

## DBMS INTRODUCTION

### Q #1) What do you understand by 'Database'?

**Answer:** Database is an organized collection of related data where the data is stored and organized to serve some specific purpose.

**For Example,** A librarian maintain a database of all the information related to the books that are available in the library.

### Q #2) Define DBMS.

**Answer:** DBMS stands for Database Management System. It is a collection of application programs which allow the user to organize, restore and retrieve information about data efficiently and as effectively as possible.

Some of the popular DBMS's are MySql, Oracle, Sybase, etc.

### Q #3) Define RDBMS.

**Answer:** Relational Database Management System(RDBMS) is based on a relational model of data that is stored in databases in separate tables and they are related to the use of a common column. Data can be accessed easily from the relational database using Structured Query Language (SQL).

### Q #4) Enlist the advantages of DBMS.

**Answer: The advantages of DBMS includes:**

- Data is stored in a structured way and hence redundancy is controlled.
- Validates the data entered and provide restrictions on unauthorized access to the database.
- Provides backup and recovery of the data when required.
- It provides multiple user interfaces.

### Q #5) What do you understand by Data Redundancy?

**Answer:** Duplication of data in the database is known as data redundancy. As a result of data redundancy, duplicated data is present at multiple locations, hence it leads to wastage of the storage space and the integrity of the database is destroyed.

### Q #6) What is DDL interpreter?

**Answer:** DDL Interpreter interprets the DDL statements and records the generated statements in the table containing metadata.

### Q #7) Define DML Compiler.

**Answer:** DML compiler translates DML statements in a query language into a low-level instruction and the generated instruction can be understood by Query Evaluation Engine.

### Q #8) Explain the terms 'Record', 'Field' and 'Table' in terms of database.

**Answer:**

**Record:** Record is a collection of values or fields of a specific entity. **For Example,** An employee, Salary account, etc.

**Field:** A field refers to an area within a record that is reserved for specific data. **For Example,** Employee ID.

**Table:** Table is the collection of records of specific types. **For Example,** the Employee table is a collection of records related to all the employees.

### Q #9) What do you understand by Data Independence? What are its two types?

**Answer:** Data Independence refers to the ability to modify the schema definition in one level in such a way that it does not affect the schema definition in the next higher level.

**The 2 types of Data Independence are:**

- **Physical Data Independence:** It modifies the schema at the physical level without affecting the schema at the conceptual level.
- **Logical Data Independence:** It modifies the schema at the conceptual level without affecting or causing changes in the schema at the view level.

### Q #10) Define the relationship between 'View' and 'Data Independence'.

**Answer:** View is a virtual table that does not have its data on its own rather the data is defined from one or more underlying base tables.

Views account for logical data independence as the growth and restructuring of base tables are not reflected in views.

### Q #11) What are the advantages and disadvantages of views in the database?

**Answer: Advantages of Views:**

- As there is no physical location where the data in the view is stored, it generates output without wasting resources.
- Data access is restricted as it does not allow commands like insertion, updation, and deletion.

**Disadvantages of Views:**

- The view becomes irrelevant if we drop a table related to that view.
- Much memory space is occupied when the view is created for large tables.

**Q #12) What do you understand by the E-R model?**

**Answer:** E-R model is an Entity-Relationship model which defines the conceptual view of the database.

The E-R model basically shows the real-world entities and their association/relations. Entities here represent the set of attributes in the database.

**Q #13) Define Entity, Entity type, and Entity set.**

**Answer:**

**Entity** can be anything, be it a place, class or object which has an independent existence in the real world.

**Entity Type** represents a set of entities that have similar attributes.

**Entity Set** in the database represents a collection of entities having a particular entity type.

**Q #14) Explain the terms ‘Attribute’ and ‘Relations’**

**Answer:**

**Attribute** is described as the properties or characteristics of an entity. For Example, Employee ID, Employee Name, Age, etc., can be attributes of the entity Employee.

**Relation** is a two-dimensional table containing a number of rows and columns where every row represents a record of the relation. Here, rows are also known as ‘Tuples’ and columns are known as ‘Attributes’.

**Q #15) What is the Database transaction?**

**Answer:** Sequence of operation performed which changes the consistent state of the database to another is known as the database transaction. After the completion of the transaction, either the successful completion is reflected in the system or the transaction fails and no change is reflected.

**Q #16) What do you understand by Join?**

**Answer:** Join is the process of deriving the relationship between different tables by combining columns from one or more tables having common values in each. When a table joins with itself, it is known as Self Join.

**Q #17) Explain the Data Dictionary.**

**Answer:** Data dictionary is a set of information describing the content and structure of the tables and database objects. The job of the information stored in the data dictionary is to control, manipulate and access the relationship between database elements.

**Q #18) Different types of Database Users**

Database users are categorized based up on their interaction with the data base.

These are seven types of data base users in DBMS.

1. Database Administrator (DBA) :  
Database Administrator (DBA) is a person/team who defines the schema and also controls the 3 levels of database. The DBA will then create a new account id and password for the user if he/she need to access the data base. DBA is also responsible for providing security to the data base and he allows only the authorized users to access/modify the data base.
  - DBA also monitors the recovery and back up and provide technical support.
  - The DBA has a DBA account in the DBMS which called a system or superuser account.
  - DBA repairs damage caused due to hardware and/or software failures.
2. Naive / Parametric End Users :  
Parametric End Users are the unsophisticated who don’t have any DBMS knowledge but they frequently use the data base applications in their daily life to get the desired results.  
For examples, Railway’s ticket booking users are naive users. Clerks in any bank is a naive user because they don’t have any DBMS knowledge but they still use the database and perform their given task.
3. System Analyst :  
System Analyst is a user who analyzes the requirements of parametric end users. They check whether all the requirements of end users are satisfied.

4. **Sophisticated Users :**  
Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database. They can develop their own data base applications according to their requirement. They don't write the program code but they interact the data base by writing SQL queries directly through the query processor.
5. **Data Base Designers :**  
Data Base Designers are the users who design the structure of data base which includes tables, indexes, views, constraints, triggers, stored procedures. He/she controls what data must be stored and how the data items to be related.
6. **Application Program :**  
Application Program are the back end programmers who writes the code for the application programs. They are the computer professionals. These programs could be written in Programming languages such as Visual Basic, Developer, C, FORTRAN, COBOL etc.
7. **Casual Users / Temporary Users :**  
Casual Users are the users who occasionally use/access the data base but each time when they access the data base they require the new information, for example, Middle or higher level manager.

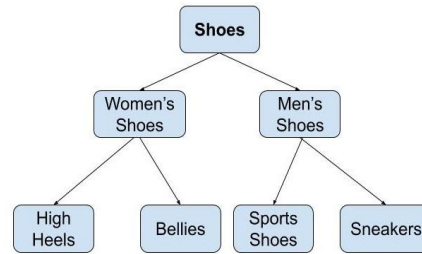
**Q #18) What is Data Model in DBMS and what are its types?**

Data Model gives us an idea that how the final system will look like after its complete implementation. 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 organisation can communicate and understand it. Though there are many data models being used nowadays but the Relational model is the most widely used model. Apart from the Relational model, there are many other types of data models about which we will study in details in this blog. Some of the Data Models in DBMS are:

1. Hierarchical Model
2. Network Model
3. Entity-Relationship Model
4. Relational Model
5. Object-Oriented Data Model
6. Object-Relational Data Model
7. Flat Data Model
8. Semi-Structured Data Model
9. Associative Data Model
10. Context Data Model

**Hierarchical Model**

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 child node to the parent node. This model easily represents some of the real-world relationships like food recipes, sitemap of a website etc. Example: We can represent the relationship between the shoes present on a shopping website in the following way:



*Hierarchical Model*

### Features of a Hierarchical Model

1. One-to-many relationship: The data here is organised in a tree-like structure where the one-to-many relationship is between the datatypes. Also, there can be only one path from 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 men's shoes 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.
3. Deletion Problem: If a parent node is deleted then the child node is automatically deleted.
4. Pointers: Pointers are used to link the parent node with the child node and are used to navigate between the stored data. Example: In the above example the 'shoes' node points to the two other nodes 'women shoes' node and 'men's shoes' node.

### Advantages of Hierarchical Model

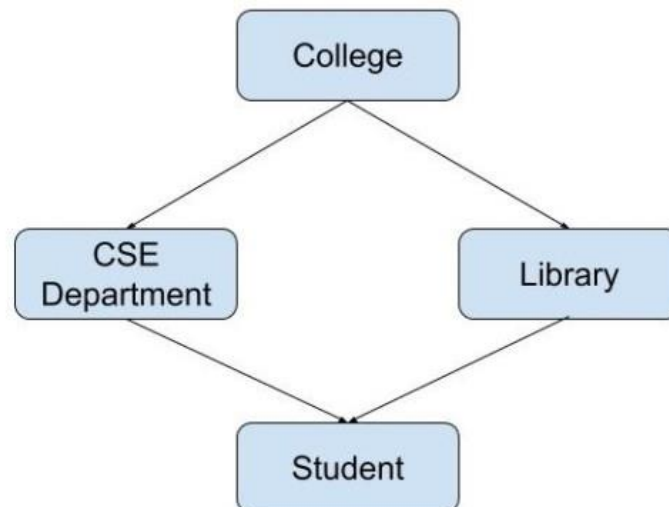
- It is very simple and fast to traverse through a tree-like structure.
- Any change in the parent node is automatically reflected in the child node so, the integrity of data is maintained.

### Disadvantages of Hierarchical Model

- Complex relationships are not supported.
- As it does not support more than one parent of the child node so if we have some complex relationship where a child node needs to have two parent node then that can't be represented using this model.
- If a parent node is deleted then the child node is automatically deleted.

### 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.



*Network 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.
3. Circular Linked List: The operations on the network model are done with the help of the circular linked list. The current position is maintained with the help of a program and this position navigates through the records according to the relationship.

#### Advantages of Network Model

- The data can be accessed faster as compared to the hierarchical model. This is because the data is more related in the network model and there can be more than one path to reach a particular node. So the data can be accessed in many ways.
- As there is a parent-child relationship so data integrity is present. Any change in parent record is reflected in the child record.

#### Disadvantages of Network Model

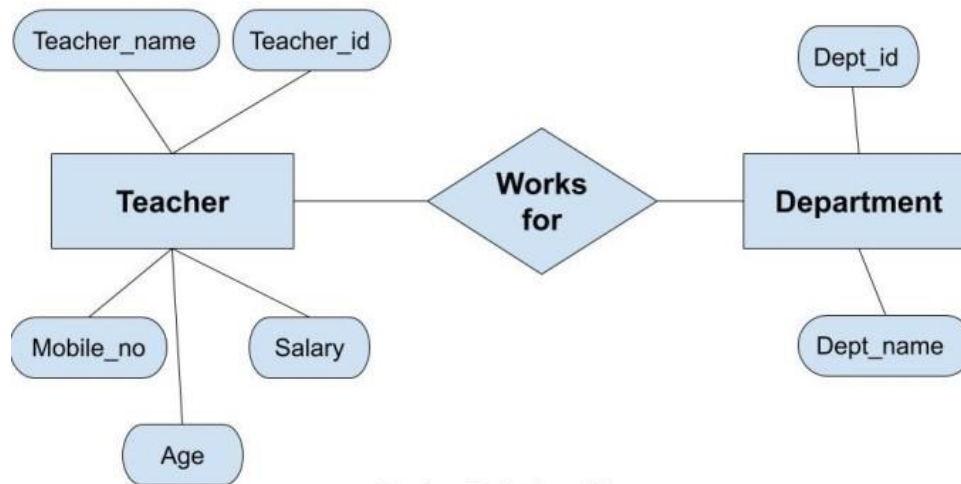
- As more and more relationships need to be handled the system might get complex. So, a user must be having detailed knowledge of the model to work with the model.
- Any change like updation, deletion, insertion is very complex.

### 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 attributes are related. Example: Teacher works for a department.

Example:



**Entity-Relationship Model**

In the above diagram, the entities are Teacher and Department. The attributes of Teacher entity are Teacher\_Name, Teacher\_id, Age, Salary, Mobile\_Number. The attributes of entity Department entity are Dept\_id, 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.

#### Advantages of ER Model

- Simple: Conceptually ER Model is very easy to build. If we know the relationship between the attributes and the entities we can easily build the ER Diagram for the model.
- Effective Communication Tool: This model is used widely by the database designers for communicating their ideas.
- Easy Conversion to any Model: This model maps well to the relational model and can be easily converted relational model by converting the ER model to the table. This model can also be converted to any other model like network model, hierarchical model etc.

#### Disadvantages of ER Model

- No industry standard for notation: There is no industry standard for developing an ER model. So one developer might use notations which are not understood by other developers.
- Hidden information: Some information might be lost or hidden in the ER model. As it is a high-level view so there are chances that some details of information might be hidden.

#### Relational Model

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 row 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.

Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	John	Engineer	100000	9111037890	2	99
AfterA002	Adam	Analyst	50000	9587569214	3	100
AfterA003	Kande	Manager	890000	7895212355	2	65

### 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.

#### Advantages of Relational Model

- Simple: This model is more simple as compared to the network and hierarchical model.
- Scalable: This model can be easily scaled as we can add as many rows and columns we want.
- Structural Independence: We can make changes in database structure without changing the way to access the data. When we can make changes to the database structure without affecting the capability to DBMS to access the data we can say that structural independence has been achieved.

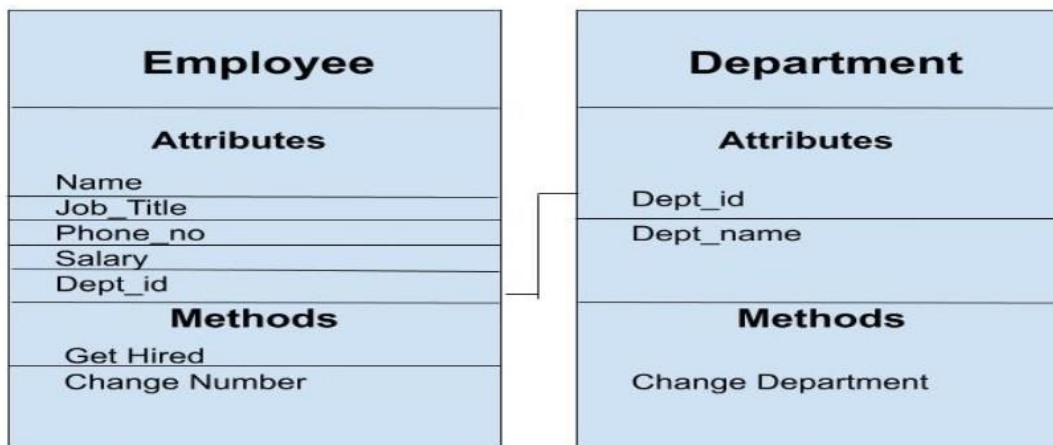
#### Disadvantages of Relational Model

- Hardware Overheads: For hiding the complexities and making things easier for the user this model requires more powerful hardware computers and data storage devices.
- Bad Design: As the relational model is very easy to design and use. So the users don't need to know how the data is stored in order to access it. This ease of design can lead to the development of a poor database which would slow down if the database grows.

But all these disadvantages are minor as compared to the advantages of the relational model. These problems can be avoided with the help of proper implementation and organisation.

### 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 advised not to store in the relational database). In this model, two or 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.



**Object\_Oriented\_Model**

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.

### Object-Relational Model

As the name suggests it is a combination of both the relational model and the object-oriented model. This model was built to fill the gap between object-oriented model and the relational model. We can have many advanced features like we can make complex data types according to our requirements using the existing data types. The problem with this model is that this can get complex and difficult to handle. So, proper understanding of this model is required.

### Flat Data Model

It is a simple model in which the database is represented as a table consisting of rows and columns. To access any data, the computer has to read the entire table. This makes the modes slow and inefficient.

### Semi-Structured Model

Semi-structured model is an evolved form of the relational model. We cannot differentiate between data and schema in this model. Example: Web-Based data sources which we can't differentiate between the schema and data of the website. In this model, some entities may have missing attributes while others may have an extra attribute. This model gives flexibility in storing the data. It also gives flexibility to the attributes. Example: If we are storing any value in any attribute then that value can be either atomic value or a collection of values.

### Associative Data Model

Associative Data Model is a model in which the data is divided into two parts. Everything which has independent existence is called as an entity and the relationship among these entities are called association. The data divided into two parts are called items and links.

- Item: Items contain the name and the identifier(some numeric value).
- Links: Links contain the identifier, source, verb and subject.

Example: Let us say we have a statement "The world cup is being hosted by London from 30 May 2020". In this data two links need to be stored:

1. The world cup is being hosted by London. The source here is 'the world cup', the verb 'is being' and the target is 'London'.
2. ...from 30 May 2020. The source here is the previous link, the verb is 'from' and the target is '30 May 2020'.

This is represented using the table as follows:

### Items

Identifiers	Name
89	The world cup
95	Is being hosted
40	By London
44	from
10	30 May 2020

### Links

Identifiers	Source	Verb	Target
70	89	95	40
75	70	44	10

## ASSOCIATIVE MODEL

### Context Data Model

Context Data Model is a collection of several models. This consists of models like network model, relational models etc. Using this model we can do various types of tasks which are not possible using any model alone.