Subject: Fundamentals of Computer Science

Unit IV- Database Systems and Computer Systems and Development

Database Systems: File-Oriented Approach, Database-oriented Approach-Components of Database system, Advantages & Disadvantages of Database approach, Applications of Database systems, Database views, Three-schema architecture, Database models-Hierarchical model, Network Model, relational Model, Object-oriented Data Model, Components of database management systems, Retrieving Data through Queries

Computer Systems and Development: Investigation, Analysis, Design, system processing and general program design, Presentation to management and users, Implementation, Documents.

Part-A: Database Systems

I. File Oriented Approach

Traditionally, data was organized in file formats

A File Oriented Approach or File Management System is a type of software that manages data files in a computer system. It has limited capabilities and is designed to manage individual or group files, such as special office documents and records. It may display report details, like owner, creation date, state of completion and similar features useful in an office environment.

A file management system is also known as a file manager

In the earlier age the computer system was used to store business records and produce different types of information. They were generally faster and more accurate than equivalent manual systems. These systems stored groups of records in separate file and so they were called file processing system.

In a typical file processing systems, each department has its own files, designed especially for those applications. The department itself works with the data processing staff, sets policies or standards for the format and maintenance of its files.

➢ Advantages of File-Oriented Approach:

1. **Backup:**
   - It is possible to take faster and automatic back-up of database stored in files of computer-based systems.
   - Computer Systems provide functionalities to serve this purpose.
   - It is also possible to develop specific application program for this purpose.
2. **Compactness**:
   - It is possible to store data compactly.

3. **Data Retrieval**:

- Computer-based systems provide enhanced data retrieval techniques to retrieve data stored in files in easy and efficient way.

4. **Editing**:

- It is easy to edit any information stored in computers in form of files.
- Specific application programs or editing software can be used for this purpose.

5. **Remote Access**:

- It is possible to access data remotely, so it is not necessary for a user to remain present at location where these data are kept.

6. **Sharing**:

- Data stored in files of computer-based systems can be shared among multiple users at a same time.

➢ Disadvantages of File oriented Approach

1) Data redundancy

In computer system many files are likely in the different formats and the programs are written in different programming languages. Moreover, the same information may be duplicated in several files, this duplication of data is known as data redundancy.

Example: The address and telephone number of a particular customer may appear in a file that consist of saving account records and in a file that consist of checking account record.

2) Data inconsistency

Various copies contain the same type of data and updates may not take place at all copies, which reflect different kinds of information.

Example: A changed customer address may be reflected in savings account records but not elsewhere in the system.

3) Difficulty in accessing data

In file processing system it is very difficult to access the data in a specific way and it also require a special application program which carry out new task.

4) Data isolation

Because data are scattered in various files and files may be in different formats, writing new applications program to retrieve the appropriate data is difficult.

5) Integrity problem

Database must satisfy a particular consistency constraint. These constraints are added in application program.

Example: The balance of a bank account may never fall below a prescribed amount.

6) Atomicity problem

A computer system, like any other mechanical or electrical devices, is subject to failure. In many applications, it is crucial that if failure occurs, the data be restored to the consistent state that existed prior to the failure.

7) Concurrent access anomalies

If two programs run concurrently it is important to has supervision. But supervision is difficult to provide because data is decentralized in file processing system. In such an environment, interaction updates may result in inconsistent data.

8) Security problems

In this type not every user of the database system should be able to access all the data.

II. Database Oriented Approach

"A Database Management System (DBMS) is a collection of interrelated data and a set of programs to access the given data".

The DBMS serves as the intermediary between the user and the database. The database structure itself is stored as a collection of files, so, we can access the data in those files through the DBMS.

➢ Functions of Database management approach

1. Database management allows us to manipulating and sharing the database among various users and applications.
2. It also provides a recovery system to restore data in consistent state.
3. It allows us to creating standard and user- friendly interface and supports
4. DBMS enhances the database programmer's productivity by supporting code reusability.
5. It also enhances the system portability and support extensibility.

➢ Advantages of Database Systems

1. Improved data sharing

DBMS helps to create an environment in which end users have better access to more and better-managed data. Such access makes it possible for end users to respond quickly to changes in their environment.

2. Improved data security

The more users access the data, the greater the risks of data security breaches. Corporations invest considerable amounts of time, effort, and money to ensure that corporate data are used properly. A DBMS provides a framework for better enforcement of data privacy and security policies.

3. Better data integration

Wider access to well-managed data promotes an integrated view of the organization's operations and a clearer view of the big picture. It becomes much easier to see how actions in one segment of the company affect other segments.

4. Minimized data inconsistency

Data inconsistency exists when different versions of the same data appear in different places. For example, data inconsistency exists when a company's sales department stores a sales representative's name as "Bill Brown" and the company's personnel department stores that same person's name as "William G. Brown," or when the company's regional sales office shows the price of a product as $45.95 and its national sales office shows the same product's price as $43.95. The probability of data inconsistency is greatly reduced in a properly designed database

5. Improved data access

The DBMS makes it possible to produce quick answers to adhoc queries. From a database perspective, a query is a specific request issued to the DBMS for data manipulation—for example, to read or update the data. Simply put, a query is a question, The DBMS sends back an answer (called the query result set) to the application. For example, end users, when dealing with large amounts of sales data, might want quick answers to questions (adhoc queries) such as:

1. What was the dollar volume of sales by product during the past six months?
2. What is the sales bonus figure for each of our salespeople during the past three months?
3. How many of our customers have credit balances of 3,000 or more?

6. Improved decision making

Better-managed data and improved data access make it possible to generate better-quality information, on which better decisions are based. The quality of the information generated depends on the quality of the underlying data. Data quality is a comprehensive approach to promoting the accuracy, validity, and timeliness of the data. While the DBMS does not

Guarantee data quality, it provides a framework to facilitate data quality initiatives.

7. Increased end-user productivity

The availability of data, combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between success and failure in the global economy.

➢ Disadvantages of Database Systems

1. Increased costs

Database systems require sophisticated hardware and software and highly skilled personnel. The cost of maintaining the hardware, software, and personnel required to operate and manage a database system can be substantial. Training, licensing, and regulation compliance costs are often overlooked when database systems are implemented.

2. Management complexity

Database systems interface with many different technologies and have a significant impact on a company's resources and culture. The changes introduced by the adoption of a database system must be properly managed to ensure that they help advance the company's objectives. Given the fact that database systems hold crucial company data that are accessed from multiple sources, security issues must be assessed constantly.

3. Maintaining currency

To maximize the efficiency of the database system, you must keep your system current. Therefore, you must perform frequent updates and apply the latest patches and security measures to all components.

Because database technology advances rapidly, personnel training costs tend to be significant. Given the heavy investment in technology and personnel training, companies might be reluctant to change database vendors.

As a consequence, vendors are less likely to offer pricing point advantages to existing customers, and those customers might be limited in their choice of database system components.

4. Frequent upgrade/replacement cycles

DBMS vendors frequently upgrade their products by adding new functionality. Such new features often come bundled in new upgrade versions of the software. Some of these versions require hardware upgrades. Not only do the upgrades themselves cost money, but it also costs money to train database users and administrators to properly use and manage the new features.

III.Applications of Database Systems

Due the evolution of Database management system, companies are getting more from their work because they can keep records of everything. Also it makes them faster to search information and records about any people or product that makes them more effective in work. So here we are sharing some of the applications and **uses of database management system (DBMS)**.

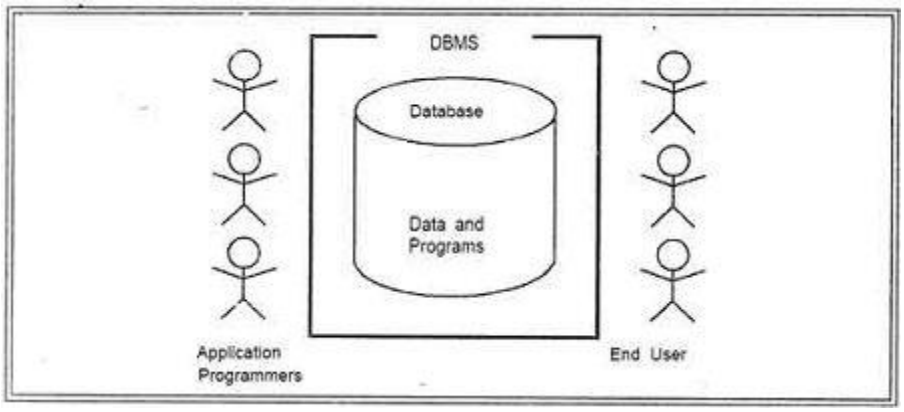
Sno	Name of the Sector	Use of DBMS
1	Banking	For customer information, account activities, payments, deposits, loans, etc.
2	Airlines	For reservations and schedule information.
3	Universities	For student information, course registrations, colleges and grades.
4	Telecommunication	It helps to keep call records, monthly bills, maintaining balances, etc.
5	Finance	For storing information about stock, sales, and purchases of financial instruments like stocks and bonds.
6	Sales	Use for storing customer, product & sales information.
7	Manufacturing	It is used for the management of supply chain and for tracking production of items. Inventories status in warehouses.
8	HR Management	For information about employees, salaries, payroll, deduction, generation of paychecks, etc.

IV. Components of Database Systems

Components of the Database System Environment

There are five major components in the database system environment and their interrelationships are.

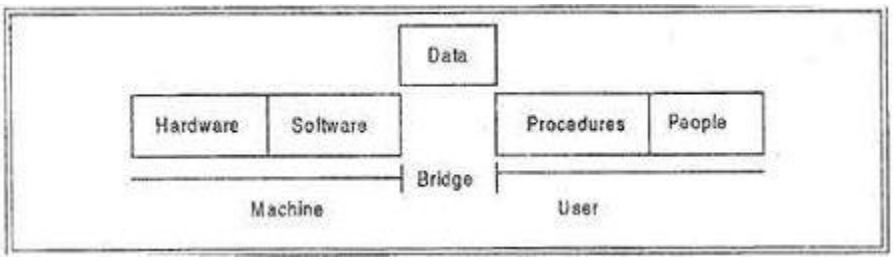
- Hardware
- Software
- Data
- Users
- Procedures



1. Hardware: The hardware is the actual computer system used for keeping and accessing the database. Databases run on a' range of machines, from Microcomputers to large mainframes. Other hardware issues for a DBMS includes database machines, which is hardware designed specifically to support a database system.

2. Software: The software is the actual DBMS. Between the physical databases itself (i.e. the data as actually stored) and the users of the system is a layer of software, usually called the Database Management System or DBMS. All requests from users for access to the database are handled by the DBMS.

The DBMS allows the users to communicate with the database i.e.it act as a mediator between the database and the users. The DBMS controls the access and helps to maintain the consistency of the data.



3. Data: It is the most important component of DBMS environment from the end users point of view. Data acts as a bridge between the machine components and the user components. The database contains the operational data and the meta-data (data about data)

The database should contain all the data needed by the organization. One of the major features of databases is that the actual data are separated from the programs that use the data. A database should always be designed, built and populated for a particular audience and for a specific purpose.

4. Users: There are a number of users who can access or retrieve data on demand using the applications and interfaces provided by the DBMS. Each type of user needs different software capabilities. The users of a database system can be classified in the following groups, depending on their degrees of expertise or the mode of their interactions with the DBMS. The users can be:

• Naive Users
• Online Users
• Application Programmers
• Sophisticated Users
• Data Base Administrator (DBA)

Naive Users: Naive Users are those users who need not be aware of the presence of the database system or any other system supporting their usage. Naive users are end users of the database who work through a menu driven application program, where the type and range of response is always indicated to the user.

A user of an Automatic Teller Machine (ATM) falls in this category. The user is instructed through each step of a transaction. He or she then responds by pressing a coded key or entering a numeric value. The operations that can be performed by valve users are very limited and affect only a precise portion of the database. For example, in the case of the user of the Automatic Teller Machine, user's action affects only one or more of his/her own accounts.

Online Users: Online users are those who may communicate with the database directly via an online terminal or indirectly via a user interface and application program. These users are aware of the presence of the database system and may have acquired a certain amount of expertise with in the limited interaction permitted with a database.

Sophisticated Users: Such users interact with the system without writing programs.

Instead, they form their requests in database query language. Each such query is submitted to a very processor whose function is to breakdown DML statement into instructions that the storage manager understands

Specialized Users: Such users are those, who write specialized database application that do not fit into the fractional data-processing framework. For example: Computer-aided design systems, knowledge base and expert system, systems that store data with complex data types (for example, graphics data and audio data).

Application Programmers: Professional programmers are those who are responsible for developing application programs or user interface. The application programs could be written using general purpose programming language or the commands available to manipulate a database.

Database Administrator: The database administrator (DBA) is the person or group in charge for implementing the database system, within an organization. The "DBA has all the system privileges allowed by the DBMS and can assign (grant) and remove (revoke) levels of access (privileges) to and from other users. DBA is also responsible for the evaluation, selection and implementation of DBMS package.

5. Procedures: Procedures refer to the instructions and rules that govern the design and use of the database. The users of the system and the staff that manage the database require documented procedures on how to use or run the system.

These may consist of instructions on how to:

• Log on to the DBMS.

• Use a particular DBMS facility or application program.

• Start and stop the DBMS.

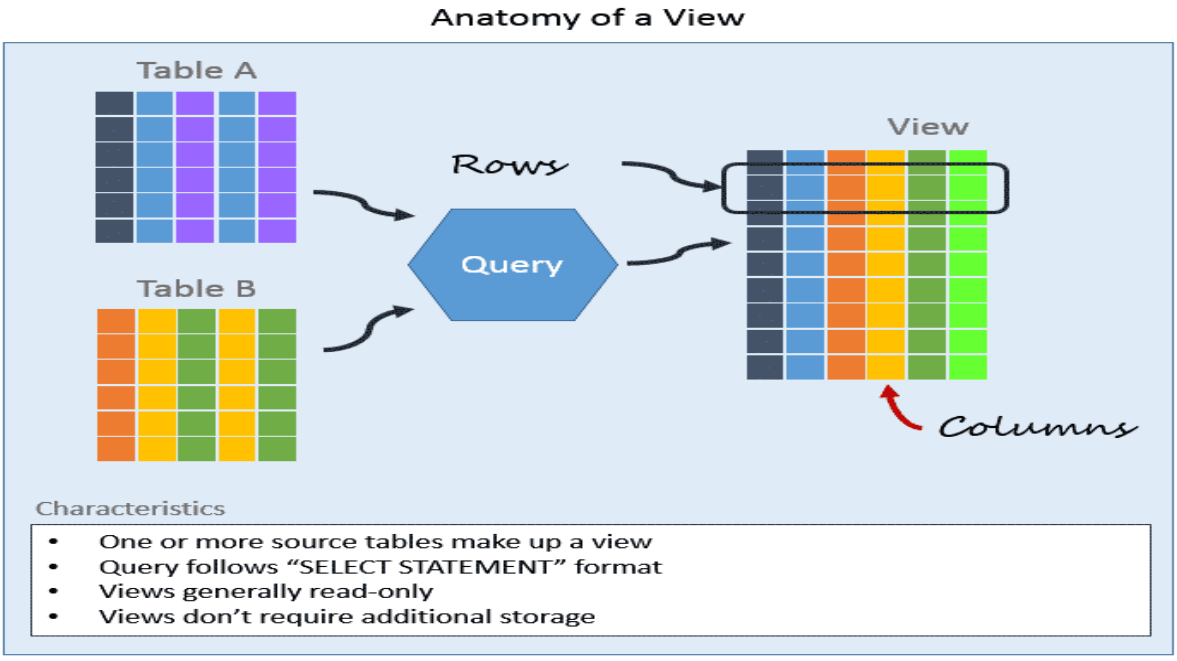
• Make backup copies of the database.

• Handle hardware or software failures.

Change the structure of a table, reorganize the database across multiple disks, improve performance, or archive data to secondary storage.

V. Database Views

A database view is a searchable object in a database which is defined by a query. Though a view doesn't store data, some refer to views as "virtual tables," A view can combine data from two or more table, using joins, and also just contain a subset of information. This makes them convenient to abstract, or hide, complicated queries.




```
CREATE VIEW PopularBooks AS SELECT ISBN, Title, Author, PublishDate

FROM Books WHERE IsPopular = 1
```

➢ Benefits of a Database View

1. Enforce Business Rules – Use views to define business rules, such as when an items is active, or what is meant by "popular." By placing complicated or misunderstood business logic into the view, you can be sure to present a unified portrayal of the data. This increases use and quality.

2. Consistency – Simplify complicated query logic and calculations by hiding it behind the view's definition. Once defined they calculations are reference from the view rather than being restated in separate queries. This makes for less mistakes and easier maintenance of code.

3. Security – Restrict access to a table, yet allow users to access non-confidential data via views. For example, you can restrict access to the employee table, that contains social security numbers, but allow access to a view containing name and phone number.

4. Simplicity – Databases with many tables possess complex relationships, which can be difficult to navigate if you aren't comfortable using Joins. Use views to provide a "flattened" view of the database for reporting or ad-hoc queries.

5. Space – Views take up very little space, as the data is stored once in the source table. Some DBMS all you to create an index on a view, so in some cases views do take up more space than the definition.

➢ Disadvantages of Views

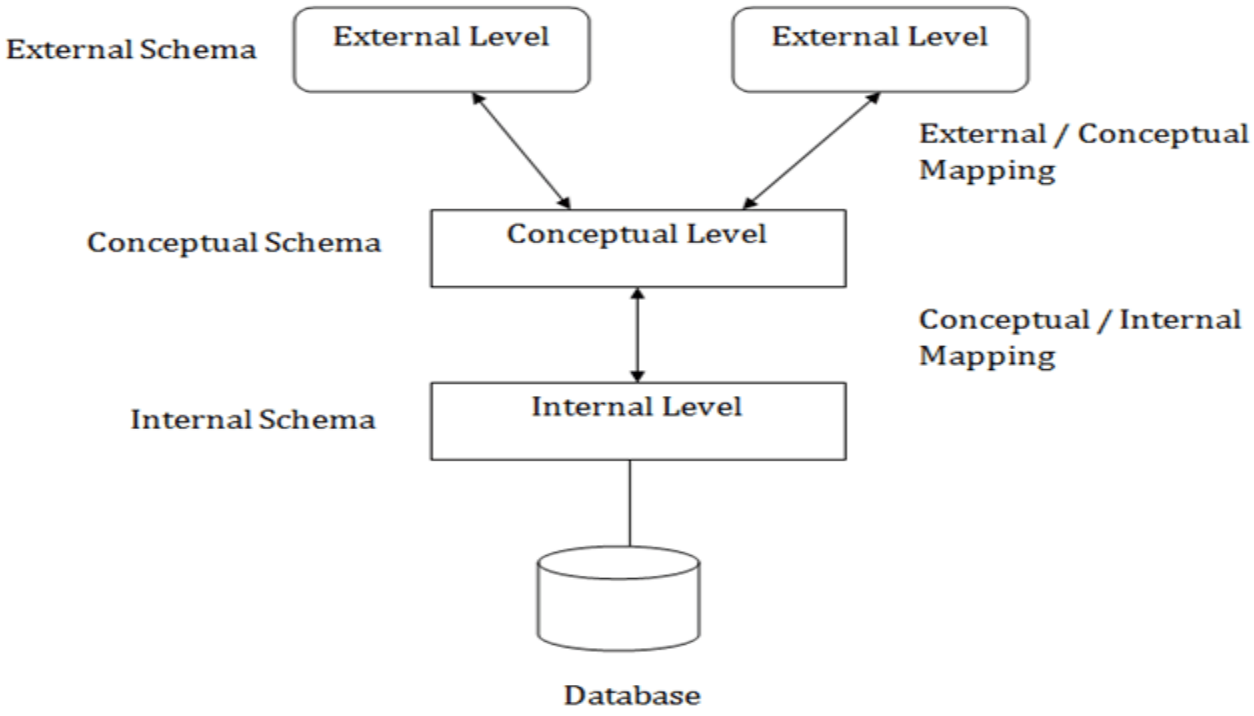
1. Performance – What may seem like a simple query against a view could turn out to be a hugely complex job for the database engine. That is because each time a view is referenced, the query used to define it, is rerun.

2. Modifications – Not all views support INSERT, UPDATE, or DELETE operations. In general, in order to support these operations, the primary key and required fields must be present in the view. Complex multi-table views are generally read only.

VI. Three Schema Architecture

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture.
- This framework is used to describe the structure of a specific database system.
- It is used to separate the user applications and physical database.
- It consists of three-levels. It breaks the database down into three different categories.

The three-schema architecture is as follows:



In the above diagram:

- It shows the DBMS architecture.
- Mapping is used to transform the request and response between various database levels of architecture.
- Mapping is not good for small DBMS because it takes more time.
- In External / Conceptual mapping, it is necessary to transform the request from external level to conceptual schema.
- In Conceptual / Internal mapping, DBMS transform the request from the conceptual to internal level.

1. Internal Level

- It describes the physical storage structure of the database.
- The internal schema is also known as a **physical schema**.
- It uses the physical data model. It is used to define that how the data will be stored in a block.
- The physical level is used to describe complex low-level data structures in detail.

2. Conceptual Level

- It describes the design of a database at the conceptual level.
- Conceptual level is also known as **logical level**.
- It describes the structure of the whole database.
- It describes what data are to be stored in the database and also describes what relationship exists among those data.
- Internal details such as an implementation of the data structure are hidden.
- Programmers and database administrators work at this level.

3. External Level

- A database contains several schemas that sometimes called as subschema. The subschema is used to describe the different view of the database.
- An external schema is also known as **view schema**.
- Each view schema describes the database part that a particular user group is interested and hides the remaining database from that user group.
- The view schema describes the end user interaction with database system

VII. Database Models

Database Model: It determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized and manipulated.

There are four common types of database model that are useful for different types of data or information. Depending upon your specific needs, one of these models can be used.

1. Hierarchical databases.
2. Network databases.
3. Relational databases.
4. Object-oriented databases

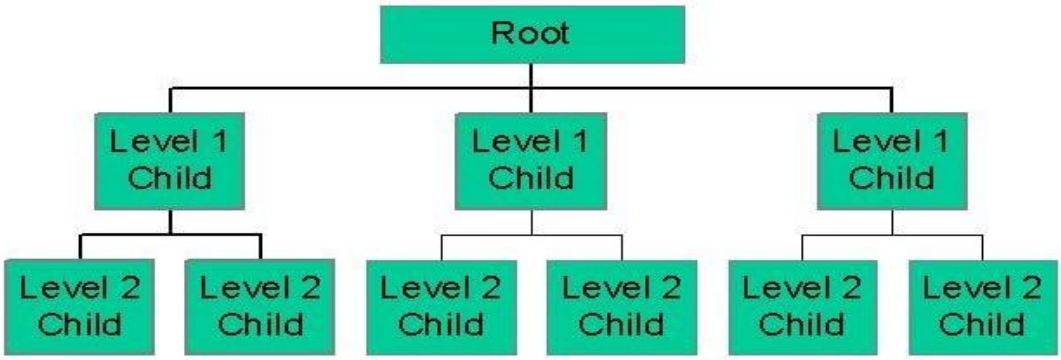
1. Hierarchical Database Model

It is one of the oldest database model developed by IBM for information Management System. In a hierarchical database model, the data is organized into a tree-like structure. In simple language we can say that it is a set of organized data in tree structure.

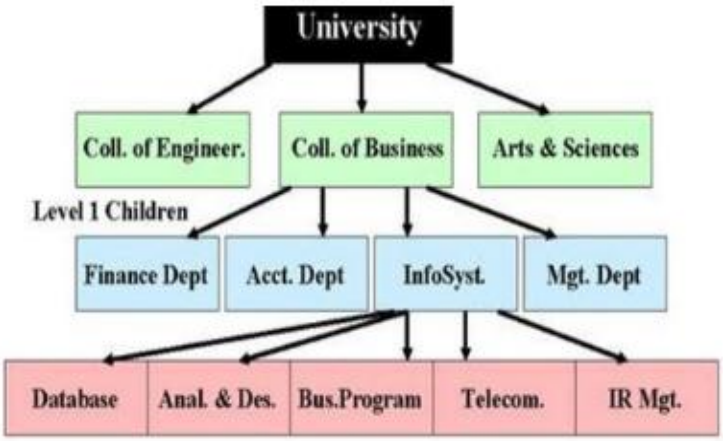
This type of Database model is rarely used now a days. Its structure is like a tree with nodes representing records and branches representing fields. The windows registry used in Windows XP is an example of a hierarchical database. Configuration settings are stored as tree structures with nodes.

The following figure shows the generalized the structure of Hierarchical database model in which data is stored in the form of tree like structure (data represented or stored in root node, parent node and child node).

Hierarchical database model



The following figure shows the example of hierarchical database model for the university management system. This type of database employs the "parent-child" relationship of storing data.



Advantages

1. The model allows us easy addition and deletion of new information.
2. Data at the top of the Hierarchy is very fast to access.
3. It worked well with linear data storage mediums such as tapes.
4. It relates well to anything that works through a one to many relationships.

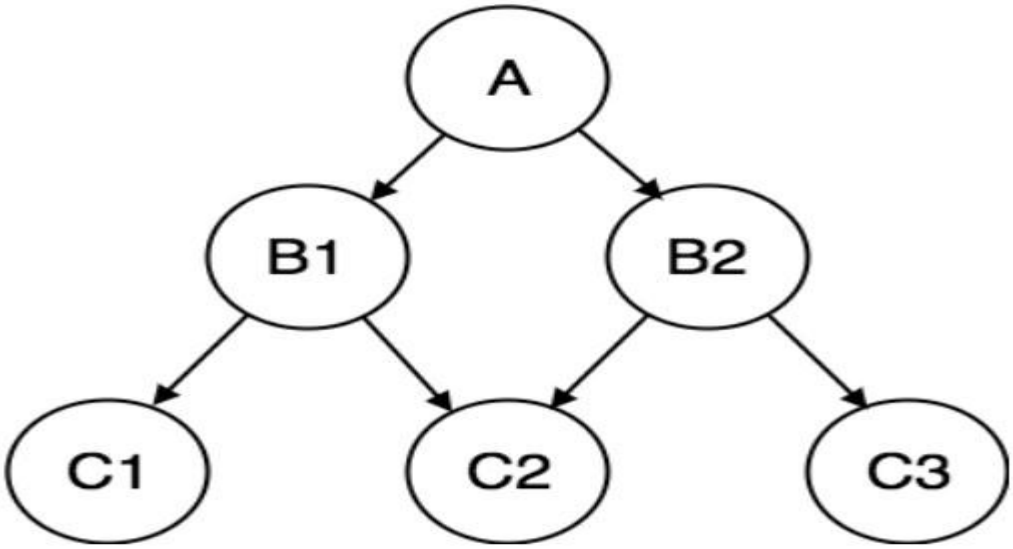
For example; there is a president with many managers below them, and those managers have many employees below them, but each employee has only one manager.

Disadvantages

1. It requires data to be repetitively stored in many different entities.
2. Now a day there is no longer use of linear data storage mediums such as tapes.
3. Searching for data requires the DBMS to run through the entire model from top to bottom until the required information is found, making queries very slow.
4. This model support only one to many relationships, many to many relationships are not supported.

2. Network Database Model

This is looks like a Hierarchical database model due to which many times it is called as modified version of Hierarchical database. Network database model organized data more like a graph and can have more than one parent node. The network model is a database model conceived as a flexible way of representing objects and their relationships



Advantages

1. The network model is conceptually simple and easy to design.
2. The network model can represent redundancy in data more effectively than in the hierarchical model.
3. The network model can handle the one to many and many to many relationships which is real help in modeling the real-life situations.
4. The data access is easier and flexible than the hierarchical model.
5. The network model is better than the hierarchical model in isolating the programs from the complex physical storage details.

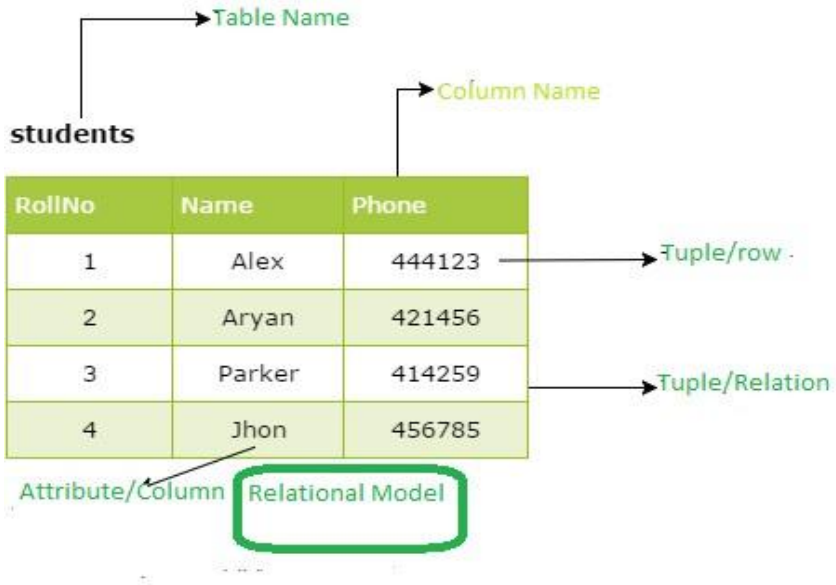
Disadvantages:

1. All the records are maintained using pointers and hence the whole database structure becomes very complex.
2. The insertion, deletion and updating operations of any record require the large number of pointers adjustments.
3. The structural changes to the database is very difficult.

3. Relational Database Model

A relational database is developed by E. F. Codd in 1970. The various software systems used to maintain relational databases are known as a relational database management system (RDBMS). In this model, data is organized in rows and columns structure i.e., two-dimensional tables and the relationship is maintained by storing a common field. It consists of three major components.

In relational model, three key terms are heavily used such as relations, attributes, and domains. A relation nothing but is a table with rows and columns. The named columns of the relation are called as attributes, and finally the domain is nothing but the set of values the attributes can take. The following figure gives us the overview of rational database model.



Terminology used in Relational Model

Tuple: Each row in a table is known as tuple.

Cardinality of a relation: The number of tuples in a relation determines its cardinality. In this case, the relation has a cardinality of 4.

Degree of a relation: Each column in the tuple is called an attribute. The number of attributes in a relation determines its degree. The relation in figure has a degree of 3.

Advantages

1. Relational model is one of the most popular used database model.
2. In relational model, changes in the database structure do not affect the data access.
3. The revision of any information as tables consisting of rows and columns is much easier to understand.
4. The relational database supports both data independence and structure independence concept which makes the database design, maintenance, administration and usage much easier than the other models.
5. In this we can write complex query to accesses or modify the data from database.
6. It is easier to maintain security as compare to other models.

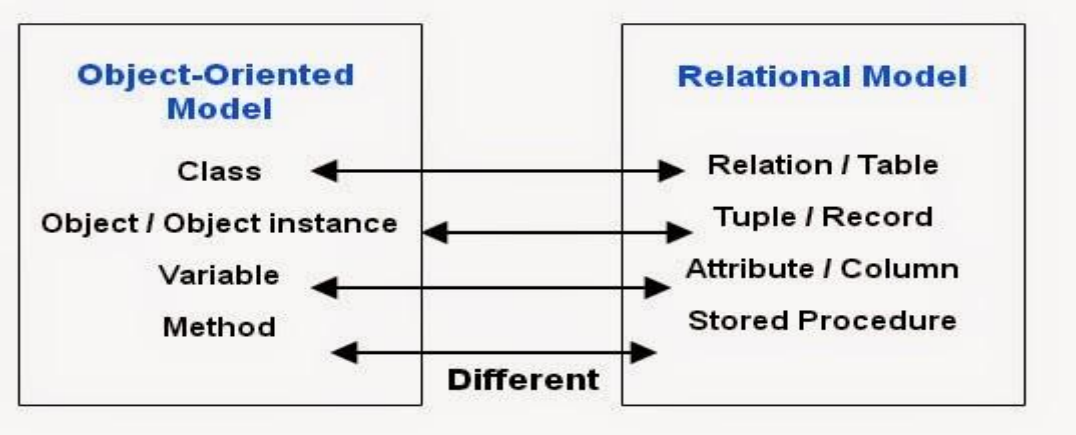
Disadvantages

1. Mapping of objects in relational database is very difficult.
2. Object oriented paradigm is missing in relation model.
3. Data Integrity is difficult to ensure with Relational database.
4. Relational Model is not suitable for huge database but suitable for small database.
5. Hardware overheads are incurred which make it costly.
6. Ease of design can lead to bad design.
7. Relational database system hides the implementation complexities and the physical data storage details from the users.

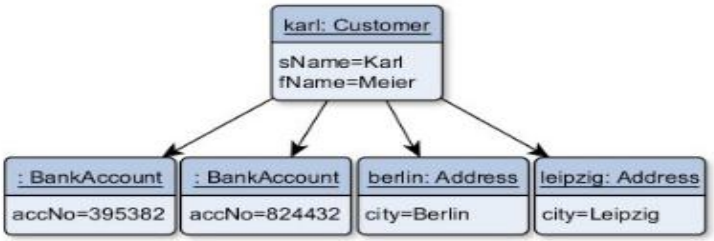
4. Object-Oriented Database Model

An object database is a system in which information is represented in the form of objects as used in object-oriented programming. Object oriented databases are different from relational databases which are table-oriented. The object-oriented data model is based on the object-oriented- programming language concept, which is now in wide use. Inheritance, polymorphism, overloading. object-identity, encapsulation and information hiding with methods to provide an interface to objects, are among the key concepts of object-oriented programming that have found applications in data modelling. The object-oriented data model also supports a rich type system, including structured and collection types.

The following figure shows the difference between relation and object-oriented database model.



The following figure shows an example of object-oriented model



.

Advantages

1. Object database can handle different types of data while relational data base handles a single data. Unlike traditional databases like hierarchical, network or relational, the object-oriented databases can handle the different types of data, for example, pictures, voice video, including text,numbers and so on.
2. Object-oriented databases provide us code reusability, real world modelling, and improved reliability and flexibility.
3. The object-oriented database is having low maintenance costs as compared to other model because most of the tasks within the system are encapsulated, they may be reused and incorporated into new tasks.

Disadvantages

1. There is no universally defined data model for an OODBMS, and most models lack a theoretical foundation.
2. In comparison to RDBMSs the use of OODBMS is still relatively limited.
3. There is a Lack of support for security in OODBMSs that do not provide adequate security mechanisms.
4. The system more complex than that of traditional DBMSs.

VIII. Retrieving Data through Queries

The Structured Query Language offers database users a powerful and flexible data retrieval mechanism — the SELECT statement. In this topic, we'll take a look at the general form of the SELECT statement and compose a few sample database queries together.

let's begin our exploration of the SELECT statement..

The General Form of the SELECT Statement

The general form of the SELECT statement appears below:

SELECT *select_list* FROM *source* WHERE *condition(s)* GROUP BY *expression* HAVING *condition* ORDER BY *expression*

The first line of the statement tells the SQL processor that this command is a **SELECT** statement and that we wish to retrieve information from a database. The *select_list* allows us to specify the type of information we wish to retrieve. The **FROM** clause in the second line specifies the specific database table(s) involved and the **WHERE** clause gives us the capability to limit the results to those records that meet the specified *condition(s)*. The final three clauses represent advanced features

The easiest way to learn SQL is by example. let's begin looking at some database queries. Throughout this article, we'll use the employee's table from the fictional XYZ Corporation human resources database to illustrate all of our queries.

Here's the entire table:

EmployeeID	LastName	FirstName	Salary	ReportsTo
1	Smith	John	32000	2
2	Scampi	Sue	45000	NULL
3	Kendall	Tom	29500	2
4	Jones	Abraham	35000	2
5	Allen	Bill	17250	4
6	Reynolds	Allison	19500	4
7	Johnson	Katie	21000	3

Retrieving an Entire Table

XYZ Corporation's Director of Human Resources receives a monthly report providing salary and reporting information for each company employee. The generation of this report is an example of the SELECT statement's simplest form. It simply retrieves all of the information contained within a database table — every column and every row. Here's the query that will accomplish this result:

```
SELECT *
FROM employees
```

The asterisk (*) appearing in the *select_list* is a wildcard used to inform the database that we would like to retrieve information from all of the columns in the employee's table identified in the FROM clause. We wanted to retrieve all of the information in the database, so it wasn't necessary to use a WHERE clause to restrict the rows selected from the table. Here's what our query results look like:

EmployeeID	LastName	FirstName	Salary	ReportsTo
----------	--------	---------	------	---------
1	Smith	John	32000	2
2	Scampi	Sue	45000	NULL
3	Kendall	Tom	29500	2
4	Jones	Abraham	35000	2
5	Allen	Bill	17250	4
6	Reynolds	Allison	19500	4
7	Johnson	Katie	21000	3

Part –B Computer Systems and Development

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

1. A system must have some **structure and behavior** which is designed to achieve a predefined objective.
2. **Interconnectivity** and **interdependence** must exist among the system components.
3. 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 −

1.Organization

Organization implies structure and order. It is the arrangement of components that helps to achieve predetermined objectives.

2.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.

3.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.

4. Integration

Integration is concerned with how 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.

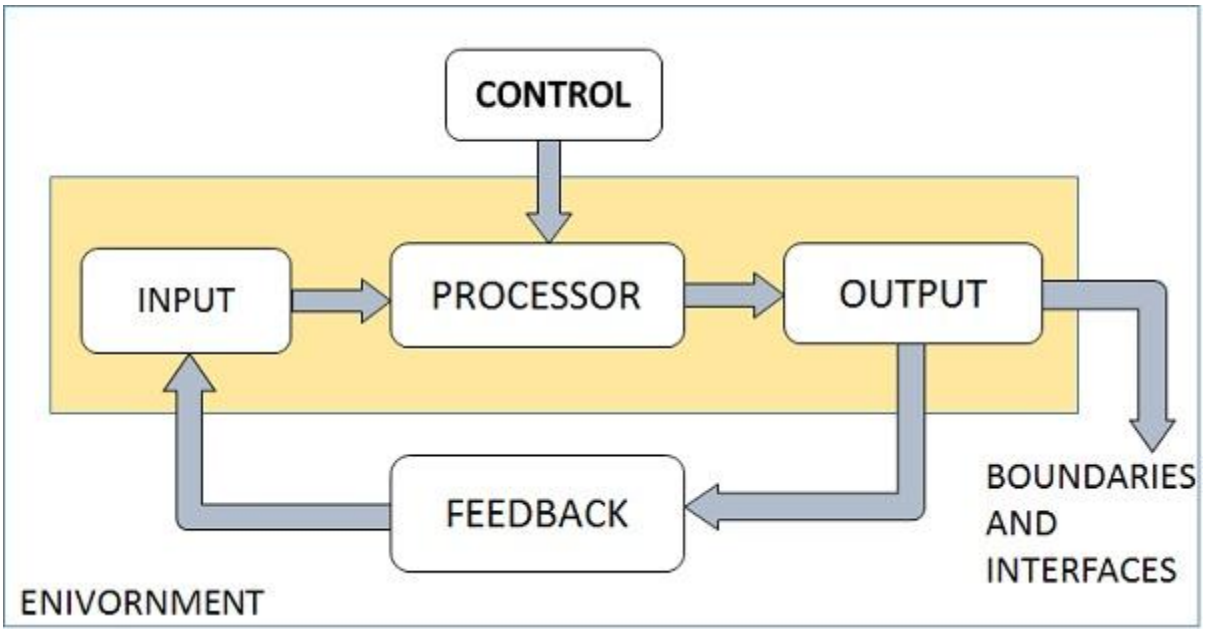
5. Central Objective

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 "super system" 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.

Categories of Information

There are three categories of information related to managerial levels and the decision managers make.

Volume of Information	Type of Information	Information Level	Management Level	System Support
Low Consensed	Unstructured	Strategic Information	Upper	DSS
Medium Moderately Processed	Moderately Structured	Management Control Information	Middle	MIS
Large Detail Reports	Highly Structured	Operational Information	Lower	DPS

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).

I. System Development Process

System Development is systematic process which includes phases such as investigation, planning, analysis, design, deployment, and maintenance. Here, in this topic, we will primarily focus on

1. Investigation
2. Analysis/System Analysis
3. Design
4. System processing and general program design
5. Presentation to management and users
6. Implementation
7. Documents

1. Investigation

In this phase, the existing system is studied closely to discover the true nature of the problem which led to the request for investigation. Normally the system is first investigated to find whether there are sufficient problems to permit a full scale integration. This preliminary investigation may also reveal that a new system isr equired or the old one could be modified to handle the problems . The following methods are used for finding the facts of the system.

I. Study system documentation
II. Conduct interviews
III. Send out Questionnaires
IV. Observe the system in operation
V. Use the automatic methods

I. **Study System Documentation**:

The analyst collects copies of all forms that are currently being used in the system. Studying a completed form enables the analyst to determine whether the form needs to be redesigned.

II.Interviews:

It is one of the most valuable methods available to the analyst for learning about an existing system. The following guidelines should be followed to conduct interviews

- Choose the right person to interview
- Get authorization from management to conduct an interview
- Prepare the interview questionnaire in advance
- Write down the points to be covered and make a list of relevant questions to be asked.
- Conduct the interview without differing the specified topic.
- Prepare a report after completion of interviews
- The main advantage of an interview is that it is a flexible method of gathering information.

III.Questionnaires:

The purpose of a questionnaire is to obtain facts or opinions. For example, the question "How Many orders do you process on an average day" needs a factual answer, where as "what do you think is the reason for the large number of returned orders? "Calls for an opinion to be expressed. The questionnaire should be designed so that:

- Directions and questions are stated clearly and unambiguously, negative wording should be avoided.
- Only relevant information is requested.
- It is brief and easy to complete.
- Controversial questions are avoided
- Questions are worded to avoid giving clues to expected answers.
- Compared to the interview, the questionnaire is economical way of obtaining similar information from large number of people.

IV. Observe the System In Operation:

It gives the analyst a chance to obtain data first hand. It enables to verify whether the data obtained from interviews or questionnaires is consistent or not. We can able to determine which one gave the reliable information. The advantages of this observation is

- Clarification
- Know the employees better

Disadvantages:

- People are behaving differently when they are being observed.
- The analyst must be physically present in order to observe operations; it may not possible in all shifts.

V. Automated Methods:

Data is normally obtained by personal observation can be obtained by automatic devices also. Many devices containing microprocessors can monitor their usage. For example, if we need to know often a word processor is used, it is not necessary for someone to stand around to obtain this information. The word processor can do this automatically; every time that someone uses it, a record is kept.

2. Systems Analysis

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

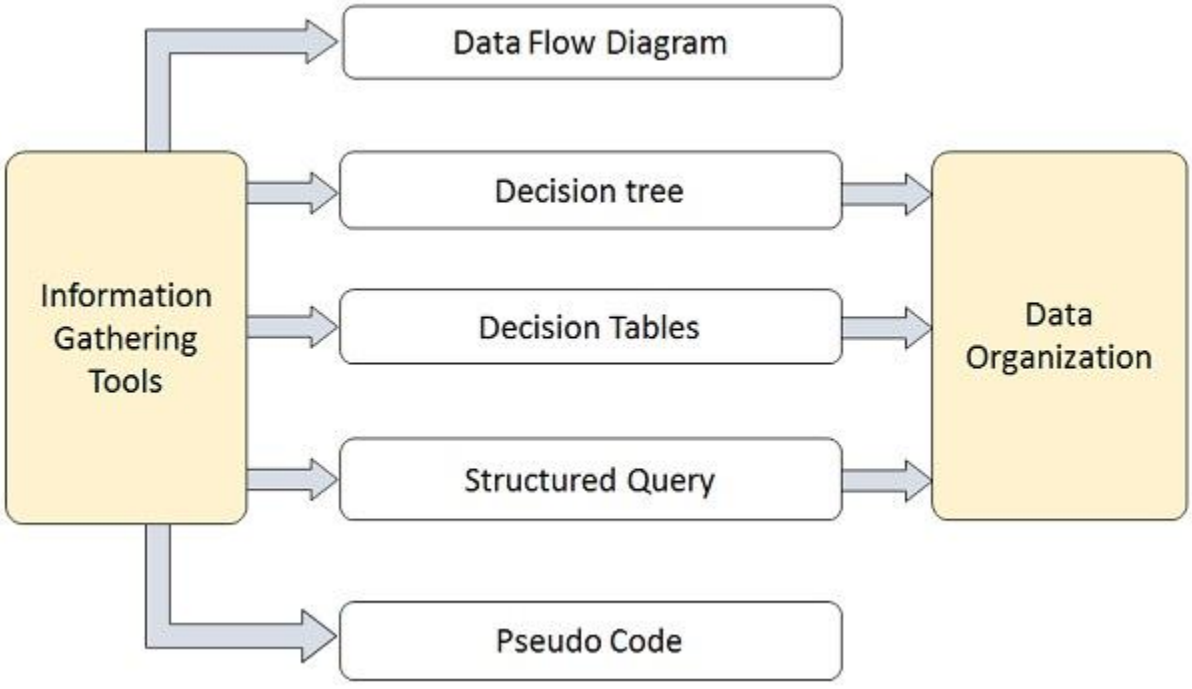
System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

Structured Analysis is a development method that allows the analyst to understand the system and its activities in a logical way.

Analysis Tools

During Structured Analysis, various tools and techniques are used for system development. They are −

a) Data Flow Diagrams
b) Data Dictionary
c) Decision Trees
d) Decision Tables
e) Structured English
f) Pseudocode



a) Data Flow Diagrams (DFD) or Bubble Chart

It is a technique developed by Larry Constantine to express the requirements of system in a graphical form.

- It shows the flow of data between various functions of system and specifies how the current system is implemented.
- It is an initial stage of design phase that functionally divides the requirement specifications down to the lowest level of detail.
- Its graphical nature makes it a good communication tool between user and analyst or analyst and system designer.

- It gives an overview of what data a system processes, what transformations are performed, what data are stored, what results are produced and where they flow.

Basic Elements of DFD

It is easy to understand and quite effective when the required design is not clear and the user wants a notational language for communication. However, it requires a large number of iterations for obtaining the most accurate and complete solution.

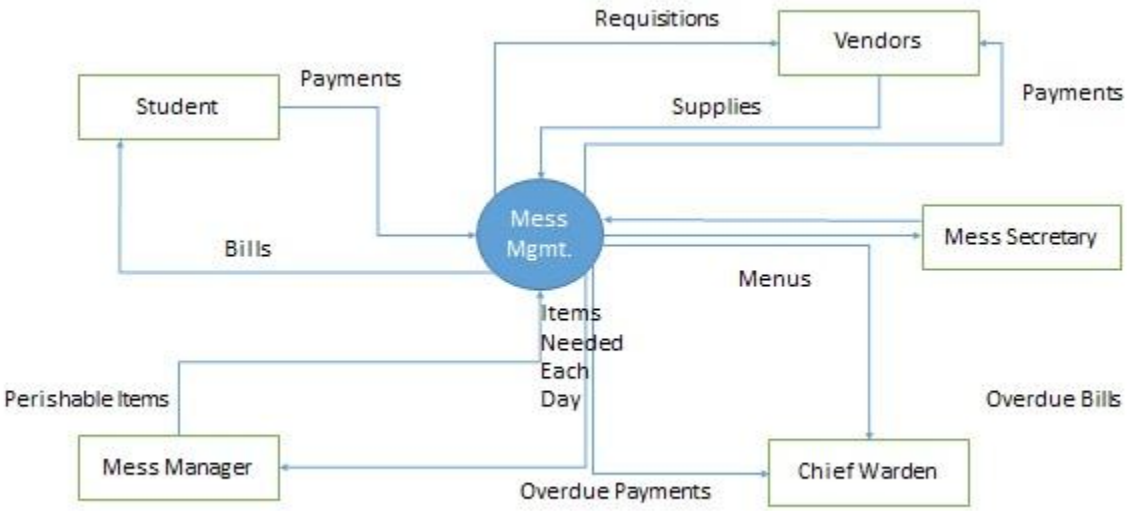
The following table shows the symbols used in designing a DFD and their significance −

Symbol Name	Symbol	Meaning
Square		Source or Destination of Data
Arrow		Data flow
Circle		Process transforming data flow
Open Rectangle		Data Store

Context Diagram

A context diagram helps in understanding the entire system by one DFD which gives the overview of a system. It starts with mentioning major processes with little details and then goes onto giving more details of the processes with the top-down approach.

The context diagram of mess management is shown below.



b).Data Dictionary

A Data Dictionary is a structured repository of data elements in the system. It stores the descriptions of all DFD data elements that is, details and definitions of data flows, data stores, data stored in data stores, and the processes.

A Data Dictionary improves the communication between the analyst and the user. It plays an important role in building a database. Most DBMSs have a data dictionary as a standard feature.

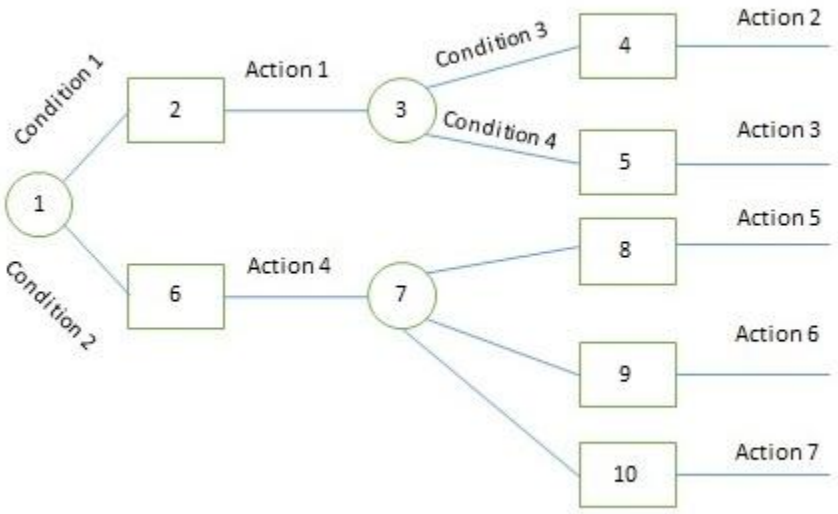
For example, refer the following table

Sr.No.	Data Name	Description	No. of Characters
1	ISBN	ISBN Number	10
2	TITLE	title	60
3	SUB	Book Subjects	80
4	ANAME	Author Name	15

c) Decision Trees

Decision Trees are a method for defining complex relationships by describing decisions and avoiding the problems in communication. A Decision Tree is a diagram that shows alternative actions and conditions within horizontal tree framework. Thus, it shows which conditions to consider first, second, and so on.

Decision trees show the relationship of each condition and their permissible actions. A square node indicates an action and a circle indicates a condition. It forces analysts to consider the sequence of decisions and identifies the actual decision that must be made.



The major limitation of a decision tree is that it lacks information in its format to describe what other combinations of conditions you can take for testing. It is a single representation of the relationships between conditions and actions.

For example, refer the following decision tree −

d) Decision Tables

Decision tables are a method of describing the complex logical relationship in a precise manner which is easily understandable.

- It is useful in situations where the resulting actions depend on the occurrence of one or several combinations of independent conditions.
- It is a matrix containing row or columns for defining a problem and the actions.

Components of a Decision Table

- **Condition Stub** − It is in the upper left quadrant which lists all the condition to be checked.
- **Action Stub** − It is in the lower left quadrant which outlines all the action to be carried out to meet such condition.
- **Condition Entry** − It is in upper right quadrant which provides answers to questions asked in condition stub quadrant.
- **Action Entry** − It is in lower right quadrant which indicates the appropriate action resulting from the answers to the conditions in the condition entry quadrant.

The entries in decision table are given by Decision Rules which define the relationships between combinations of conditions and courses of action. In rules section,

- Y shows the existence of a condition.
- N represents the condition, which is not satisfied.
- A blank - against action states it is to be ignored.
- X (or a check mark will do) against action states it is to be carried out.

For example, refer the following table −

CONDITIONS	Rule 1	Rule 2	Rule 3	Rule 4
Advance payment made	Y	N	N	N
Purchase amount = Rs 10,000/-	-	Y	Y	N
Regular Customer	-	Y	N	-
ACTIONS				
Give 5% discount	X	X	-	-
Give no discount	-	-	X	X

e) Structured English

Structure English is derived from structured programming language which gives more understandable and precise description of process. It is based on procedural logic that uses construction and imperative sentences designed to perform operation for action.

- It is best used when sequences and loops in a program must be considered and the problem needs sequences of actions with decisions.
- It does not have strict syntax rule. It expresses all logic in terms of sequential decision structures and iterations.

For example, see the following sequence of actions −

```
if customer pays advance
  then
    Give 5% Discount
  else
    if purchase amount >=10,000
      then
```

```
        if  the customer is a regular customer
          then Give 5% Discount
        else  No Discount
      end if
    else No Discount
  end if
end if
```

f) Pseudocode

A pseudocode does not follow to any programming language and expresses logic in plain English.

- It may specify the physical programming logic without actual coding during and after the physical design.
- It is used in conjunction with structured programming.
- It replaces the flowcharts of a program.

Guidelines for Selecting Appropriate Tools

Use the following guidelines for selecting the most appropriate tool that would suit your requirements −

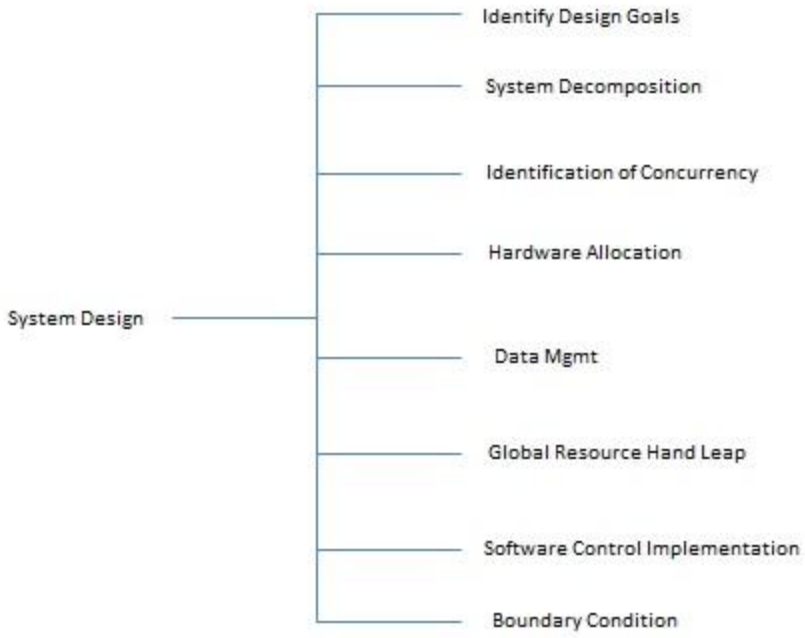
- Use DFD at high or low level analysis for providing good system documentations.
- Use data dictionary to simplify the structure for meeting the data requirement of the system.
- Use structured English if there are many loops and actions are complex.
- Use decision tables when there are a large number of conditions to check and logic is complex.
- Use decision trees when sequencing of conditions is important and if there are few conditions to be tested.

3. 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 is the phase that bridges the gap between problem domain and the existing system in a manageable way. This phase focuses on the solution domain, i.e. *"how to implement?"*

.



a) Inputs to System Design

System design takes the following inputs

- Statement of work
- Requirement determination plan
- Current situation analysis
- Proposed system requirements including a conceptual data model, modified DFDs, and Metadata (data about data).

b) Outputs for System Design

System design gives the following outputs

- Infrastructure and organizational changes for the proposed system.
- A data schema, often a relational schema.
- Metadata to define the tables/files and columns/data-items.
- A function hierarchy diagram or web page map that graphically describes the program structure.
- Actual or pseudo code for each module in the program.
- A prototype for the proposed system.

c) File 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.

Comparison

	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

File Access

One can access a file using either Sequential Access or Random Access. File Access methods allow computer programs read or write records in a file.

a) Sequential Access

Every record on the file is processed starting with the first record until End of File (EOF) is reached. It is efficient when a large number of the records on the file need to be accessed at any given time. Data stored on a tape (sequential access) can be accessed only sequentially.

b) Direct (Random) Access

Records are located by knowing their physical locations or addresses on the device rather than their positions relative to other records. Data stored on a CD device (direct-access) can be accessed either sequentially or randomly.

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 Zip code.
- **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.

Types of Documentations

When it comes to System Design, there are following four main documentations

- Program documentation
- System documentation
- Operations documentation
- User documentation

Program Documentation

- It describes inputs, outputs, and processing logic for all the program modules.
- The program documentation process starts in the system analysis phase and continues during implementation.
- This documentation guides programmers, who construct modules that are well supported by internal and external comments and descriptions that can be understood and maintained easily.

System Documentation

System documentation serves as the technical specifications for the IS and how the objectives of the IS are accomplished. Users, managers and IS owners need never reference system documentation. System documentation provides the basis for understanding the technical aspects of the IS when modifications are made.

- It describes each program within the IS and the entire IS itself.
- It describes the system's functions; the way they are implemented, each program's purpose within the entire IS with respect to the order of execution, information passed to and from programs, and overall system flow.
- It includes data dictionary entries, data flow diagrams, object models, screen layouts, source documents, and the systems request that initiated the project.
- Most of the system documentation is prepared during the system analysis and system design phases.
- During systems implementation, an analyst must review system documentation to verify that it is complete, accurate, and up-to-date, and including any changes made during the implementation process.

Operations Documentation

Operations documentation contains all the information needed for processing and distributing online and printed output. Operations documentation should be clear, concise, and available online if possible.

It includes the following information −

- Program, systems analyst, programmer, and system identification.
- Scheduling information for printed output, such as report, execution frequency, and deadlines.
- Input files, their source, output files, and their destinations.
- E-mail and report distribution lists.
- Special forms required, including online forms.
- Error and informational messages to operators and restart procedures.
- Special instructions, such as security requirements.

User Documentation

It includes instructions and information to the users who will interact with the system. For example, user manuals help guides, and tutorials. User documentation is valuable in training users and for reference purpose. It must be clear, understandable, and readily accessible to users at all levels.

The users, system owners, analysts, and programmers, all put combined efforts to develop a user's guide.

User documentation should include −

- A system overview that clearly describes all major system features, capabilities, and limitations.

- Description of source document content, preparation, processing, and, samples.
- Overview of menu and data entry screen options, contents, and processing instructions.
- Examples of reports that are produced regularly or available at the user's request, including samples.
- Security and audit trail information.
- Explanation of responsibility for specific input, output, or processing requirements.
- Procedures for requesting changes and reporting problems.
- Examples of exceptions and error situations.
- Frequently asked questions (FAQs).
- Explanation of how to get help and procedures for updating the user manual.

4. System Processing and general Program Design:

After the outputs, inputs and files have been designed, the analyst must decide how the various parts of the systems will interact with one another to produce the outputs from inputs by using the appropriate files. The processing of data through the system is done through people and programs. The computer programs required for the system must be determined. For each program, the analyst must specify what is accomplished by it.

A widely used tool for specifying systems processing is the systems flow chart.

Flowchart is a diagrammatic representation of sequence of logical steps of a program. Flowcharts use simple geometric shapes to depict processes and arrows to show relationships and process/data flow.

Flowchart Symbols

Symbol	Symbol Name	Purpose
	Start/Stop	Used at the beginning and end of the algorithm to show start and end of the program.
	Process	Indicates processes like mathematical operations.
	Input/ Output	Used for denoting program inputs and outputs.
	Decision	Stands for decision statements in a program, where answer is usually Yes or No.
	Arrow	Shows relationships between different shapes.
	On-page Connector	Connects two or more parts of a flowchart, which are on the same page.

	Off-page Connector	Connects two parts of a flowchart which are spread over different pages.

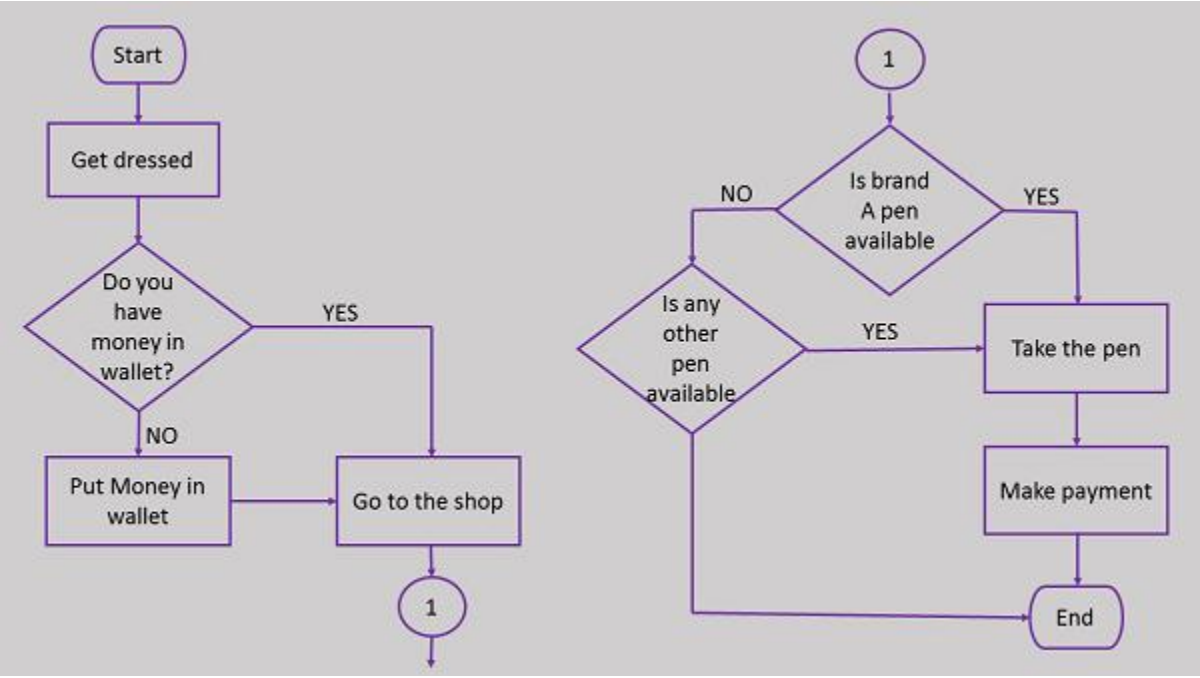
Guidelines for Developing Flowcharts

These are some points to keep in mind while developing a flowchart −

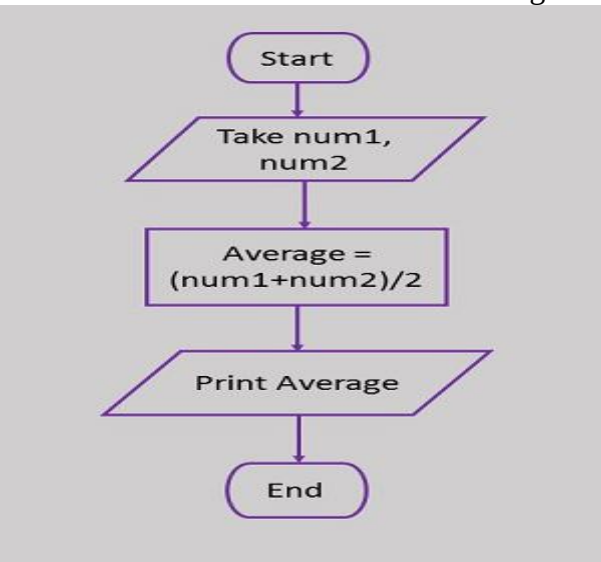
- Flowchart can have only one start and one stop symbol
- On-page connectors are referenced using numbers
- Off-page connectors are referenced using alphabets
- General flow of processes is top to bottom or left to right
- Arrows should not cross each other

Example Flowcharts

Here is the flowchart for going to the market to purchase a pen.



Here is a flowchart to calculate the average of two numbers.



5. Presentation to Management and Users:

The end of the design phase of a systems development project is an important milestone. At this point, we have to decide whether the newly designed system should be implemented. It is a crucial decision because implementation will require much more of the company resources (like time, personnel, money) than any of the previous phases.

Before approval is given to proceed with implementation, the users will have to agree that the system will meet their requirements. As far as possible, all disagreements should be sorted out this stage. Users should point out any changes which they feel are needed. It is much easier to make a change now rather than after the system has been implemented.

Assuming that the users give their approval of the various features of the system, a top level management decision still has to make about implementation. The analyst should not assume that management will automatically improve the new system.

The analyst should follow the following guidelines before presenting to Management and Users

Review the new System by specifying all advantages

Specify the estimation cost for implementing this new system, which includes- number of people required, hardware components cost, suppliers and stationery, etc.

6. Implementation

Implementation is a process of ensuring that the information system is operational. It involves −

- Constructing a new system from scratch
- Constructing a new system from the existing one.

Implementation allows the users to take over its operation for use and evaluation. It involves training the users to handle the system and plan for a smooth conversion.

The following are the major activities which comprise the implementation process

a) Develop detailed programming specifications
b) Develop test specifications and test data
c) Write computer programs
d) User training
e) System testing
f) File conversion
g) Changeover to new system

a) Develop detailed programming specifications:

Internal logic of the various modules is developed. Algorithms are specified in enough detail for the computer programs to be written directly from them.

b) Develop test specifications and test data:

The analyst should decide how program test should be done, which modules to be tested first. The development of test data should begin when the detailed programming logic is being written.as a general guide, test data should check every possible path in a program, and should wide ranging enough to ensure that invalid data would not get processed by the program.

c) Write Computer Programs:

The analyst chooses an appropriate programming language for writing the program. This depends on the type of processing involued. Most business systems are written in COBOL.

Programs are individually tested using the test data developed in the previous step.after ensuring that programs work by themselves, the next step is test together those modules which are interdependent.

d) User training:

The personnel in the system must know in detail what their roles will be, how they can use the system, and what the system will or will not do. The success or failure of well designed and technically elegant systems can depend on the way they are operated and used.

Training Systems Operators

Systems operators must be trained properly such that they can handle all possible operations, both routine and extraordinary. The operators should be trained in what common malfunctions may occur, how to recognize them, and what steps to take when they come.

Training involves creating troubleshooting lists to identify possible problems and remedies for them, as well as the names and telephone numbers of individuals to contact when unexpected or unusual problems arise.

Training also involves familiarization with run procedures, which involves working through the sequence of activities needed to use a new system.

User Training

- End-user training is an important part of the computer-based information system development, which must be provided to employees to enable them to do their own problem solving.
- User training involves how to operate the equipment, troubleshooting the system problem, determining whether a problem that arose is caused by the equipment or software.
- Most user training deals with the operation of the system itself. The training courses must be designed to help the user with fast mobilization for the organization.

Training Guidelines

- Establishing measurable objectives
- Using appropriate training methods
- Selecting suitable training sites
- Employing understandable training materials

Training Methods

Instructor-led training

It involves both trainers and trainees, who have to meet at the same time, but not necessarily at the same place. The training session could be one-on-one or collaborative. It is of two types −

Virtual Classroom

In this training, trainers must meet the trainees at the same time, but are not required to be at the same place. The primary tools used here are: video conferencing, text based Internet relay chat tools, or virtual reality packages, etc.

Normal Classroom

The trainers must meet the trainees at the same time and at the same place. They primary tools used here are blackboard, overhead projectors, LCD projector, etc.

Self-Paced Training

It involves both trainers and trainees, who do not need to meet at the same place or at the same time. The trainees learn the skills themselves by accessing the courses at their own convenience. It is of two types −

Multimedia Training

In this training, courses are presented in multimedia format and stored on CD-ROM. It minimizes the cost in developing an in-house training course without assistance from external programmers.

Web-based Training

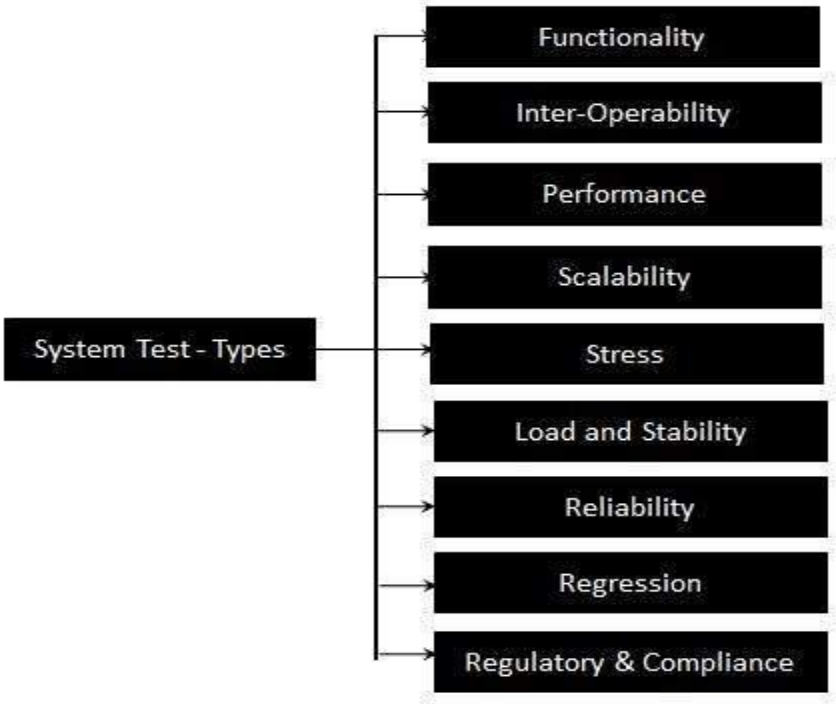
In this training, courses are often presented in hyper media format and developed to support internet and intranet. It provides just–in-time training for end users and allow organization to tailor training requirements.

e)System Testing:

System Testing (ST) is a black box testing technique performed to evaluate the complete system the system's compliance against specified requirements. In System testing, the functionalities of the system are tested from an end-to-end perspective.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

Types of System Tests:



f)File Conversion:

t is a process of converting one file format into another. For example, file in WordPerfect format can be converted into Microsoft Word.

For successful conversion, a conversion plan is required, which includes −

- Knowledge of the target system and understanding of the present system
- Teamwork
- Automated methods, testing and parallel operations
- Continuous support for correcting problems
- Updating systems/user documentation, etc

Many popular applications support opening and saving to other file formats of the same type. For example, Microsoft Word can open and save files in many other word processing formats.

Change over to new system:

It contains description of all the activities that must occur during implementation of the new system and put it into operation. It anticipates possible problems and solutions to deal with them.

It includes the following activities −

- Name all files for conversions.
- Identifying the data requirements to develop new files during conversion.
- Listing all the new documents and procedures that are required.
- Identifying the controls to be used in each activity.
- Identifying the responsibility of person for each activity.
- Verifying conversion schedules.

Conversion Methods

The four methods of conversion are −

- Parallel Conversion
- Direct Cutover Conversion
- Pilot Approach
- Phase-In Method

Method	Description	Advantages	Disadvantages
Parallel Conversion	Old and new systems are used simultaneously.	Provides fallback when new system fails. Offers greatest security and ultimately testing of new system.	Causes cost overruns. New system may not get fair trail.
Direct Cutover Conversion	New system is implemented and old system is replaced completely.	Forces users to make new system work Immediate benefit from new methods and control.	No fall back if problems arise with new system Requires most careful planning
Pilot Approach	Supports phased approach that gradually implement system across all users	Allows training and installation without unnecessary use of resources. Avoid large contingencies from risk management.	A long term phasein causes a problem of whether conversion goes well or not.
Phase-In Method	Working version of system implemented in one part of organization based on feedback, it is installed throughout the organization all alone or stage by stage.	Provides experience and line test before implementation When preferred new system involves new technology or drastic changes in performance.	Gives impression that old system is erroneous and it is not reliable.

8. Documentation:

Any written text, illustrations or video that describe a software or program to its users is called **program or software document**. User can be anyone from a programmer, system analyst and administrator to end user. At various stages of development multiple documents may be created for different users. In fact, **software documentation** is a critical process in the overall software development process.

In modular programming documentation becomes even more important because different modules of the software are developed by different teams. If anyone other than the development team wants to or needs to understand a module, good and detailed documentation will make the task easier.

These are some **guidelines** for creating the documents −

- Documentation should be from the point of view of the reader
- Document should be unambiguous
- There should be no repetition
- Industry standards should be used
- Documents should always be updated
- Any outdated document should be phased out after due recording of the phase out

Advantages of Documentation

These are some of the advantages of providing program documentation −

- Keeps track of all parts of a software or program
- Maintenance is easier

- Programmers other than the developer can understand all aspects of software
- Improves overall quality of the software
- Assists in user training
- Ensures knowledge de-centralization, cutting costs and effort if people leave the system abruptly

Example Documents

A software can have many types of documents associated with it. Some of the important ones include −

- **User manual** − It describes instructions and procedures for end users to use the different features of the software.
- **Operational manual** − It lists and describes all the operations being carried out and their inter-dependencies.
- **Design Document** − It gives an overview of the software and describes design elements in detail. It documents details like **data flow diagrams, entity relationship diagrams**, etc.
- **Requirements Document** − It has a list of all the requirements of the system as well as an analysis of viability of the requirements. It can have user cases, reallife scenarios, etc.
- **Technical Documentation** − It is a documentation of actual programming components like algorithms, flowcharts, program codes, functional modules, etc.
- **Testing Document** − It records test plan, test cases, validation plan, verification plan, test results, etc. Testing is one phase of software development that needs intensive documentation.
- **List of Known Bugs** − Every software has bugs or errors that cannot be removed because either they were discovered very late or are harmless or will take more effort and time than necessary to rectify. These bugs are listed with program documentation so that they may be removed at a later date. Also they help the users, implementers and maintenance people if the bug is activated.

II. 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.

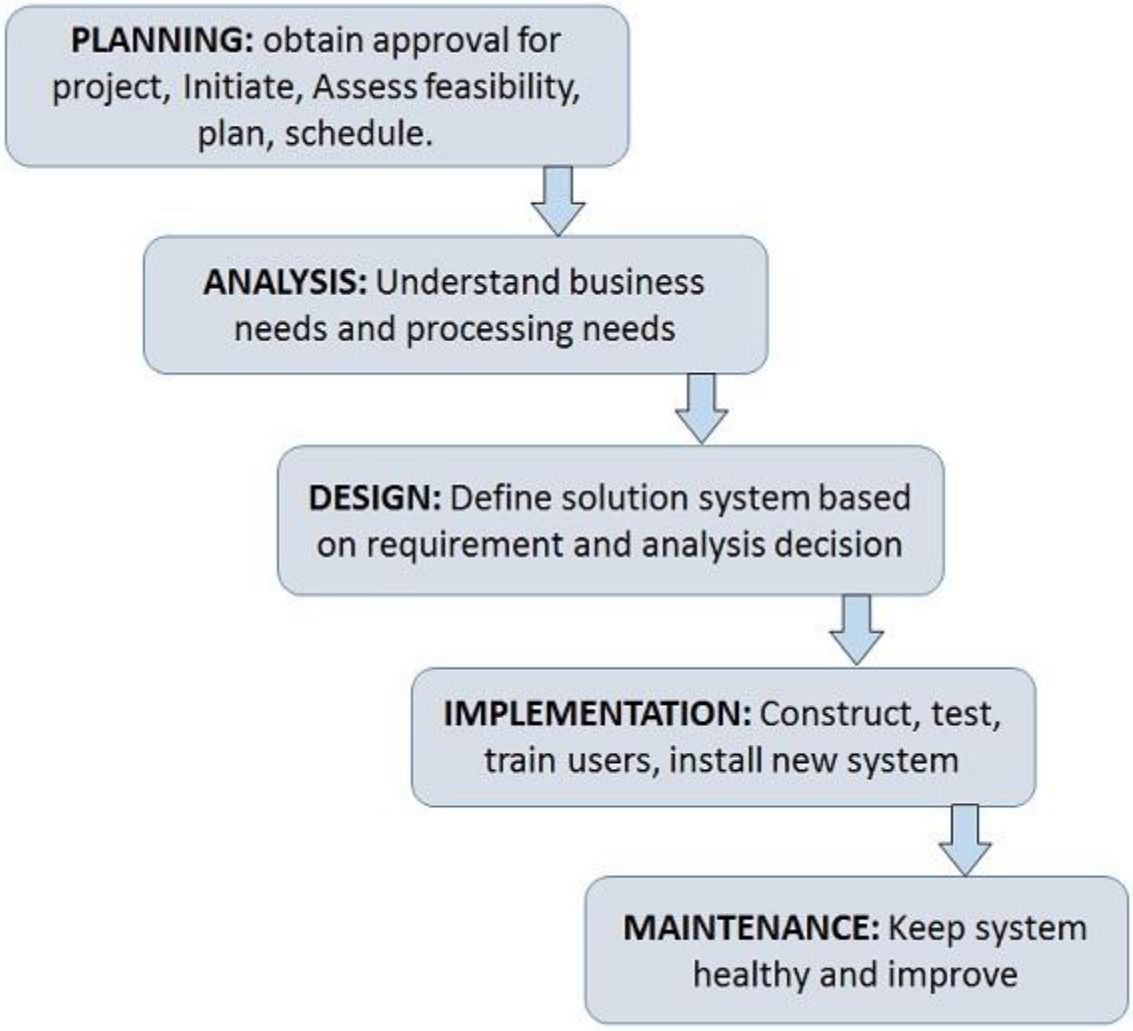
System Development Life Cycle (SDLC) is a conceptual model which includes policies and procedures for developing or altering systems throughout their life cycles.

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.