1 Introduction to Database Management System

UNIT I

Syllabus

Basic concepts, Advantages of DBMS over file processing systems, Data abstraction, Database languages, Data models, Data independence, Components of a DBMS, Overall structure of DBMS, Multi-user DBMS architecture, System catalogs, Data Modeling: Basic concepts, Entity, attributes, relationships, constraints, keys

1.1 Database Concept

Q. Write a short note on Data.
Q. Write a short note on Database.
Q. Write a short note on : DBMS.

- Many of us are very much familiar with the term *called as data*. We come across term data regularly in our day-to-day life. The name of a person, price of book, number of students in a college, pin code of a city, etc. are some examples of data.
- In our daily life, we have to remember bulk amount of data. However, it is easier for us to remember only some amount of data.

Example :

- We may be in a position to tell accurately the age, height, income, educational qualification, residential address, etc. of our close friends.
- But it could be very difficult for us to memorize all this information for a large number of individuals in an organization.

1. Data

- The facts and figures that can be recorded in system (computer system) and that have some special meaning assigned to it is called as data.

Example : Data of a customer like name, telephone number, address, product purchased date etc.

- As need of data increased, there was need to develop a computer-based system for storing and managing data as a file system or information system.

2. Database

- A database is a collection of related data items stored at one place

Example : College database stores information about students, teachers, classes, subjects (All related data)

- A database is nothing but set of data having some relation between them.
- Database acts like a logical collection of relevant data. It is designed to offer an organized mechanism for storing, managing and retrieving stored information.
- Sample database structure,

Student table

Sid	Name	Class	Major

Course Table

Cid	Name	Hours

Department Table

Did	Name

Marks Table

Sid	Cid	Marks	Grade

Fig. 1.1.1 : Sample student database

3. Database Management System (DBMS)

- A Database Management System (DBMS) is a collection of software or programs which helps user in creation and maintenance of a database (set of information). Hence, it is also known as a **computerized record-keeping system.**
- DBMS is software system that helps in the process of defining, constructing, manipulating the database.

- Database management system has become an integral part of the information system of many organizations as it is used to handle huge amount of data.
- Computer-based Information Systems (IS) is capable of serving many complex tasks in a coordinated manner. Such systems handle large volume of data, multiple users and several applications in a centralized database environment.
- The heart of an Information System (IS) is database management system. This is because most IS have to handle huge amounts of data. This core module of an Information System is also called as Database Management System (DBMS).
- Examples :
  - MS Access, FoxPro by Microsoft.
  - Oracle by Oracle corp.
  - SQL Server by Microsoft.
  - Ingres, DB2 by IBM.

1.1.1 Process of Database Management

(a) Defining data :

In this step we decide data structures and constrains for storing data in database.

Example : Data type for age is integer.

(b) Constructing data storage device :

Saving data using some data storage device which will be managed by DBMS.

Example : Database may be stored on server hard disk.

(c) Manipulating data :

Represent data in software and produce user friendly reports from the available data.

Example : Attendance report from employee database.

(d) Sharing data among users :

It should be allowed to access database concurrently by multiple users in order to make application multi-user.

Example : At a given point of time, multiple customers may access the database of a particular bank.

1.1.2 History of Database System

Relational databases have gone through three major steps of evolution as given below.

Stage 1 : Beginning of databases

- The basis for relational databases was formulated by Dr. E. F. Codd at IBM research lab, with a paper **"A relational model of data for large shared data banks"**.
- Oracle was the first commercial product based on above paper.

Stage 2 : Middle age of databases

- IBM, Informix and Sybase entered the Arena of DBMS.
- Microsoft bought Sybase's technology to compete all DBMS products.

Stage 3 : Future of databases

- The future will evolve DBMS that will be meet the requirements of the Internet.
- The most likely scenario gives idea about survival of only three Vendors i.e. Oracle, IBM and Microsoft.

1.1.3 Purpose of Database

Now a days data have become an integral part of the Information systems of many organizations, so it is very important to manage data in the system.

1. **Data As a Corporate Resource**

   Data management and control is very important for efficient working of an organization.

2. **Data Availability**

   As most of the organizational functions are computerized there is increases the need to keep data available for user.

3. **Sharing Data**

   As many users are accessing data simultaneously, sharing of data is required for improving availability of data.

4. **Maintaining Complex data**

   As complexity of the data grows complex relationships between them need to be managed in simpler way.

5. **Data consolidation**

   There should be fixed approach toward consolidation of data in any organizations.

6. Reducing cost

Many organizations are reducing their development costs by allowing user to perform their business transactions. This is used in industries like outlet and C2B (Customer to Business) electronic commerce examples like indiatimes.com

1.1.4 Characteristics of DBMS

Q. Explain the Characteristics of DBMS.

Q. Explain various advantages of Databases.

- The database approach has many characteristics due to which now a days database has become an integral part of software industry.
- The characteristics we are going to learn in many different units are mentioned below:

1. Data integrity
2. Data security
3. Data independence
4. Transaction control - rollback
5. Concurrency control
6. Data recovery - backup and restore

1. Data integrity

- Integrity constraints provide a way of ensuring that changes made to the database by authorized users do not result in a loss of data consistency and correctness.
- Database integrity concern with the correctness and completeness of data in the database.
- This objective can never be guaranteed, one cannot ensure that every entry made in the database is accurate.
- Some examples of incorrect data are as below :
  1. Student taking admission to branch which is not available in college.
  2. Employee assigned with non existing department.
  3. Sometime inconsistency introduced due to system failures.

2. Data security

- A DBMS system always has a separate system for security which is responsible for protecting database against accidental or intentional loss, destruction or misuse.
- Data in database should be given to only authorized users.
- Only authorized users should be allowed to modify the data.
- Authorized users are able to access data any time they want.

3. Data independence

Data Independence can be defined as the capacity to change data kept at one place without changing data kept at other locations.

4. Transaction control – rollback

- The changes made in database can be reverted back with help of rollback command.
- The changes can be saved successfully with help of commit data command.

5. Concurrency control

- The data in database can be accessed by multiple users at same point of time.
- Such operations allowed by sharing same data between multiple users.

6. Data recovery backup and restore

- Database recovery is the process of restoring the database to original (correct) state after database failure.
- The main element of database recovery is the most recent database backup.
- If you maintain database backup efficiently, then database recovery is very straight forward process.

1.1.5 File System Vs Database System

Q. State & explain the disadvantages of File Processing System. **SPPU - May 18**

Q. List significant differences between file processing system and Database Management System.

1. Redundancy can be reduced

- As we are using relational approach for data organization, data is not stored in more than one location.
- Repetition of information can be avoided which in turn saves storage space.

2. **Inconsistency can be avoided**

   With the usage of database it is assured that all the users access actual or true data present in the database.

3. **Data can be shared**

- Multiple users can login at a time into the database to access information.
- They can manipulate the database in a controlled environment.

  Example : In various parts many users are accessing data in database in a controlled manner.

4. With a **centralized control** of data, the database system may be designed for an overall optimal performance for entire organization.

5. **Standards can be enforced**

- Standards (rules and regulations for coding and designing) can be enforced on the database to regulate the access to the database.
- Primary key constraint or foreign key constraint can be enforced on database which will be helpful for accessing data from database.

6. **Security restrictions can be applied**

- Security is for restricting user access to the database server.
- Security of data is very crucial and needs to be carefully handled.

7. **Integrity can be maintained**

   Through integrity, one can ensure only accurate data is stored within the database.

8. **Data independence can be provided**

- None of the users need to know the technical aspects of the database to access it.
- They are physically as well as logically independent to access the database.

9. **New applications** may be developed using the existing database.

Table 1.1.1 Comparison

Sr. No.	Parameter	Database Management System	File Processing System
1.	**Redundancy**	All data at one location, minimal redundancy	Data is at multiple locations causes more data redundancy.
2.	**Inconsistency**	All the users access actual or consistent data.	All the users access local data which can introduce inconsistencies in data.
3.	**Data Sharing**	DBMS provides multiple user interface by sharing data.	Data is isolated in the file system and limited sharing.
4.	**Centralized control**	Unauthorized access is restricted in DBMS.	Unauthorized access cannot be restricted .
5.	**Data Standards**	Data at one location, so enforcing standard is simple.	Data is at distributed location, so enforcing standard is simple.
6.	**Record Storage**	Computerized record keeping system.	Collection of individual files accessed by applications programs.
7.	**Security restrictions**	It is easy to impose security restrictions.	It is bit complex to assign security restrictions.
8.	**Integrity**	It is easy to impose various integrity constraints.	It is difficult or not possible to assign some integrity constraints.
9.	**Data independence**	It's co-ordinates both the physical and logical independence.	It's co-ordinates only the physical access to data independence.
10.	**New applications**	It will be simple to design and connect new applications.	New application needs newer file system or sharing may reduce access speed of data.
11.	**Flexibility of Access**	It allows flexible access to data.	It is designed to allow predetermined access to data.

1.2 Overall structure of DBMS

1.2.1 Three Level Architecture for Database

Q. Describe the three level architecture of DBMS. Explain how it is useful for achieving data independence. **SPPU - Dec. 17**

Q. State and explain various levels of database abstraction.

1.2.1(A) Introduction

- The goal of the three schema architecture is to separate the front end (Presentation Tier) and the back end (Database Tier)
- The three-schema architecture is a tool with which the user can visualize the schema levels in a DBMS.
- Many DBMS systems do not separate the three levels completely, but support the three-schema architecture to some extent.
- A description of data in terms of a data model is called a schema.
- The description of a database is called **database schema**, which is specified during database design and it is not expected to change frequently.

1.2.1(B) Database System architecture

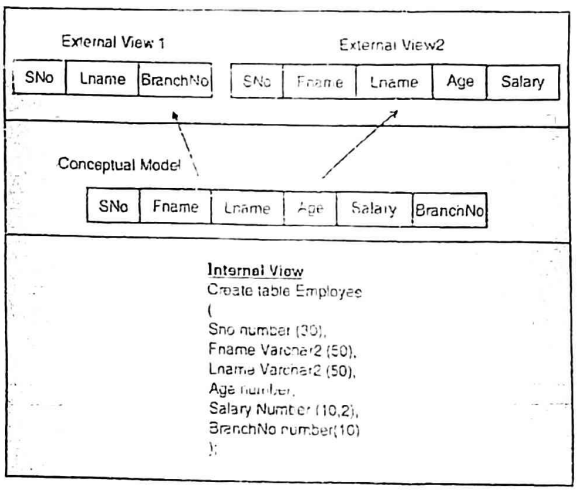

Fig. 1.2.1 : Database Schema Levels

(I) Internal Level or Physical Level

- The internal level is very close to physical storage of data.
- The internal (or physical) database is stored on secondary storage devices, mainly the magnetic disk.
- At its ground level, it is stored in the form of bits with the physical addresses on the secondary storage device.
- At its highest level, it can be viewed in the form of files and simple data structures.

Internal view/ schema

- The internal view defines the various stored data types and specifies the indexes which exist, how that stored fields are represented and so on.
- The internal schema uses a physical data model.

Example :

```
Create table Employee
{
    Sno number (30),
    Fname varchar2 (50),
    Lname varchar2 (50),
    Age number,
    Salary number (10, 2),
    BranchNo number (10)
};
```

(II) Conceptual Level

- This level describes the structure of the whole database for a group of users.
- The conceptual model is also called as the data model or we can say data model is used to describe the conceptual schema when a database system is implemented.
- The conceptual schema hides the internal details of physical storage and targets on describing entities, data types, relationships and constraints.
- The conceptual schema contains all the information to build relevant external records. As the conceptual model is derived from the physical model.

Conceptual view / schema

- The conceptual view is a representation of the entire content of the database.
- The conceptual view includes definitions of each of the various conceptual data types.

(III) External Level or View Level

- The external level is the one closest to the user, i.e., it is the related with the way data is viewed by individual end users.
- The external level includes a number of user views or external schemas.
- Each external schema describes the segment of the database that is required for a particular user group and hides the rest of the database from that user group.

- External views are the proper interface between the user and the database, as an individual user can hardly be expected to be interested in the entire database.
- The external model is derived from the conceptual model

External view / schema

External schema consists of definitions of each of the various external data types in that external view.

1.2.1(C) Data Abstraction

Q. Explain abstraction of DBMS

Q. Explain physical, conceptual and view level abstraction of DBMS.

- A description of data in terms of a data model is called a schema.
- The description of a database is called **database schema**, which is specified during database design and it does not expect to change frequently.

(I) Internal Schema

- This level describes the physical storage structure of the data in database
- Describes the complete details of data storage and various available access methods for the database.

Internal View/Schema :

The internal view defines the various stored data types and specifies the indexes which exist, how that stored fields are represented and so on.

(II) Conceptual Schema

- The conceptual schema hides the internal details of physical storage and targets on describing entities, data types, relationships and constraints.
- The conceptual schema contains all the information to build relevant external records. As the conceptual model is derived from the physical model.

Conceptual view / schema

- The conceptual view is a representation of the entire contents of the database.
- The conceptual view includes definitions of each of the various conceptual data types.

(III) External Schema

- The external level is the one closest to the user, i.e., it is the related with the way data is viewed by individual end users
- External schema describes the segment of the database that is required for a particular user group and hides the rest of the database from that user group.
- External views are the proper interface between the user and the database, as an individual user can hardly be expected to be interested in the entire database.

External view / Schema:

External schema consists of definitions of each of the various external data types in that external view.

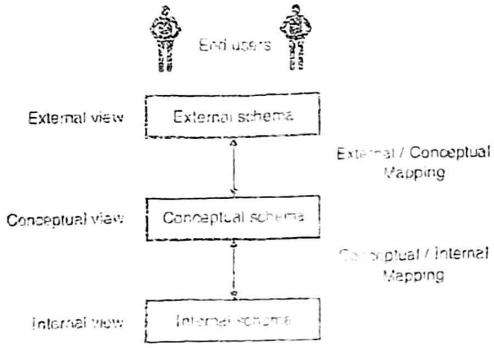

Fig. 1.2.2 : Data Abstraction

(IV) Mapping

- The processes of transforming requests and results between various levels of architecture are called mappings.
- These mappings may be time-consuming, so small databases do not support external views
  - **External / conceptual Schema Mapping :** The DBMS must transform a request on an external schema into a request against the conceptual schema.
  - **Conceptual / Internal Schema Mapping :** A certain amount of mapping is necessary to transform requests between the conceptual and internal levels.

1.2.1(D) Data Independence

Q. What do you mean by data independence?

Q. Explain type of data independence in details.

- Concept of data independence can be explained with help of 3 schema architecture
- The three-schema architecture can make it easier to achieve true data independence

Definition

Data Independence can be defined as the capacity to change one level of schema without changing the schema at the next higher level.

Types

(a) Logical data independence

(b) Physical data independence

(a) Logical data independence

- Logical data independence is a capacity to change the conceptual schema without any changes to external schemas. (or application programs)
- Separating the external views from the conceptual view enables us to change the conceptual view without affecting the external views. This separation is sometimes called logical data independence.

  Example : We may change the conceptual schema by removing a data item. In this case the external schemas that refer only to the remaining data should not be affected.

(b) Physical data independence

- Physical data independence is a capacity to change the internal schema without any changes to conceptual schema.
- The separation of the conceptual view from the internal view enables us to provide a logical description of the database without the need to specify physical structures. This is often called physical data independence.

  Examples : By creating additional access paths to improve the performance of retrieval. If the data remains same as before in the database, then we should not change the conceptual schema

1.2.1(E) Difference Between Logical and Physical data Independence

Q. Differentiate between Logical data independence and Physical data Independence **SPPU - Oct.-19**

Table 1.2.1 : Comparison

Sr. No.	Parameter	Logical data independence	Physical data independence
1	Definition	The capacity to change the conceptual schema without any changes to external schemas.	The capacity to change the internal schema without any changes to conceptual schema.
2	Modification	Modify conceptual schema when structure of database is	Modification are performed to improve performance.
3	Immunity		It provides independence and immunity to conceptual and external schema
4	Complexity		It is easy

1.3 Database System Detailed(Multiuser DBMS) Architecture

Q. Draw and list components of database system structure **SPPU - Dec.-17**

Q. Describe working of detailed system architecture of DBMS

1. Introduction

- A database system is partitioned into two different modules that interact with each other in order to the overall system
- The database system is divided into modules that deal with each of the responsibilities of the overall system that includes
  - Hardware
  - Software
  - Data
  - Procedures
  - Database access language
- Every DBMS is supposed performing significant tasks in the database management system

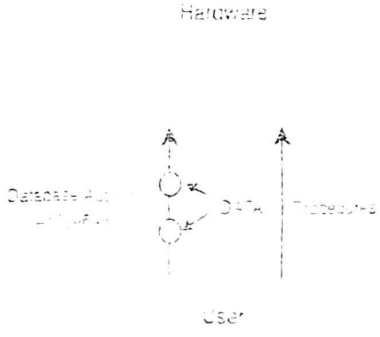

Fig. 1.3.1

2. Hardware Component

- Hardware component includes computer, hard disks, I/O channels for data and physical components involved.

- When we run MySQL (DBMS environment) on our personal computer, then our computer's Hard Disk and Keyboard using which we type in all the commands, our computer's RAM, ROM all become integral part for DBMS to work well.

3. Components of a database system

1. Query Processor Components
2. Storage Manager/ Storage Management
3. Transaction Management

The storage manager is important because databases typically require a huge amount of storage space.

1.3.1 Query Processor Components

Q. Write a short notes on : Query processor.

1. Introduction

The query processor will accept query from user and solves it by accessing the database.

(i) DDL interpreter
(ii) DML compiler
(iii) Query evaluation engine

(i) DDL interpreter

This will interpret DDL statements and fetch the definitions in the data dictionary.

(ii) DML compiler

- This will translate DML statements in a query language into low level instructions that the query evaluation engine understands.
- A query can usually be translated into any of a number of alternative evaluation plans for same query result DML compiler will select best plan for **query optimization.**

(iii) Query evaluation engine

This engine will execute low-level instructions generated by the DML compiler on DBMS.

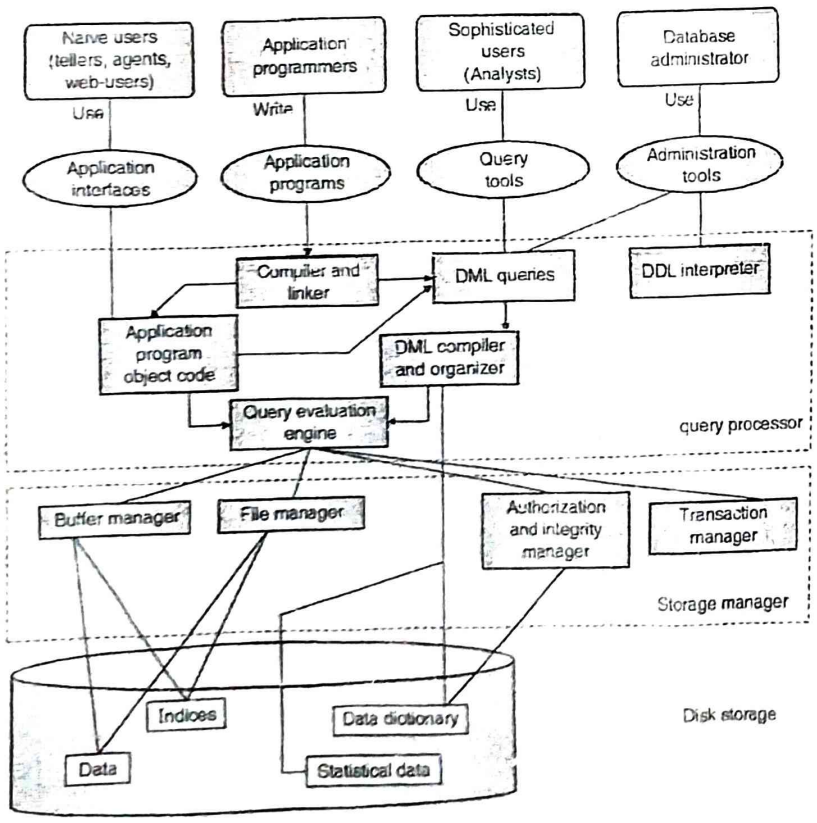

Fig. 1.3.2 : Components of DBMS

1.3.2 Storage Manager / Storage Management

Q. Write a short notes on : Storage management

- A storage manager is a program module which acts like interface between the data stored in the database and the application programs and queries submitted to the system.
- The data is stored on the disk using the file system
- The storage manager is program which is responsible for the interaction with the file manager.
- The storage manager translates the various databases language statements into low level file system commands.
- Thus, the storage manager is responsible for storing, retrieving and updating data in the database.
- The storage manager components include :
  - **Authorization and integrity manager :** Checks for integrity constraints and authority of users to access data.
  - **Transaction manager**, which ensures that the database remains in a consistent (correct) state although there is system failures.
  - **File manager**, which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.
  - **Buffer manager**, which is responsible for retrieving data from disk storage into main memory. The buffer manager is a important part of the database system, as it enables the database to handle data sizes that are much larger than the size of main memory.
- Data structures implemented by storage manager,
  - **Data files :** Stored in the database itself.
  - **Data dictionary** : Stores metadata about the structure of the database.
  - **Indices :** Provide fast access to data items.

1.3.3 Transaction Management

Q. Write a short note on : Transaction management.

- A transaction is a series of small database operations that together form a single large operation.
- A transaction is started by issuing a BEGIN TRANSACTION command. Once this command is executed the DBMS starts recording the transaction.
  All operations executed after BEGIN TRANSACTION command are treated as a single large operation.
- Application programs use transactions to execute sequences of operations. It is important that all the operations are successfully completed.
- Transaction management component will ensure the atomicity and durability properties.

1.4 Database Users

Q. What are the different types of database system users?

Q. State various functions of each type of database users.

There are four broad categories of users as per needs to access data :

1. Naive users
2. Application programmers
3. Sophisticated users
4. Specialized users

1. Naive users

- Naive users are users who interact with the system using application programs that have been developed previously.

  Example : Student wants to pay fees Rs.50 then accountant will invokes a program called fees_payment. This program asks the accountant for the amount of fees to be paid.
- The typical graphical user interface for naive users is a kind of form interface, where the user can fill in appropriate fields of the form.
- A given end user can access the database via one of the applications or can use an interface provided as an integral part of the database system software (such interfaces are also supported by means of applications, of course, but those applications are built-in, not user-written, e.g., query language processor)
- Naive users can read reports generated from the database.

2. Application programmers

- Application programmers responsible for writing application programs that use the database.
- Application programmers are **developers or computer professionals** who write application programs.
- Application programmers develop user interfaces using any preferred language.
- Rapid Application Development (RAD) tools are available nowadays that enable an application programmer to construct application without writing code.
- Some programming languages combine control structures with database language statements. Such languages, sometimes called fourth-generation languages.

3. Sophisticated users

- Sophisticated users interact with application without writing programs by using a database query language.
- This query will be solved by query processor.
- Online Analytical Processing (OLAP) tools is used to view summaries of data in different ways which helps analysts (e.g. sales of region city etc.) with OLAP analysts can use data mining tools, which help them find certain kinds of patterns in data.

4. Specialized users

- Creates the actual database and implements technical controls needed to enforce various policy decisions.
- Specialized users are sophisticated users who develop database applications.
- The DBA is also responsible for ensuring that the system operates with adequate performance and for providing a variety of other related technical services.

1.4.1 Database Administrator (DBA)

Q. What is role of DBA in database management system?

Q. List the responsibilities of DBA **SPPU - Dec. 18**

Q. Explain the problems that may arrive if the DBA does not discharge the responsibilities properly ? **SPPU - May 19**

- The database administrator is responsible for the overall planning of the company's data resources, for the design of data and for the day-to-day operational aspects of data management.
- A database administrator is a person responsible for the installation, configuration, up gradation, maintenance and monitoring databases in an organization.
- The overall planning of corporate data is the strategic aspect of the database administration function and involves company-wide planning of existing data and assessment of organization-wise data standards.

Responsibilities of DBA in enterprises

- Designing schema.
- Deciding on the storage and access methods.
- Selecting database software and hardware.
- Designing the means of reorganizing databases periodically.
- Designing database searching strategies.
- Designing authorization checks and validation procedures.
- Designing restart and recovery procedures to take care of system crashes.
- Specifying techniques for monitoring database performance.

- The operations management of database administration deals with data problems arising on a day-to-day basis. Specifically, the responsibilities include :
  - Investigation of errors found in the data.
  - Supervision of restart and recovery procedures in the event of a failure.
  - Supervision of reorganization of databases.
  - Initiation and control of all periodic dumps of data.

Skills required for DBA

- Good communication skills.
- Excellent knowledge of databases architecture and design and RDBMS (Oracle, SQL Server etc.)
- Knowledge of Structured Query Language (SQL).

- In addition, this aspect of database administration includes maintenance of data security, which involves maintaining security authorization tables, conducting periodic security audits, investigating all known security breaches.
- To carry out all these functions, it is crucial that the DBA has all the accurate information about the

company's data readily on hand. For this purpose he maintains a *data dictionary*.

- The data dictionary contains definitions of all data items and structures, the various schemes, the relevant authorization and validation checks and the different mapping definitions.
- It should also have information about the source and destination of a data item and the flow of a data item as it is used by a system. This type of information is a great help to the DBA in maintaining centralized control of data.

1.5 Database Model

Q. What is data model?

- Data model will give you idea how your final system or software will look like after development is completed.
- This concept is exactly like real world modeling in which before constructing any project (Bridges, Buildings, Towers etc.) engineers create a model for it, this model gives you idea about how your project will look like after construction.
- A data model is an overview of a software system which describes how data can be represented and accessed from software system after its complete implementation.
- Data models define data elements and relationships among various data elements for a specifi system.
- If we need to design a system which keeps track of student and classes information we can use model given below,

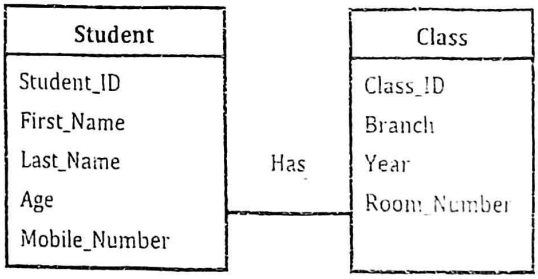

Fig. 1.5.1 : Simple logical data model for student and class

- **According to Hoberman (2009)** "A data model is a way finding tool for both business and IT professionals, which uses a set of symbols and text to precisely explain a subset of real information to improve communication within the organization and thereby lead to a more flexible and stable application environment."
- Data model is a simple abstraction of complex real-world data gathering environment.

1.5.1 Benefits of Data Modeling

- A data model is a set of concepts that can be used to **describe the structure of data** in a database.

  In Fig. 1.5.1 (Simple Logical Data model) we have described structure of student data and class data.
- Data models are used to **support the development** of information systems by providing the definition and format of data to be involved in future system.
- Data model is acting like a guideline for development also gives idea about possible alternatives to achieve targeted solution.
- A data model can be sometimes referred to as a **data structure**, especially in the context of programming languages.
- In Fig. 1.5.1 (Simple Logical Data model) student and class are data structure of type class.

a. Reduced risk

- Data model prevents system from **future risk and failure** by defining structure of data in advance.
- As we got idea of final system in the beginning of development itself so if need to have any revision or improvement we can do it in system, as actual system is not yet developed.

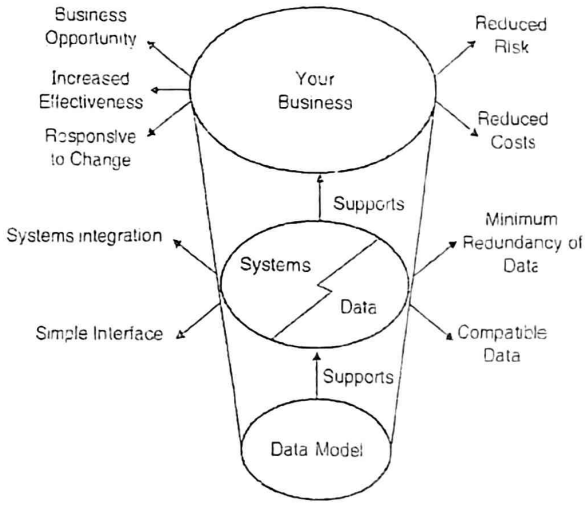

Fig. 1.5.2 : Important roles of Data Model in DBMS

b. Reduced cost

As we got an idea of final system at the beginning of development itself, so we can reduce cost of project by

proper planning and cost estimations as actual system is not yet developed

c. Minimizes redundancy and data compatibility

Data repetition and data type compatibility can be checked and removed with the help of data model.

d. Improves effectiveness of system

We can improve Graphical User Interface (GUI) of system by making its model and get it approved by its future user (user of that system) so it will be simple for them to operate system and make entire system effective.

1.5.2 Basic Building Block of Data Model

1. Introduction

The basic building block for each and every model is Entities, Attributes, relationships and constraints.

2. Entity

- A fundamental component of a model.
- An entity is having its own independent existence in real world.

  Example : A Student, Faculty, Subject having independent existence.
- An entity may be an object with a physical existence or it may have logical existence.

  Example : Entities like Department, Section, adult (age>18) may have physical existence or it may have only logical existence.

3. Attributes

- Each entity has its own properties which describes that entity such properties are known as **attributes.**

  Example : For an employee of ID 30 value of name attribute value 'Jayendra'
- A particular entity will have some value for each of its attributes.

  Example : Employee entity may be described by attributes name, age, phone etc.

4. Tuple / records

- A single row / tuple contain all the information (in the form of attributes) about a single entity.
- Each horizontal row of the STUDENT table represents a single entity a single Student.
- A Table can have any number of rows in it can be zero or thousand.
- If Zero number of rows are present in a table, such table is called as Empty table.

5. Relationships

- A relationship is an association among several entities.

  Example : Employee works for Department.
- **Degree :** The degree of relationship type is number of participating entity types in a particular relation.
- Data model uses three types of relationships as below :

  (a) One is to one

  - One entity is associated with at most one other entity
  - **Example** : One department can have only one manager

  (b) One is to Many

  - One entity is associated with any number of entities in other entity.
  - **Example** : One teacher may teach to many students.

  (c) Many is to Many

  - One entity is associated with any number of entities in other entity.
  - **Example** : Books in library issued by students.

6. Constraints

Constraints are nothing but limitations or conditions imposed on data of database in order to keep database in consistent or correct state.

Example : Employee can have phone number or cannot have phone number, so we can define phone number attribute as a NULL attribute using NULL constraint,

phone_no char (10) NULL

7. Keys

- The column value that uniquely identifies a single record in a table called as Key of table.
- An attribute or set of attributes whose values uniquely identify each entity in an entity set is called a key for that entity set.

- ID is a key of student table. It is possible to have only one student with a one ID (Say only one student 'Mahesh' with ID = 1

Table 1.5.1 : Types of keys in RDBMS

Key type	Definition
Super key	An attribute (or combination of attributes) that uniquely identifies each row in a table
Candidate key	A minimal (irreducible) superkey. A superkey that does not contain a subset of attributes that is itself a superkey.
Primary key	A candidate key selected to uniquely identify all other attribute values in any given row.
Secondary key	An attribute (or combination of attributes) used strictly for data retrieval purposes.
Foreign key	An attribute (or combination of attributes) in one table whose values must either match the primary key in another table or be null.

1.5.3 Types of Data Models

Q. List the different data models . Explain any two.

SPPU : Dec.-19

1.5.3(A) Hierarchical Model

1. Introduction

- This was developed by joint efforts of IBM and North American Rockwell known as Information management system.
- It was the first DBMS model.
- The data is stored hierarchically either in top down or bottom up approach of designing.
- This model uses pointers to navigate between stored data.
- This model represents data as a *hierarchical tree*.

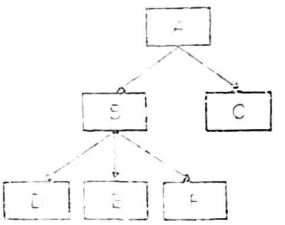

Fig. 1.5.3 Hierarchical model

2. Basic building blocks / structure

- Let us consider simple organizational structure as shown in Fig. 1.5.4

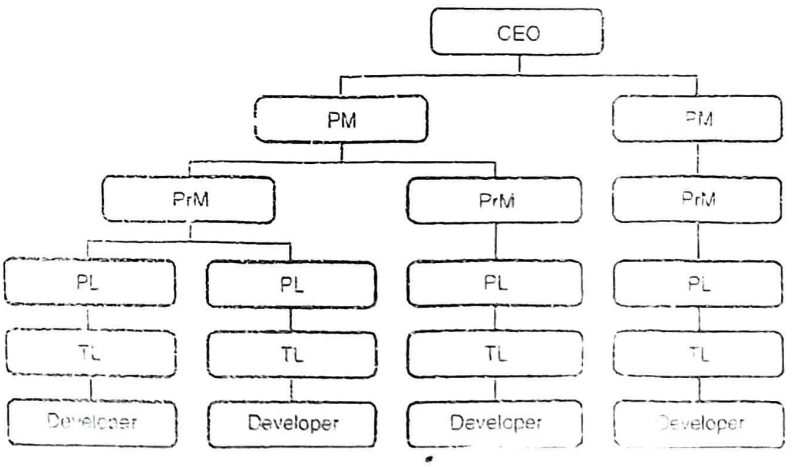

Fig. 1.5.4 : Organizational hierarchy

- CEO is root node having many DU Heads below that mangers who manages multiple project leads who organizes developer as shown below :
- CEO → Program Manager (PM) → Project Manager (PrM) → Project Leader (PL) → Team Leader (TL) → Developer.

3. Business rule

One parent node may have many child nodes, but one child cannot have more than one parent.

4. Example

One of the popular DBMS based on hierarchical model is Information Management System (IMS) from IBM.

5. Advantages

(i) Conceptual simplicity

Relationship between various levels is logically very simple. Hence database structure becomes easier to view.

(ii) Database security

Security is given by DBMS system itself it does not depends on weather programmer has given security or not.

(iii) Simple creation, updation and access

- Hierarchical model is simple to construct with help of pointers or similar concepts and very simple to understand also adding and deleting records is easy in tree structure using pointers.
- This file system is faster and easy data retrieval through higher level records in tree structure.

(iv) Database integrity

- There is always parent child association between different levels of records in files.
- Hence child record is attached with the parent record which maintains the integrity.

(v) Data independence

- DBMS itself maintains data independence so it reduces program efforts and its maintenance.
- If one part of database code is changed no need to change other part of database coding.

(vi) Efficiency

This model having good performance when database contains large amount of data in which one record has many related records like a class contains many students studying in it.

6. Disadvantages

(i) Complex implementation

Only data independence is not enough for designer and programmers to build database system they need to have knowledge of physical data storage which may be complex.

(ii) Difficult to manage

- Any change in a location of data needs change in all application programs that accesses changed data.
- Data access is restricted by Pointer path.

(iii) Lack of structural independence

- Change in database structure does not affects data access is called as structural independence.
- Advantage of data independence may be restricted by structural independence.

(iv) Complex application programming

- Programmers must know how physical data is stored in order to access data.
- Even programmer knows path of data storage.

(v) Limitations in implementation

- Generally 1:N relationship can be implemented in hierarchical model.

  Example : one teacher teaches to many students.
- It is very difficult to implement M:N relationship in hierarchical model.

  Example : many students can have many books.
- Query optimization is generally not possible or possible up to certain extent.

1.5.3(B) Network Database Model

1. Introduction

- Like the hierarchical model, this model also uses pointers toward data but there is no need of parent to child association so it does not necessarily use a downward tree structure.
- This model used in network databases.

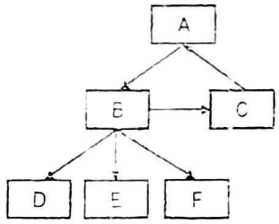

Fig. 1.5.5 : Network model

2. Basic building blocks / structure

- This database model is similar to hierarchical model up to some aspect.
- There are some concepts involved in network model as below :
  - SET : A relationship between any two record types is called as a set
  - RECORD :

  (i) Owner record is like parent record in hierarchical model

  Example : Professor teaches to students in this case professor is owner entity.

  (ii) Parent record is like child record in hierarchical model.

  Example : Professor teaches to students in this case student is parent entity.
- A relationship can be 1:N or in this model we can show M:N relationship.
- **Example :** Many Professors in college teaches too many students.

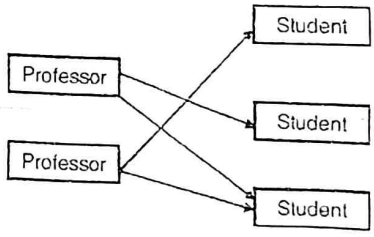

Fig. 1.5.6 : College hierarchy

3. Example :

IDS (Integrated Data Store) is one of the DBMS product based on network models. This was developed by joined efforts of IBM and North American Rockwell known as Information Management System.

4. Advantages

(i) Simple design

The network model is simple and easy to design and understand.

(ii) Ability to handle many types of relationship

- The network model can handle the one-to-many or many-to-many relationships.
- Hence network model manages multi user environment.

(iii) Ease of data access

- In a network model, an application can access a root (parent) record and all the member records within a set (child).
- Provide very efficient and high speed retrieval.

(iv) Data integrity

In a network model, no member can exist without a parent entity. A user must therefore first define the root record and then the child record.

(v) Data independence

- In network model, application programs work independently of the data.
- Any changes made in the data do not affect the application program.

(vi) Conformance to standards

In a network model, facilitates the administrator's portability by offering data creation by DDL (Data Definition Language) and DML (Data Manipulation Language).

5. Disadvantages

(i) System complexity

- In a network model, data are accessed one record at a time.
- This can increase the complexity of system for accessing multiple records at a time.

(ii) Lack of structural independence

Any changes made to the database structure (or data) require the application programs to be modified before it can access data.

1.5.3(C) Relational Model

1. Introduction

- The relational model first proposed by E. F. Codd hence he is known as father of Relational model.
- Relational database was an attempt to simplify database structure by making use of tables and columns.

- A relational database is a collection of 2-dimensional tables which consists of rows and columns.

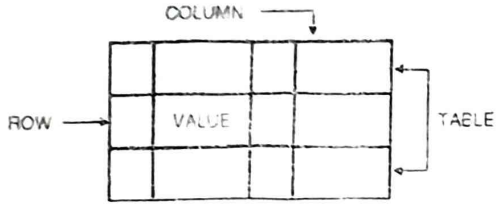

Fig. 1.5.7 : Relational model

2. Basic building blocks / structure

- Tables are known as "relations". Columns are known as **"attributes"** and rows (or records) are known as **"tuples"**.
- This model uses collection of tables to represent relationships amongst the data.
- In this model, each database item is viewed as a record with attributes. A set of records with similar attributes is called a ***table***. Each table contains a record of a particular type.
- The database uses relational model called as RDBMS (Relational Database Management System)
- Let us consider a college database it contains information about,
  - **Student table** : Contains information of all students.
  - **Faculty table** : Contains information of all faculties (Teaching and non teaching).
  - **Accounts tables** : Contains information about all fees transactions happened in college, salary records for faculties and all other accounts details.
  - **Class table :** Maintains information about various classes available in college.

Student table				
STUD_ID	**Name**	**Age**	**Std**	**Div**
105	Mahesh	25	BE	A
106	Suhas	28	FE	B
107	Jay	29	SE	A
108	Sachin	30	TE	D

Faculty table				
FAC_ID	**Name**	**Type**	**Age**	**Salary**
205	Jay	Teaching	25	20000
206	Om	NONT	26	12000
207	Yogesh	NONT	24	15000

Account table				
ACC_ID	**Tran**	**Date**	**Receiver**	**Amt**
1	Cash	1/1/2011	Jay	12000
2	Cheque	1/1/2011	Mahesh	15000
3	Cash	1/1/2011	Yogesh	1000

Class table			
CLASS_ID	**Year**	**Branch**	**Room**
1	FE	IT	102
2	SE	IT	103
3	FE	CS	105

3. Example :

Most of the popular commercial DBMS products like Oracle, Sybase, MySQL etc are based on relational model.

4. Advantages

(i) **Relational algebra** : A relational database supports relational algebra and various operations of the set theory (like union, intersection etc.)

(ii) **Dynamic views :** In a RDBMS, a view is not a part of the physical schema, it is always dynamic. Hence changing the data in a table also changes the data present in view.

(iii) **SQL (Structured Query Language)** : For data access in RDBMS we have English like query language called as Structured Query Language (SQL) which can be used for accessing data from RDBMS. Most of the database vendors support the SQL standard.

(iv) **Excellent data security** : Relational databases support the concept of user rights (every user is assigned with some database permission called as user rights), thus meeting the security needs of databases.

(v) The other advantages of relational databases are performance and support to new hardware technologies and also flexible for all types of data needs.

(vi) Relational databases are scalable and provide good support for the implementation of distributed systems and other advanced database systems.

1.5.3(D) Entity Relationship Model

1. Introduction

- In 1976, Chen developed the **Entity-Relationship (ER) model**, a high level data model that is useful in developing a conceptual design for a database.
- ER model define data elements and relationships among various data elements for a specified system.
- The ER data model is based on perception of real world data that consists of set of entities (data items) and relationship among these entities.
- ER model is a popular high level conceptual model used for conceptual design of database.

Example : In Fig. 1.5.8 of ER model depicts relationship between professor and students (Professor teaches students).

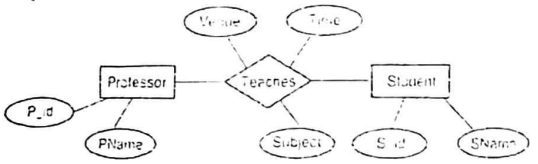

Fig. 1.5.8 : ER model

2. Basic building blocks / structure

- **Entity :**
  - A fundamental component of ER model.
  - An entity is a thing in real world with its own independent existence.

    Example : A Student, Faculty, Subject having independent existence.
- **Attributes :**
  - Various properties that describe an entity are known as attribute.
  - The attribute value that describes each entity becomes a major part of data stored in database.
  - A particular entity will have some value for each of its attributes.

    Example : For an employee of ID 30 value of name attribute is 'Jayendra'.
- **Relationships :**
  - A relationship is an association among several entities.
  - We use diamond to illustrate relationships in ER diagrams. It is to be read from left to right.

    Example : Employee works for Department.

Fig. 1.5.9 : ER model

3 Advantages

(i) Very simple design

The ER model is simple and easy to design logical view of data.

(ii) Ease of representation

In ER model designer, programmer, developer can understand system very easily by looking at an ER model constructed.

(iii) Integrated with relational databases

ER model is completely integrated with relational model which can offer us well structured design process.

4. Disadvantages

(i) Limited constraint representation

- The model can display constraint those are directly tied by relationships.
- **Example** : professor can teach to one or more students.
- But it is not possible to represent some constraints like average marks of all students of a professor may schedule for more than 10 hours.

(ii) Limited relationship representation

- Relationship between attributes of same entity cannot be represented.
- **Example :** There is no way to represent relationship between class wise students list and marks scored by them.

(iii) No DML

ER model does not have any language by which we can insert data in database.

(iv) Loss of information contents

ER model dose not deal with actual data values so we may not be able to represent information content of system.

1.5.3(E) Object Model

1. Introduction

- The data is stored in the form of objects, which are structures called *classes* that display the data within it.
- The fields are instances of these classes called as objects.
- This model is used in File management systems.
- The DBMS (Database Management System) developed with help of such model is called as OODBMS (Object Oriented Database Management System).

- Object oriented databases evolved to handle more complex applications such as databases for scientific experiments, geographic information system, engineering design and manufacturing.
- This model represents real world objects.
- Object oriented approach deals with data at next level.

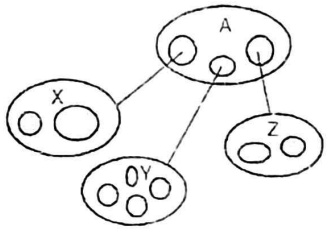

Fig. 1.5.10 : Object model

- This model represents DB in terms of objects, their attributes and their behaviors.

2. Advantages

(i) OO (Object Oriented) features provide a clear modular structure which is good for defining abstract data types where internal implementation details are hidden.

(ii) This model is easy to maintain and modify existing code as we can create new model with small change in existing.

3. Disadvantages

(i) This model is often provided through object-oriented languages such as C++ and Java.

(ii) Practically very complex and inapplicable many a times.

1.6 Comparison of All Data Models

Sr. No.	Parameter	File system model	Hierarchical model	Network model	Relational model	ER model	Object oriented model
1	Data Independence	No	Yes	Yes	Yes	Yes	Yes
2	Structural Independence	No	No	No	Yes	Yes	Yes
3	Storage Type	File	Segment	Record	Relation (Table)	Entity	Class
4	Single row storage	Record	Segment occurrence	Current Record	Row(tuple)	Entity Occurrence	Object – instance of class
5	Basic storage	Field	Segment field	Record field	Relation Attribute	Entity attribute	Object attribute
6	Storage Identifier	Index	Sequence Field	Record key	Key	Entity key attribute	Object identifier
7	Advantages	*Simple to implement, *Low cost implementation	*Promotes Data sharing, *conceptual simplicity, *Handle simple relationships (1:N) *Flexible data access	*conceptual simplicity, *Handle complex relationships (M:N)	*Tabular view, *Adhoc query capability, *improves management and implementation simplicity	*Very good *conceptual simplicity, *Effective *communication tool, *Integrated with dominant relational tools	*Semantic contents, *Promotes data integrity
8	Disadvantages	*Limited implementations *Lack of standards	*Complex implementation, *Lack of standards *Limited implementations (No DML)	*Simplicity limits efficiency *Complex navigational system	*Hardware and software required	*Limited constraint representation *Limited relationship representation * No DML	* Slow development Of standards * complex navigation * Slow transactions if overload on system
9	Examples	Operating system, Notepad, CSF(Comma Separated Files)	IMS (Information Management System)	IDS (Integrated Data Store)	Oracle, DB2, SQL SERVER etc.	ER diagrams	Database using C, java etc

1.7 System catalog

- A DBMS needs to maintain a description of all the data items that it contains. Like a RDBMS contains information about all relation, database objects, views and index present in it.
- The DBMS should also maintains information about views (a definition of the view) to compute the tuples that belong in the view whenever the view is queried or called.
- The multiple relations which stores information about data in database. Such tables are created and maintained by the system itself. They are called as **catalog relations**.
- The catalog relations are also called the **system catalog** or the **data dictionary**.
- The system catalog is sometimes also referred to as **metadata** (data about data)
- The information in the system catalog is generally used only for query optimization.

Relation
Name of given relation
Name of Attribute and type of attributes of relation
Index name for all indexes on the relation.
Integrity constraints imposed on the relation
Index
Name of index and structure of the index (e.g., hash index)
Search key attributes
View
Name of view name
View definition.

Fig. 1.7.1 : Information stored in System Catalog for each data objects

1.8 Database Language

There are four database languages

1. DDL
2. DML
3. DCL
4. TCL

Data Definition Language (DDL)

- To create database schema and database objects like table, view, trigger we need to use Data Definition Language (DDL).
- The set of DDL commands are as follows

1. CREATE Statement
2. ALTER Statement
3. DROP Statement
4. RENAME Statement
5. TRUNCATE Statement

Data Manipulation Language (DML)

- Data Manipulation Language (DML) statements are used for manipulating or managing data in database.
- The set of DDL commands are as below,

1. INSERT Statement
2. DELETE Statement
3. UPDATE Statement

- Data Control Language (DCL)
- Data Control Language (DCL) is used to control various user actions like inserting data, updating data, deleting data or viewing data.
- DCL is set of commands used to,

1. Grant
2. Revoke

- Transaction Control Language (TCL)
- TCL is set of commands used to,

1. Commit
2. Abort

Review Questions

Q. 1 Write advantages of DBMS over a file system

Q. 2 What are the different types of database system users?

Q. 3 List the functions of a Database Administrator (DBA).

Q. 4 Write short note on responsibilities of database administrator.

Q. 5 Describe the overall architecture of DBMS with diagram.

Q. 6 List significant differences between file processing system and Database Management System.

Q. 7 Write short note on data Independence and its types.

Q. 8 What is data model ? Explain various types of data models.

Q. 9 Give evaluation of various data models of DBMS.

Q. 10 Explain hierarchical and network database model.

Q. 11 Compare various data models available.

Q. 12 Explain various data model with their advantages and disadvantages.

□□□