AN INTRODUCTION TO DBMS (CHAPTER -7)

1) What is Database ?

- A database is a self-describing, shared collection of interrelated data from where users can efficiently retrieve information in response to specific queries.

2) What is DBMS ?

- Full form is Database Management System. It is used to managed, processed & sorted data interrelated data in a meaningful format.

3) What is Metadata ?

- The data related to the description of the database is called Metadata or Data Dictionary or Data Directory. Metadata keeps changing dynamically as & when files are modified, added or deleted.

4) What is database scheme ?

- Data dictionary stores information related to the database structure is called database scheme.

5) What are the components that made metadata ?

- i) Name of the tables that make up the database.
  ii) Name of the diff. columns in a table.
  iii) The properties of each column in a table, like the data-type, the length of each data-type etc.
  iv) Information related to various keys of the database like primary keys, foreign keys etc.
  v) Information related to various constraints or rules that the data should follow.
  vi) Information regarding the various indices used like an index at the end of a book.

6) What is System Table ?

- Metadata is usually stored in the form of tables called system tables.

7) What is interrelated data ?

- It implies that the data stored as diff. relations or tables are not independent but are related to each other.
  Eg. Suppose there is a table viz. **student** in which there is a column teacher_id & in **teacher** table there is a column teacher_name. So, here teacher_name & teacher_id columns are interrelated to each other though they are from diff. table.

8) What are the advantages of DBMS ?

- **Minimised data duplication –** In DBMS, a particular data is stored centrally in one place only & shared by all applications. Whenever a user runs a query through an application program, DBMS retrieves the data. Moreover, when an update is required data needs to be updated at one place only. This eliminates the problem of data integrity & data inconsistency.
- **Data Independence –** In a database system, the data is stored independent of any application program. Application programs interact with the DBMS software, which in turn interacts with the data in the database. This makes application programs independent of any internal changes.
- **Better Sucurity –** The central control of a DBMS makes it easier to control physical access to terminals, storage devices & specific interface forms for updating or deletion of records.
- **User Friendly Interfaces –** Database technology makes it easier to represent data in a user friendly manner. The complexity of data & implementation details are hideen by the DBMS & end user can access the required data with minimal of technical knowledge.
- **Concurrency Problem –** In case a DBMS is not designed for multiple users, problems can arise when more than one user wants to access a particular record simultaneously. This is known as concurrency problem. Eg. – When 2 joint account holders want to access their account simultaneously for transaction.

9) What is data abstraction ?

- The general user of the database may not be a computer expert & hence should not be concerned about the internal storage and working of the database. He should concerned only with the manipulation of the data like data entry, data modification or data query.
  To overcome the problem, the database approach allows user to handle the data without being concerned about the underlying mechanism by which the data is actually stored or accessed. It presents user a view of the data that the user can readily use & understand without going into the internal complexities.

10) Write short notes on 3 schema architecture of a Database ?

- The overall design & description of a database is called database schema. Database scheme are of 3 parts –
  i) **Physical/Internal Schema -** This is the lowest level of a data abstraction. In this layer schema describes in details **how the raw data is actually stored at the byte level.** It also describes the access paths for the database. The database system hides many of these lowest level storage & access details from the database programmers & end users.
- ii) **Logical/Conceptual Schema –** The next higher level of schema, here, the database administrator describe **what data are stored in the database & the relationship between those data.** The entire database is described in terms of relatively simple, easy to understand structures like data tables, keys, indices etc.
- **View/External Schema -** This is the highest level of data abstraction. This schema deals with the way a particular user application program **views the data from the database.** Majority of the users will not be concerned about the whole database but will be concerned only a part of the data in the database.

11) What is Data Independence ?

- An important advantage of 3-schema architecture is that any modification made at lower level schema, does not affect the structure at higher level schema. This is called data independence.
  There are 2 types of data independence –
  i) **Physical data independence –** It is the ability to change the internal or storage schema of a database without having to change the logical schema. Thus changes made in this level, does not require any change in the logical level, which in turn does not require any change in the view level where application programs interacts with the DBMS.
  ii) **Logical data independence –** It is the ability to change the logical or conceptual schema of a database without having to change the view schema of the database. Applications programs that interact at the external

level do not get affected by changes made at the logical level. However, application program depend mainly on conceptual model of the database it is difficult to achieve logical data independence.

12) What is data mapping ?

- Translating request at one level to a request at another level is called mapping. Mapping helps in maintaining data independence , whereby if the schema changed at one level, the schema remains unchanged at a higher level. Only the mapping between 2 levels is changed.

13) Write short notes on various types of Database languages ?

- i) **DDL(Data Definition Language) –** The DDL determines the way the data is logically arranged & the way data is physically stored.
  DDL consists of – a) **Create** b) **Alter** c) **Drop** – query
  Create – is used to create a new table.
  Alter – is used to change the structure of the table.
  Drop – is used to delete the table.
- ii) **DML(Data Manipulation Language) –** The database can be accessed, filled & manipulated by the end user using DML.
  DML consists of – a) **Insert** b) **Select** c) **Delete** d) **Update –** query.
  Insert – add new information or add new row within the table.
  Select – retrieve or fetch the information or data within the table.
  delete – delete the information or row within the table.
  update – update the information or row/columns value of previously created rows.
- iii) **DCL(Data Control Language) –** These are used to control acces privileges to users. If defines when a proposed change to a database can be made irreversibly. Only admisistrator can execute DCL commands.
  DCL consists of – a) **Rollback** b) **CALL** c) **Return**
  Rollback –undo the previous work or undo previous mistake.
  Call – to execute an SQL function.
  Return – to return value from a function.

14) What is data models ?

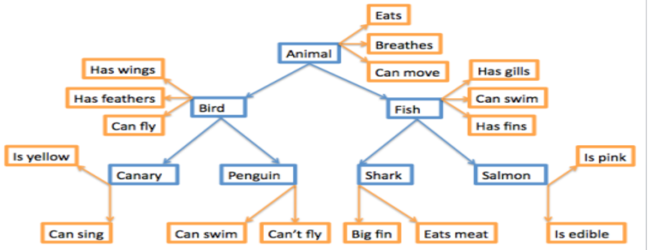
- Data models are a collection of conceptual tools for describing the data, the relationships between the data, the constraints applicable on the data etc.
  Data models are 3 types –
  - **Physical Models.**
  - **Record Based Logical Models.**
  - **Object Based Logical Models.**

15) Write briefly on Physical Models ?

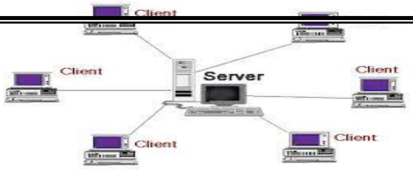
- These data models are used to describe at the lowest level of data abstraction i.e. the way data is physically stored in the database. 2 popular data models are there –
  - **Unifying Model**
  - **Frame Memory Model**

16) Write all types of Record Based Logical Models ?

- These data models are used to describe data at the logical & view levels. It uses concepts that may be understood by the end users. There are 3 types of record based logical models –
  - **Hierarchical Model –** This is the oldest model. Here records are logically organized into a hierarchy of relationships forming an inverted tree pattern.
    All records in a hierarchy are called nodes with each node related to the next in a parent-child relationship.



  - **Advantage –** i) This model allows easy addition & deletion of information.
    ii) Data at the top of the hierarchy is very fast to access.
    iii) It is good for entities that have a one-to-many relationship between them.
  - **Disadvantage –** i) It requires data to be repetitively stored in many diff. entities.
    ii) The database can be very slow when searching for information on the lower entities.
    iii) Many-to-many relationships are not supported & can have only one-to-many relationship.
    iv) Searching for data requires the DBMS to run through the entire model from top to bottom until the required information is found.
  - **Network Model –** This model is used to store data similar to the hierarchical model's parent-child relationship. A hierarchical model is a subset of network model. This model allows a record to be a child of more than one parent records.
  - **Advantage –** i) The model retains almost all the advantages of the hierarchical model.
    ii) This model is conceptually simple & easy to design.
    iii) The model supports both one-to-many & many-to-many relationships.
    iv) Data access is easier than the hierarchical model.



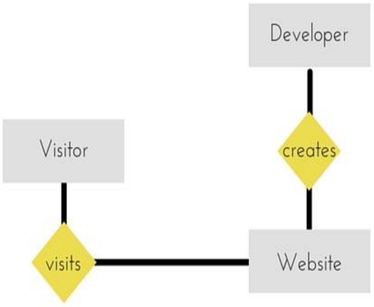
- **Disadvantage –** i) The model is difficult to implement & maintains.
  ii) The whole database structure is complex one as it involves a lot of link.
- **Relational Model –** This is the most popular database model in use today. In this model data & the relationship between them is represented as a collection of tables. The tables are joined by relationships using various keys. Each table has multiple column & rows with each column having a unique name that represents an attribute or property of the object represented by table.
- **Advantage –** i) The representation of information in the form of tables consisting of rows & columns.
  ii) Diff. tables can be easily combined or manipulated to extract the required information.
  iii) The use of relational algebra & relational calculus to manipulate the relations between tables, removes any ambiguity which usually may arise in the network model.
- **Disadvantage –** i) If the no. of tables is large then querying the database may take some time.
  ii) Join operations to combine data from diff. tables may require large data storage.

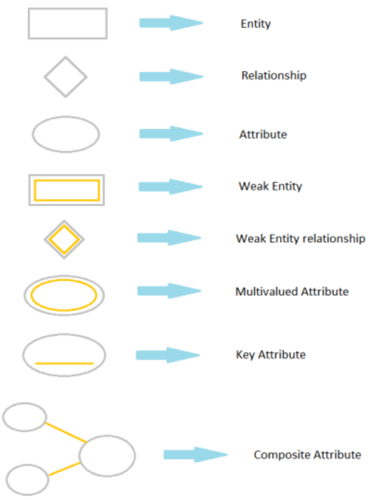
17) Write briefly on types of object based logical models ?

- These data models are used in describing data at the ogical & view levels. These models are closer to human perception & farther from system perception.

  Object based Logical Models are of 2 types –
  - **Object Oriented Model.**
  - **Entity Relationship Model.**
  - **Entity Relationship(ER) Model –**

    ER-Diagram is a visual representation of data that describes how data is related to each other.





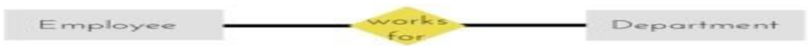
Symbols and Notations

Components of E-R Diagram

The E-R diagram has three main components.

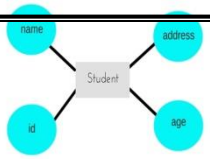
1) Entity

An **Entity** can be any object, place, person or class. In E-R Diagram, an **entity** is represented using rectangles. Consider an example of an Organisation. Employee, Manager, Department, Product and many more can be taken as entities from an Organisation.



2) Attribute

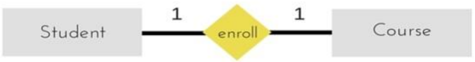
An **Attribute** describes a property or characterstic of an entity. For example, Name, Age, Address etc can be attributes of a Student. An attribute is represented using eclipse.



Binary Relationship

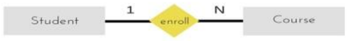
Binary Relationship means relation between two Entities. This is further divided into three types.

1. **One to One :** This type of relationship is rarely seen in real world.



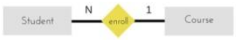
The above example describes that one student can enroll only for one course and a course will also have only one Student. This is not what you will usually see in relationship.

2. **One to Many :** It reflects business rule that one entity is associated with many number of same entity. The example for this relation might sound a little weird, but this menas that one student can enroll to many courses, but one course will have one Student.

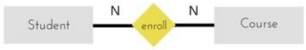


The arrows in the diagram describes that one student can enroll for only one course.

3. **Many to One :** It reflects business rule that many entities can be associated with just one entity. For example, Student enrolls for only one Course but a Course can have many Students.



4. **Many to Many :** The below diagram represents that many students can enrol for more than one courses.


Student
N
enroll
N
Course