Unit 1

Database Systems Introducing the database and DBMS, Files and File Systems, Problems with file System and advantages of Database Management systems. Data Models: The importance of Data models, Data Model Basic Building Blocks, Business Rules, The evolution of Data Models, Degree of Data Abstraction.

1) What is DBMS?

DBMS is a collection of programs that facilitates users to create and maintain a database. In other words, DBMS provides us an interface or tool for performing different operations such as the creation of a database, inserting data into it, deleting data from it, updating the data, etc. DBMS is software in which data is stored in a more secure way as compared to the file-based system. Using DBMS, we can overcome many problems such as- data redundancy, data inconsistency, easy access, more organized and understandable, and so on. There is the name of some popular Database Management System- MySQL, Oracle, SQL Server, Amazon simple DB (Cloud-based), etc.

2) What is a database?

A Database is a logical, consistent and organized collection of data that it can easily be accessed, managed and updated. Databases, also known as electronic databases are structured to provide the facility of creation, insertion, updating of the data efficiently and are stored in the form of a file or set of files, on the magnetic disk, tapes and another sort of secondary devices. Database mostly consists of the objects (tables), and tables

include of the records and fields. Fields are the basic units of data storage, which contain the information about a particular aspect or attribute of the entity described by the database.
DBMS is used for extraction of data from the database in the form of the queries.

3) What is a database system?
The collection of database and DBMS software together is known as a database system. Through the database system, we can perform many activities such asThe data can be stored in the database with ease, and there are no issues of data redundancy and data inconsistency.
The data will be extracted from the database using DBMS software whenever required. So, the combination of database and DBMS software enables one to store, retrieve and access data with considerate accuracy and security.

4) What are the advantages of DBMS?
o Redundancy control
o Restriction for unauthorized access
o Provides multiple user interfaces
o Provides backup and recovery
o Enforces integrity constraints
o Ensure data consistency
o Easy accessibility
o Easy data extraction and data processing due to the use of

queries

5. Difference between File System and DBMS:

Basis	File System	DBMS
Structure	The file system is software that manages and organizes the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It doesn't provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.
Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
Data Independence	There is no data independence.	In DBMS data independence exists.

User Access	Only one user can access data at a time.	Multiple users can access data at a time.
Meaning	The user has to write procedures for managing databases	The user not required to write procedures.
Sharing	Data is distributed in many files. So, not easy to share data	Due to centralized nature sharing is easy
Data Abstraction	It give details of storage and representation of data	It hides the internal details of Database
Integrity Constraints	Integrity Constraints are difficult to implement	Integrity constraints are easy to implement
Example	Cobol, C++	Oracle, SQL Server

6. Advantages of DBMS over File system:

Data redundancy and inconsistency: Redundancy is the concept of repetition of data i.e. each data may have more than a single copy. The file system cannot control the redundancy of data as each user defines and maintains the needed files for a specific application to run. There may be a possibility that two users are maintaining the data of the same file for different applications. Hence changes made by one user do not reflect in files used by second users, which leads to inconsistency of data. Whereas DBMS controls redundancy by maintaining a single repository of data that is defined once and is accessed by many users. As there is no or less redundancy, data remains consistent.

Data sharing: The file system does not allow sharing of data or sharing is too complex. Whereas in DBMS, data can be shared easily due to a centralized system.

Data concurrency: Concurrent access to data means more than one user is accessing the same data at the same time. Anomalies occur when changes made by one user get lost because of changes made by another user. The file system does

not provide any procedure to stop anomalies. Whereas DBMS provides a locking system to stop anomalies to occur.

Data searching: For every search operation performed on the file system, a different application program has to be written. While DBMS provides inbuilt searching operations. The user only has to write a small query to retrieve data from the database.

Data integrity: There may be cases when some constraints need to be applied to the data before inserting it into the database. The file system does not provide any procedure to check these constraints automatically. Whereas DBMS maintains data integrity by enforcing user-defined constraints on data by itself.

System crashing: In some cases, systems might have crashed due to various reasons. It is a bane in the case of file systems because once the system crashes, there will be no recovery of the data that's been lost. A DBMS will have the recovery manager which retrieves the data making it another advantage over file systems.

Data security: A file system provides a password mechanism to protect the database but how long can the password be protected? No one can guarantee that. This doesn't happen in the case of DBMS. DBMS has specialized features that help provide shielding to its data.

Backup: It creates a backup subsystem to restore the data if required.

Interfaces: It provides different multiple user interfaces like graphical user interface and application program interface.

Easy Maintenance: It is easily maintainable due to its centralized nature.

7. What is Data model & Importance of Data Model?

Data models can facilitate interaction among the designer, the applications programmer, and the end user.

A well-developed data model can even foster improved understanding of the organization for which the database design is developed. In short, data models are a communication tool.

The Importance of the Data Modelling

Data constitute the most basic information units employed by a system. Applications are created to manage data and to help transform data into information.

But data are viewed in different ways by different people. So that there is a huge importance of data modeling in DBMS.

For example, contrast the (data) view of a company manager with that of a company clerk. Although the manager and the clerk both work for the same company, the manager is more likely to have an enterprise-wide view of company data than the clerk.

Applications programmers have yet another view of data, being more concerned with data location, formatting, and specific reporting requirements.

Basically, applications programmers translate company policies and procedures from a variety of sources into appropriate interfaces, reports, and query screens.

When a good database blueprint is available, it does not matter that an applications programmer's view of the data is different from that of the manager and/or the end user. Conversely, when a good database blueprint is not available, problems are likely to occur.

For instance, an inventory management program and an order entry system may use conflicting product-numbering schemes, thereby costing the company thousands (or even millions) of dollars. The data model is an abstraction; you cannot draw the required data out of the data model.

8. Data Model Basic Building Blocks

Building blocks of a Data Model

A data model is a structure of the data that contains all the required details of the data like the name of the data, size of the data, relationship with other data and constraints that are applied on the data. It is a communication tool.

A data model is essential in order to store the database in a sorted manner. It will provide the interaction between the system analyst, designer and application programmer. It improves the understanding of designing of the database in which the organization is interested.

A data model constitutes of building blocks. They are:

1. Entities
2. Attributes
3. Relationships
4. Constraints

These are explained as following below in brief.

1. **Entities:**
   Entities are real time objects that exist.It can be a person, place, object, event, concept. Entities are represented by a rectangle box containing the entity name in it.
   Example: Student, employee.
2. **Attributes:**
   It is the set of characteristics representing an entity.It is represented by a ellipse symbol with attribute name on it.
   Example: A student has attributes like name, roll number, age and much more.
3. **Relationship:**
   It describes the association between two entities.It is represented using diamond symbol containing relationship name with it.The data model generally uses three kinds of relationships:one to many, many to many, one to one.
   Example: The relationship between two entities Student and Class has many to many relationship.
4. **Constraints**:
   Constraints are conditions applied on the data. It provides the data integrity.
   Example: A student can take a maximum of 2 books from the library is applied as a constraint on the student database.

9. Business rules can apply to many aspects of an organization and can be expressed in a variety of ways. In general, business rules define specific instructions or constraints on how certain day-to-day actions should be performed.

For example, business rules can include:

1. A decision-making approval structure for invoice processing where only certain managers can sign off on invoices totaling a specific amount
2. Calculations in which a formula may be used to calculate revenue or expenses
3. Policies where an organization requires its employees to work with a preferred list of vendors
4. When business rules are designed separately from process implementations, they provide a powerful and flexible approach to help organizations move more quickly in meeting their goals and better respond to changing business needs.

10. Evolution of data Model or types of Data Models

Managing data was the key and was essential. Therefore, data model originated to solve the file system issues. Here are the Data Models in DBMS −

Hierarchical Model

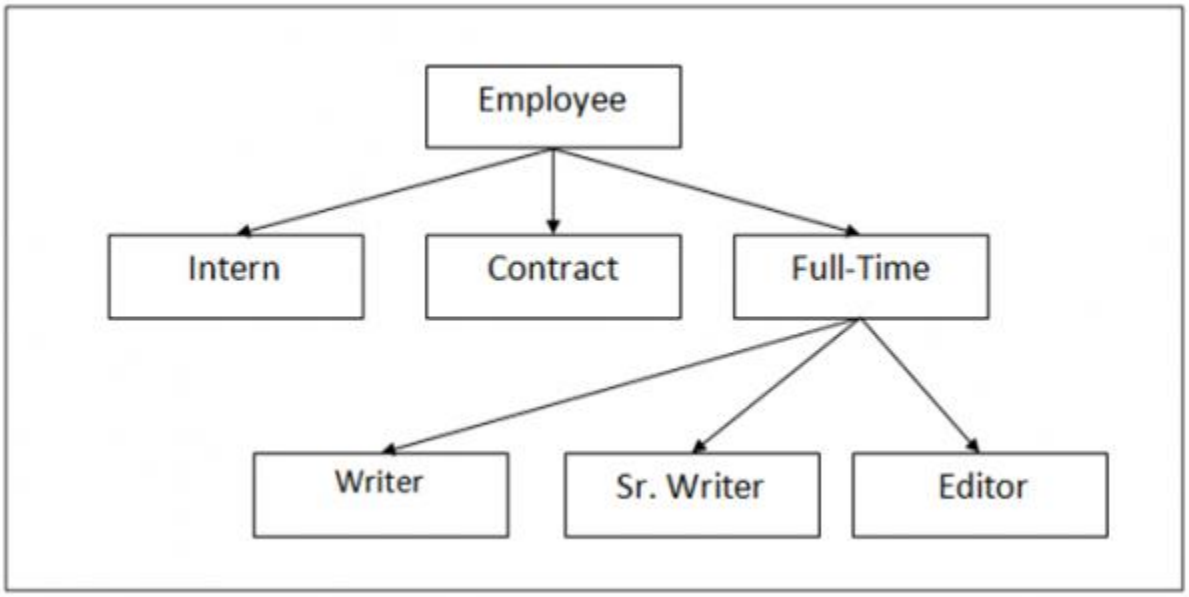
In Hierarchical Model, a hierarchical relation is formed by collection of relations and forms a tree-like structure.

The relationship can be defined in the form of parent child type.

One of the first and most popular Hierarchical Model is Information Management System (IMS), developed by IBM.

Example

The hierarchy shows an Employee can be an Intern, on Contract or Full- Time. Sub-levels show that Full-Time Employee can be hired as a Writer, Senior Writer or Editor:



Advantages

- The design of the hierarchical model is simple.
- Provides Data Integrity since it is based on parent/ child relationship
- Data sharing is feasible since the data is stored in a single database.
- Even for large volumes of data, this model works perfectly.

Disadvantages

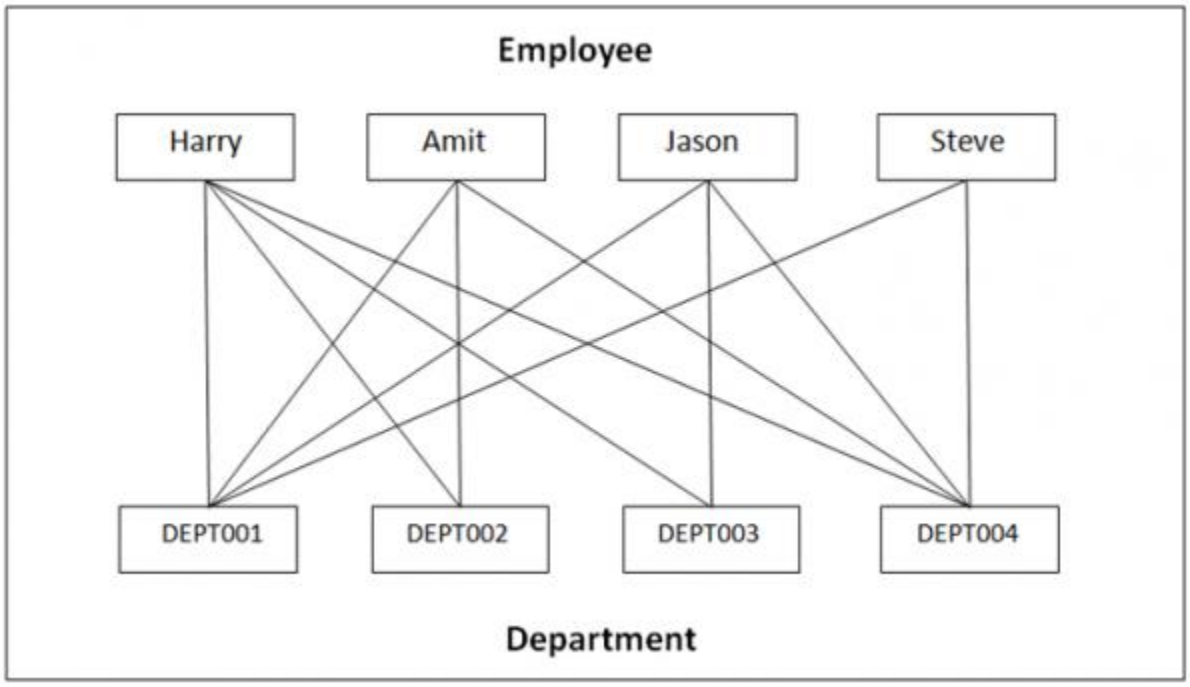
- Implementation is complex.
- This model has to deal with anomalies like Insert, Update and Delete.
- Maintenance is difficult since changes done in the database may want you to do changes in the entire database structure.

Network Model

The Hierarchical Model creates hierarchical tree with parent/ child relationship, whereas the Network Model has graph and links.

The relationship can be defined in the form of links and it handles many-to-many relations. This itself states that a record can have more than one parent.

Example



Advantages

- Easy to design the Network Model
- The model can handle one-one, one-to-many, many-to-many relationships.
- It isolates the program from other details.
- Based on standards and conventions.

Disadvantages

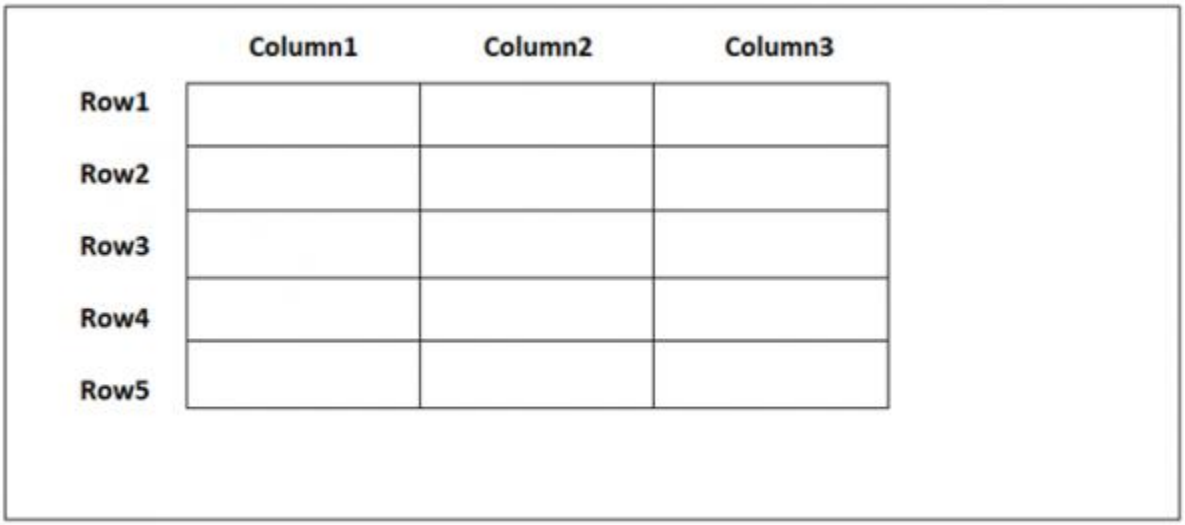
- Pointers bring complexity since the records are based on pointers and graphs.

- Changes in the database isn't easy that makes it hard to achieve structural independence.

Relational Model

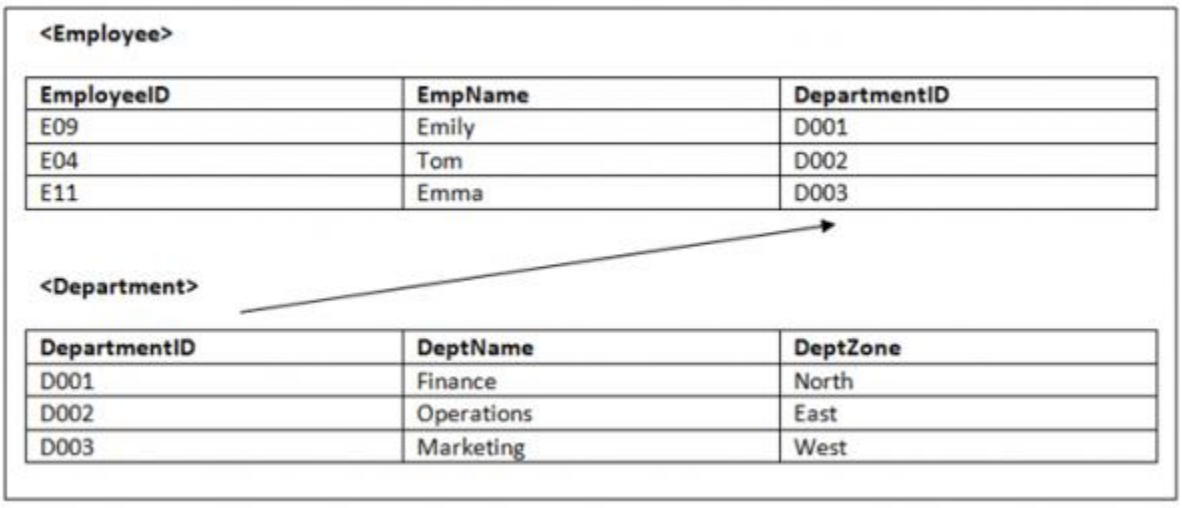
A relational model groups data into one or more tables. These tables are related to each other using common records.

The data is represented in the form of rows and columns i.e. tables:



Example

Let us see an example of two relations **<Employee>** and **<Department>** linked to each other, with **DepartmentID**, which is Foreign Key of **<Employee>** table and Primary key of **<Department>** table.



Advantages

- The Relational Model does not have any issues that we saw in the previous two models i.e. update, insert and delete anomalies have nothing to do in this model.
- Changes in the database do not require you to affect the complete database.
- Implementation of a Relational Model is easy.
- To maintain a Relational Model is not a tiresome task.

Disadvantages

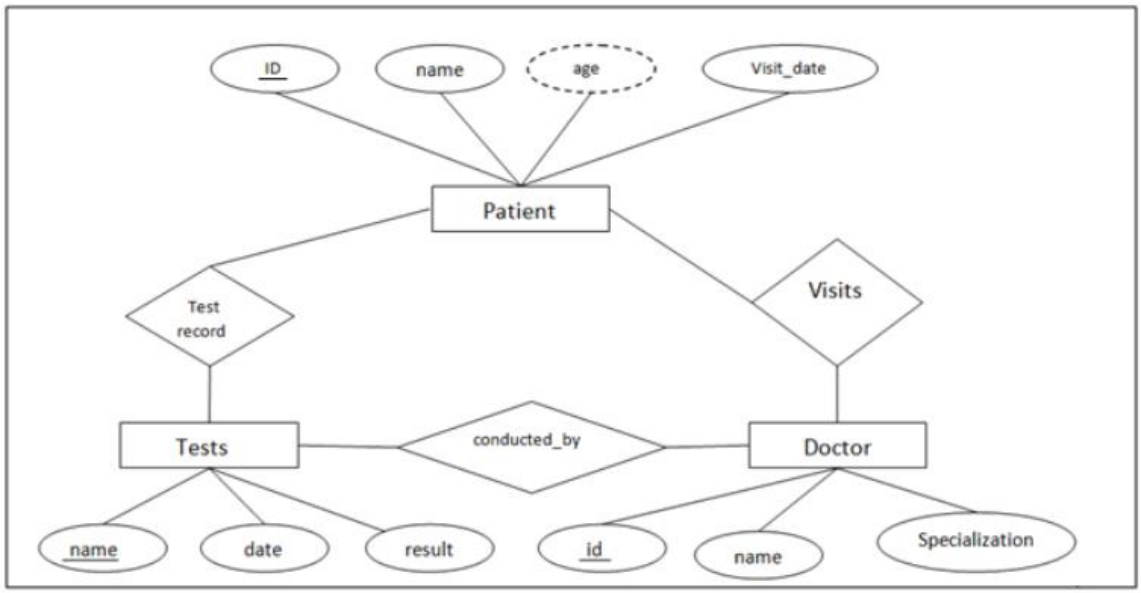
- Database inefficiencies hide and arise when the model has large volumes of data.
- The overheads of using relational data model come with the cost of using powerful hardware and devices.

Entity Relationship Data Model

ER model is used to represent real life scenarios as entities. The properties of these entities are their attributes in the ER diagram and their connections are shown in the form of relationships.

An ER model is generally considered as a top down approach in data designing.

An example of ER model is −



This is an ER model of a Hospital. The entities are represented in rectangular boxes and are Patient, Tests and Doctor.

Each of these entities have their respective attributes which are −

- Patients - ID (primary key), name, age,visit_date
- Tests- Name (primary key), date, result
- Doctor- ID (primary key), name, specialisation

11. Data Abstraction is a process of hiding unwanted or irrelevant details from the end user. It provides a different view and helps in achieving data independence which is used to enhance the security of data.

The database systems consist of complicated data structures and relations. For users to access the data easily, these complications are kept hidden, and only the relevant part of the database is made accessible to the users through data abstraction.

Levels of abstraction for DBMS

Database systems include complex data-structures. In terms of retrieval of data, reduce complexity in terms of usability of users and in order to make the system efficient, developers use levels of abstraction that hide irrelevant details from the users. Levels of abstraction simplify database design.

Mainly there are three levels of abstraction for DBMS, which are as follows −

Physical or Internal Level

Logical or Conceptual Level

View or External Level

Physical or Internal Level

It is the lowest level of abstraction for DBMS which defines how the data is actually stored, it defines data-structures to store data and access methods used by the database. Actually, it is decided by developers or database application programmers how to store the data in the database.

So, overall, the entire database is described in this level that is physical or internal level. It is a very complex level to understand. For example, customer's information is stored in tables and data is stored in the form of blocks of storage such as bytes, gigabytes etc.

Logical or Conceptual Level

Logical level is the intermediate level or next higher level. It describes what data is stored in the database and what relationship exists among those data. It tries to

describe the entire or whole data because it describes what tables to be created and what are the links among those tables that are created.

It is less complex than the physical level. Logical level is used by developers or database administrators (DBA). So, overall, the logical level contains tables (fields and attributes) and relationships among table attributes.

View or External Level

It is the highest level. In view level, there are different levels of views and every view only defines a part of the entire data. It also simplifies interaction with the user and it provides many views or multiple views of the same database.

View level can be used by all users (all levels' users). This level is the least complex and easy to understand.

For example, a user can interact with a system using GUI that is view level and can enter details at GUI or screen and the user does not know how data is stored and what data is stored, this detail is hidden from the user.