Q.1 The difference between the file system and DBMS system.

1.	File system is a software that manages and organizes the files in a storage medium within a computer.	DBMS is a software for managing the database.
2.	Redundant data can be present in a file system.	In DBMS there is no redundant data.
3.	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.
4.	There is no efficient query processing in file system.	Efficient query processing is there in DBMS.
5.	There is less data consistency in file system.	There is more data consistency because of the process of normalization.
6.	It is less complex as compared to DBMS.	It has more complexity in handling as compared to file system.
7.	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file system.
8.	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.

Q.2 What is data Abstraction in DBMS system. And write their level of abstraction.

- For the system to be usable, it must retrieve data efficiently.
- The need for efficiency has led designers to use complex data structures to represent data in the database.
- Since many database-systems users are not computer trained, developers hide the complexity from users through several levels of abstraction, to simplify users' interactions with the system

Level of Abstraction for DBMS

Physical level

- The lowest level of abstraction describes *how* the data are actually stored.
- The physical level describes complex low-level data structures in detail.

Logical level

- The next-higher level of abstraction describes *what* data are stored in the database, and what relationships exist among those data.
- The logical level thus describes the entire database in terms of a small number of relatively simple structures.
- Although implementation of the simple structures at the logical level may involve complex physical-level structures, the user of the logical level does not need to be aware of this complexity.
- Database administrators, who must decide what information to keep in the database, use the logical level of abstraction.

View level

- The highest level of abstraction describes only part of the entire database.
- Even though the logical level uses simpler structures, complexity remains because of the variety of information stored in a large database.
- Many users of the database system do not need all this information.
- They need to access only a part of the database.
- The view level of abstraction exists to simplify their interaction with the system.
- The system may provide many views for the same database.

Example: view of data

- At the physical level, a *customer*, *account*, or *employee* record can be described as a block of consecutive storage locations (for example, words or bytes).
- The language compiler hides this level of detail from programmers.

Q.3 What are the various data model is available in DBMS system explain anyone with example.

Data models

- A **data model** is a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.
- The entity–relationship (E-R) model is a high-level data model. It is based on a perception of a real world that consists of a collection of basic objects, called *entities*, and of *relationships* among these objects.
- The relational model is a lower-level model. It uses a collection of tables to represent both data and the relationships among those data.
- Today a vast majority of database products are based on the relational model.
- Designers often formulate database schema design by first modeling data at a high level, using the E-R model, and then translating it into the relational model.

- **A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system.**
- **While the Relational Model is the most widely used database model, there are other models too:**

DBMS Database Models

- **Hierarchical Model**
- **Network Model**
- **Entity-relationship Model**
- **Relational Model**

- The structure of a database is the **data model: a collection of conceptual** Tools for describing data, data relationships, data semantics, and consistency constraints.
- Data models provide a way to describe the design of a database at the logical level.

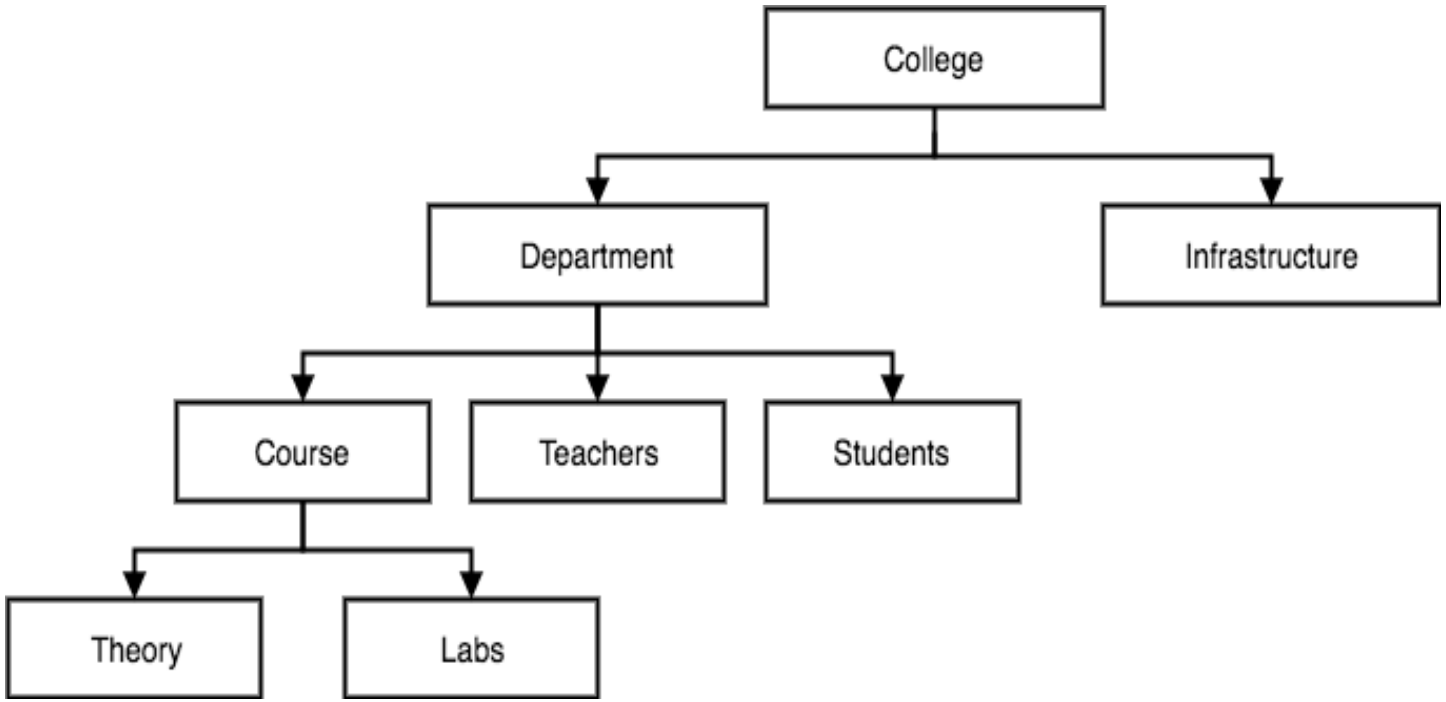
Hierarchical Model

- This database model organizes data into a tree-like-structure, with a single root, to which all the other data is linked.
- The hierarchy starts from the **Root** data, and expands like a tree, adding child nodes to the parent nodes.

In this model, a child node will only have a single parent node.

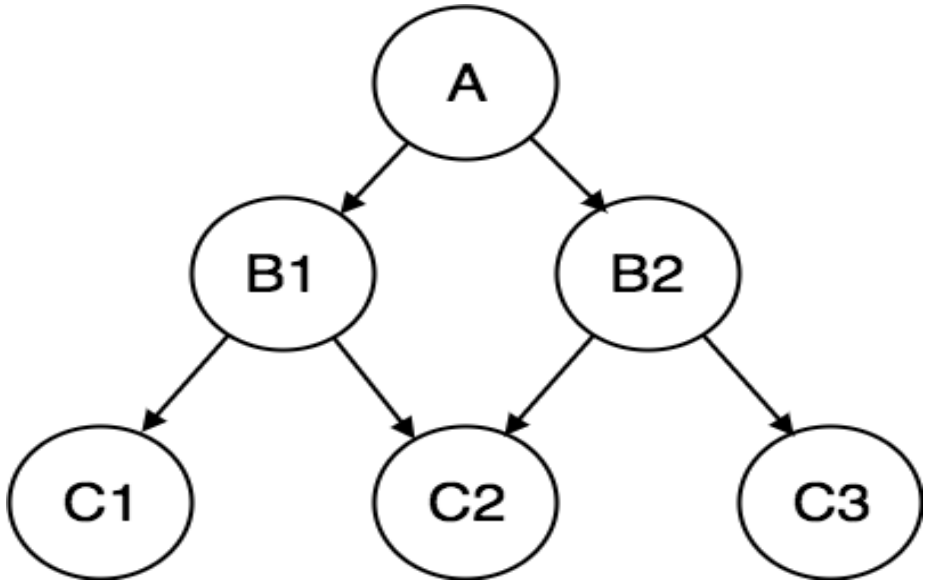
This model efficiently describes many real-world relationships like index of a book, recipes etc.

In hierarchical model, data is organized into tree-like structure with one one-to-many relationship between two different types of data, for example, one department can have many courses, many professors and of-course many students.



Network Model

- This is an extension of the Hierarchical model. In this model data is organized more like a graph, and are allowed to have more than one parent node.
- In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast.
- This database model was used to map many-to-many data relationships.
- This was the most widely used database model, before Relational Model was introduced.



Relational Model

- The relational model uses a collection of tables to represent both data and the relationships among those data.
- Each table has multiple columns, and each column has a unique name.
- In this model, data is organised in two-dimensional **tables** and the relationship is maintained by storing a common field.
- This model was introduced by E.F Codd in 1970, and since then it has been the most widely used database model, infact, we can say the only database model used around the world.
- The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table.
- Hence, tables are also known as **relations** in relational model.

- presents a sample relational database comprising three tables:
- One shows details of bank customers, the second shows accounts, and the third shows which accounts belong to which customers.
- Each table contains records of a particular type.
- Each record type defines a fixed number of fields, or attributes.
- The columns of the table correspond to the attributes of the record type
- a special character (such as a comma) may be used to delimit the different attributes of a record, and another special character (such as a newline character) may be used to delimit records.
- The relational model hides such low-level implementation details from database developers and users.

Q.4 What is E-R model. Wirte various keyword used in E-R diagram. With example

- An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram).

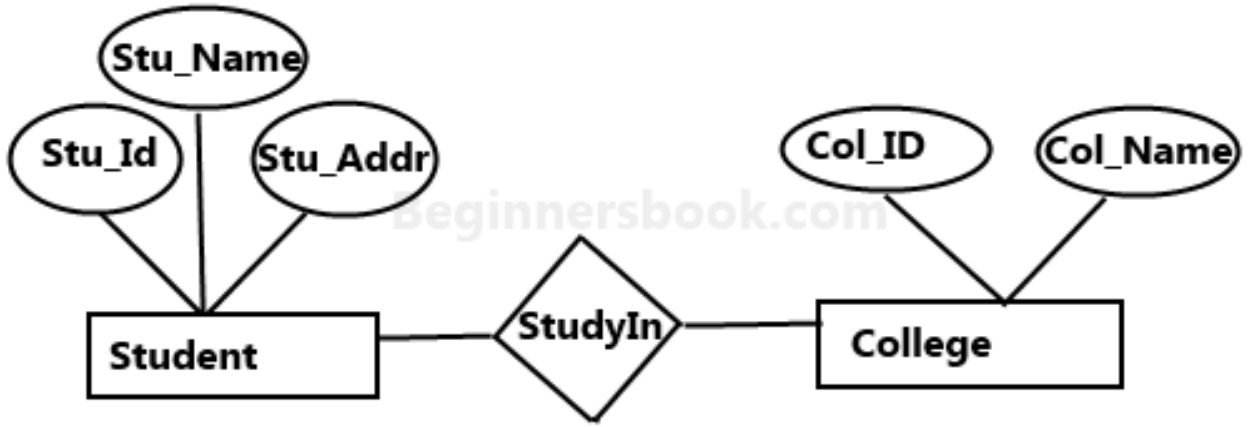
- An ER model is a design or blueprint of a database that can later be implemented as a database.

- The main components of E-R model are: entity set and relationship set.
- An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes.

- In terms of DBMS, an entity is a table or attribute of a table in database.

- So by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

Sample E-R Diagram

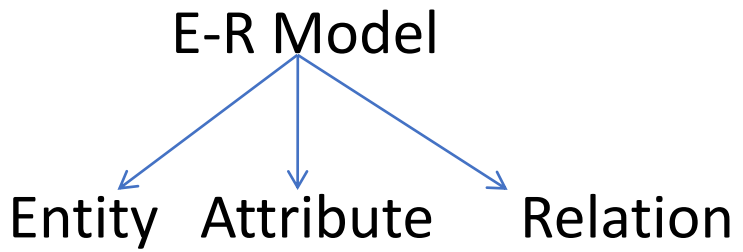


Sample E-R Diagram

- In the following diagram we have two entities Student and College and their relationship.
- The relationship between Student and College is many to one as a college can have many students however a student cannot study in multiple colleges at the same time.
- Student entity has attributes such as Stu_Id, Stu_Name & Stu_Addr and College entity has attributes such as Col_ID & Col_Name.

Components of E-R Diagram

- **Rectangle**: Represents Entity sets.
  Ellipses: Attributes
  Diamonds: Relationship Set
  Lines: They link attributes to Entity Sets and Entity sets to Relationship Set
  Double Ellipses: Multivalued Attributes
  Dashed Ellipses: Derived Attributes
  Double Rectangles: Weak Entity Sets
  Double Lines: Total participation of an entity in a relationship set.



Q.5 Define attribute with their various type.

Attribute : An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

- Key attribute
- Composite attribute
- Multivalued attribute
- Derived attribute

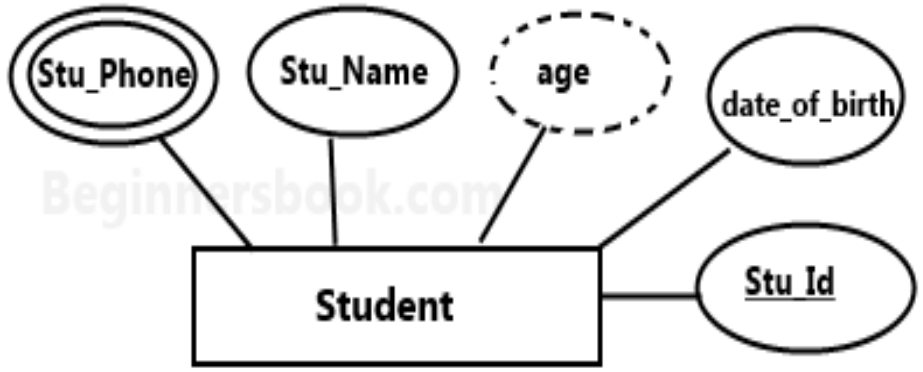
- **Key Attribute :** A key attribute can uniquely identify an entity from an entity set. For example, student roll number can uniquely identify a student from a set of students. Key attribute is represented by oval same as other attributes however **the text of key attribute is underlined.**

- **Composite Attribute :** An attribute that is a combination of other attributes is known as composite attribute. For example, In student entity, the student address is a composite attribute as an address is composed of other attributes such as pin code, state, country.

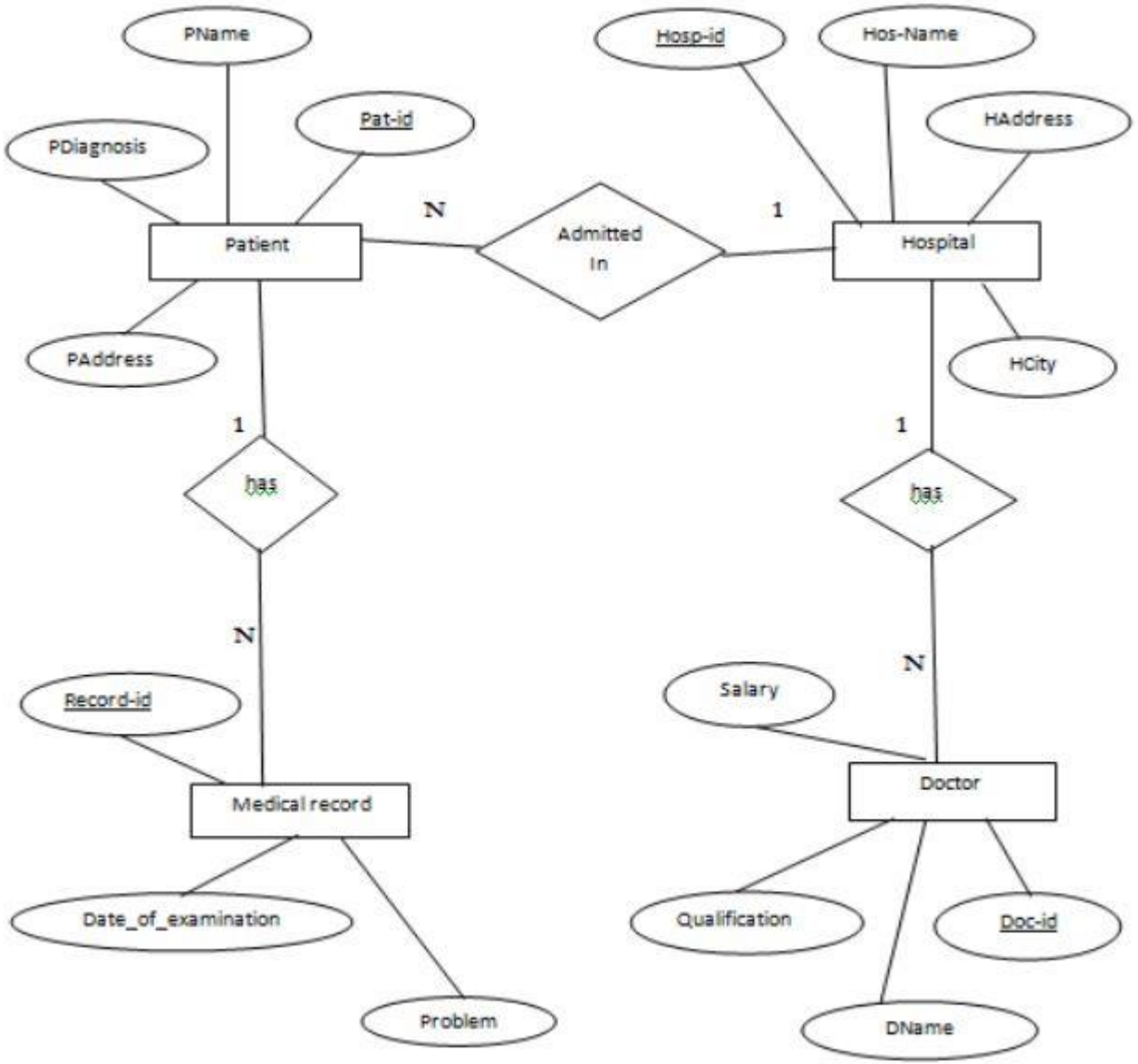
Multivalued Attribute : An attribute that can hold multiple values is known as multivalued attribute. It is represented with **double ovals** in an ER Diagram. For example – A person can have more than one phone numbers so the phone number attribute is multivalued.

Derived Attribute : A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by **dashed oval** in an ER Diagram. For example – Person age is a derived attribute as it changes over time and can be derived from another attribute (Date of birth).

Example : Student Entity with various attributes



Q.6 Make an E-R diagram of Hospital Management System.



Q.7 Define Database language with their type.

Database Languages

- A database system provides a **data definition language** to specify the database schema and a **data manipulation language** to express database queries and updates.
- The data definition and data manipulation languages are not two separate languages.
- instead they simply form parts of a single database language, such as the widely used SQL language

Data-Definition Language

- Specify a database schema by a set of definitions expressed by a special language called a **data-definition language** (**DDL**).
- the following statement in the SQL language defines the account table:
- Create table account(account-number char(10),balance integer).
- DDL statement creates the *account* table.
- it updates a special set of tables called the **data dictionary** or **data directory.**

DBMS Languages

Data Definition Language-DDL

- Data Definition Language (DDL) statements are used to define the database structure or schema.

Some examples:

- CREATE - to create objects in the database
- ALTER - alters the structure of the database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
- COMMENT - add comments to the data dictionary
- RENAME - rename an object

Database Languages

- **Data Manipulation Language (DML)**
  - **Provides basic data manipulation operations on data held in the database.**
  - DML is a language for retrieving and updating (insert, delete, & modify) the data in the DB.
  - **Types of DML:**
    - **Procedural Language (3GL):** user specifies *what* data is required and *how* to get those data(allows user to tell system exactly how to manipulate data.) Ex:Java
    - **Nonprocedural Language(4GL):** user specifies *what* data is required without specifying *how* to get those data(allows user to state what data is needed rather than how it is to be retrieved.) Ex:SQL

Data-Manipulation Language

- **Data manipulation** is
- The retrieval of information stored in the database
- The insertion of new information into the database
- The deletion of information from the database
- The modification of information stored in the database
- A **data-manipulation language (DML)** is a language that enables users to access or manipulate data as organized by the appropriate data model.
- There are basically two types:

- **Procedural DMLs** require a user to specify *what* data are needed and *how* to get those data.
- **Declarative DMLs** (also referred to as **nonprocedural** DMLs) require a user to specify *what* data are needed *without* specifying how to get those data.

- Declarative DMLs are usually easier to learn and use than are procedural DMLs.
- since a user does not have to specify how to get the data, the database system has to figure out an efficient means of accessing data.
- The DML component of the SQL language is nonprocedural