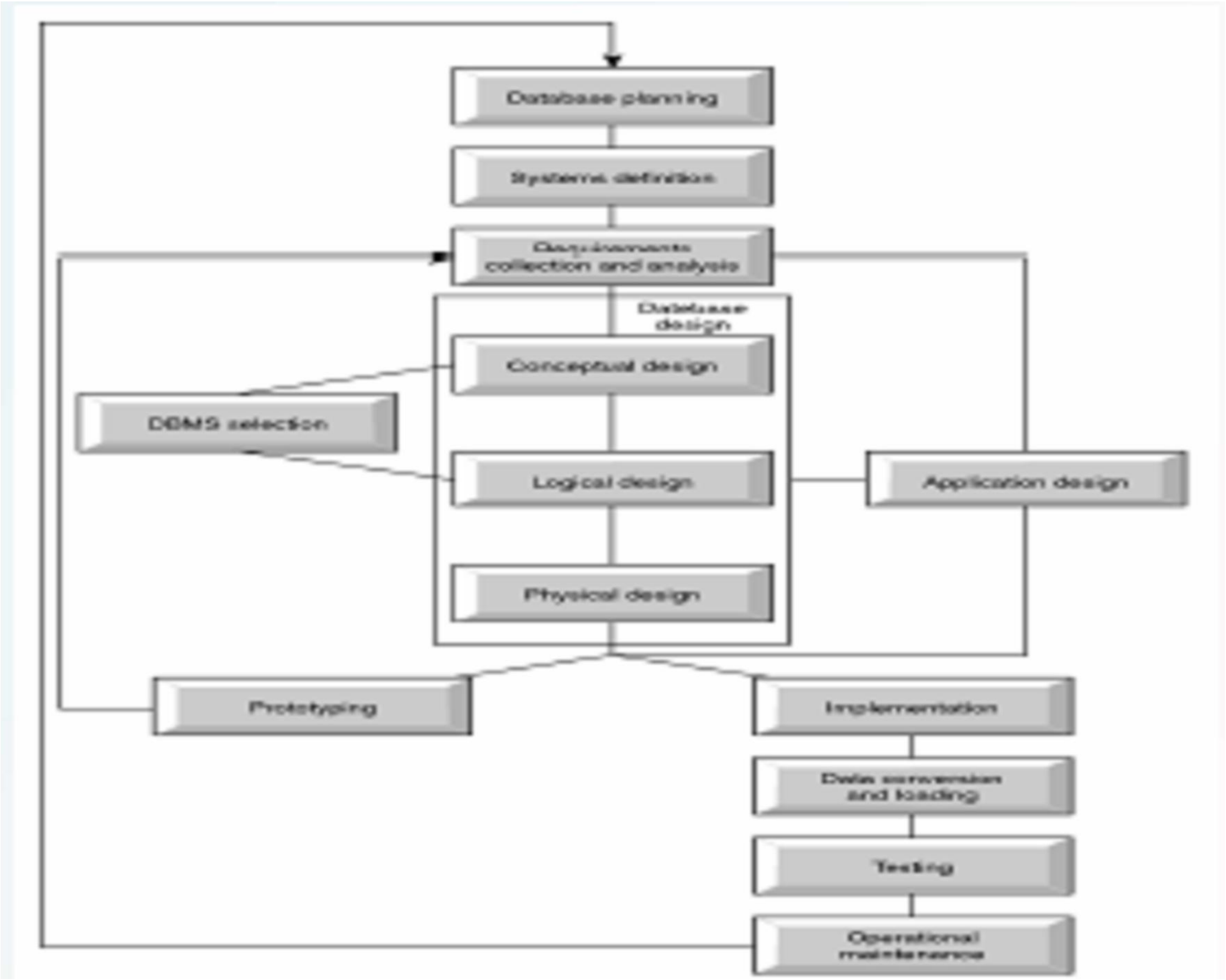
2.1.3 Advantages of Database systems

- Control of data redundancy
- Data consistency
- More information from the same amount of data.
- Sharing of data
- Improved data integrity
- Improved security
- Enforcement of standards
- Economy of scale
- Improved data accessibility and responsiveness
- Balanced conflicting requirements
- Increased productivity
- Improved maintenance through data independence
- Increased concurrency
- Improved backup and recovery services

2.1.4 Disadvantages of Database systems

- Complexity
- Size
- Cost of DBMS
- Additional hardware costs
- Cost of conversion
- Performance
- Higher impact of a failure

2.1.5 Lifecycle of Database system development



- Planning phase
    - Prior to the first phase of our requirement's elicitation, a plan must be put forward showing each stage of the life cycle and how it can be performed efficiently and effectively.
- System Definition phase
    - This is where the scope and boundaries of the database application, which includes the major application areas and user groups, are defined.
- Requirement collection and analysis Phase
    - It is concerned with collecting the data and analyzing the user requirements in order to build the database application

- Conceptual model
    - Create a conceptual schema (*High-Level*), A pictorial representation of the database, showing the main data items and relationships among them. e..g. ERD
- Logical Design
    - Map conceptual database schema to logical database schema
    - (*Representational*), it is also called an *Implementation* model e.g. Relational
- Physical design
    - Internal storage structures and file organisations are specified (*Physical*), e.g. B-tree
- Prototype
    - This is a working model of the database application, which allows the designers or users to visualize and evaluate how the final system will look and function
- Implementation
    - The phase involves the creation of the external, conceptual and internal definitions and the application programs.
- Data conversion and Loading
    - If we have a legacy system at this stage, then we need to transfer any existing data into the new database and converting any existing application.
- The testing phase
    - Validating user requirements against the application programs then tests the system. The testing phase aims at executing the application programs with intention to find errors
- Operational maintenance
    - This process involves monitoring & maintaining the system.

LECTURE 2

- Introduction to Relational Databases
- Introduction to Relational databases
- Terminology of relational model
- Properties of database relations.
- Relational Keys.
- Meaning of entity integrity and referential integrity.

2.2 Introduction to Relational Databases

- The relational approach was originally proposed in 1970’s .
- The first project that proved the practicality of the relational model is System R, developed at IBM’s San Jose Research Laboratory in 1976. examples oracle, Sybase, access.
- The model was proposed as a disciplined way of handling data using the rigour of mathematics, particularly set theory.
- This would enhance the concept of program-data independence and improve programmer activities.
- The relational model will have only values.
- Even references between data in different sets (relations) are represented by means of values.
- In the hierarchical and network model there are explicit references (pointers), which make them more complicated.
- The Relational approach is based on elementary mathematical relation theory.
- Its basic construct is a relation. A relation is also called a table.
- The data is organized in tables. The table has columns and rows.

2.2.1 Relational Model Terminologies

- A relation is a table with columns and rows.
- Only applies to logical structure of the database, not the physical structure.
- Attribute is a named column of a relation.
- Domain is the set of allowable values for one or more attributes.
- Tuple is a row of a relation.
- Degree is the number of attributes in a relation.
- Cardinality is the number of tuples in a relation.
- Relational Database is a collection of normalized relations with distinct relation names.
- Relation schema is a named relation defined by a set of attribute and domain name pairs.
- Relational database schema is a set of relation schemas, each with a distinct name.

2.2.2 Properties of Relations

- Relation name is distinct from all other relation names in relational schema.
- Each cell of relation contains exactly one atomic (single) value.
- Each attribute has a distinct name.
- Values of an attribute are all from the same domain.
- Each tuple is distinct; there are no duplicate tuples.
- Order of attributes has no significance.
- Order of tuples has no significance, theoretically.

Relational Keys

- Superkey
  - An attribute, or a set of attributes, that uniquely identifies a tuple within a relation.
- Candidate Key
  - Superkey (K) such that no proper subset is a superkey within the relation.
  - In each tuple of R, values of K uniquely identify that tuple (uniqueness).
  - No proper subset of K has the uniqueness property (irreducibility).
- Primary Key
  - Candidate key selected to identify tuples uniquely within relation.
- Alternate Keys
  - Candidate keys that are not selected to be primary key.
- Foreign Key
  - Attribute, or set of attributes, within one relation that matches candidate key of some (possibly same) relation.

2.2.3 Relational Database Integrity Constraints

- Database Integrity, Refers to the ability for the database to give correct and accurate results all the time.
- For a database to be able to achieve this integrity, it is subjected to the following integrity constraints
- Null

  - Represents value for an attribute that is currently unknown or not applicable for tuple
  - Deals with incomplete or exceptional data.
  - Represents the absence of a value and is not the same as zero or spaces, which are values.

- Null integrity Constraints
  - In a base relation, no attribute of a primary key can be null.
- Referential Integrity Constraints
  - If foreign key exists in a relation, either foreign key value must match a candidate key value of some tuple in its home relation or foreign key value must be wholly null.
- Enterprise (integrity) Constraints
  - Additional rules specified by users or database administrators.

Lecture 3 Database Design

- Approaches for data model development
- E-R Modeling (Conceptual design)
- Logical design

2.3 Approaches for data model development

- Once all the requirements have been collected and analyzed, the next step is to create a conceptual schema for the database, using a high-level conceptual data model.
- This step is called conceptual design.
- The conceptual schema is a concise description of the data requirements of the users and includes detailed descriptions of the entity types, relationships, and constraints;
- these are expressed using the concepts provided by the high-level data model.
- Because these concepts do not include implementation details, they are usually easier to understand and can be used to communicate with nontechnical users.
- The high-level conceptual schema can also be used as a reference to ensure that all users' data requirements are met and that the requirements do not conflict.

- This approach enables the database designers to concentrate on specifying the properties of the data, without being concerned with storage details. Consequently, it is easier for them to come up with a good conceptual database design
- ER modeling process is independent of the development platform (or software) Enables unambiguous, accurate communication of understanding of the data resource in an *abstract level*
- It is used for communications between database designer & users during systemanalysis & design process

Top-down Approach

- identify data entities
- determine attributes of the entities
- determine the nature of the relationships
- – usually results in a data model that is well organized
- but details can be easily overlooked.

Bottom-up Approach

- gather information on data used by the organization by data collection techniques..
- group into entities of which these data are attributes
- determine the nature of the relationships
- insures that no important data is overlooked but overall organization may not be so apparent

2.4 Entity-Relationship Modeling

- E/R model consists of:
  - – Entity type
  - – Attribute type
  - – Relationship type
- Entity type
  - A collection of objects (or a concept) that is identified by the enterprise as having an independent existence & share common properties (characteristics).
  - Most entities are recognizable business concepts, either concrete or abstract, about which various data are stored.

- In our usage, the term 'entity' will be synonymous with the terms 'entity type' or 'entity class'.
- – it refers to the generalization of occurrences
- Entity Occurrence or an instance
  - An object or concept that is uniquely identifiable.
  - Entity occurrence: Mr Zulu is a lecturer at the Copperbelt University
  - Entity type: Lecturer
- Weak & Strong Entity
  - Weak Entity Type is an entity type that is existence-dependent on some other entity type. Eg. Children of an employee in a company DB.
- Strong Entity Type
  - An entity type that is not existence-dependent on some other entity type. Eg. An employee entity type in a company DB

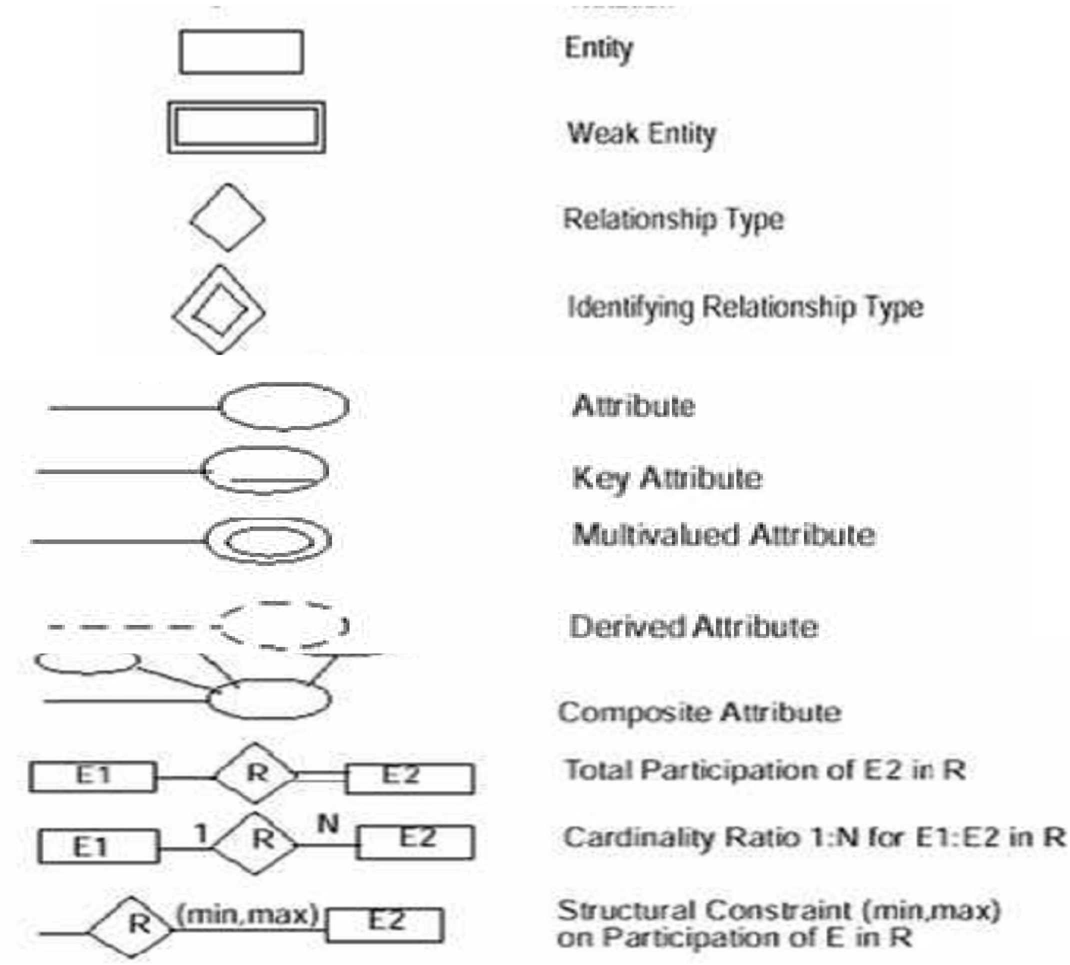
- Attributes
  - An attribute is a piece of information at the atomic level - that is it cannot be subdivided into meaningful component pieces
  - Property of an entity that is of interest to the organization
  - We are usually only interested in a subset of an entity's attributes which is directly related to the application
  - it is a good practice to have for each attribute a brief description
  - Attributes take on particular values in occurrences E.g. VehicleID of Car
  - Each entity will have one (or more) attribute that distinguishes it from all other entities, called an *Identifier or a Primary Key*
  - Where two or more attributes comprise the identifier it is called a *composite identifier.* E.g. Patient_Id (Patient_Name, Date_Of_Birth)
  - Types of attributes
    - Simple:
      - Each entity has a single atomic value for the attribute, for example SSN, CourseNo.
    - Composite:
      - The attribute may be composed of several components

        - Multivalued:
            - An entity may have multiple values for that attribute; for example Color of a Car or Previous Degrees of a Student
        - Derived:
            - The domain value of attribute can be determined from one or more other attributes.

- Relationship
    - Relationship = An association among entities
    - STUDENT Taught by LECTURER
- Unary or Recursive Relationships
    - A relationship where the same entity participates more than once in a different roles.
        - Eg. PERSON Married to PERSON
        - Staff **Supervises** Staff
    - Relationship Participation & Structural Constraints
        - Two types:
        - Cardinality and participation constraints.
            - Cardinality Constraints (Ratio)
                - Determines the number of possible relationships for each participating entity.
                - The number of allowed instances of entity B that can (or must) be associated with each instance entity A.
                - Most common degree for relationships is binary with cardinality ratios of one-to-one (1:1), one-to-many (1:M) or many-to-many (M:N).
            - Participation Constraints
                - is about the importance of the instances' *participation* in a specific relationship.
                - If it is applied to every instance of an entity, then it is called a *total* or a *mandatory participation (*it is a *must* for each instance belongs to that entity type).

- If only parts of the instances participate in a relationship or in other words, an instance may or may not participate in that relationship, then this is called a *partial* or *optional participation*.

ER-diagram Chen Notation


Entity
Weak Entity
Relationship Type
Identifying Relationship Type
Attribute
Key Attribute
Multivalued Attribute
Derived Attribute
Composite Attribute
E1
R
E2
Total Participation of E2 in R
E1
1
R
N
E2
Cardinality Ratio 1:N for E1:E2 in R
R
(min,max)
E2
Structural Constraint (min,max) on Participation of E in R

- Lets consider an example database application, called COMPANY, it serves to illustrate the basic ER model concepts and their use in schema design.
- We first list the data requirements for the database here, and then create its conceptual schema step by step as we introduce the modeling concepts of the ER model.
- **Details To be discussed in class**

Requirements specification

The COMPANY database keeps track of a company's employees, departments, and projects.

1. The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.

2. A department controls a number of projects, each of which has a unique name, a unique number, and a single location.

3. We store each employee's name, social security number. i address, salary, sex, and birth date. An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same department. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.

4. We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, sex, birth date, and relationship to the employee.

2.5 Logical Design

- Mapping from conceptual model (EER-M) into a relational schema
- This approach involves applying transformation rules (or steps) to the EER model in order to achieve a relational logical schema.
- NB: the steps below should not be taken as golden rules.
- Application requirements vary from one domain to another and from one user to another.
- Hence, the following steps should be only taken as guidelines

Step 1

- For each regular entity type (ignore those with subclasses at this point) on your diagram create a table and nominate a primary key (PK) for that relation.

Step 2

- For each weak entity type, create a relation.
- The PK is a combination of the identifier of the parent entity and the identifier of the weak or dependent entity (i.e. a composite PK).

Step 3

- Each unary or binary type relationship with a one-to-one cardinality is mapped by placing a foreign key attribute in one of the relational as the linking attribute.

- If the relationship is mandatory OR optional from both sides, then it does not make a difference as to where to place the foreign key
- If it is mandatory (total participation) from one side and optional (partial participation) from the other side, then the PK of the optional side is inserted as a foreign key at the mandatory side.

Step 4

- For Each unary or binary one-to-many relationship type.
- The PK from the 1-end is inserted as a FK at the N-end.

Step 5

- Each N:M relationship of any degree is mapped to a new linking relation whose PK includes the keys of all participating relations (a composite PK).
- Include all relationship's attributes in the new relation.

Step 6

- Each multi-value attribute mapped to a new relation.
- The new relation should include an attribute pertaining to the main relation as a foreign key

Example to be discussed in class

3. Part 3. System implementation techniques

Database Management, Database recovery, Data Security, transaction processing and concurrent control.

LECTURE 1.

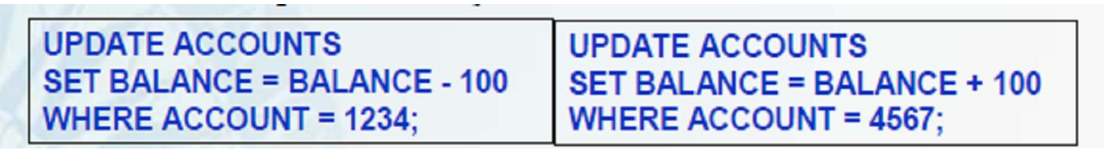
- Transaction Processing
- Transaction management

3.1 Transaction Processing

- A Transaction is a program unit (deletion, creation, updating etc) whose execution preserves the consistency of a database.
- To ensure that the above is met a transaction must be
  - Atomic
  - Execute to completion
  - Not execute at all

- A transaction has 4 Basic properties also known as the ACID properties
  - Atomicity
  - Consistency
  - Isolation
  - Durability
- atomicity
  - Also known as the all nothing property
  - A transaction is an individual unit that is either performed in its entirety or not performed at all.
- Consistency
  - The transaction must transform the database from one consistency state to another consistency state
- Isolation
  - Transactions execute independently of one another.
- Durability
  - The effects of a successfully completed transaction are permanently recorded in the database and must never be lost due to subsequent failure

3.1.1 Transactions Management

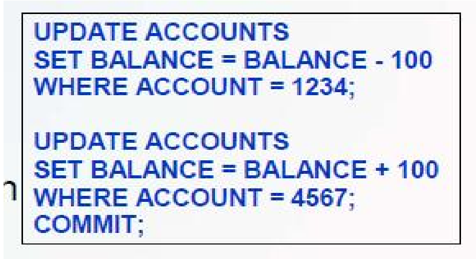
- Scenario:
  - transferring money from one account to another in one bank requires the SQL commands:
  - 

```
UPDATE ACCOUNTS
SET BALANCE = BALANCE - 100
WHERE ACCOUNT = 1234;
```

```
UPDATE ACCOUNTS
SET BALANCE = BALANCE + 100
WHERE ACCOUNT = 4567;
```

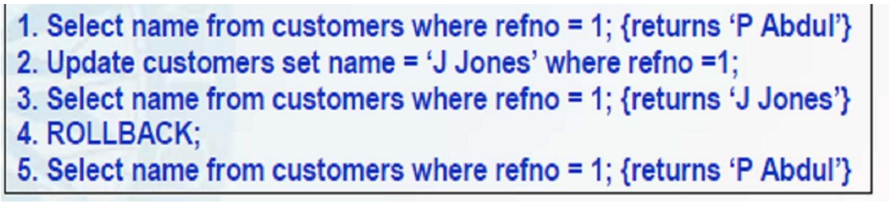
  - Transferring £100 from account 1234 to Account 4567.
  - Together they comprise a single transaction.
  - Potential problem:
    - Database crash !!
  - Crash may leave the Database in inconsistent state

  - in the example, it would be better if neither of the commands had been executed.
  - Transaction integrity demands that the effects of a transaction should be either complete or not enacted at all.
- Commit/Rollback protocol
  - exist to support transaction integrity
  - Commit is when changes to a Database are made permanent when a Database crashes, any changes that have not been committed will be lost
- We can issue an explicit commit command when both of these update commands have been issued



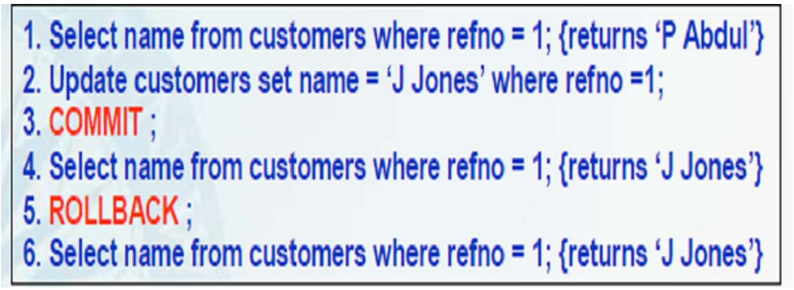
```
UPDATE ACCOUNTS
SET BALANCE = BALANCE - 100
WHERE ACCOUNT = 1234;

UPDATE ACCOUNTS
SET BALANCE = BALANCE + 100
WHERE ACCOUNT = 4567;
COMMIT;
```

- Rollback
  - a mechanism to undo the effects of a transaction.
  - when issued all of the Database changes since last commit are undone



```
1. Select name from customers where refno = 1; {returns 'P Abdul'}
2. Update customers set name = 'J Jones' where refno =1;
3. Select name from customers where refno = 1; {returns 'J Jones'}
4. ROLLBACK;
5. Select name from customers where refno = 1; {returns 'P Abdul'}
```

  - the Rollback in 4 undoes the effect of the Update in 2
  - because the Update has not been committed
  - – Suppose we issue a Commit command



```
1. Select name from customers where refno = 1; {returns 'P Abdul'}
2. Update customers set name = 'J Jones' where refno =1;
3. COMMIT ;
4. Select name from customers where refno = 1; {returns 'J Jones'}
5. ROLLBACK ;
6. Select name from customers where refno = 1; {returns 'J Jones'}
```

  - The Commit command in 3 makes the change permanen

LECTURE 2

- Database Recovery
- Recovery Facilities
- Recovery Techniques
- Database Security
- Database security measures

3.2 Database Recovery

- This is the process of restoring the database to a consistency state after a failure
- Types of failure
  - System failure – system entering an undesirable state, like an infinite loop or deadlock.
  - Logic Errors – Bad programmes
  - Hardware failures – eg hard disk crashes

3.2.1 Recovery Facilities

- The DBMS provides the following facilities to recover from failure.
- Backup Mechanism -: Periodical backup of the system
- Logging Facility -: Keeps truck of the current state of the transaction and the database
- Checkpoint Facility -: enables update to the database to be made permanent
- Recovery manager -: Allows the system t o restore the database to a consistency state following a failure

3.2.2 Recovery Techniques

- Deferred updates :
  - This were you use a log to record all new transactions and the log will be used to update the database at a later stage.
- Immediate updates :
  - This is where updates are made to the records immediately and the update is kept in both the log and the database
- Shadow Paging
  - Two page tables are maintained during the life of a transaction

        - The current page and the shadow page
        - When the transaction starts the two tables are the same
        - The shadow page is not changed and is used to restore the database in the event of a failure
        - The current page is used to record all updates to the database
        - When the transaction completes the current page becomes the shadow page and the shadow page is garbage collected.

3.3 Database Security

- Mechanism that protects the database against intentional or accidental threats.
- It encompasses hardware , software , people and data
- It is considered in relation to the following situations:
    - Theft/Fraud
    - Loss of Confidentiality
    - Loss of privacy
    - Loss of Integrity
    - Loss of Availability
- Theft/Fraud
    - This is the acquisition of data illegally without permission. The acquired data is normally used for malicious activities. Eg. Bank account details
- Confidentiality
    - Refers to the need to maintain secrecy over the data usually that which is critical to an organization
- Secrecy
    - Refers to the need to protect data about individuals , loss would lead to legal action taken against the organization
- Integrity
    - Loss results in invalid and corrupted data
- Availability
    - Data must be available to authorized persons at an appropriate time (when as required)

- Loss leads to the inability to access data.

3.3.1 Database Security measures

- Measures that can be used to safeguard databases from anticipated threats
  - Authorization
  - Authentication
  - Views or subschema
  - encryption
- Authentication
  - Mechanisms that determines whether a user is s/he what s/he claims to be
  - Establishing proof of identity
    - Physical traits
    - Pin codes
    - Cards etc
- Authorization
  - Also known as Access control
  - This is the granting of rights and privileges that enables a user to have access to the system
- Views or subschema
  - A view is a virtual table that does not exist in the database but is produced upon request by particular user
- Encryption
  - This is the encoding of the data by a special algorithm that renders the data unreadable by any program without the decryption key.

LECTURE 3.

- Concurrency control
- Concurrency problems
- Simple locking protocols
- Two phase locking protocol

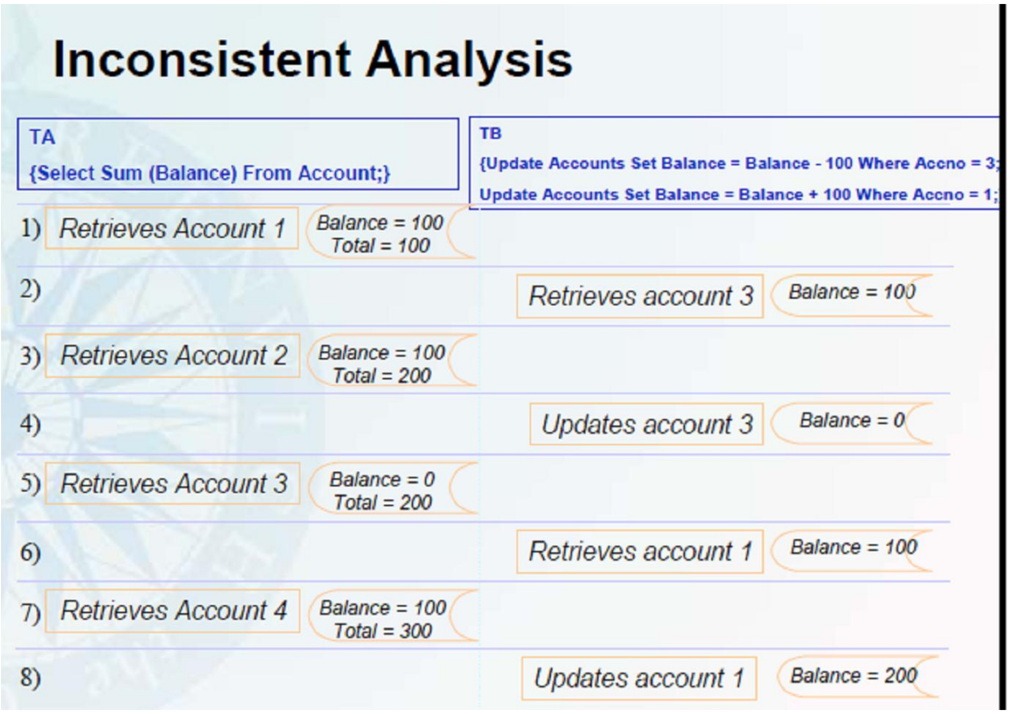
3.4 Concurrency Control

- Concurrency
    - The process describing two or more users accessing the database at the same time and transactions are interleaved.
    - Undesirable results may occur, hence the need for concurrency control
- Concurrency problems
- The Lost Update Problem
    - The following situation might arise:
        - 1) TA reads Account record 1234. Value of balance is 150.
        - 2) TB reads Account record 1234. Value of balance is 150.
        - 3) TA increases to 250 (150+100).
        - 4) TB increases to 350 (150+200).
        - 5) TA writes back balance of 250.
        - 6) TB writes back balance of 350.
    - The account should have a balance of 450, not 350.
    - The update performed by TA has been lost
- The uncommitted dependency problem
    - When does it occur?
    - Another transaction may start using data that has not yet been committed.
    - – Effects: the 2nd transaction will use false information.
    - – Example
        - TA
            - Update Accounts
            - Set Balance = Balance - 100
            - Where Accno = 1234;
            - If Balance < 0.00 Then Rollback Else Commit;
        - TB
            - Delete from Accounts

        - Where Balance < 0.00;

    - 1) TA retrieves Account 1234. Value of balance is 50.
    - 2) TA reduces balance by 100. Leaving it as -50.
    - 3) TA writes back value of -50.
    - 4) TB retrieves Account 1234. Balance is -50.
    - 5) TB deletes Account 1234 as it has negative balance.
    - 6) TA rolls back update. Too late! the account has been deleted
    - TB used uncommitted data.

- Inconsistent Analysis problem
    - A transaction accesses records while are they being updated by another transaction.
    - Example
    - TA
        - Select Sum (Balance)
        - From Account;
    - TB
        - Update Accounts
        - Set Balance = Balance - 100 Where Accno = 3;
        - Update Accounts
        - Set Balance = Balance + 100 Where Accno = 1;
    - 2nd transaction transfers money from one account to another.
    - Hence, should have no effect on **TA** result

- 

Inconsistent Analysis

	TA
	{Select Sum (Balance) From Account;}
			TB
			{Update Accounts Set Balance = Balance - 100 Where Accno = 3;
			Update Accounts Set Balance = Balance + 100 Where Accno = 1;	

1)	Retrieves Account 1	Balance = 100
		Total = 100		

2)			Retrieves account 3	Balance = 100

3)	Retrieves Account 2	Balance = 100
		Total = 200		

4)			Updates account 3	Balance = 0

5)	Retrieves Account 3	Balance = 0
		Total = 200		

6)			Retrieves account 1	Balance = 100

7)	Retrieves Account 4	Balance = 100
		Total = 300		

8)			Updates account 1	Balance = 200

3.4.1 Locking Protocols

- How to avoid all previous problems?

- ***Lock*** the object to prevent access by other transactions

- A transaction releases the object when it finishes with it

- Other transactions need to queue until the object is released

- The lock could be shared or exclusive

  - A Shared lock ***S*** is placed on an object that is being accessed for read only purposes

    - many S locks may be placed

    - an X lock must wait

  - An exclusive lock *X may be placed*, when an object is being altered

    - No other lock may be placed

    - All transactions must wait

- The Lost Update Problem:

  - TA will place an X lock on Account 1234 before it starts update

- The uncommitted dependency:

- TA will lock TB out from Account 1234 until it has completed the rollback

➢ The inconsistent analysis:

- TA will place an S lock on all of the account records

3.4.2 Problems with Locking

➢ Appropriate locking can guarantee correctness, However, it also introduces potential undesirable effects:

➢ Deadlock

- No transactions can proceed; each waiting on lock held by another.

➢ Starvation

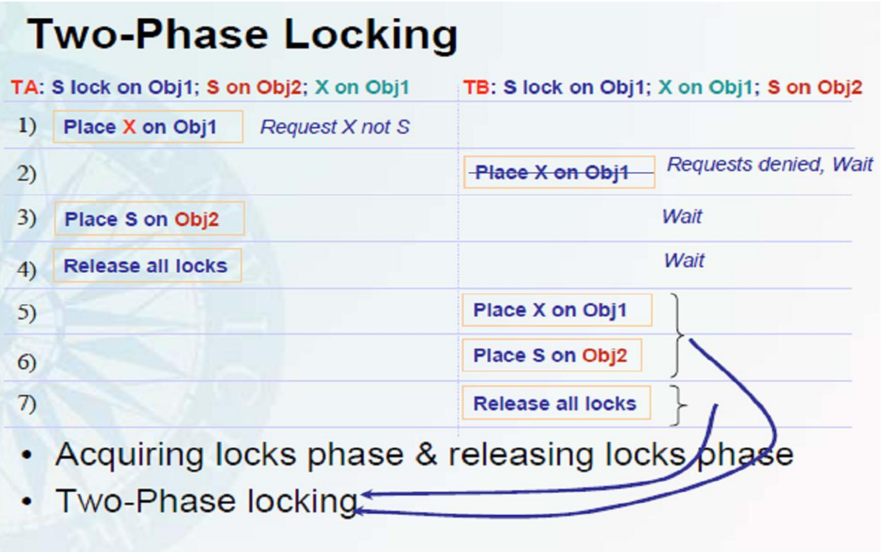
- One transaction is permanently "frozen out" of access to data.
- Results into reduced performance
  - Locking introduces delays while waiting for locks to be released.

3.4.3 Two-Phase Locking

➢ A transaction follows a 2 phase locking protocol if all operations precede the first unlock operations in the transaction.

➢ According to this protocol every transaction can be divided into two phases

- Growing phase
  - A transaction acquires all the locks needed but cannot release any locks
- Shrinking Phase
  - A transaction releases its locks but cannot acquire any locks



Two-Phase Locking

	TA: S lock on Obj1; S on Obj2; X on Obj1		TB: S lock on Obj1; X on Obj1; S on Obj2	
1)	Place X on Obj1	*Request X not S*		
2)			~~Place X on Obj1~~	*Requests denied, Wait*
3)	Place S on Obj2			*Wait*
4)	Release all locks			*Wait*
5)			Place X on Obj1	
6)			Place S on Obj2	
7)			Release all locks	

- Acquiring locks phase & releasing locks phase
- Two-Phase locking