INTRODUCTION TO DATABASE

1. What is DataBase? Explain with an example on why should we need a database.

**Ans**- A database is a collection of data that is organized so the information within can be easily accessed later. Your data will be more accurate, reliable, and easy to use if you have a database.

A good database is crucial to any company or organization. This is because the database stores all the pertinent details about the company such as employee records, transactional records, salary details etc.

**The various reasons a database is important are −**

Manages large amounts of data

A database stores and manages a large amount of data on a daily basis. This would not be possible using any other tool such as a spreadsheet as they would simply not work.

Accurate

A database is pretty accurate as it has all sorts of build in constraints, checks etc. This means that the information available in a database is guaranteed to be correct in most cases.

Easy to update data

In a database, it is easy to update data using various Data Manipulation languages (DML) available. One of these languages is SQL.

Security of data

Databases have various methods to ensure security of data. There are user logins required before accessing a database and various access specifiers. These allow only authorised users to access the database.

Data integrity

This is ensured in databases by using various constraints for data. Data integrity in databases makes sure that the data is accurate and consistent in a database.

Easy to research data

It is very easy to access and research data in a database. This is done using Data Query Languages (DQL) which allow searching of any data in the database and performing computations on it.

1. Write a short note on the File base storage system. Explain the major challenges of a File base storage system.

**Ans-** A file-based storage system is a type of data storage system that stores data in individual files rather than in a database. It is one of the simplest and oldest methods of storing data and is still widely used today.

In a file-based storage system, each piece of data is stored in a separate file with a unique filename. The files are stored in a hierarchical directory structure that allows for easy organization and retrieval of data.

This type of storage system is commonly used for storing text files, images, videos, and other multimedia files. It is also used for storing software and program files.

One of the main advantages of file-based storage is its simplicity. It is easy to understand and implement and requires no special software or equipment. Additionally, it allows for the easy sharing of data between different systems and users.

**Although file-based storage systems are simple and easy to use, they can present some challenges as the amount of data stored in them grows. Here are some of the major challenges of a file-based storage system:**

**1. Data redundancy**: In a file-based system, data is stored in individual files. This can lead to data redundancy, where the same data is stored in multiple files, consuming unnecessary disk space.

**2. Data inconsistency:** With a file-based system, there is no centralized control over data. This can lead to data inconsistency, where the same data is stored in different formats in different files. This can make it difficult to ensure that data is accurate and up-to-date.

**3. Limited scalability:** As the amount of data stored in a file-based system grows, it can become difficult to manage and organize. This can limit the scalability of the system, making it challenging to store and retrieve large amounts of data efficiently.

**4. Security:** File-based systems may be vulnerable to security breaches, as individual files can be more easily accessed and manipulated. This can put sensitive data at risk, particularly if proper security measures are not in place.

**5. Limited functionality:** File-based systems have limited functionality compared to database systems. They lack the ability to perform complex queries and other advanced data manipulation tasks.

Overall, file-based storage systems are simple and easy to use, but they may not be the best choice for organizations that need to manage large amounts of data efficiently and securely.

1. What is DBMS? What was the need for DBMS?

Ans**- DBMS stands for Database Management System, which is a software system used to manage, organize, and manipulate data stored in a database. It is a collection of programs that enables users to access and manipulate data in an organized and efficient manner.**

**A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.**

The need for Database Management Systems (DBMS) arises from the limitations of traditional file-based systems. Some of the key reasons why organizations need DBMS are:

1. **Data Management:** DBMS provides a centralized approach to data management that ensures data integrity, consistency, and security. It allows defining constraints and rules to ensure that data is accurate and consistent across different applications and users.
2. **Data Sharing:** DBMS allows sharing data across different applications and users, reducing the need for duplicating data. It enables sharing data between different departments and systems within an organization, improving collaboration and decision-making.
3. **Data Integration:** DBMS allows integrating data from different sources, providing a comprehensive view of the data. It enables data integration from different systems and platforms, improving the quality of data analysis.
4. **Scalability:** DBMS provides a scalable solution for managing and processing large amounts of data. It allows adding new data and users without affecting the performance or stability of the system.
5. **Data Backup and Recovery:** DBMS provides backup and recovery mechanisms that ensure data is not lost in case of a system failure. It allows restoring data to a specific point in time, ensuring data consistency.
6. **Reduced Data Redundancy:** DBMS eliminates data redundancy by storing data in a structured way. It allows sharing data across different applications and users, reducing the need for duplicating data.
7. **Improved Data Access and Availability:** DBMS provides efficient data access and retrieval mechanisms that enable quick and easy data access. It allows multiple users to access the data simultaneously, ensuring data availability.

Overall, DBMS was developed to address the challenges of managing large amounts of data efficiently and accurately. It has become an essential tool in organizations of all sizes and across all industries.

1. Explain 5 challenges of file-based storage system which was tackled by DBMS.

Ans- **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.

DBMS is continuously evolving from time to time. It is a powerful tool for data storage and protection.

1. List out the different types of classification in DBMS and explain them in depth.

Ans- Classification of Database Management System

**1.** **Based on Data Model**

a) **Relational Data Model**

The relational data model is a database model that represents data in a tabular format, with each row in the table representing a unique instance of the data, and each column representing a specific attribute or characteristic of the data.

One of the main benefits of the relational data model is its simplicity and ease of use. Tables can be easily created, modified, and queried using SQL (Structured Query Language), which is a standard language used for managing relational databases. Additionally, the model allows for easy maintenance and scaling of databases, making it a popular choice for a wide range of applications and industries.

**b) Entity-Relationship Model**

The Entity-Relationship (ER) model is a database model that is used to design and represent relationships between entities in a database.

In the ER model, an entity is a real-world object or concept that has properties or attributes that describe it. For example, in a university database, a student is an entity with attributes such as name, ID number, and GPA.

Entities are connected by relationships, which represent the connections between them. Relationships can be one-to-one, one-to-many, or many-to-many. For example, a student can have one or many courses, while a course can have many students.

**c) Object-Based Data Model**

An object-based data model is a database model that represents data as objects, with each object consisting of data and its associated methods or functions. It is based on object-oriented programming principles and allows for the representation of complex data structures.

In an object-based data model, objects are organized into classes, which define the attributes and methods that all objects of that class will have. For example, in a database for a library, there could be a class called "Book," with attributes such as title, author, and publication date, and methods such as check-out and return.

Objects can be created from these classes, with each object having its own values for the attributes and the ability to perform the methods defined in the class. For example, an object could be created for a specific book, with its own values for the attributes, such as the title "To Kill a Mockingbird," and the ability to perform methods such as check-out and return.

The object-based data model allows for complex relationships between objects, such as inheritance, where one class can inherit attributes and methods from another class, and composition, where one class can be composed of one or more other classes.

Overall, the object-based data model is a powerful tool for representing complex data structures and is commonly used in scientific, engineering, and other complex applications.

**2. Classifications based on Number of Users**

The database management system can also be classified on the basis of its user. So, a DBMS can either be used by a single user or it can be used by multiple users.  The database system that can be used by a single user at a time is referred to **as a single-user system** and the database system that can be used by multiple users at a time is referred to as a **multiple user system**.

**3. Based on the Database Distribution**

Depending on the distribution of the database over numerous sites we can classify the database as:

**a) Centralized DBMS**

In the centralized DBMS, the entire database is stored in a single computer site. Though the centralized database supports multiple users still the DBMS software and the data both are stored on a single computer site.

**b) Distributed DBMS**

In the distributed DBMS (DDBMS) the database and the DBMS software are distributed over many computer sites. These computer sites are connected via a computer network.

**4. Classification Based on Application**

OLTP (Online Transaction Processing) DBMS: Used for transaction processing, where a large number of small transactions are performed in real-time. These systems are optimized for high-speed transaction processing, with low latency and high throughput.

OLAP (Online Analytical Processing) DBMS: Used for data analysis and decision-making. These systems are optimized for complex queries and data analysis, with high query throughput and the ability to handle large volumes of data.

**5.Based on Architecture**

**Centralized DBMS**: Stores data on a single server, with all processing performed on that server. This type of architecture is useful for small to medium-sized applications, where data access requirements are not too complex.

**Distributed DBMS:** Stores data on multiple servers, with processing distributed across those servers. This type of architecture is useful for large-scale applications, where data access requirements are complex and data needs to be accessed from different locations.

**6.Based on Data Storage Type**

**In-memory DBMS:** Stores data in memory, allowing for extremely fast data access and processing.

**Disk-based DBMS:** Stores data on disk, which is slower than memory-based storage but allows for larger volumes of data to be stored.

The choice of DBMS type depends on the specific needs of the organization or application, and factors such as data access requirements, data volume, and performance requirements.

1. What is the significance of Data Modelling and explain the types of data modeling.

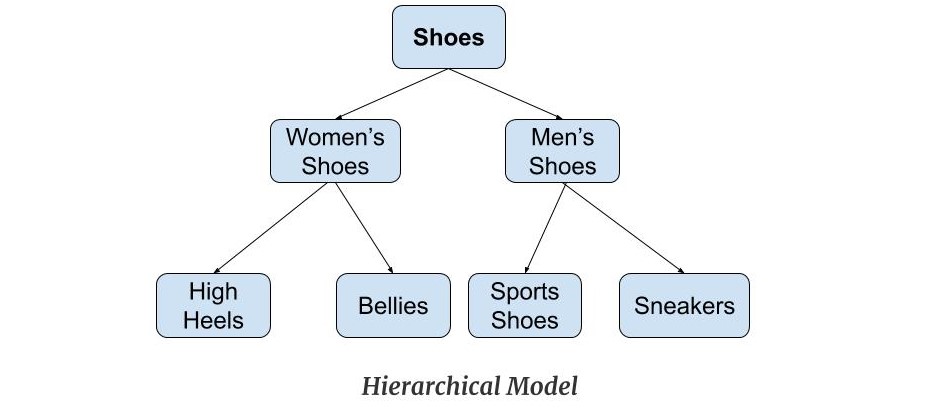
Ans- Data modeling is the process of creating a conceptual representation of data structures and relationships between them, which helps in understanding and organizing data in a clear and efficient manner. It provides a way to define, manipulate, and manage data to support business objectives, data analysis, and decision-making.

It defines the data elements and the relationships between the data elements. Data Models are used to show how data is stored, connected, accessed, and updated in the database management system. Here, we use a set of symbols and text to represent the information so that members of the organization can communicate and understand it. Though there are many data models being used nowadays but the Relational model is the most widely used model.

Some of the Data Models in DBMS are:

**Hierarchical Model**

The hierarchical Model was the first DBMS model. This model organises the data in the hierarchical tree structure. The hierarchy starts from the root which has root data and then it expands in the form of a tree adding a child node to the parent node. This model easily represents some of the real-world relationships like food recipes, sitemaps of a website, etc. Example: We can represent the relationship between the shoes present on a shopping website in the following way:

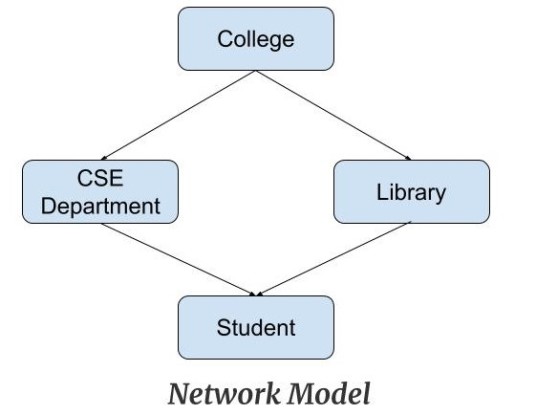


**Features of a Hierarchical Model**

1. **One-to-many relationship*:***The data here is organized in a tree-like structure where the one-to-many relationship is between the datatypes. Also, there can be only one path from the parent to any node. ***Example:***In the above example, if we want to go to the node *sneakers*we only have one path to reach there i.e through the men's shoe node.
2. **Parent-Child Relationship*:***Each child node has a parent node but a parent node can have more than one child node. Multiple parents are not allowed.

**Network Model**

This model is an extension of the hierarchical model. It was the most popular model before the relational model. This model is the same as the hierarchical model, the only difference is that a record can have more than one parent. It replaces the hierarchical tree with a graph. ***Example:***In the example below we can see that node student has two parents i.e. CSE Department and Library. This was earlier not possible in the hierarchical model.



**Features of a Network Model**

1. **Ability to Merge more Relationships*:***In this model, as there are more relationships so data is more related. This model has the ability to manage one-to-one relationships as well as many-to-many relationships.
2. **Many paths*:***As there are more relationships so there can be more than one path to the same record. This makes data access fast and simple.

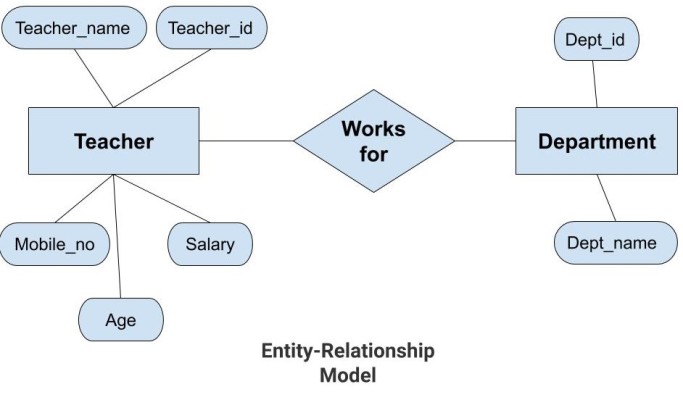
**Entity-Relationship Model**

Entity-Relationship Model or simply ER Model is a high-level data model diagram. In this model, we represent the real-world problem in the pictorial form to make it easy for the stakeholders to understand. It is also very easy for the developers to understand the system by just looking at the ER diagram. We use the ER diagram as a visual tool to represent an ER Model. ER diagram has the following three components:

**Entities:** Entity is a real-world thing. It can be a person, place, or even a concept. Example: Teachers, Students, Course, Building, Department, etc are some of the entities of a School Management System.

**Attributes:** An entity contains a real-world property called attribute. This is the characteristics of that attribute. Example: The entity teacher has the property like teacher id, salary, age, etc.

**Relationship**: Relationship tells how two entities are related. Example: Teacher works for a department.



In the above diagram, the entities are Teacher and Department. The attributes of ***Teacher***entity are Teacher\_Name, Teacher\_id, Age, Salary, and Mobile\_Number. The attributes of entity ***Department***entity are Dept\_id, and Dept\_name. The two entities are connected using the relationship. Here, each teacher works for a department.

**Features of ER Model**

* **Graphical Representation for Better Understanding*:***It is very easy and simple to understand so it can be used by the developers to communicate with the stakeholders.
* **ER Diagram*:***ER diagram is used as a visual tool for representing the model.
* **Database Design:**This model helps the database designers to build the database and is widely used in database design.

**Relational Model**

The relational Model is the most widely used model. In this model, the data is maintained in the form of a two-dimensional table. All the information is stored in the form of rows and columns. The basic structure of a relational model is tables. So, the tables are also called relations in the relational model. Example: In this example, we have an Employee table.



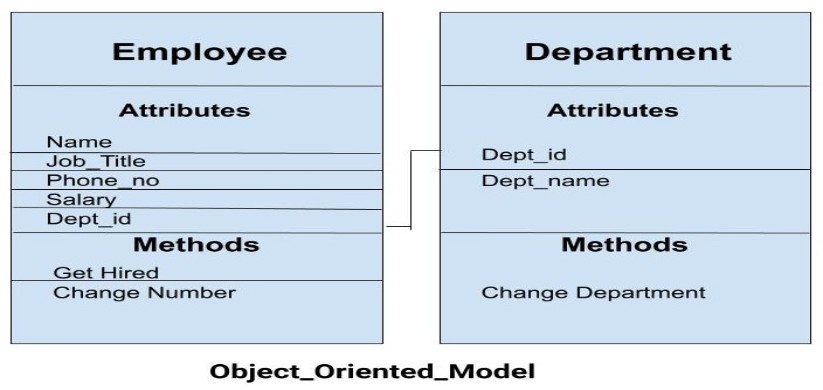
**Features of Relational Model**

**Tuples :** Each row in the table is called tuple. A row contains all the information about any instance of the object. In the above example, each row has all the information about any specific individual like the first row has information about John.

**Attribute or field:** Attributes are the property which defines the table or relation. The values of the attribute should be from the same domain. In the above example, we have different attributes of the employee like Salary, Mobile\_no, etc.

**Object-Oriented Data Model**

The real-world problems are more closely represented through the object-oriented data model. In this model, both the data and relationship are present in a single structure known as an object. We can store audio, video, images, etc in the database which was not possible in the relational model(although you can store audio and video in relational database, it is adviced not to store in the relational database). In this model, two are more objects are connected through links. We use this link to relate one object to other objects. This can be understood by the example given below.

 In the above example, we have two objects Employee and Department. All the data and relationships of each object are contained as a single unit. The attributes like Name, Job\_title of the employee, and the methods which will be performed by that object are stored as a single object. The two objects are connected through a common attribute i.e the Department\_id and the communication between these two will be done with the help of this common id.

7.Explain 3 schema architecture along with its advantages.

Ans- The three-schema architecture divides the database into three-level used to create a separation between the physical database and the user application. In simple terms, this architecture hides the details of physical storage from the user.

The database administrator (DBA) responsible is to change the structure of database storage without affecting the user’s view. It deals with the data, the relationship between them and the different access methods implemented on the database. The logical design of database is called a schema

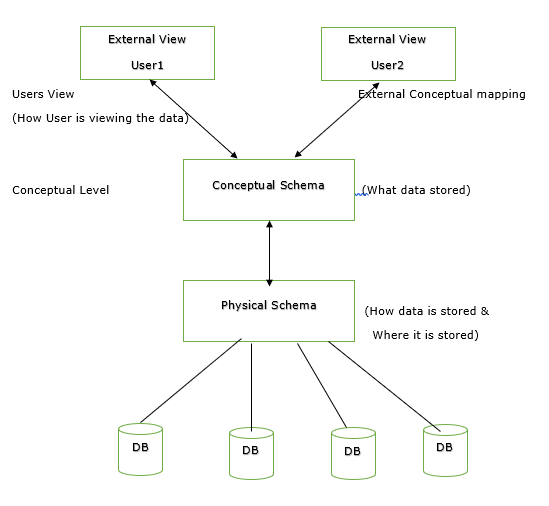
This architecture contains three layers of database management system, which are as follows −

* External level
* Conceptual level
* Internal level

The ANSI-SPARC database architecture is the basis of most of the modern databases.

The three levels present in this architecture are Physical level, Conceptual level and External level.

**The three level schema architecture in DBMS is given below −**



The details of these levels are as follows −

## **Physical Level**

This is the lowest level in the three level architecture. It is also known as the internal level. The physical level describes how data is actually stored in the database. In the lowest level, this data is stored in the external hard drives in the form of bits and at a little high level, it can be said that the data is stored in files and folders. The physical level also discusses compression and encryption techniques.

## **Conceptual Level**

The conceptual level is at a higher level than the physical level. It is also known as the logical level. It describes how the database appears to the users conceptually and the relationships between various data tables. The conceptual level does not care for how the data in the database is actually stored.

## **External Level**

This is the highest level in the three level architecture and closest to the user. It is also known as the view level. The external level only shows the relevant database content to the users in the form of views and hides the rest of the data. So different users can see the database as a different view as per their individual requirements.