D.N.R COLLEGE (AUTONOMOUS), BHIMAVARAM

DEPARTMENT OF COMMERCE

II B.COM (CA) – IV SEMESTER

DATABASE MANAGEMENT SYSTEM

R. RADHA RANI

LECTURER IN COMMERCE

I UNIT

1.a)What is Database management system? Explain its objectives.

Database: A data base is an organized collection of information.
A database is a collection of data or information that is organized for rapid search and retrieval by a computer, also known as an electronic database.

Database Management System: DBMS is a collection of programs that enable users to create and maintain a database. The DBMS is hence a general purpose software system that facilitates the processes of defining, constructing and manipulating databases for various applications.

Objectives of DBMS:

Mass Storage

DBMS can store a lot of data in it. So for all the big firms, DBMS is really ideal technology to use. It can store thousands of records in it and one can fetch all that data whenever it is needed.

Removes Duplicity

If you have lots of data then data duplicity will occur for sure at any instance. DBMS guarantee it that there will be no data duplicity among all the records. While storing new records, DBMS makes sure that same data was not inserted before.

Multiple Users Access

No one handles the whole database alone. There are lots of users who are able to access database. So this situation may happen that two or more users are accessing database. They can change whatever they want, at that time DBMS makes it sure that they can work concurrently.

Data Protection

Information such as bank details, employee’s salary details and sale purchase details should always be kept secured. Also all the companies need their data secured from unauthorized use. DBMS gives a master level security to their data. No one can alter or modify the information without the privilege of using that data.

Data Back up and recovery

Sometimes database failure occurs so there is no option like one can say that all the data has been lost. There should be a backup of database so that on database failure it can be recovered. DBMS has the ability to backup and recover all the data in database.

Query Language:

DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and different filtering options, as he or she wants. Traditionally it was not possible where file-processing system was used.

Everyone can work on DBMS

There is no need to be a master of programming language if you want to work on DBMS. Any accountant who is having less technical knowledge can work on DBMS. All the definitions and descriptions are given in it so that even a non-technical background person can work on it.

Integrity

Integrity means your data is authentic and consistent. DBMS has various validity checks that make your data completely accurate and consistence.

Platform Independent

One can run DBMS at any platform. No particular platform is required to work on database management system.

Sharing of Data
To share the data to its users under the controlled environment is the objective of DBMS.
Users can share the data to authorized persons with utmost security.
Resolvability: The ability of the DBMS to change in response to growing user needs and advancing technology.

b) Explain the various Services of Database Management System.

Database Management System: DBMS is a collection of programs that enable users to create and maintain a database. The DBMS is hence a general purpose software system that facilitates the processes of defining, constructing and manipulating databases for various applications.
Database management system (DBMS) consists of a collection of interrelated data and a set of programs to access that data.
Services of Database Management System:

Data Storage Management:
It provides a mechanism for management of permanent storage of the data. The internal schema defines how the data should be stored by the storage management mechanism and the storage manager interfaces with the operating system to access the physical storage.
Data Manipulation Management:
A DBMS furnishes users with the ability to retrieve, update and delete existing data in the database.
Data Definition Services:
The DBMS accepts the data definitions such as external schema, the conceptual schema, the internal schema, and all the associated mappings in source form.
Data Dictionary/System Catalog Management:
The DBMS provides a data dictionary or system catalog function in which descriptions of data items are stored and which is accessible to users.
Database Communication Interfaces:
The end-user’s requests for database access are transmitted to DBMS in the form of communication messages.
Authorization / Security Management:
The DBMS protects the database against unauthorized access, either international or accidental. It furnishes mechanism to ensure that only authorized users an access the database.
Backup and Recovery Management:
The DBMS provides mechanisms for backing up data periodically and recovering from different types of failures. This prevents the loss of data,
Concurrency Control Service:
Since DBMSs support sharing of data among multiple users, they must provide a mechanism for managing concurrent access to the database. DBMSs ensure that the database kept in consistent state and that integrity of the data is preserved.
Transaction Management:
A transaction is a series of database operations, carried out by a single user or application program, which accesses or changes the contents of the database.
Database Access and Application Programming Interfaces:

All DBMS provide interface to enable applications to use DBMS services. They provide data access via Structured Query Language (SQL). The DBMS query language contains two components: (a) a Data Definition Language (DDL) and (b) a Data Manipulation Language (DML).

c)What are the Characteristics of Database Management System?

Database Management System: DBMS is a collection of programs that enable users to create and maintain a database. The DBMS is hence a general purpose software system that facilitates the processes of defining, constructing and manipulating databases for various applications.

Modern DBMS has the following characteristics:

Real-world entity:

Modern DBMS are more realistic and uses real world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use student as entity and their age as their attribute

Self Explaining Nature

A DBMS contains one database and along with that it also contains metadata about that one database. Metadata is the data about data.

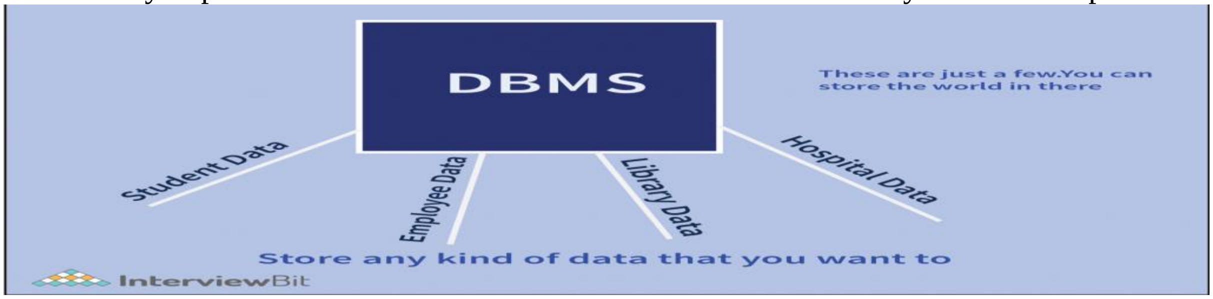
Concurrent Access without Anomalies

Multiple users can access the database at the same time without any anomalies

Stores Any Kind of Structured Data

A database can store any kind of structured data.

DBMS can store practically any data that exists in the world and is structured and this is yet another very important characteristic because we need to work on every kind of data present.


DBMS
These are just a few.You can store the world in there
Student Data
Employee Data
Library Data
Hospital Data
Store any kind of data that you want to
InterviewBit

Relation-based tables:

DBMS allows entities and relations among them to form as tables. This eases the concept of data saving. A user can understand the architecture of database just by looking at table names etc.

Isolation of data and application:

A database system is entirely different than its data. Where database is said to active entity, data is said to be passive one on which the database works and organizes. DBMS also stores metadata which is data about data, to ease its own process.

Less redundancy: DBMS follows rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Following normalization, which itself is a mathematically rich and scientific process, make the entire database to contain as less redundancy as possible.

Consistency:

DBMS always enjoy the state on consistency where the previous form of data storing applications like file processing does not guarantee this. Consistency is a state where every relation in database remains consistent.

Query Language:

DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and different filtering options, as he or she wants. Traditionally it was not possible where file-processing system was used.

ACID Properties:

DBMS follows the concepts for ACID properties, which stands for Atomicity, Consistency, Isolation and Durability. These concepts are applied on transactions, which manipulate data in database.

Ease of Access

Before DBMS, the conventional file system (i.e. files and folders) was used to store complex and large data. Searching for a Name in thousands of Students was a very difficult task. DBMS provides ease of access to the data inside the database.

Security

Unauthorized users should not be allowed to access the database.

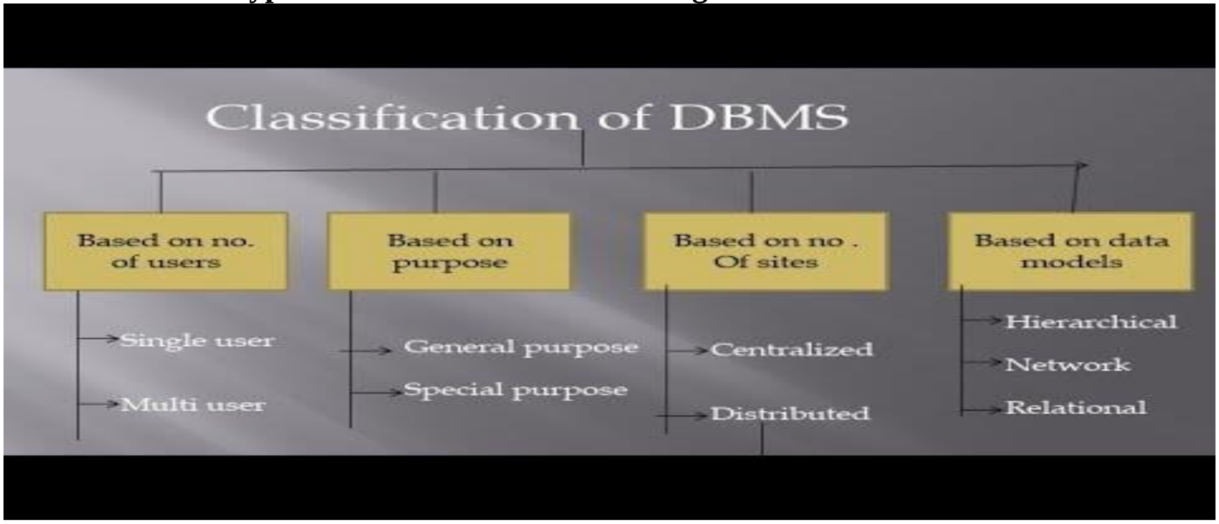
Authentication: The DBMS has authentication for various users that directly refers to the limit to which the user can access the Database.

Users: DBMS is used by various users for various purposes. Some may involve in retrieving data and some may involve in backing it up. Some of them are administrators, designers and end users.

a)Write the classification of Database Management System.

Database Management System: DBMS is a collection of programs that enable users to create and maintain a database. The DBMS is hence a general purpose software system that facilitates the processes of defining, constructing and manipulating databases for various applications.

There are various types of databases used for storing different varieties of data:



Classification of DBMS

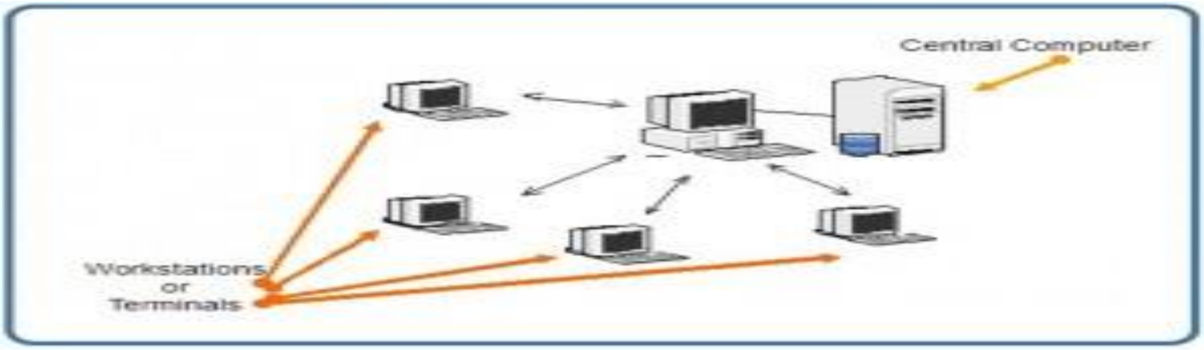
Based on no. of users
- Single user
- Multi user

Based on purpose
- General purpose
- Special purpose

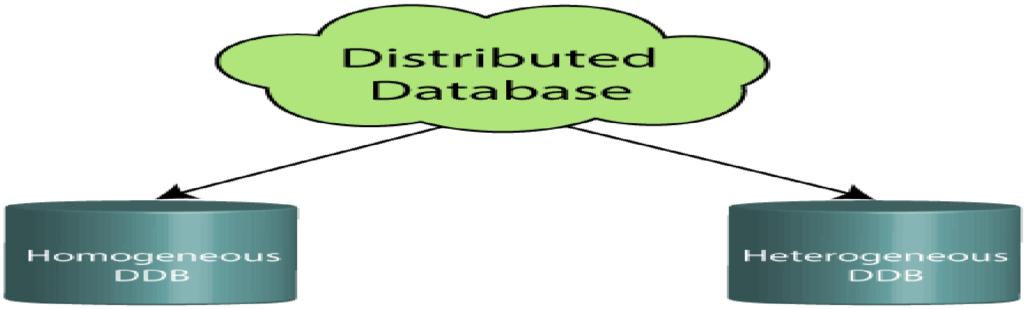
Based on no . Of sites
- Centralized
- Distributed

Based on data models
- Hierarchical
- Network
- Relational

1) **Centralized Database:** It is the type of database that stores data at a centralized database system. It comforts the users to access the stored data from different locations through several applications. An example of a Centralized database can be Central Library that carries a central database of each library in a college/university.



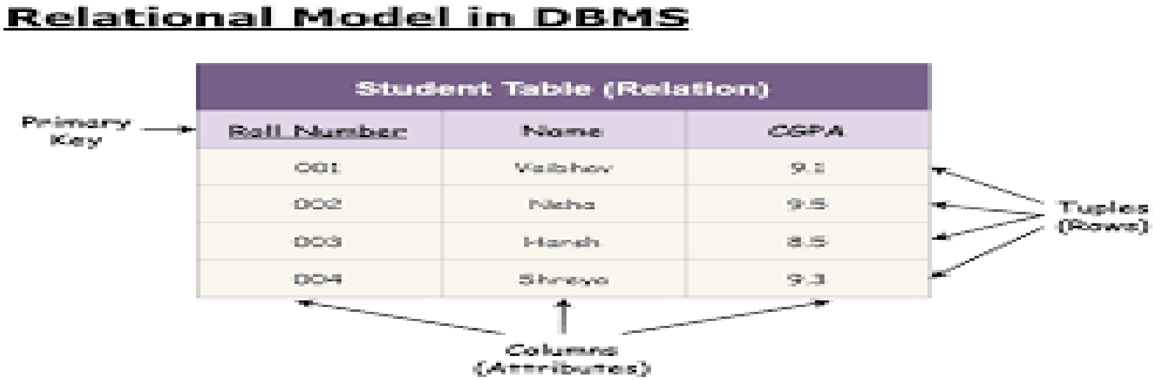
2) Distributed Database: Unlike a centralized database system, in distributed systems, data is distributed among different database systems of an organization. These database systems are connected via communication links. Such links help the end-users to access the data easily. We can further divide a distributed database system into:



Homogeneous DDB: Those database systems which execute on the same operating system and use the same application process and carry the same hardware devices.

Heterogeneous DDB: Those database systems which execute on different operating systems under different application procedures, and carries different hardware devices.

3) Relational Database: This database is based on the relational data model, which stores data in the form of rows(tuple) and columns(attributes), and together forms a table(relation). A relational database uses SQL for storing, manipulating, as well as maintaining the data.



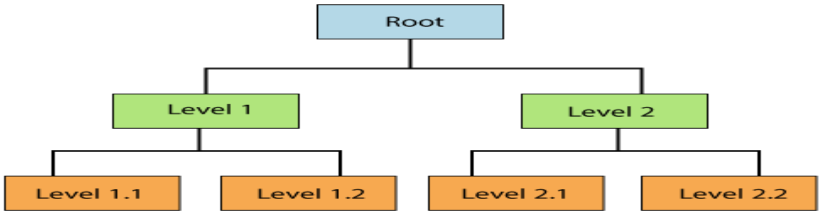
Student Table (Relation)		
Roll Number	Name	CGPA
001	Vaibhav	9.1
002	Neha	9.5
003	Harsh	8.5
004	Shreya	9.3

4) No SQL Database: Non-SQL/Not Only SQL is a type of database that is used for storing a wide range of data sets. It is not a relational database as it stores data not only in tabular form but in several different ways.

5) Cloud Database: A type of database where data is stored in a virtual environment and executes over the cloud computing platform. It provides users with various cloud computing services for accessing the database.

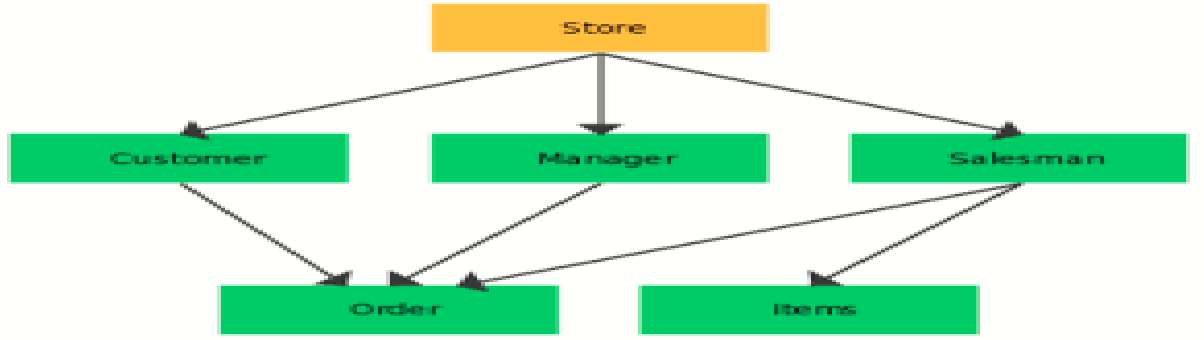
6) Object-oriented Databases: The type of database that uses the object-based data model approach for storing data in the database system. The data is represented and stored as objects which are similar to the objects used in the object-oriented programming language.

7) Hierarchical Databases: It is the type of database that stores data in the form of parent-children relationship nodes. Here, it organizes data in a tree-like structure.



Hierarchical Database

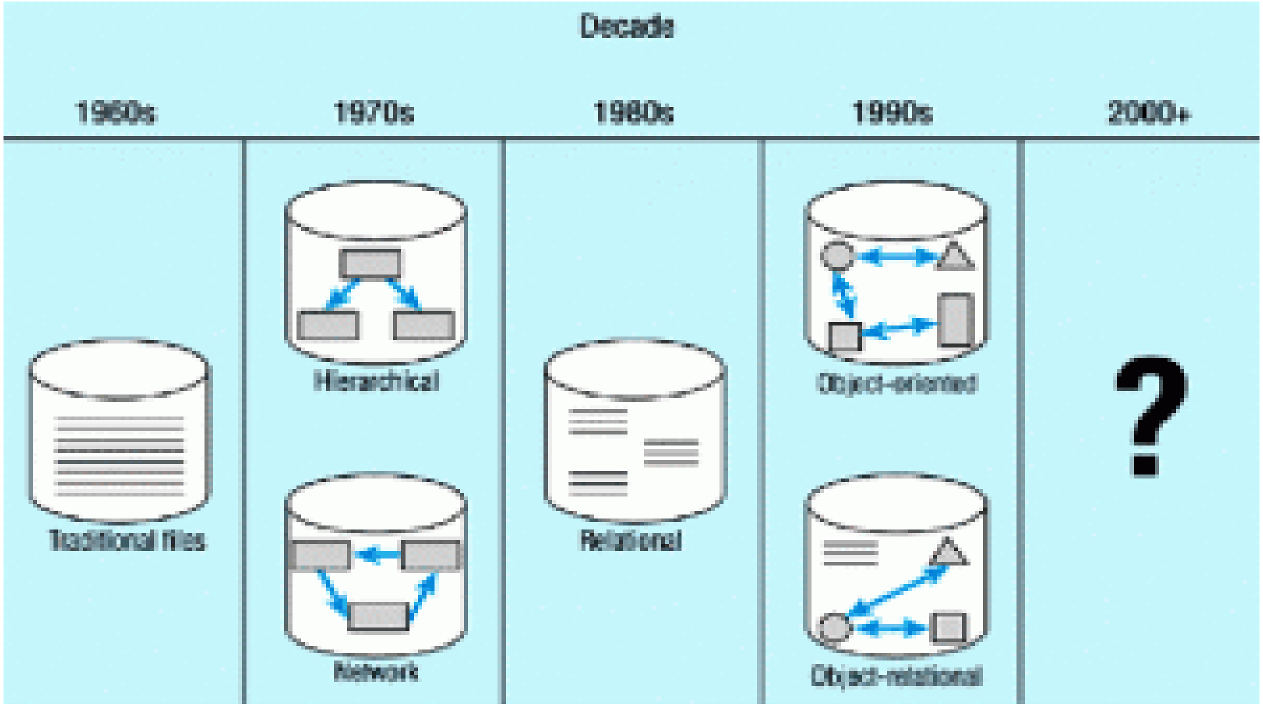
8) Network Databases: It is the database that typically follows the network data model. Here, the representation of data is in the form of nodes connected via links between them.



9) Personal Database: Collecting and storing data on the user's system defines a Personal Database. This database is basically designed for a single user.
10) Enterprise Database: Large organizations or enterprises use this database for managing a massive amount of data. It helps organizations to increase and improve their efficiency. Such a database allows simultaneous access to users.
11) General purpose software - software that can be used for multiple purposes.
12) Special purpose software - software built for a specific purpose.

b)Discuss about evolution of Database Management System.
The evolution of Database Management system

The main objective of the database is to ensure that data can be stored and retrieved easily and effectively. It is a compilation of data (records) in a structured way. In a database, the information is stored in a tabular form where data may or may not inter linked. Hence we can say that basically database is a compilation of database files and each database file is further a collection of records.



The chronological order of the development of DBMS is as follows
1. Flat Files (1970s-1990s)
2. Hierarchical (1970s-1990s)
3. Network (1970s- 1990s)
4. Relational (1980s-present)
5. Object-Oriented (1990s- present)
6. Object-Relational (1990s-present)

1.Flat Files (1970s-1990s) : Flat files database is a database that stores information in a single file or table. In text file, every line contains one record where fields either have fixed length or they are separated by commas, whitespaces, tabs or any records and they cannot contain multiple tables as well.

2. Hierarchical (1970s-1990s) : As the name indicates, hierarchical database contains data in a hierarchically-arranged data. More Perceptively it can parent can have many children but one child can only have one parent i. e.,; one-to-many relationship.
3. Network database (1970s -1990s) : The inventor of network model is Charles Bachmann. Unlike hierarchical database model, network database allows multiple parent and child relationships i. e., it maintains many-to many relationship. Network database is basically a graph structure. To represent complex data relationship more effectively. **.** To improve the performance of the database. **.** To improve a database standard.
4. Relational database (1980s-present) : Relationship database model was proposed by E. F. Codd. A relational database is a type of database that stores and provides access to data points that are related to one another. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables. It allows the entities to be related through a common attribute.
5. Object – oriented database (1990s -present) : Object-Oriented database management system is that database system in which the data or information is presented in the form of objects, much like in object-Oriented programming language.
6. Object- relationship database (1990s-present) : Defined in simple teams, an object relationship database management system displays a modified object- oriented user- display over the already implemented relationship database management system.

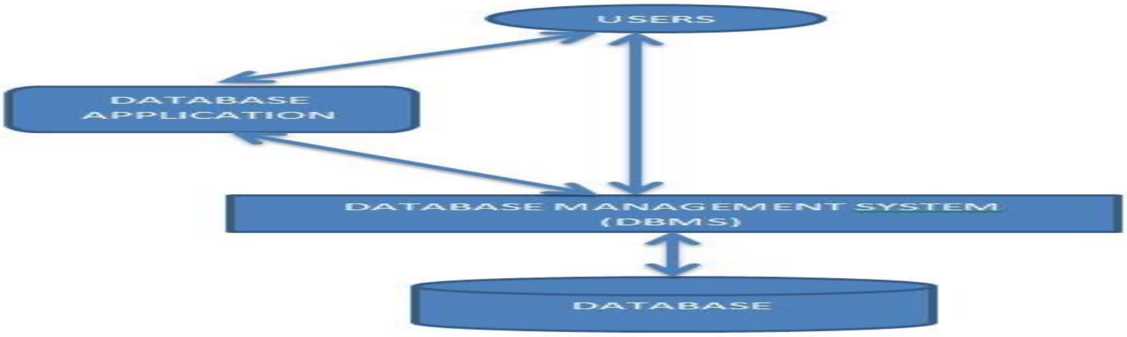
c)Discuss the Overview of Database Management System.

Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

Overview of DBMS:

A **database management system** stores data in such a way that it becomes easier to retrieve, manipulate, and produce information. Traditionally, data was organized in file formats. DBMS was a new concept then, and all the research was done to make it overcome the deficiencies in traditional style of data management. The following are the main components of DBMS.



Real-world entity:

Modern DBMS are more realistic and uses real world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use student as entity and their age as their attribute

Self Explaining Nature

A DBMS contains one database and along with that it also contains metadata about that one database. Metadata is the data about data.

Concurrent Access without Anomalies

Multiple users can access the database at the same time without any anomalies

Stores Any Kind of Structured Data

A database can store any kind of structured data.

DBMS can store practically any data that exists in the world and is structured and this is yet another very important characteristic because we need to work on every kind of data present

Relation-based tables:

DBMS allows entities and relations among them to form as tables. This eases the concept of data saving. A user can understand the architecture of database just by looking at table names etc.

Isolation of data and application:

A database system is entirely different than its data. Where database is said to active entity, data is said to be passive one on which the database works and organizes. DBMS also stores metadata which is data about data, to ease its own process.

Less redundancy: DBMS follows rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Following normalization, which itself is a mathematically rich and scientific process, make the entire database to contain as less redundancy as possible.

Consistency:

DBMS always enjoy the state on consistency where the previous form of data storing applications like file processing does not guarantee this. Consistency is a state where every relation in database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state.

Query Language:

DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and different filtering options, as he or she wants. Traditionally it was not possible where file-processing system was used.

Ease of Access

Before DBMS, the conventional file system (i.e. files and folders) was used to store complex and large data. However, DBMS provides ease of access to the data inside the database.

Security

Unauthorized users should not be allowed to access the database.

Authentication: The DBMS has authentication for various users that directly refers to the limit to which the user can access the Database.

Users: DBMS is used by various users for various purposes. Some may involve in retrieving data and some may involve in backing it up. Some of them are administrators, designers and end users.

5 Marks

6)a)What is the need of Database Management Systems.

A **D**ata **B**ase **M**anagement **S**ystem is a system software for easy, efficient and reliable data processing and management. It can be used for:

- Creation of a database.
- Retrieval of information from the database.
- Updating the database.
- Managing a database.

It provides us with the many functionalities and is more advantageous than the traditional file system in many ways listed below:

1) Processing Queries and Object Management:

In traditional file systems, we cannot store data in the form of objects. We can directly store data in the form of objects in a database management system. A DBMS gives us the ability to query the database.

2) Controlling redundancy and inconsistency:

Redundancy refers to repeated instances of the same data. A database system provides redundancy control whereas in a file system, same data may be stored multiple times. A DBMS uses **data normalization** to avoid redundancy and duplicates.

3) Efficient memory management and indexing:

DBMS makes complex memory management easy to handle. In file systems, files are indexed in place of objects so query operations require entire file scans

4) Concurrency control and transaction management:

Several applications allow user to simultaneously access data. This may lead to inconsistency in data in case files are used.

5) Access Control and ease in accessing data:

A DBMS can grant access to various users and determine which part and how much of the data can they access from the database thus removing redundancy.

6) Integrity constraints: Data stored in databases must satisfy integrity constraints. A DBMS ensures that it is only out of one of the programs offered schema , hence, database integrity is preserved.

7)Other benefits: Apart from the above mentioned features a database management also provides the following:

- Multiple User Interface
- Data scalability, expandability and flexibility:
- Overall the time for developing an application is reduced.
- Security: Simplifies data storage as it is possible to assign security permissions allowing restricted access to data.

(b)Explain Database Management System with an example.

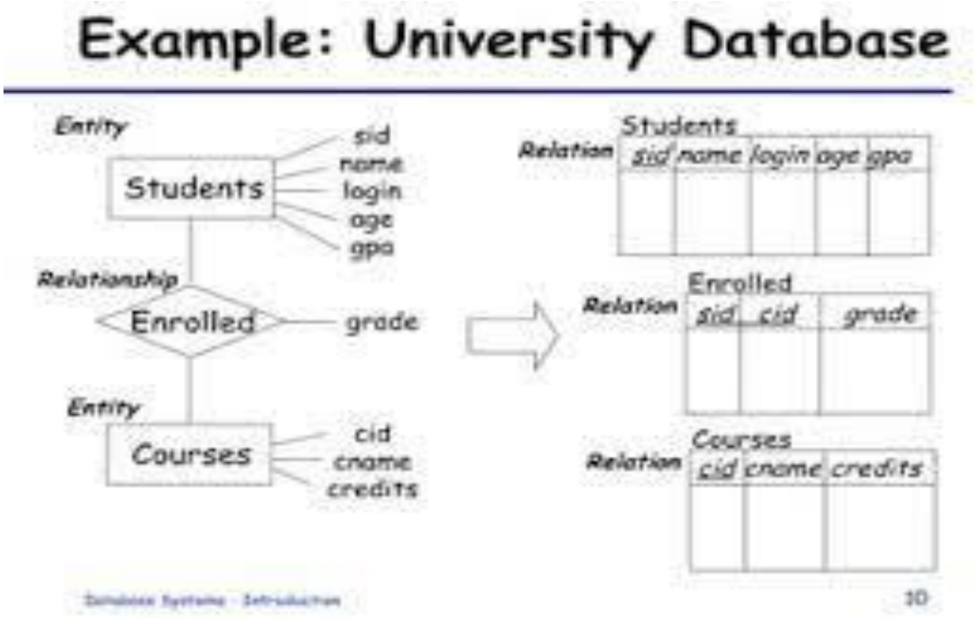
Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information.

Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

Database Management System

- Database management system is a software which is used to manage the database. For example: My SQL, Oracle, etc are a very popular commercial database which is used in different applications.
- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, retrieval of data, creating a table in the database and a lot more.
- It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

Example: The following example shows the Database Management System of a University



Here Students and Courses are the Entities and the relationship is enrollment in different courses. In DBMS database is maintained in Relations or tables, here we can observe students, enrolled, courses tables and their attributes. We can easily access the data and retrieve data of our requirement by making Queries on related Tables.

(c)What are the limitations of Database Management Systems?

Database Management Systems:

A **database management system** stores data in such a way that it becomes easier to retrieve, manipulate, and produce information.

Limitations of DBMS: Database Management System is quite useful compared to the file based management system. However, it does have some disadvantages. Some of those are as follows −

More Costly

Creating and managing a database is quite costly. High cost software and hardware is required for the database. Also highly trained staff is required to handle the database and it also needs continuous maintenance.

High Complexity

A Database Management System is quite complex as it involves creating, modifying and editing a database. Consequently, the people who handle a database or work with it need to be quite skilled or valuable data can be lost.

Database handling staff required

As discussed in the previous point, database and DBMS are quite complex. Hence, skilled personnel are required to handle the database so that it works in optimum condition. This is a costly venture as these professionals need to be very well paid.

Database Failure

All the relevant data for any company is stored in a database. So it is imperative that the database works in optimal condition and there are no failures. A database failure can be catastrophic and can lead to loss or corruption of very important data.

High Hardware Cost

A database contains vast amount of data. So a large disk storage is required to store all this data. Sometimes extra storage may even be needed. All this increases hardware costs.

Huge Size

A database contains a large amount of data, especially for bigger organisations. This data may even increase as more data is updated into the database. All of these leads to a large size of the database.

Upgradation Costs

Often new functionalities are added to the database.This leads to database upgradations. All of these upgradations cost a lot of money.

Cost of Data Conversion

If the database is changed or modified in some manner, all the data needs to be converted to the new form. This cost may even exceed the database creation and management costs sometimes.

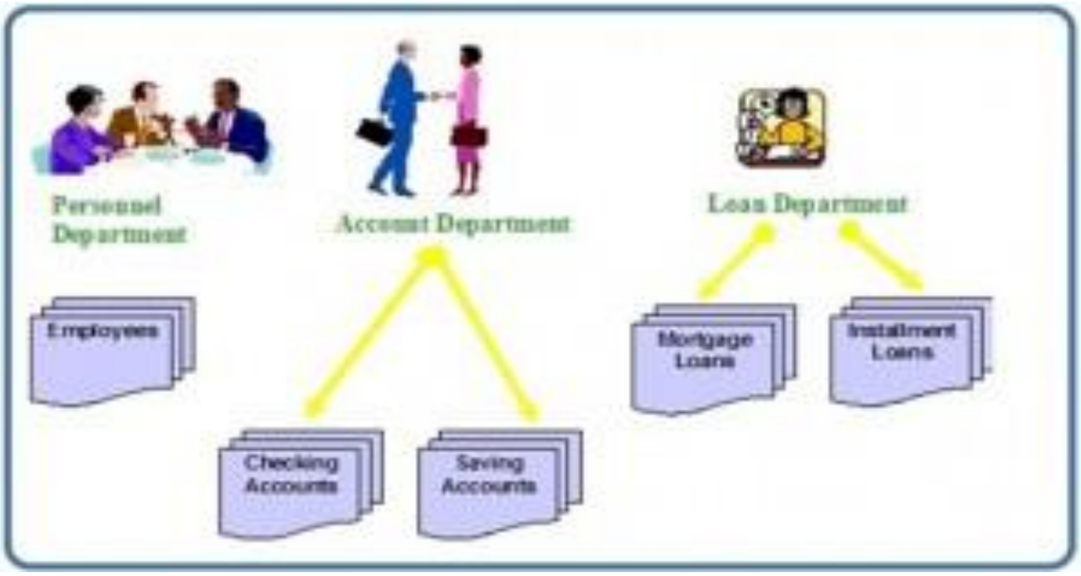
11.a)Write the types of Databases

(b) Explain any five services of DBMS.

(c) What are the characteristics that are involved in DBMS?

II UNIT

1. (a) What are the drawbacks of File based System?
   File based System:
   The systems that are used to organize and maintain data files are known as file based data systems. These file systems are used to handle a single or multiple files and are not very efficient.
   A file-based data management system (also called a file system) is a type of software that allows users to access and organize small groups of data. It is usually integrated into a computer's operating system and is responsible for storing and retrieving files from a storage medium, such as a hard disk or flash drive.



Drawbacks of File-Based Systems

Data Integrity Problems –
The data present in the database should be consistent and correct. To achieve this, the data should must satisfy certain constraints.

Slow access time –
Direct access of files is very difficult and one needs to know the entire hierarchy of folders to get to a specific file. This involves a lot of time.

Presence of redundant data –
The same data can be present in two or more files which takes up more disc space.

Inconsistent Data –
Due to data redundancy, same data stored at different places might not match to each other.

Difficulty in recovery of corrupt data –
Recovery or backup of lost and corrupt data is nearly impossible in case of File Processing System.

Lack of Atomicity –
Operations performed in the database must be atomic i.e. either the operation takes

place as a whole or does not take place at all.

Problem in Concurrent Access –

When a number of users operates on a common data in database at the same time then anomalies arise, due to lack of concurrency control.

Unanticipated Queries-

In a file-based system, handling sudden/ad-hoc queries can be difficult, since it requires changes in the existing programs.

Unauthorized Access –

Anyone who gets access to the file can read or modify the data.

Smaller size-

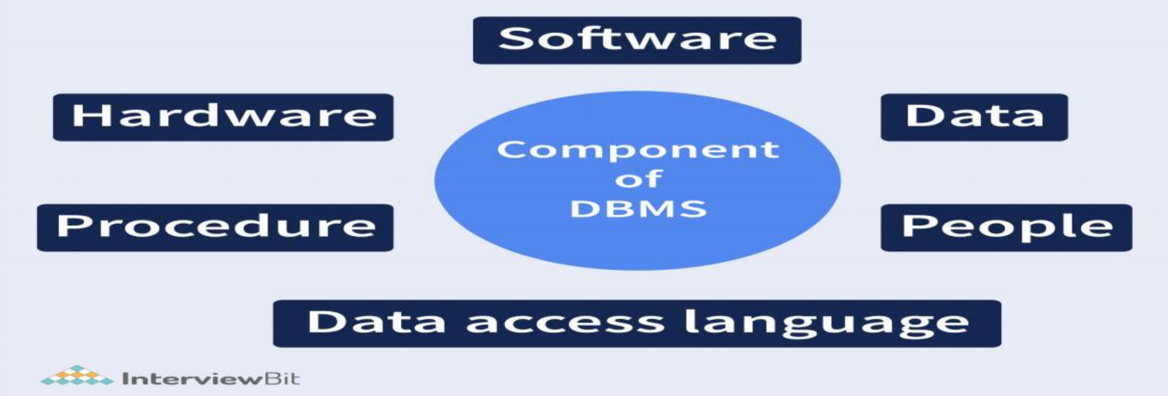
The File based system is limited to a smaller size and cannot store large amounts of data.

. **Difficult to Share-**The data files in the file based system may be stored across multiple locations. Consequently, it is difficult to share the data easily with multiple users.

b) Explain the various Components of Database Management Systems.

Components of DBMS: Components of DBMS describe what are the different parts that work together for creating, managing the database that forms a complete system, which consist of software, hardware, people, techniques of handling database, and the data also.

The DBMS can be classified into five main components which are explained as follows:



Hardware

The hardware is the actual computer system used for keeping and accessing the database. The conventional DBMS hardware consists of secondary storage devices such as hard disks. Servers and workstations are used to transfer data.

Software

Software is a collection or set of programs, **instructions that tell a computer what to do**. Software comprises the entire set of programs, procedures, and routines associated with the operation of a computer system.Some DBMS software examples include MySQL, Microsoft Access, SQL Server, FileMaker, Oracle, RDBMS, dBASE, Clipper, and FoxPro**.**

Data

It is an important component of the database management system. The main task of DBMS is to process the data. Databases are used to store the data, retrieved, and updated to and from the databases.

- **Meta data and Actual data:**It is the most important component of the database management system. The typical database contains both the **metadata**(data about data) and the **actual(operational) data.**

Procedures

Procedures refer to general instructions to use a database management system. This includes procedures to setup and install a DBMS, To login and logout of DBMS software, to manage databases, to take backups, generating reports etc.

Users

There are a number of users who can access or retrieve the data on demand using the application and the interfaces provided by the DBMS.

The users of the database can be classified into different groups −

- Native Users
- Online Users
- Sophisticated Users
- Specialized Users
- Application Users
- DBA- Database Administrator

Database Access Language

Database Access Language is a simple language that allows users to write commands to perform the desired operations on the data that is stored in the database.

- Through utilizing the language, users can create new databases, tables, insert data, and delete data.
- The examples of database languages are **SQL(structured query language), My Access, Oracle**, etc.

People

The people who control and manage the databases and perform different types of operations on the database in the DBMS. The people include database administrator, software developer and End user.

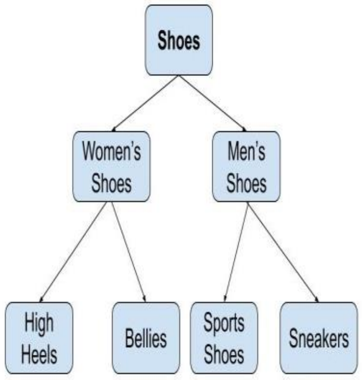
(c) Explain about different Data Models.

Data Model: Data models define how the logical structure of a database is modeled. Data Models are used to show how data is stored, connected, accessed and updated in the database management system. Therefore, there are following five data models used for understanding the structure of the database:

1. Hierarchical model
2. Relational model
3. Network model
4. Entity-Relationship (ER) Model
5. Object-oriented data model

1. Hierarchical Model: Hierarchical Model was the first DBMS model. This model organizes 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.

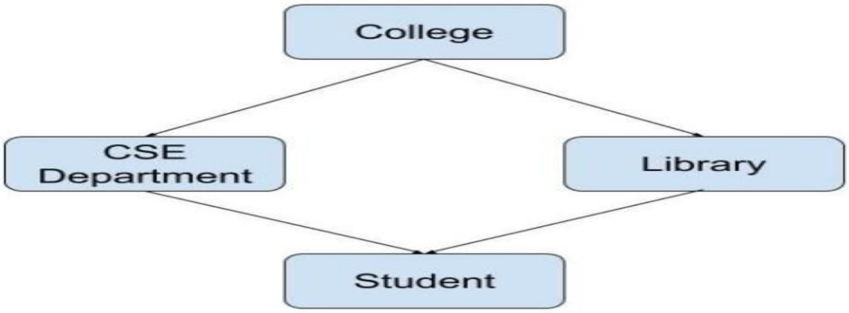
Example: We can represent the relationship between the shoes present on a shopping website in the following way:



Hierarchical Model

2. 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.

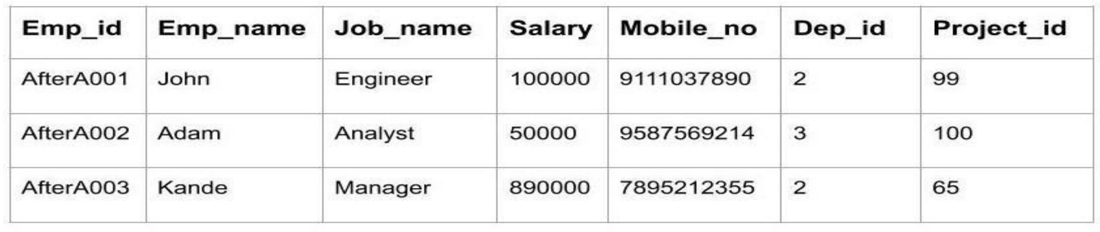
Example: In the example below we can see that node student has two parents i.e. CSE Department and Library. This was earlier not possible in the hierarchical model.



Network Model

3. Relational Model: Relational Model is the most widely used model. In this model, the data is maintained in the form of 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.

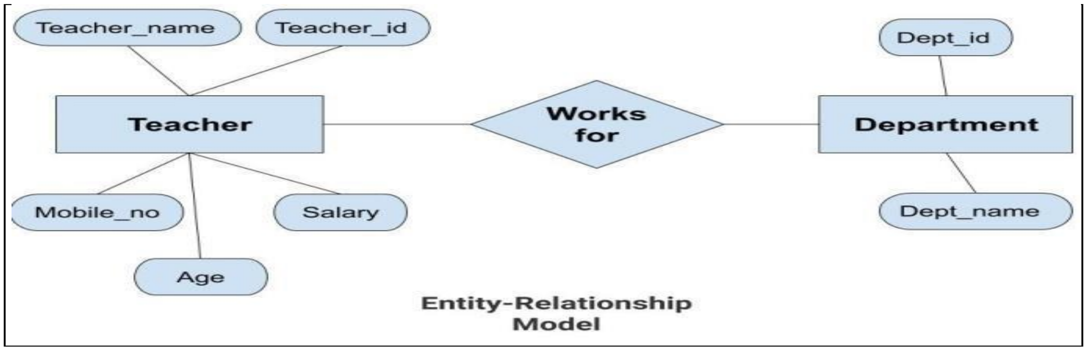


Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	John	Engineer	100000	9111037890	2	99
AfterA002	Adam	Analyst	50000	9587569214	3	100
AfterA003	Kande	Manager	890000	7895212355	2	65

EMPLOYEE TABLE

4. 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.

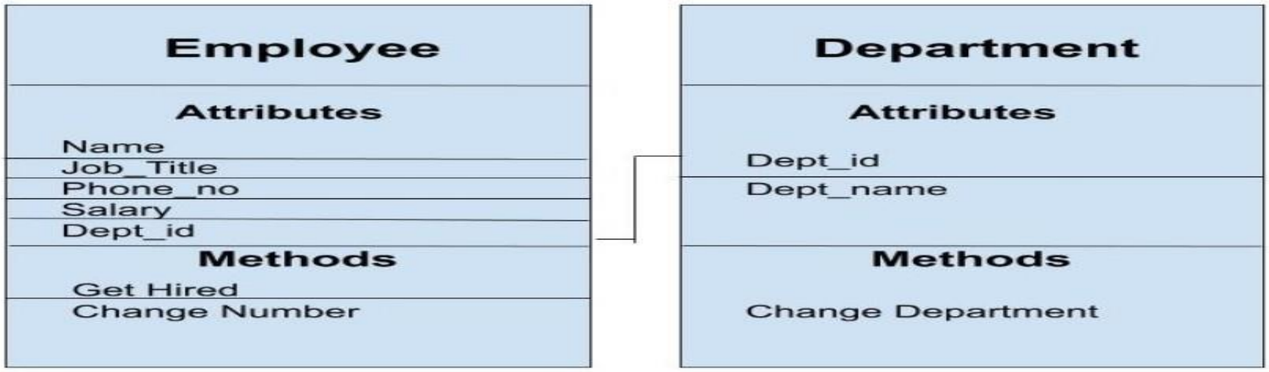
Example: In the below diagram, the entities are Teacher and Department. The attributes of **Teacher** entity are Teacher_Name, Teacher_id, Age, Salary, Mobile_Number. The attributes of entity **Department** entity are Dept_id, Dept_name. The two entities are connected using the relationship. Here, each teacher works for a department.



Entity-Relationship Model

5. Object-Oriented Data Model: The object-Oriented data model is that model representing the data and relationship as the object. This model stores the videos, audios, and graphical files which can’t be stored in the relationship model.

This model is based on the three main components whose names are object class, object identity, and object structure.



Object_Oriented_Model

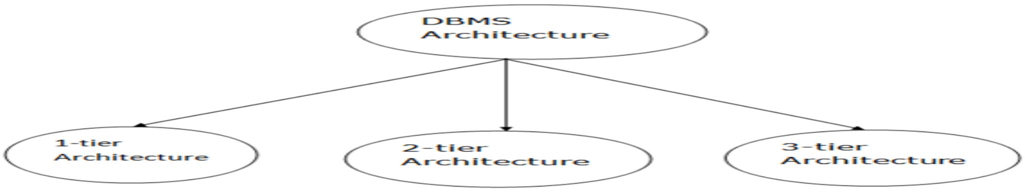
a)Discuss about the architecture of Database Management Systems.

Architecture of Database Management Systems.

The DBMS design depends upon its architecture. The basic client/server architecture is used to deal with a large number of PCs, web servers, database servers and other components that are connected with networks.

DBMS architecture depends upon how users are connected to the database to get their request done.

Types of DBMS Architecture

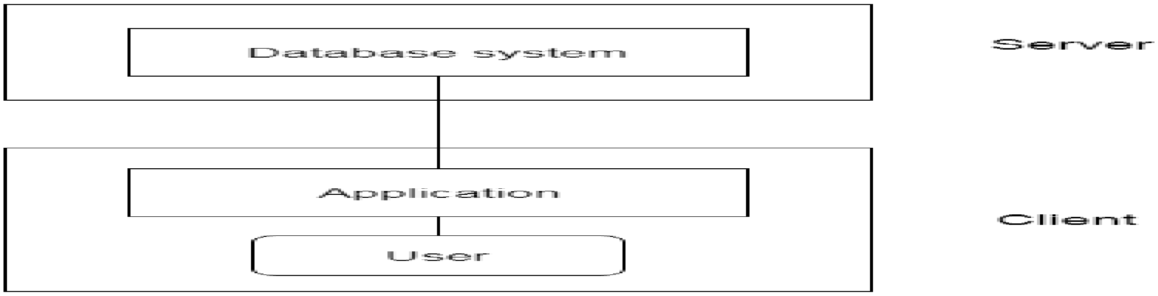
- 

1-Tier Architecture

In this architecture, the database is directly available to the user. It means the user can directly sit on the DBMS and uses it. Any changes done here will directly be done on the database itself. It doesn't provide a handy tool for end users. The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

2-Tier Architecture

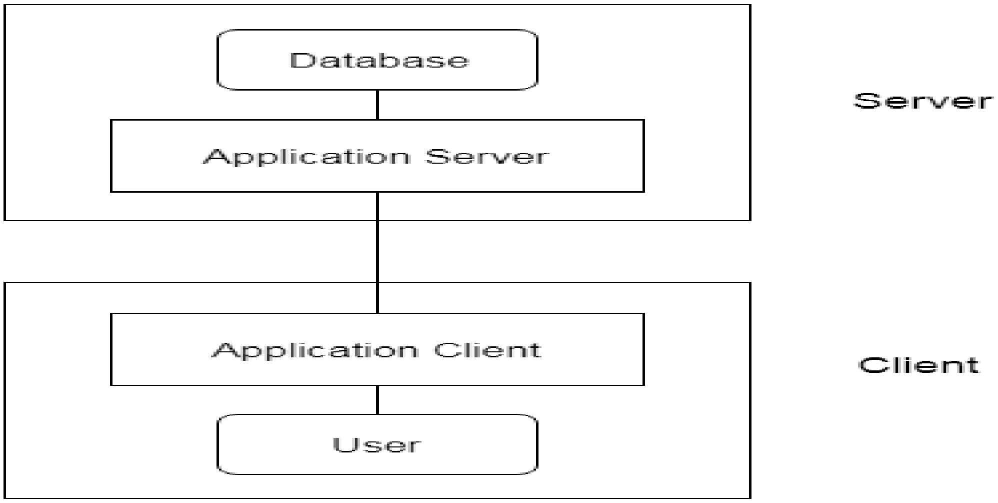
The 2-Tier architecture is same as basic client-server. In the two-tier architecture, applications on the client end can directly communicate with the database at the server side. The user interfaces and application programs are run on the client-side. The server side is responsible to provide the functionalities like: query processing and transaction management.



Three tier Architecture:

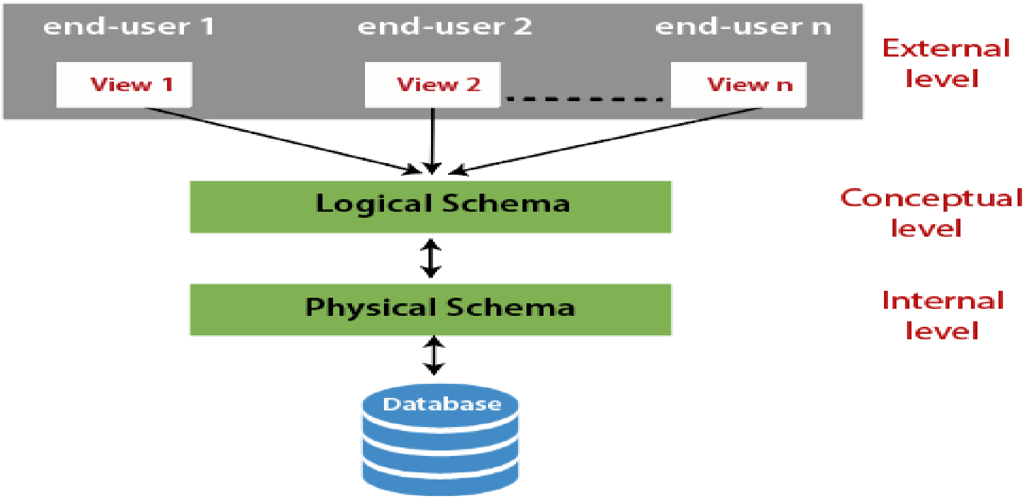
The three schema architecture describes how the data is represented or viewed by the user in the database. The three schema architecture divides the database into three-level to create a separation between the physical database and the user application. In simple words, this architecture hides the details of physical storage from the user. The database administrator (DBA) should be able to change the structure of database storage without affecting the users view

- The 3-Tier architecture is used in case of large web application.



This architecture contains three layers or levels of the database management system:

2. External level
3. Conceptual level
4. Internal level



DBMS Architecture

1. External or View level: his 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.

2. Conceptual or Logical level: The conceptual level describes the structure of the whole database. This level acts as a middle layer between the physical storage and user view. It explains what data to be stored in the database, what relationship exists among those data, and what the data types are. There is only one conceptual schema per database.

3. Internal or Physical level: This is the lowest level of database abstraction. The internal schema is also known as a physical schema. It describes how the data is actually stored in the database and provides methods to access data from 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.

b)Explain the Advantages of DBMS.

DATABASE MANAGEMENT SYSTEM (DBMS): A database management system (DBMS) is a software for creating and managing databases. A DBMS makes it possible for end users to create, read, update and delete data in a database.

1. Eliminates Data Redundancy: Unlike traditional file-system storage, Data Redundancy in DBMS is very less or not present. Data Redundancy occurs when the same data are stored unnecessarily at different places. Data Redundancy is reduced or eliminated in DBMS because all data are stored at a centralized location rather than being created by individual users and for each application.

2. Data Inconsistency: Data inconsistency exists when different versions of the same data appear in different places. The probability of data inconsistency is greatly reduced in a properly designed database.

3. Data Sharing: Data Sharing is the primary advantage of Database management systems. DBMS system allows users and applications to share Data with multiple applications and users. Data are stored in one or more servers in the network and that there is some software locking mechanism that prevents the same set of data from being changed by two people at the same time.

4. Data Security: DBMS systems provide a strong framework to protect data privacy and security. DBMS ensures that only authorized users have access to data and there is a mechanism to define access privileges.

5. Data Concurrency: In DBMS, Data are stored in one or more servers in the network and that there is some software locking mechanism that prevents the same set of data from being changed by two people at the same time.

6. Data Access: Data Access great advantage of DBMS because we can write small queries which will search the Database for you and it will retrieve the information in the fastest way possible due to its inbuilt searching operations.

7. Data Backup and Recovery: This is another advantage of DBMS as it provides a strong framework for Data backup, users are not required to back up their data periodically and manually, it is automatically taken care by DBMS

8. Data Atomicity: An atomic transaction is one in which all of the database actions occur or none of them do. It is the duty of DBMS to store a complete transaction in the database. If any transaction is partially completed, then it rolls backs them**.**

9.Data Searching(queries): Searching and retrieving of data is very easy in DBMS systems. In DBMS, we can write small queries to search for multiple information at a time from the data from DB servers.

10.Mass Storage

DBMS can store a lot of data in it. So for all the big firms, DBMS is really ideal technology to use. It can store thousands of records in it and one can fetch all that data whenever it is needed.

11.Multiple Users Access

No one handles the whole database alone. There are lots of users who are able to access database. So this situation may happen that two or more users are accessing database. They can change whatever they want, at that time DBMS makes it sure that they can work concurrently.

c)Explain the basic components of Database System.

5Marks

7.a)Write the features of File based System.

File based system: The systems that are used to organize and maintain data files are known as file based data systems. These file systems are used to handle a single or multiple files and are not very efficient.

Features:

a)The File based system is limited to a smaller size and cannot store large amounts of data. This system is relatively uncomplicated but this means it cannot support complicated queries, data recovery etc.

b)There may be redundant data in the file based system as it does not have a complex mechanism to get rid of it.

c)A file based system helps in basic data management for any user.

d)The data stored in the file based system should remain consistent. Any transactions done in the file based system should not alter the consistency property.

d)The file based system should not allow any illegal or potentially hazardous operations to occur on the data.

e)The file based system should allow concurrent access by different processes and this should be carefully coordinated.

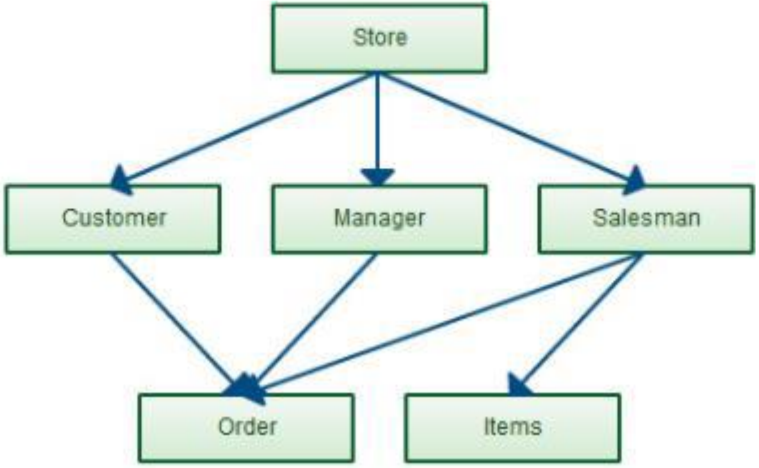
f)The file based system should make sure that the data is uniformly structured and stored so it is easier to access it.

(b)Explain the features of Network Data Model.

Network Data Model:

The network model was created to represent complex data relationships more effectively when compared to hierarchical models, to improve database performance and standards.

Given below is the pictorial representation of the network model in DBMS −



Features

The features of a Network Model are as follows −

- **Ability to Merge Relationships** − In this model, because of more relationships the data is more related. It has an ability to manage one-to-one relationships as well as many-to-many relationships.

- **1to Many:** M It has entities which are organized in a graphical representation and some entities are accessed through several paths. A User perceives the network model as a collection of records in 1:M relationships.
- **Data access fast and simple:** Many paths − There can be more than one path to the same record because of more relationships. It makes data access fast and simple.
- **Circular Linked List** − The operations in this model are done with the help of the circular linked list. The current position is maintained with the help of a program and navigates through the records based on relationships.
- **Include DDL and DML commands:** It includes Data Definition Language (DDL) and Data Manipulation Language (DML) commands.

(c)Discuss about File based System Vs Data based System.

File System:

File system is a method of organising the files with a hard disk or other medium of storage. File system arranges the files and helps in retrieving the files, when required. It is compatible with different file types

DBMS:

DBMS, meanwhile, is the acronym for Database Management System. It is also a software used to store and regain user's data, while also maintaining the required security measures. This includes a group of programmes that can help to manipulate the database

Difference between File System and DBMS

FILE SYSTEM	**DBMS**
Used to manage and organize the files stored in the hard disk of the computer	A software to store and retrieve the user's data
Redundant data is present	No presence of redundant data
Query processing is not so efficient	Query processing is efficient
Data consistency is low	Due to the process of normalization, the data consistency is high
Less complex, does not support complicated transactions	More complexity in managing the data, easier to implement complicated transactions
Less security	Supports more security mechanisms
Less expensive in comparison to DBMS	Higher cost than the File system
Does not support crash recovery	Crash recovery mechanism is highly supported

12. (a) What is File based System

(b) Explain DBMS Vendors and their Products.

DBMS Vendors : The major DBMS vendors are Oracle, IBM, Microsoft and Sybase (see Oracle Database, DB2, SQL Server and ASE). MySQL and SQLite are very popular open source products (see MySQL and SQLite).

List Of The Top Database Management Software

Given below is the list of most popular database management systems:

SolarWinds Database Performance Analyzer

1. Db Visualizer
2. Manage Engine Applications Manager
3. Oracle RDBMS
4. IBM DB2
5. Microsoft SQL Server
6. SAP Sybase ASE
7. Teradata
8. ADABAS
9. MySQL
10. FileMaker
11. Microsoft Access

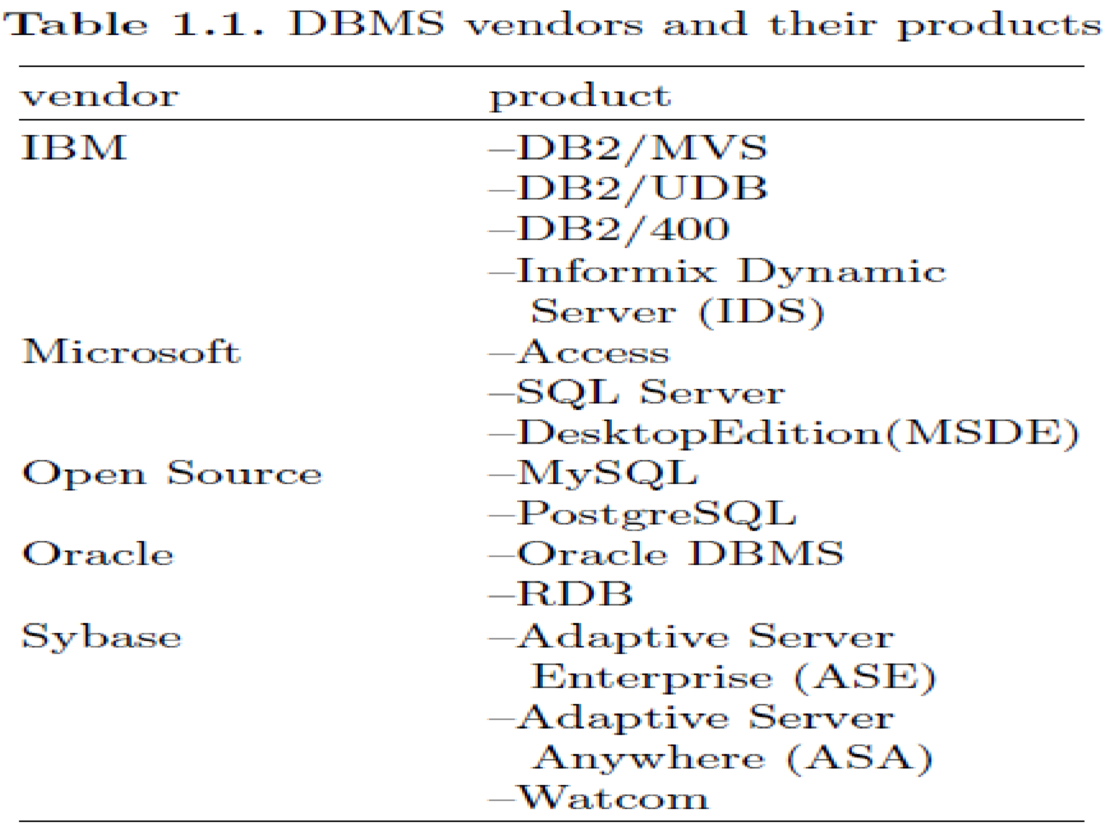


Table 1.1. DBMS vendors and their products

vendor	product
IBM	–DB2/MVS –DB2/UDB –DB2/400 –Informix Dynamic Server (IDS)
Microsoft	–Access –SQL Server –DesktopEdition(MSDE)
Open Source	–MySQL –PostgreSQL
Oracle	–Oracle DBMS –RDB
Sybase	–Adaptive Server Enterprise (ASE) –Adaptive Server Anywhere (ASA) –Watcom

c)Explain any two different data models.

III UNIT

3.(a) Explain the Features of Entity Relationship Model.

Entity Relationship Model.

An Entity Relationship model or ER Diagram is a diagram that represents relationships among entities in a database. It is commonly known as an ER Diagram. An ER Diagram in DBMS plays a crucial role in designing the database.

Features of ER model:

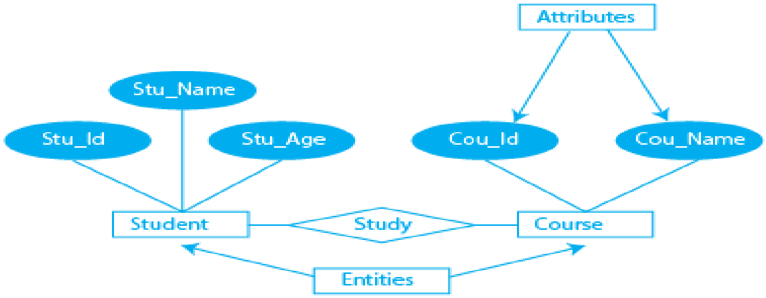
The features of the ER model are as follows –

High-level data model ER model in DBMS is the high-level data model. It stands for the Entity-relationship model and is used to represent a logical view of the system from a data perspective.

Provides Unified and graphical view: It describes data as a set of entities, attributes, and relationships. It provides the graphical view of the database design and easy to understand.

Easy implementation: it is widely used to develop the initial design of the database. It provides a collection of basic concepts that can be easily implemented in a database system.

ER diagrams: It is the basic concept of ER model. ER model makes use of ER diagrams, which are the diagrams sketched to design a database. ER diagrams are built on three basic concepts: entities, attributes, and relationships between them. It develops a conceptual design for a database that provides a very simple and straightforward view of the data. An ER diagram in DBMS defines entities, associated attributes, and relationships between entities. This helps visualize the logical structure of the database. The following is the example for ER diagram.



Real-world Entities: objects E-R diagrams are used to model real-world objects like a person, a car, a company and the relation between these real-world objects.

Easy understand the database system to users: Creating an ER Model in DBMS is considered a best practice before implementing your database because it makes it easier for the developers to understand the database system just by looking at the ER model.

Simple and straightforward view It develops a conceptual design for a database that provides a very simple and straightforward view of the data.

Defines relationships An ER diagram in DBMS defines entities, associated attributes, and relationships between entities. This helps visualize the logical structure of the database.

Easy convertible to other models: The E-R diagram used for representing E-R model can be easily converted into relations(tables) in relation model.

Effective communication tool: ER-Model is an effective communication tool among users and database designers.

Visual representation of data logical structure: ER model makes use of ER diagrams, which are the diagrams sketched to design a database. ER diagrams are built on three basic concepts: entities, attributes, and relationships between them. An ER diagram in DBMS defines entities, associated attributes, and relationships between entities. This helps visualize the logical structure of the database.

Provide description of items: Entity-Relationship models are useful for providing descriptions of items.

Require no technical knowledge:E-R diagrams require no technical knowledge and no hardware support.

b)Write about the building blocks of Entity Relationship model.

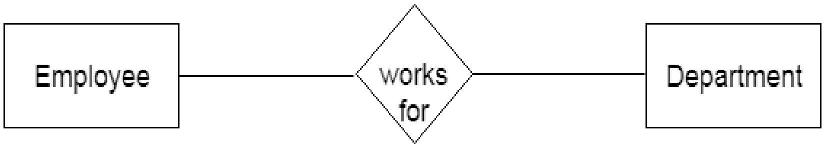
Entity–relationship model: An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database.

The Building blocks of ER Diagram are:

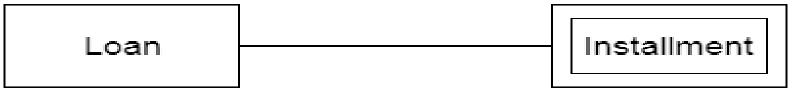
1. Entities
2. Attributes
3. Relationships
4. Constraints

1. Entityy**:** An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

E.g. Painter, Employee, Student.

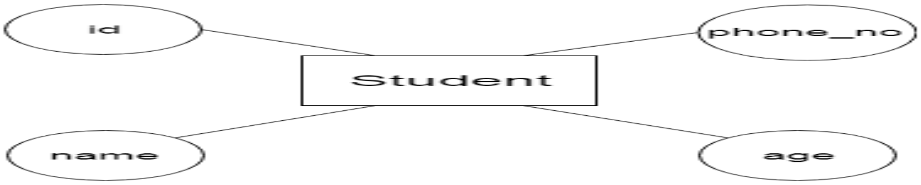


Weak Entity: An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.

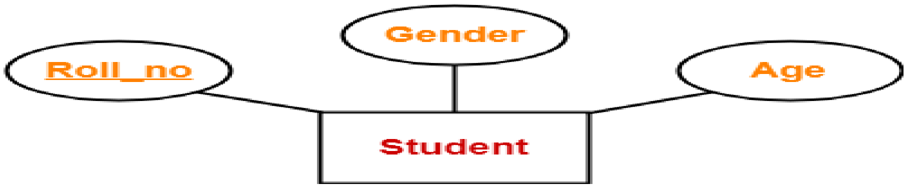


2. Attribute: The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

For example, id, age, contact number, name, etc. can be attributes of a student.

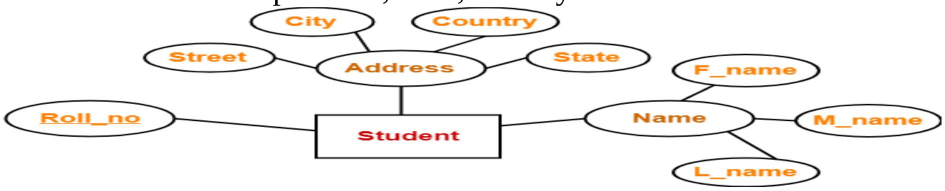


a. Key Attribute: The key attribute is used to represent the main characteristics of an entity. It represents a primary key. Key attribute is used to identify an entity uniquely in an entity set. The key attribute is represented by an ellipse with the text underlined.



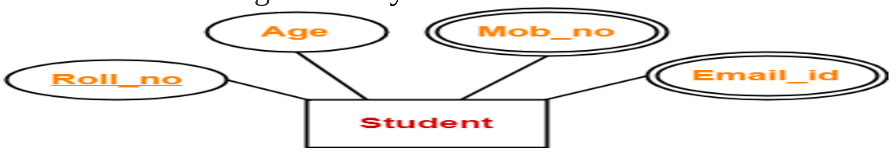
b. Composite attribute:: An attribute that is a combination of other attributes is known as composite attribute.(Or) Composite attributes are those attributes which can be further divided into many other simpleattributes is called composite attribute.

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.



c. Multi valued Attribute: An attribute can have more than one value. These attributes are known as a multi valued attribute. The double oval is used to represent multi valued attribute.

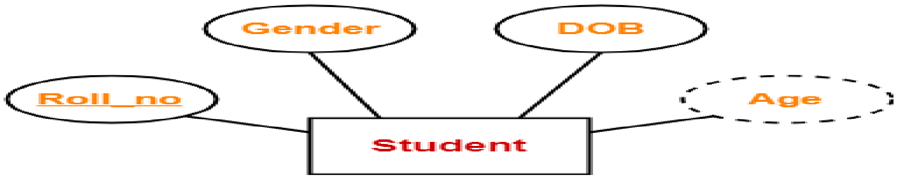
For Example Here, the attributes "Mob_no" and "Email_id" are multi valued attributes as they can take more than one values for a given entity.



d. Derived Attribute

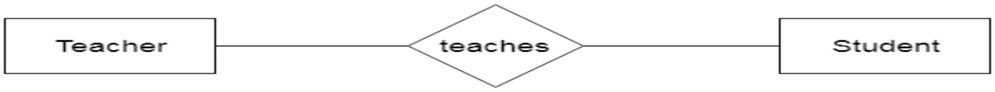
An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

For example, A person's age changes over time and can be derived from another attribute like Date of birth.



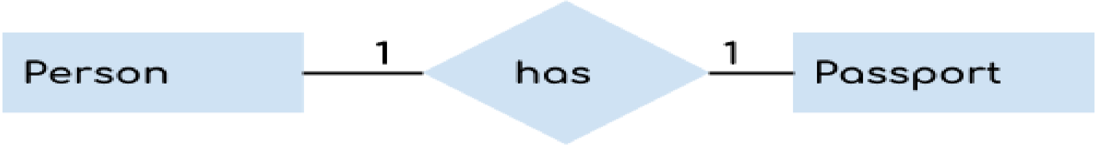
3. Relationship

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

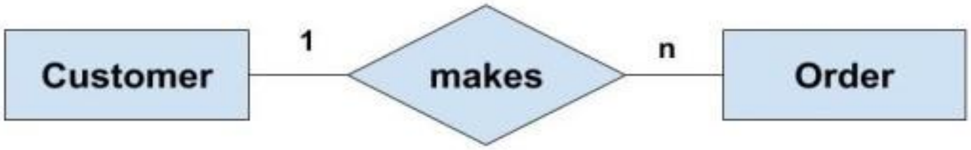


Types of relationship are as follows:

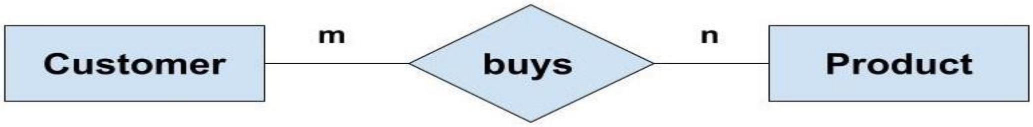
.1One to One Relationship: When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship. For example, a person has only one passport and a passport is givento one person.



.2One to Many Relationship: When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship. For example – a customer can place many orders but aorder cannot be placed by many customers.



3 .Many to Many Relationships : When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship. For example, a many-to-many relationship exists between customers and products: customers can purchase various products, and products can be purchased by many customers.



4. **Constraints:** Constraints are the rules that we can apply on the type of data in a table. That is, we can specify the limit on the type of data that can be stored in a particular column in a table using constraints. For example NOT NULL - Ensures that a column cannot have a NULL value, UNIQUE - Ensures that all values in a column are different

(c) Write about classification of Entity sets.

Entity–relationship model: 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 Building blocks of ER Diagram are:

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

Entity : An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

E.g. Painter, Employee, Student

Entity Set: It is a collection or a group of ‘entities’ sharing exactly the ‘same set of attributes’. All **entities** can be **distinctly identified** in an entity set. This is because all the entities have a different set of value for some set of attributes. We further classify the entity set into two basic categories **Strong** and **Weak entity set.**

Entity set can be classified into two categories as shown below:

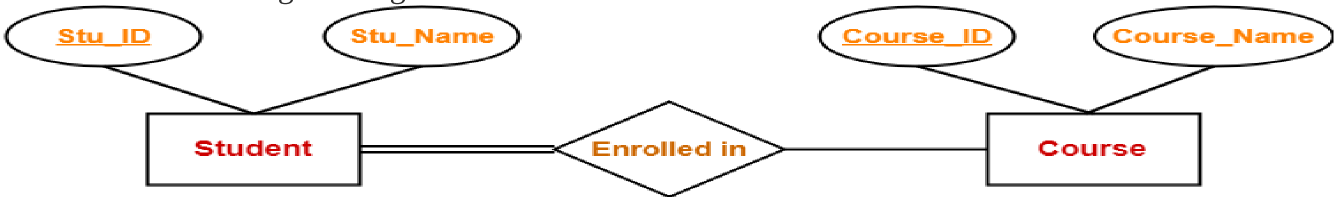
1. Strong Entity Set
2. Weak Entity Set

1.Strong Entity:.

A strong entity set is an entity set that contains sufficient attributes to uniquely identify all its entities.

- In other words, a primary key exists for a strong entity set.
- Primary key of a strong entity set is represented by underlining it.
- A single rectangle is used for representing a strong entity set.

Consider the following ER diagram-



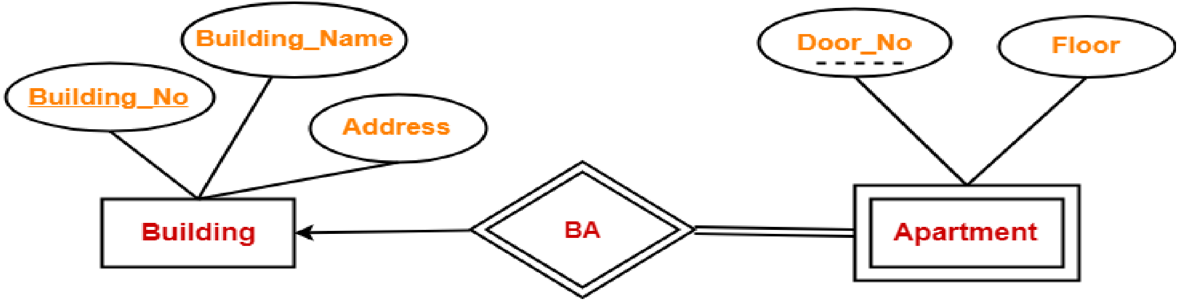
In this ER diagram,

- Two strong entity sets "**Student**" and "**Course**" are related to each other.
- Student ID and Student name are the attributes of entity set "Student".
- Student ID is the primary key using which any student can be identified uniquely.
- Course ID and Course name are the attributes of entity set "Course".
- Course ID is the primary key using which any course can be identified uniquely.

2.Weak Entity Set-

A weak entity set is an entity set that does not contain sufficient attributes to uniquely identify its entities.

- In other words, a primary key does not exist for a weak entity set.
- A double rectangle is used for representing a weak entity set.
- A double diamond symbol is used for representing the relationship that exists between the strong and weak entity sets and this relationship is known as **identifying relationship**.



In this ER diagram,

- One strong entity set "**Building**" and one weak entity set "**Apartment**" are related to each other.
- Strong entity set "Building" has building number as its primary key.
- Door number is the discriminator of the weak entity set "Apartment".
- This is because door number alone can not identify an apartment uniquely as there may be several other buildings having the same door number.

(Or)

a)Explain the features of Relational Model.

INTRODUCTION: Relational Model was proposed by E.F. Codd to model data in the form of relations or tables. After designing the conceptual model of Database using ER diagram, we need to convert the conceptual model in the relational model which can be implemented using any RDMBS languages like Oracle SQL, MySQL etc.

Features of a relational database

ACID characteristics.

ACID refers to four essential properties: Atomicity, Consistency, Isolation, and Durability.
These features are the key difference between a relational database and a non-relational database.

Atomicity

Atomicity keeps data accurate. It makes sure all data is compliant with the rules, regulations, and policies of the business.

Consistency

The state of the database must remain consistent throughout the transaction.
Consistency defines the rules for maintaining data points. This ensures they remain in a correct state after a transaction.

Isolation

With a relational database, each transaction is separate and not dependent on others. This is made possible by isolation.

Durability

Durability means that you can recover data from a failed transaction.
It also ensures that data changes are permanent.

Data flexibility

Relational databases allow for flexibility. Users can change what they see. And it’s easy to add additional data at a later time.

Relations:

Relational Model represents how data is stored in Relational Databases. A relational database stores data in the form of relations (tables). Consider a relation STUDENT with attributes ROLL_NO, NAME, ADDRESS, PHONE and AGE shown in Table 1.

STUDENTROLL_	**NAME**	**ADDRESS**	**PHONE**	**AGE**
1	RAM	DELHI	9455123451	18
2	RAMESH	GURGAON	9652431543	18
3	SUJIT	ROHTAK	9156253131	20
4	SURESH	DELHI		18

1. **Attribute:** Attributes are the properties that define a relation. e.g.; **ROLL_NO**, **NAME**
2. **Relation Schema:** A relation schema represents name of the relation with its attributes. e.g.; STUDENT (ROLL_NO, NAME, ADDRESS, PHONE and AGE) is relation

schema for STUDENT. If a schema has more than 1 relation, it is called Relational Schema.

3. **Tuple:** Each row in the relation is known as tuple. The above relation contains 4 tuples, one of which is shown as:

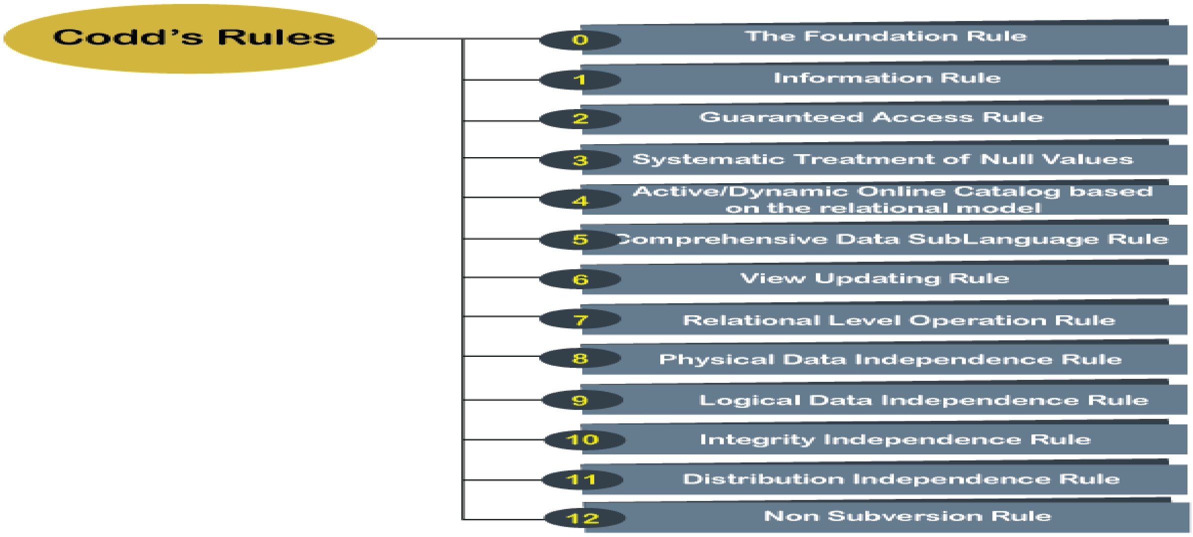
1	RAM	DELHI	9455123451	18

4. **Relation Instance:** The set of tuples of a relation at a particular instance of time is called as relation instance. Table 1 shows the relation instance of STUDENT at a particular time. It canchange whenever there is insertion, deletion or updation in the database.
5. **Degree:** The number of attributes in the relation is known as degree of the relation. The **STUDENT** relation defined above has degree 5.
6. **Cardinality:** The number of tuples in a relation is known as cardinality. The **STUDENT** relation definedabove has cardinality 4.
7. **Column:** Column represents the set of values for a particular attribute. The column **ROLL_NO** isextracted from relation STUDENT.
8. **NULL Values:** The value which is not known or unavailable is called NULL value. It is represented byblank space. e.g.; PHONE of STUDENT having ROLL_NO 4 is NULL.

b)Discuss about CODD's rules.

CODD's rules.

E.F Codd was a Computer Scientist who invented the **Relational model** for Database management. Based onrelational model, the **Relational database** was created. Codd proposed 13 rules popularly known
as **Codd's 12 rules** to test DBMS's concept against his relational model. Codd's rule actualy define what quality a DBMS requires in order to become a Relational Database Management System (RDBMS).


Codd's Rules
0 The Foundation Rule
1 Information Rule
2 Guaranteed Access Rule
3 Systematic Treatment of Null Values
4 Active/Dynamic Online Catalog based on the relational model
5 Comprehensive Data SubLanguage Rule
6 View Updating Rule
7 Relational Level Operation Rule
8 Physical Data Independence Rule
9 Logical Data Independence Rule
10 Integrity Independence Rule
11 Distribution Independence Rule
12 Non Subversion Rule

Rule 0: The Foundation Rule

The database must be in relational form. So that the system can handle the database through its relational capabilities.

Rule 1: Information Rule

A database contains various information, and this information must be stored in each cell of a table in the form of rows and columns.

Rule 2: Guaranteed Access Rule

Every single or precise data (atomic value) may be accessed logically from a relational database using the combination of primary key value, table name, and column name.

Rule 3: Systematic Treatment of Null Values

This rule defines the systematic treatment of Null values in database records. The null value has various meanings in the database, like missing the data, no value in a cell, inappropriate information, unknown data and the primary key should not be null.

Rule 4: Active/Dynamic Online Catalog based on the relational model

It represents the entire logical structure of the descriptive database that must be stored online and is known as a database dictionary. It authorizes users to access the database and implement a similar query language to access the database.

Rule 5: Comprehensive Data Sub Language Rule

The relational database supports various languages, and if we want to access the database, the language must be the explicit, linear or well-defined syntax, character strings and supports the comprehensive:

Rule 6: View Updating Rule

All views table can be theoretically updated and must be practically updated by the database systems.

Rule 7: Relational Level Operation (High-Level Insert, Update and delete) Rule

A database system should follow high-level relational operations such as insert, update, and delete in each level or a single row. It also supports union, intersection and minus operation in the database system.

Rule 8: Physical Data Independence Rule

All stored data in a database or an application must be physically independent to access the database. Each data should not depend on other data or an application. If data is updated or the physical structure of the database is changed, it will not show any effect on external applications that are accessing the data from the database.

Rule 9: Logical Data Independence Rule

It is similar to physical data independence. It means, if any changes occurred to the logical level (table structures), it should not affect the user's view (application).

Rule 10: Integrity Independence Rule

A database must maintain integrity independence when inserting data into table's cells using the SQL query language. All entered values should not be changed or rely on any external factor or application to maintain integrity.

Rule 11: Distribution Independence Rule

The distribution independence rule represents a database that must work properly, even if it is stored in different locations and used by different end-users. The end users can access the database, and these access data should be independent for every user to perform the SQL queries.

Rule 12: Non Subversion Rule

The non-submersion rule defines RDBMS as a SQL language to store and manipulate the data in the database. If a system has a low-level or separate language other than SQL to access the database system, it should not subvert or bypass integrity to transform data.

c)Explain the concept of Relational Integrity.

Relational Integrity:

Referential integrity or Relational Integrity refers to the relationship between tables. Because each table in a database must have a primary key, this primary key can appear in other tables because of its relationship to data within those tables. When a primary key from one table appears in another table, it is called a *foreign key*.

Integrity constraints: In database management systems (DBMS) there is a certain set of rules which are used to maintain the quality and consistency of data in the database. Every time there is an insertion, deletion, or updating of data in the database it is the responsibility of these integrity constraints to maintain the integrity of data and thus help to prevent accidental damage to the database.

types of integrity constraints: There are four types of integrity constraints in DBMS:

1. Domain Constraint
2. Entity Constraint
3. Referential Integrity Constraint
4. Key Constraint

Ensure that the overall validity: In Database Management Systems, integrity constraints are pre-defined set of rules that are applied on the table fields(columns) or relations to ensure that the overall validity, integrity, and consistency of the data present in the database table is maintained.

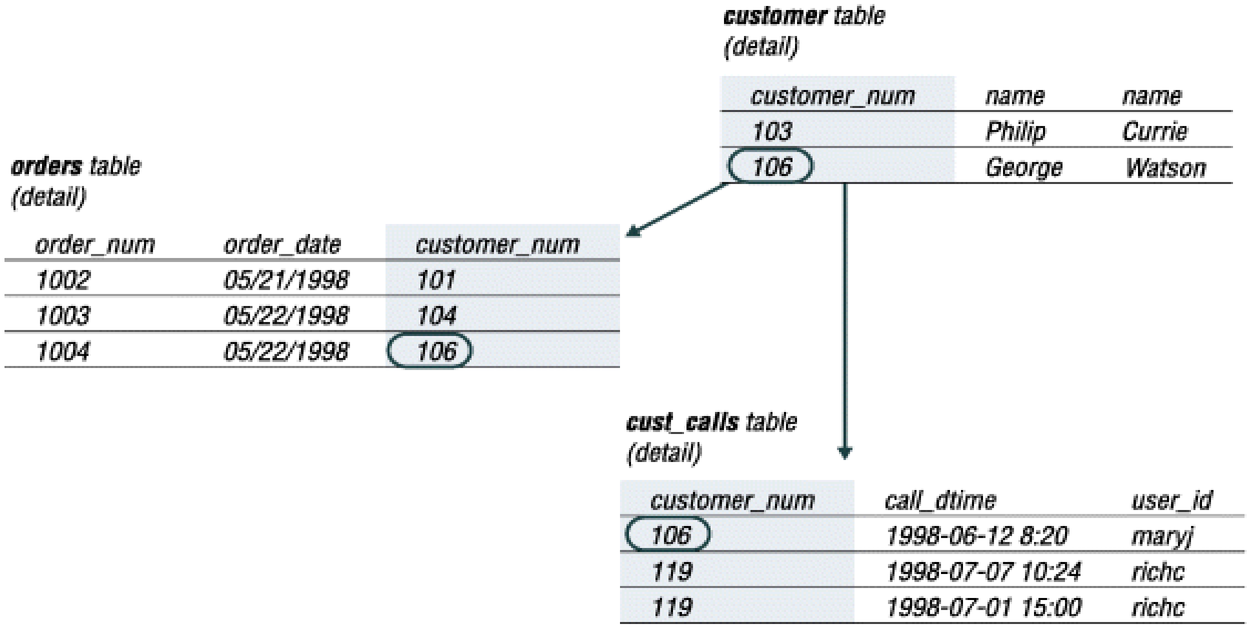
Foreign keys: Foreign keys join tables and establish dependencies between tables. Tables can form a hierarchy of dependencies in such a way that if you change or delete a row in one table, you destroy the meaning of rows in other tables.

Define primary and foreign keys: To define primary and foreign keys, and the relationship between them, use the CREATE TABLE and ALTER TABLE statements.

Example: To understand the concept of relational integrity we can take this example.

The following figure shows that the **customer_num** column of the **customer** table is a primary key for that table and a foreign key in the **orders** and **cust_call** tables. Customer number 106, George Watson is *referenced* in both the **orders** and **cust_calls** tables. If customer 106 is deleted from the **customer** table, the link between the three tables and this particular customer is destroyed.

Figure 1. Referential integrity in the demonstration database



customer table
(detail)

customer_num	name	name
103	Philip	Currie
106	George	Watson

orders table
(detail)

order_num	order_date	customer_num
1002	05/21/1998	101
1003	05/22/1998	104
1004	05/22/1998	106

cust_calls table
(detail)

customer_num	call_dtime	user_id
106	1998-06-12 8:20	maryj
119	1998-07-07 10:24	richc
119	1998-07-01 15:00	richc

When you delete a row that contains a primary key or update it with a different primary key, you destroy the meaning of any rows that contain that value as a foreign key.

Logical dependency of a foreign key on a primary key: Referential integrity is the logical dependency of a foreign key on a primary key. The integrity of a row that contains a foreign key depends on the integrity of the row that it references—the row that contains the matching primary key.

Error message: By default, the database server does not allow you to violate referential integrity and gives you an error message if you attempt to delete rows from the parent table before you delete rows from the child table.

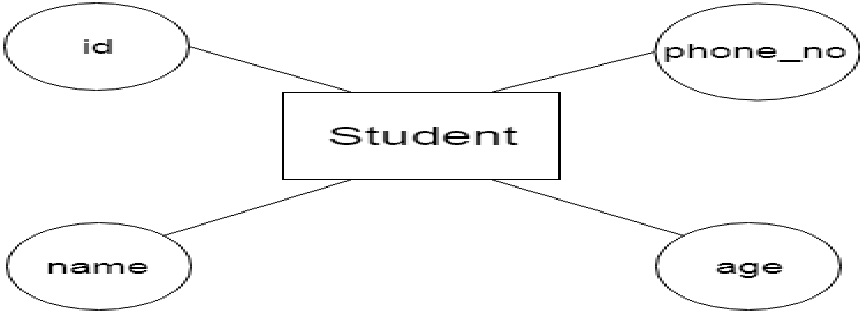
5 Marks

8.(a) Explain about Attribute classification.

Attribute:

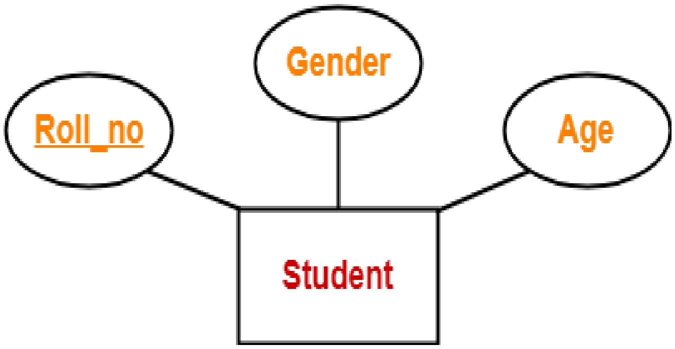
The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

For example, id, age, contact number, name, etc. can be attributes of a student.



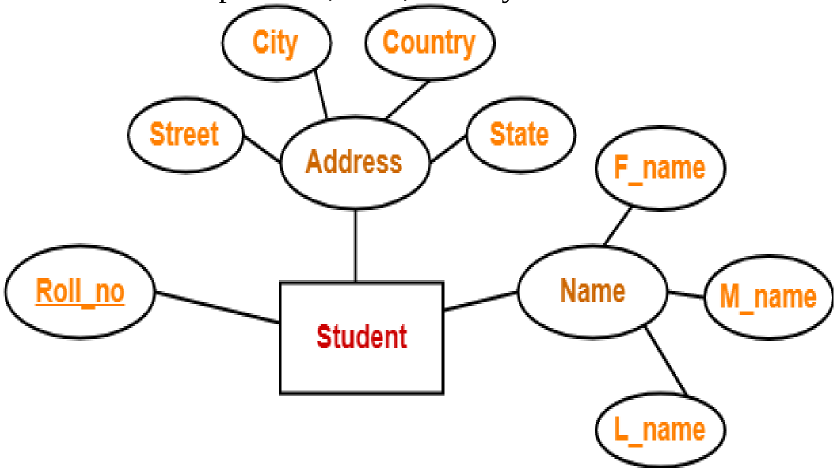
Attribute classification:

a. Key Attribute: The key attribute is used to represent the main characteristics of an entity. It represents a primary key. Key attribute is used to identify an entity uniquely in an entity set. The key attribute is represented by an ellipse with the text underlined.



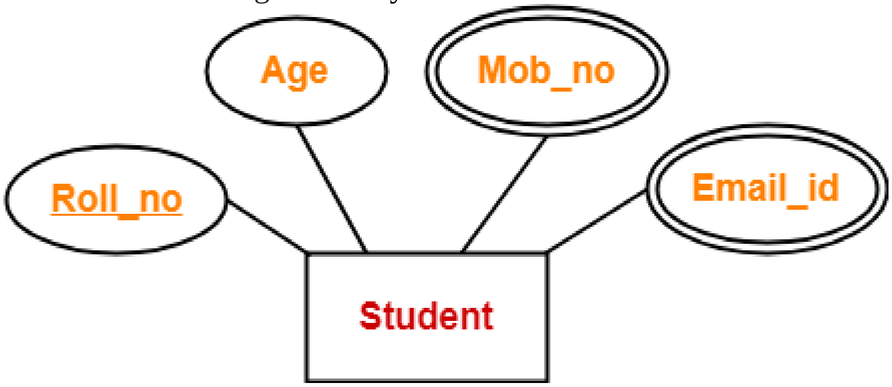
b. Composite attribute:: An attribute that is a combination of other attributes is known as composite attribute.(Or) Composite attributes are those attributes which can be further divided into many other simpleattributes is called composite attribute.

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.



c. **Multi valued Attribute:** An attribute can have more than one value. These attributes are known as a multi valued attribute. The double oval is used to represent multi valued attribute.

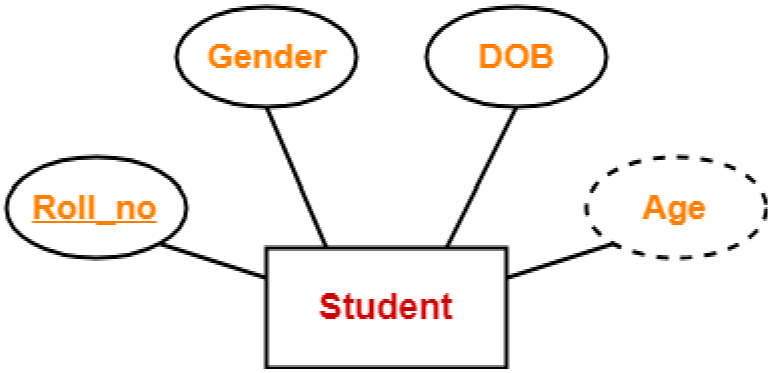
For Example Here, the attributes “Mob_no” and “Email_id” are multi valued attributes as they can take more than one values for a given entity.



d. **Derived Attribute**

An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

For example, A person's age changes over time and can be derived from another attribute like Date of birth.



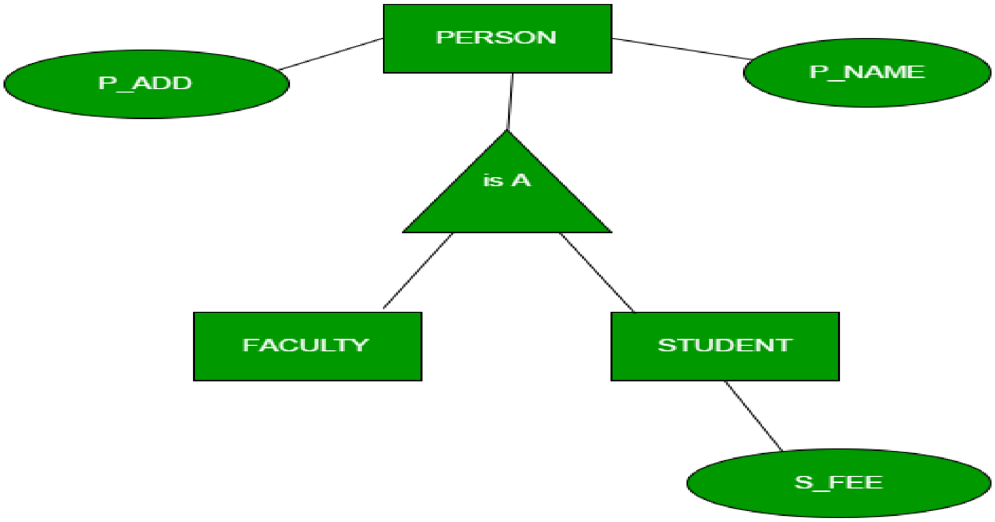
(b) Write about transforming Generalization and Specialization.

Generalization and Specialization Generalization, Specialization and Aggregation in ER model are used for data abstraction in which abstraction mechanism is used to hide details of a set of objects.

Generalization –

Generalization is the process of extracting common properties from a set of entities and create a generalized entity from it. It is a bottom-up approach in which two or more entities can be generalized to a higher level entity if they have some attributes in common.

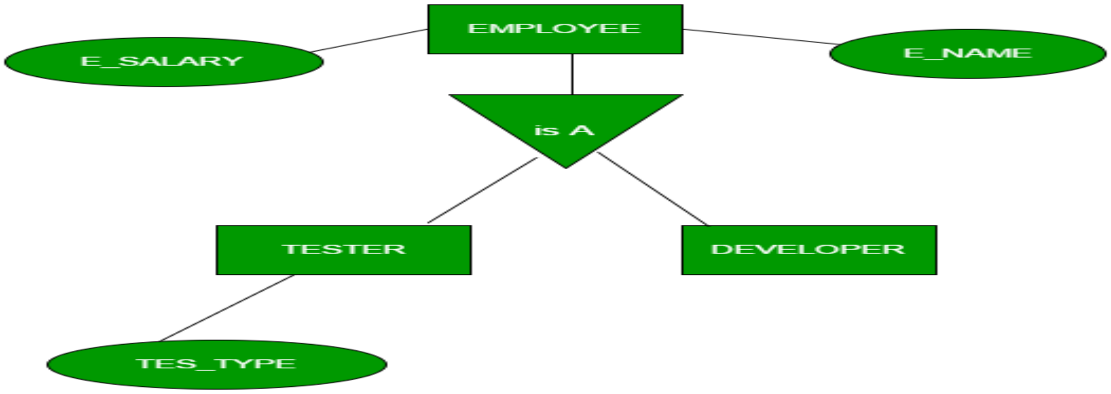
For Example, STUDENT and FACULTY can be generalized to a higher level entity called PERSON as shown in Figure 1. In this case, common attributes like P_NAME, P_ADD become part of higher entity (PERSON) and specialized attributes like S_FEE become part of specialized entity (STUDENT).



Specialization –

In specialization, an entity is divided into sub-entities based on their characteristics. It is a top-down approach where higher level entity is specialized into two or more lower level entities. For Example, EMPLOYEE entity in an Employee management system can be specialized into DEVELOPER, TESTER etc. as shown in Figure 2. In this case, common attributes like E_NAME, E_SAL etc. become part of higher entity (EMPLOYEE) and specialized attributes like TES_TYPE become part of specialized entity (TESTER).

(c) Explain the merits of Relational Data Model.



Specialization

13.(a) Explain Relationship Degree.

Relationship Degree:

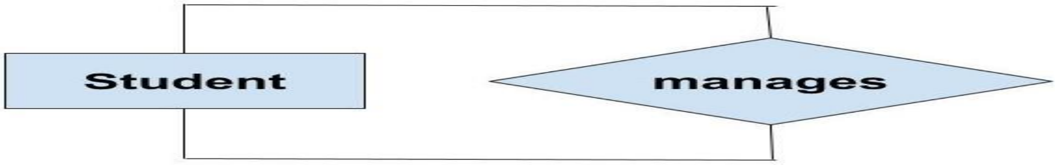
The degree of a relationship is the number of entity types that participate(associate) in a relationship. By seeing an E-R diagram, we can simply tell the degree of a relationship i.e the number of an entity type that is connected to a relationship is the degree of that relationship.

Based on the number of entity types that are connected we have the following degree of relationships:

a. Unary
b. Binary
c. Ternary
d. N-ary

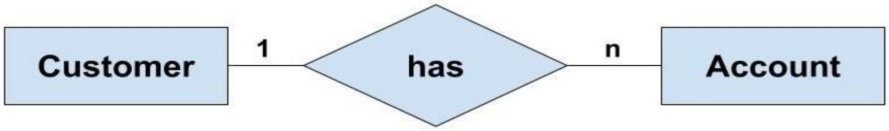
a). Unary (degree 1): *A unary relationship exists when both the participating entity type are the same. When such a relationship is present we say that the degree of relationship is 1.*

So, the ‘Student’ is the only entity participating here. We can represent this relationship using the E-R diagram as follows:



Unary Relationship
(degree 1)

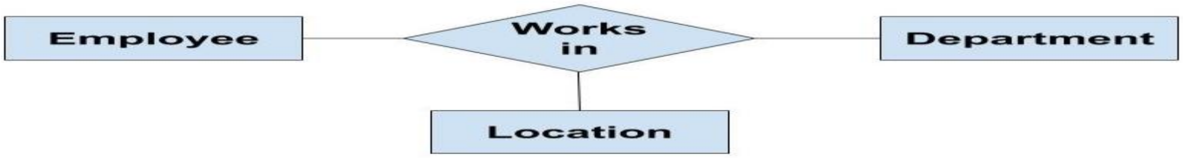
b). Binary (degree 2): *A binary relationship exists when exactly two entity type participates. When such a relationship is present we can say that the degree is 2. This is the most common degree of relationship. For example customer has a account.*



Binary Relationship
(degree 2)

c). Ternary (degree 3): *A ternary relationship exists when exactly three entity type participates. When such a relationship is present we say that the degree is 3.*

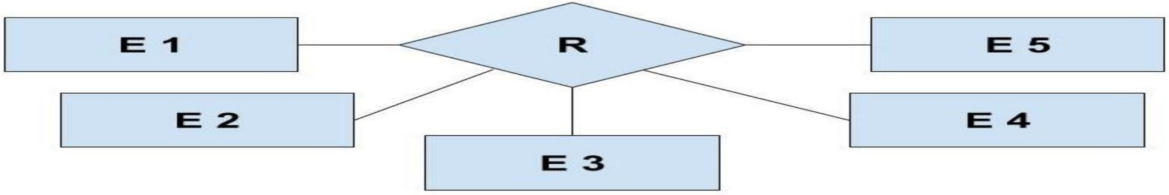
For example, We have three entity type 'Employee', 'Department' and 'Location'. The relationship between these entities are defined as an employee works in a department, an employee works at a particular location.



Ternary Relationship
(degree 3)

d). N-ary (n degree): *An N-ary relationship exists when 'n' number of entities are participating. So, any number of entities can participate in a relationship. There is no limitation to the maximum number of entities that can participate.*

We represent an N-ary relationship as follows:



N-ary Relationship

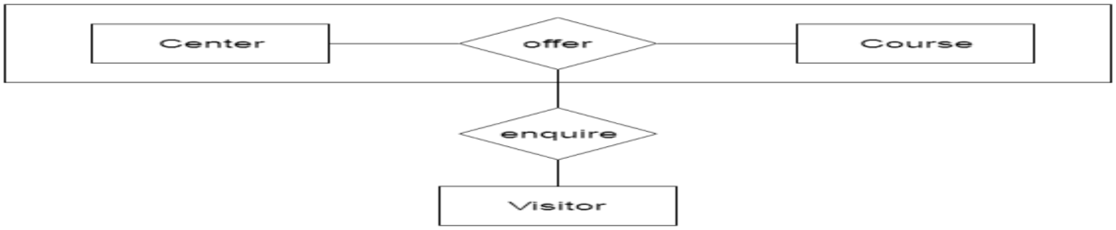
(b) Write about Aggregation and Composition.

Aggregation and Composition:

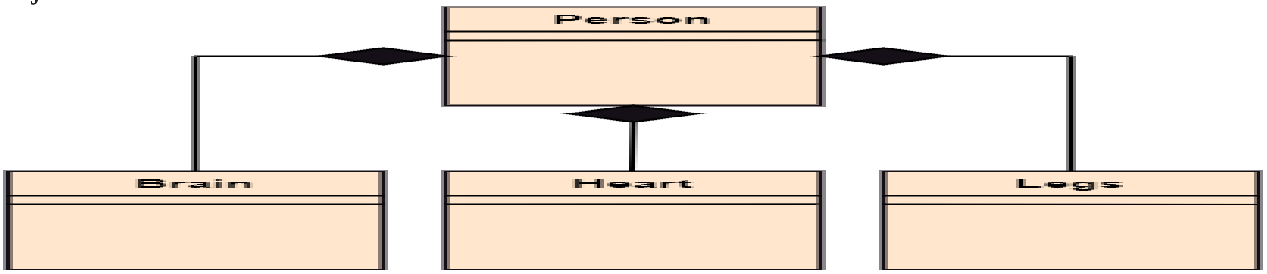
Both Composition and Aggregation are types of association which are used to represent the relationship between two classes. But they are absolutely different from each other. The basic difference between the two is that composition is a strong association, while aggregation is a weak association.

Aggregation: Aggregation in DBMS (Database Management System) is a process of combining two or more entities to form a more meaningful new entity. To establish a single entity, aggregation creates a relationship that combines these entities.

For example: Center entity offers the Course entity act as a single entity in the relationship which is in a relationship with another entity visitor. In the real world, if a visitor visits a coaching center then he will never enquiry about the Course only or just about the Center instead he will ask the enquiry about both.



Composition: The composition is a part of aggregation, and it portrays the whole-part relationship. It depicts dependency between a composite (parent) and its parts (children), which means that if the composite is discarded, so will its parts get deleted. It exists between similar objects.



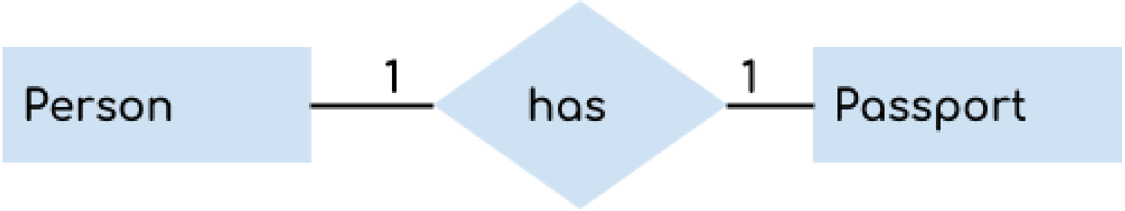
(c) Discuss about Relationship classification.

A relational database collects different types of data sets that use tables, records, and columns. It is used to create a well-defined relationship between database tables so that relational databases can be easily stored. A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

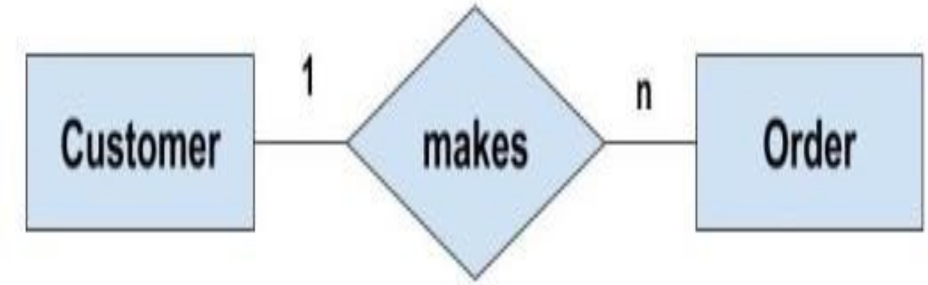
There are four types of relationships:

1. One to One
2 One to Many
3 Many to One
4. Many to Many

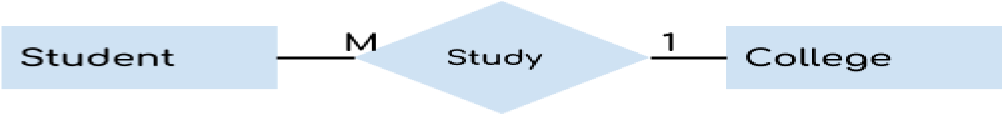
.1One to One Relationship: When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship. For example, a person has only one passport and a passport is givento one person.



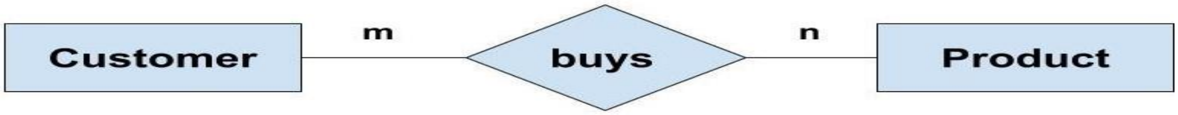
.2One to Many Relationship: When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship. For example – a customer can place many orders but aorder cannot be placed by many customers.



.3Many to One Relationship: When more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship. For example – many students can study in a single college but a student cannot study in many colleges at the same time.



.4Many to Many Relationships : When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship. For example, a many-to-many relationship exists between customers and products: customers can purchase various products, and products can be purchased by many customers.



IV UNIT

4.a)Define Structured Query Language? Write about various commands in SQL.

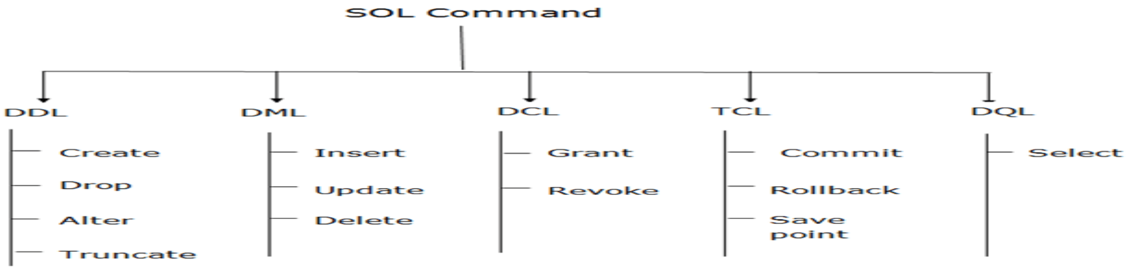
Structured Query Language: SQL: Structured query language (SQL) is a standard programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values. You can use SQL

SQL Commands

SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data. SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

Types of SQL Commands

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.



1.Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

For example

CREATE It is used to create a new table in the database.

Syntax:

CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,....]);

2. Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.
- DQL commands are included in DML

Here are some commands that come under DML:

- INSERT
- UPDATE
- DELETE
- SELECT

For example

INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Syntax:

```
INSERT  INTO TABLE_NAME
(col1, col2, col3,.... col N)
VALUES (value1, value2, value3, .... valueN);
```

3.Data Control Language

DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL:

- Grant
- Revoke

For example

Grant: It is used to give user access privileges to a database.

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

4.Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only. These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

Example:

Commit: Commit command is used to save all the transactions to the database.

Syntax:

```
COMMIT;
```

(b) What are the various data types available in SQL?

SQL data types: SQL data types define the nature of data that can be stored in database objects, like tables. Every table has columns, and each column has a name and data type associated with it. You can define both of these things during the creation of the table. Data types are used to represent the nature of the data that can be stored in the database table.

Data types mainly classified into three categories for every database.

- String Data types
- Numeric Data types
- Date and time Data types

1. String Data types

CHAR(Size)	It is used to specify a fixed length string that can contain numbers, letters, and special characters. Its size can be 0 to 255 characters. Default is 1.
VARCHAR(Size)	It is used to specify a variable length string that can contain numbers, letters, and special characters. Its size can be from 0 to 65535 characters.

TEXT(Size)	It holds a string that can contain a maximum length of 255 characters.
TINYTEXT	It holds a string with a maximum length of 255 characters.
MEDIUMTEXT	It holds a string with a maximum length of 16,777,215.
LONGTEXT	It holds a string with a maximum length of 4,294,967,295 characters.

2.Numeric Data types

BIT(Size)	It is used for a bit-value type. The number of bits per value is specified in size. Its size can be 1 to 64. The default value is 1.
INT(size)	It is used for the integer value. Its signed range varies from -2147483648 to 2147483647 and unsigned range varies from 0 to 4294967295. The size parameter specifies the max display width that is 255.
INTEGER(size)	It is equal to INT(size).
FLOAT(size, d)	It is used to specify a floating point number. Its size parameter specifies the total number of digits. The number of digits after the decimal point is specified by **d** parameter.
DOUBLE(size, d)	It is a normal size floating point number. Its size parameter specifies the total number of digits. The number of digits after the decimal is specified by d parameter.

3.Date and time Data types

DATE	It is used to specify date format YYYY-MM-DD. Its supported range is from '1000-01-01' to '9999-12-31'.
DATETIME(fsp)	It is used to specify date and time combination. Its format is YYYY-MM-DD hh:mm:ss. Its supported range is from '1000-01-01 00:00:00' to 9999-12-31 23:59:59'.
TIME(fsp)	It is used to specify the time format. Its format is hh:mm:ss. Its supported range is from '-838:59:59' to '838:59:59'
YEAR	It is used to specify a year in four-digit format. Values allowed in four digit format from 1901 to 2155, and 0000.

BOOLEAN Data Type

This data type stores the logical values. Oracle Boolean Data Type represents either TRUE or FALSE and mainly used in conditional statements. Values need not enclose within quotes while assigning for this data type.

(c) What are the commands in Data Manipulation Language?

DML Commands in SQL: Data Manipulation Language.

The DML commands in Structured Query Language change the data present in the SQL database. We can easily access, store, modify, update and delete the existing records from the database using DML commands.

Following are the four main DML commands in SQL:

1. SELECT Command
2. INSERT Command
3. UPDATE Command
4. DELETE Command

1.SELECT Command

SELECT is the most important data manipulation command in Structured Query Language. The SELECT command shows the records of the specified table. It also shows the particular record of a particular column by using the WHERE clause.

Syntax :

SELECT column_Name_1, column_Name_2, ….., column_Name_N **FROM** Name_of_table;

SQL>SELECT Emp_Id, Emp_Salary **FROM** Employee;

This SELECT statement displays all the values of **Emp_Salary** and **Emp_Id** column of **Employee** table:

Emp_Id	**Emp_Salary**
201	25000
202	45000

2.If we want to retrieve the data from all the columns of the table, we have to use the following SELECT command:

Syntax: SELECT * **FROM** table_name;

SELECT * **FROM** Student;

This SQL statement displays the following values of the student table:

Student_ID	**Student_Name**	**Student_Marks**
BCA1001	Abhay	85
BCA1002	Anuj	75

3.If you want to access all the records of those students whose marks is 80 from the above table, then you have to write the following DML command in SQL:

SELECT * FROM Student **WHERE** Stu_Marks = 80;
The above SQL query shows the following table in result:

Student_ID	**Student_Name**	**Student_Marks**
BCA1001	Abhay	80
BCA1003	Bheem	80

2.INSERT Command:
INSERT is another most important data manipulation command in Structured Query Language, which allows users to insert data in database tables.
We can insert values into a table in two ways.
Syntax:
Insert into tablename values(value1,value2,value3…….valuen);
By this method we can directly insert values into a existing table
a)For example if we want to insert values in Student table
SQL>insert into Student values(101,'ramesh',92,20);

tu_Id	**Stu_Name**	**Stu_Marks**	**Stu_Age**
101	Ramesh	92	20

b) Insert into table name values(&column1,&column2……&column n);
By this method it asks the values for columns and will enter values.

SQL> Insert into Student values(&stu_id,'&stu_name',&stu_marks',&stu_age);
Enter value for tu_id: 201
Enter value for stu_name: jatin
Enter value for stu_marks: 83
Enter value for stu_age: 19
We enter value and the record will be inserted.

Stu_Id	**Stu_Name**	**Stu_Marks**	**Stu_Age**
101	Ramesh	92	20
201	Jatin	83	19

3.UPDATE Command
UPDATE is another most important data manipulation command in Structured Query Language, which allows users to update or modify the existing data in database tables.
Syntax of UPDATE Command
UPDATE Table_name **SET** [column_name1= value_1, ….., column_nameN = value_N] **WHERE** condition
1.Update the value of a single field.

Product_Id	Product_Name	Product_Price	Product_Quantity
P101	Chips	20	20
P102	Chocolates	60	40

Suppose, you want to update the Product_Price of the product whose Product_Id is P102. To do this, you have to write the following DML UPDATE command:

SQL>UPDATE Product **SET** Product_Price = 80 **WHERE** Product_Id = 'P102' ;

2.Update the value of multiple fields of the database table.

Suppose, you want to update Stu_Marks and Stu_Age of that student whose Stu_Id is 103 and 202. To do this, you have to write the following DML Update command:

SQL>UPDATE Student **SET** Stu_Marks = 80, Stu_Age = 21 **WHERE** Stu_Id = 103 AND Stu_ Id = 202;

3.Update commond is used to perform calculations such as total and average.

SQL>update student set total=eng+tel+acc;

By this command the total column of student table will be updated and all the records get total marks.

4.DELETE COMMAND: is a DML command which allows SQL users to remove single or multiple existing records from the database tables.

This command of Data Manipulation Language does not delete the stored data permanently from the database. We use the WHERE clause with the DELETE command to select specific rows from the table.

1.Delete a single record from the table.

SQL>DELETE FROM Product WHERE Product_Id = 'P202' ;

The record will be deleted.

2.Delete the multiple records or rows from the database table.

Suppose, you want to delete the record of those students whose Marks is greater than 70. To do this, you have to write the following DML Update command:

SQL>DELETE FROM Student WHERE Stu_Marks > 70 ;

(Or)

(a) Explain the Commands available in Data Definition Language.

Data Definition Language(DDL):

Data Definition Language(DDL) is a subset of SQL and a part of DBMS(Database Management System). DDL consist of Commands to commands like CREATE, ALTER, TRUNCATE and DROP. These commands are used to create or modify the tables in SQL.

Data Definition Language(DDL) commands in DBMS are used to define the database objects. We can create, delete and alter tables using DDL commands. Since a database structure is designed by DDL commands, they are called Data Definition Languages (DDLs).

DDL Commands :

DDL commands are as follows.

1. Create
2. Alter
3. truncate

4. drop
5. Rename
6. Desc

1.CREATE :
This command is used to create a new table in SQL. The user has to give information like table name, column names, and their datatypes.

Syntax –
CREATE table tablename(column1 datatype,column2 datatype,column n…));

Example –
We need to create a table for storing Student information of a particular College. Create syntax would be as below.
CREATE TABLE Student_info(College_Id number(2),College_name varchar(30),Branch varchar(10));

2.ALTER :
This command is used to add, delete or change columns in the existing table. The user needs to know the existing table name and can do add, delete or modify tasks easily.

a)**To add a column to an existing table.**

Syntax
ALTER TABLE tablename ADD (columnname datatype);

Example –
In our Student_info table, we want to add a new column for CGPA. The syntax would be as below as follows.
ALTER TABLE Student_info ADD (CGPA number(9);

b) **To modify the column width**

Syntax
ALTER TABLE tablename modify(columnname datatype);

Example
Alter table student modify(Stu_name varchar(15));
By this we can increase the column witdth

3.TRUNCATE
This command is used to remove all rows from the table, but the structure of the table still exists.

Syntax
TRUNCATE TABLE table_name;

Example
The College Authority wants to remove the details of all students for new batches but wants to keep the table structure. The command they can use is as follows.
TRUNCATE TABLE Student_info;

4.DROP
This command is used to remove an existing table along with its structure from the Database.

Syntax
Syntax to drop an existing table.
DROP TABLE table_name;

Example
If the College Authority wants to change their Database by deleting the Student_info Table.

DROP TABLE Student_info;

5.RENAME:

It is possible to change name of table with or without data in it using simple RENAME command.

Syntax –

RENAME TABLE <Table Name> To <New_Table_Name>;

Example:

If you want to change the name of the table from Employee to Emp we can use **rename** command as

RENAME TABLE Employee To EMP;

6.DESC:

This command is used to describe the structure of the table.

Syntax –

Desc tablename:

Example:

Desc emp:

The structure of emp table will be displayed.

(b) What is a function? Write any five Aggregate functions.

Aggregate functions.

An aggregate function in SQL returns one value after calculating multiple values of a column. Aggregate functions are a vital component of database management systems. They allow us to perform calculations on large data sets quickly and efficiently.

For example, these functions generate statistical reports, perform financial analysis, and manage inventory levels. In addition, we can better understand the data we are working with by using aggregate functions. We can easily calculate the average price of all products in our inventory or find the total sales for a particular time. Without aggregate functions, we would need to manually sort through each data point, which would be time-consuming and error-prone.

Various types of SQL aggregate functions are:

- Count()
- Sum()
- Avg()
- Min()
- Max()

Stuname	Eng	Tel	Acc
A	30	50	40
B	20	30	30
C	40	30	20

For example we take the above Student table to apply Aggregate functions

1.COUNT() Function

The COUNT() function returns the number of rows in a database table.

*Syntax:**COUNT(*)***
Example:
SQL>select count(*) from student:
3

2.SUM() Function

The SUM() function returns the total sum of a numeric column.
*Syntax:**SUM()***
Example:
SQL>select sum(eng) from student;
90

3.AVG() Function

The AVG() function calculates the average of a set of values.
*Syntax:**AVG()***
Example:
SQL>select Avg(eng) from student;
30

4.MIN() Function

The MIN() aggregate function returns the lowest value (minimum) in a set of non-NULL values.
*Syntax:**MIN()***
Example:
SQL>select Min(eng) from student;
20

5.MAX() Function

The MAX() aggregate function returns the highest value (maximum) in a set of non-NULL values.
*Syntax:**AVG()***
Example:
SQL>select Max(eng) from student;
4

(c) Write about Selection operation and Projection operation.

Selection operation and Projection operation.

Projections and Selections are two unary operations in Relational Algebra and has practical applications in RDBMS (relational database management systems). In practical sense, Projection means selecting specific columns (attributes) from a table and Selection means filtering rows (tuples).

1.<u>Selection</u> :

This operation chooses the subset of tuples from the relation that satisfies the given condition mentioned in the syntax of selection.

Notation –

σ_c (R)

Here, ‘c’ is selection condition and ‘σ (sigma)’ is used to denote Select Operator**.**

Example-:

Select all the student of Team A :

$\sigma_{Team = 'A'}$ (Student)

Roll	Name	Department	Fees	Team
1	Bikash	CSE	22000	A
2	Josh	CSE	34000	A

2.Projection operation

It displays the specific column of a table. It is denoted by pie (∏). It is a vertical subset of the original relation. It eliminates duplicate tuples.

Syntax

$\prod_{regno}$(student)

Example

Consider the student table:

Regno	Branch	Section
1	CSE	A
2	ECE	B
3	CIVIL	B
4	IT	A

To display regno column of student table, we can use the following command −

$\prod_{regno}$(student)

Output

RegNo
1
2
3
4

9. (a) Write any three update quarries with examples

(b) Explain how to Create a Database Table.

Create a Database Table :Creating a basic table involves naming the table and defining its columns and each column's data type.

The SQL **CREATE TABLE** statement is used to create a new table.

Syntax

The basic syntax of the CREATE TABLE statement is as follows −

CREATE TABLE table_name(column1 datatype,   column2 datatype,   column3 datatype,
   columnN datatype,…   ));

Example

Create table student(regdno number(7),tsunami varchar2(10),age number(2));

Table will be created.

CREATE TABLE is the keyword telling the database system what you want to do. In this case, you want to create a new table. The unique name or identifier for the table follows the CREATE TABLE statement.

Then in brackets comes the list defining each column in the table and what sort of data type it is. The syntax becomes clearer with the following example.

A copy of an existing table can be created using a combination of the CREATE TABLE statement and the SELECT statement.

(c) Write the process of Inserting values into a Table

V UNIT

5.(a) List out various Data types of PL/SQL.

Data Types in PL/SQL are used to define how the data will be stored, handled, and treated by Oracle during the data storage and processing. Data types are associated with the specific storage format and range constraints. In Oracle, each value or constant is assigned with a data type.

Data types mainly classified into three categories for every database.

- String Data types
- Numeric Data types
- Date and time Data types

1. String Data types

CHAR(Size)	It is used to specify a fixed length string that can contain numbers, letters, and special characters. Its size can be 0 to 255 characters. Default is 1.
VARCHAR(Size)	It is used to specify a variable length string that can contain numbers, letters, and special characters. Its size can be from 0 to 65535 characters.
TEXT(Size)	It holds a string that can contain a maximum length of 255 characters.
TINYTEXT	It holds a string with a maximum length of 255 characters.
MEDIUMTEXT	It holds a string with a maximum length of 16,777,215.
LONGTEXT	It holds a string with a maximum length of 4,294,967,295 characters.
Varchar2 **LONG and LONGRAW**	This data type stores the string, but the length of the string is not fixed.The size restriction for this data type is 1-4000 bytes for table column size and 1-32767 bytes for variables. This data type is used to store large text or raw data up to the maximum of 2GB.

2.Numeric Data types

BIT(Size)	It is used for a bit-value type. The number of bits per value is specified in size. Its size can be 1 to 64. The default value is 1.
INT(size)	It is used for the integer value. Its signed range varies from -2147483648 to 2147483647 and unsigned range varies from 0 to 4294967295. The size parameter specifies the max display width that is 255.
INTEGER(size)	It is equal to INT(size).

FLOAT(size, d)	It is used to specify a floating point number. Its size parameter specifies the total number of digits. The number of digits after the decimal point is specified by **d** parameter.
DOUBLE(size, d)	It is a normal size floating point number. Its size parameter specifies the total number of digits. The number of digits after the decimal is specified by d parameter.

3.Date and time Data types

DATE	It is used to specify date format YYYY-MM-DD. Its supported range is from '1000-01-01' to '9999-12-31'.
DATETIME(fsp)	It is used to specify date and time combination. Its format is YYYY-MM-DD hh:mm:ss. Its supported range is from '1000-01-01 00:00:00' to 9999-12-31 23:59:59'.
TIME(fsp)	It is used to specify the time format. Its format is hh:mm:ss. Its supported range is from '-838:59:59' to '838:59:59'
YEAR	It is used to specify a year in four-digit format. Values allowed in four digit format from 1901 to 2155, and 0000.

BOOLEAN Data Type

This data type stores the logical values. Oracle Boolean Data Type represents either TRUE or FALSE and mainly used in conditional statements. Values need not enclose within quotes while assigning for this data type.

(b) Write about the structure of PL/SQL program.

PL/SQL: PL/SQL stands for "Procedural Language extensions to the Structured Query Language". SQL is a popular language for both querying and updating data in the relational database management systems (RDBMS). It combines the SQL with procedural language to create SQL Queries. It allows the programmers to instruct the compiler 'what to do' through SQL and 'How to do' through its procedural way.

- ➢ It is one statement specific language. But PL/SQL is block structure language
- ➢ Block is logical grouping of procedural and non procedural language statements
- ➢ SQL is a universal accepted language but PL./SQL only strict to oracle
- ➢ SQL is tool , but PL/SQL is utility

- PL/SQL is a collection of procedural &amp; non Procedural statements
- PL/SQL is used for to do data base activities in the server side
- PL/SQL Block can have any Number of statements which reduces network projects
- SQL is a data manipulation language and PL/SQL is data processing language
- It accepts only one parameter
- The parameter may be number, char, date

Structure of PL/SQL Program:

```
[
        Declare
-----------------
        Variable
declaration ];
[
        Begin
-----------
Executable statements ];
[
        Exception
--------------
        Error handling
statements ];
End ;
```

Sample Program

```
 begin
dbms_output.put_line('welcome to L/SQL');
 end;
/
```

Output:

welcome to PL/SQL

PL/SQL procedure successfully completed.

(c) Define Cursor? What are the Cursor processing commands?

Cursor:

A **cursor** is a pointer to this context area. PL/SQL controls the context area through a cursor. A cursor holds the rows (one or more) returned by a SQL statement. The set of rows the cursor holds is referred to as the **active set**.

You can name a cursor so that it could be referred to in a program to fetch and process the rows returned by the SQL statement, one at a time. There are two types of cursors −

- Implicit cursors
- Explicit cursors

Implicit Cursors

Implicit cursors are automatically created by Oracle whenever an SQL statement is executed, when there is no explicit cursor for the statement. Programmers cannot control the implicit cursors and the information in it.

Explicit cursors

Explicit cursors are defined by programmers to gain more control over the context area. It is defined in the declaration section of the PL/SQL block. It is created on a SELECT statement which returns more than one row.

Whenever a DML statement (INSERT, UPDATE and DELETE) is issued, an implicit cursor is associated with this statement. For INSERT operations, the cursor holds the data that needs to be inserted. For UPDATE and DELETE operations, the cursor identifies the rows that would be affected.

In PL/SQL, you can refer to the most recent implicit cursor as the **SQL cursor**, which always has attributes such as **%FOUND, %ISOPEN, %NOTFOUND**, and **%ROWCOUNT**.

S.No	Attribute & Description
1.%FOUND	Returns TRUE if an INSERT, UPDATE, or DELETE statement affected one or more rows or a SELECT INTO statement returned one or more rows. Otherwise, it returns FALSE.
2.**%NOTFOUND**	The logical opposite of %FOUND. It returns TRUE if an INSERT, UPDATE, or DELETE statement affected no rows, or a SELECT INTO statement returned no rows. Otherwise, it returns FALSE.
3.**%ISOPEN**	Always returns FALSE for implicit cursors, because Oracle closes the SQL cursor automatically after executing its associated SQL statement.
4.**%ROWCOUNT**	Returns the number of rows affected by an INSERT, UPDATE, or DELETE statement, or returned by a SELECT INTO statement.

1.Declare the cursor:

It defines the cursor with a name and the associated SELECT statement.

Syntax for explicit cursor declaration

CURSOR name IS
SELECT statement;

2) Open the cursor:

It is used to allocate memory for the cursor and make it easy to fetch the rows returned by the SQL statements into it.

Syntax for cursor open:

OPEN cursor_name;

3) Fetch the cursor:

It is used to access one row at a time. You can fetch rows from the above-opened cursor as follows:

Syntax for cursor fetch:

Syntax for cursor fetch:

FETCH cursor_name **INTO** variable_list;

4) Close the cursor:

It is used to release the allocated memory. The following syntax is used to close the above-opened cursors.

Syntax for cursor close:

Close cursor_name;

(Or)

a)Discuss the steps to create PL/SQL program.

PL/SQL: PL/SQL stands for "Procedural Language extensions to the Structured Query Language". SQL is a popular language for both querying and updating data in the relational database management systems (RDBMS). It combines the SQL with procedural language to create SQL Queries. It allows the programmers to instruct the compiler 'what to do' through SQL and 'How to do' through its procedural way.

- It is one statement specific language. But PL/SQL is block structure language
- Block is logical grouping of procedural and non procedural language statements
- PL/SQL is a collection of procedural & non Procedural statements
- PL/SQL is used for to do data base activities in the server side
- PL/SQL Block can have any Number of statements which reduces network projects

Steps to Create PL/SQL Program:

1. To create a PL/SQL program first click on oracle 100 icon
2. Type CONNECT and type user id: SYSTEM and Password manager.
3. You will receive a message Connected.
4. Now type Edit program name.SQL (For ex: Edit Sum.sql)
5. Note pad will be opened.
6. Then type the PL/SQL program to find sum of 2 numbers as shownbelow.

```
Declare
a number(5):=40;
b
number(5):=5
0;
c number(5);
Begin
c:=a+b;
dbms_output.put_line(‘sum’||c);
end;
/
```

1. Click on File → Save, File → Exit.
2. To execute the program, type @ Program name.SQL.(for ex: @Sum.sql)
3. To view the output we must give "SET SERVEROUTPUT ON".
4. We will receive a message PL/SQL procedure successfully executed, if there are noerrors in the program.
5. Finally we get the output

Output:

```
sum90
```

(b) Explain the elements of PL/SQL language.

PL/SQL language:

The PL/SQL programming language was developed by Oracle Corporation in the late 1980s as procedural extension language for SQL and the Oracle relational database.

Elements of PL/SQL:
Character Set
You write a PL/SQL program as lines of text using a specific set of characters. The PL/SQL character set includes

the upper- and lower-case letters A .. Z and a .. z
the numerals 0 .. 9
the symbols ( ) + - * / < > = ! ~ ^ ; : . ' @ % , " # $ & _ | { } ? [ ]
tabs, spaces, and carriage returns

PL/SQL is not case sensitive, so lower-case letters are equivalent to corresponding upper-case letters except within string and character literals.

Data types:

PL/SQL supports data types of SQL and also support composite data types that means they represent individually. For example char, varchar, long data, number, date, raw etc.

Lexical Units

A line of PL/SQL text contains groups of characters known as *lexical units*, which can be classified as follows:

delimiters (simple and compound symbols)
identifiers, which include reserved words
literals
comments

Delimiters

- A *delimiter* is a simple or compound symbol that has a special meaning to PL/SQL. For example, you use delimiters to represent arithmetic operations such as addition and subtraction. Simple symbols consist of one character. A list follows:

Symbol	**Meaning**
+	addition operator
%	attribute indicator
'	character string delimiter
.	component selector
/	division operator
(	expression or list delimiter
)	expression or list delimiter
:	host variable indicator
,	item separator
*	multiplication operator
"	quoted identifier delimiter
=	relational operator
<	relational operator
>	relational operator
@	remote access indicator
;	statement terminator
-	subtraction/negation operator

Identifiers

You use identifiers to name PL/SQL program items and units, which include constants, variables, exceptions, cursors, cursor variables, subprograms, and packages.

An identifier consists of a letter optionally followed by more letters, numerals, dollar signs, underscores, and number signs.

Literals

A *literal* is an explicit numeric, character, string, or Boolean value not represented by an identifier. The numeric literal 147 and the Boolean literal FALSE are examples.

Comments

The PL/SQL compiler ignores comments, but you should not. Adding comments to your program promotes readability and aids understanding. Generally, you use comments to describe the purpose and use of each code segment. PL/SQL supports two comment styles: single-line and multi-line.

Reserved Words

Some identifiers, called *reserved words*, have a special syntactic meaning to PL/SQL and so should not be redefined. For example, the words BEGIN and END, which bracket the executable part of a block or subprogram, are reserved.

(c) Define Procedure? How to create a Procedure with an example?

PL/SQL subprograms are named PL/SQL blocks that can be invoked with a set of parameters. PL/SQL provides two kinds of subprograms −

- **Functions** − These subprograms return a single value; mainly used to compute and return a value.
- **Procedures** − These subprograms do not return a value directly; mainly used to perform an action.

PROCEDURE:

1. It is a group of elements of perform specific task
2. Procedure should not return any value
3. It returns one or more values to the calling environment.
4. It allows to perform DML operation

Creating a Procedure:

A procedure is created with the **CREATE OR REPLACE PROCEDURE** statement. The simplified syntax for the CREATE OR REPLACE PROCEDURE statement is as follows −

```
CREATE [OR REPLACE] PROCEDURE procedure_name
[(parameter_name [IN | OUT | IN OUT] type [, ...])]
{IS | AS}
BEGIN
  < procedure_body >
END procedure_name;
```

Where,

- *procedure-name* specifies the name of the procedure.
- [OR REPLACE] option allows the modification of an existing procedure.
- The optional parameter list contains name, mode and types of the parameters. **IN** represents the value that will be passed from outside and OUT represents the parameter that will be used to return a value outside of the procedure.
- *procedure-body* contains the executable part.
- The AS keyword is used instead of the IS keyword for creating a standalone procedure.

Example

The following example creates a simple procedure that displays the string 'Hello World!' on the screen when executed.

```
CREATE OR REPLACE PROCEDURE greetings
AS
BEGIN
```

```
   dbms_output.put_line('Hello World!');
END;
/
```

When the above code is executed using the SQL prompt, it will produce the following result −
Procedure created.

5 Marks

10.(a) Write about merits of PL/SQL program.

PL/SQL language:

PL/SQL: The PL/SQL programming language was developed by Oracle Corporation in the late 1980s as procedural extension language for SQL and the Oracle relational database.

1. It is a standard database language and PL/SQL is strongly integrated with SQL. PL/SQL supports both static and also dynamic SQL.
2. It allows sending an entire block of statements to the database at one time. It reduces network traffic and also provides high performance for the applications.
3. It gives high productivity to programmers as it can query, transform and also update the data in the database.
4. This is said to save time on the design and also the debugging by strong features, like the exception handling, encapsulation, data hiding and also object-oriented data types.
5. Applications that are written in PL/SQL languages are portable.
6. It provides high security level.
7. It also provides access to the predefined SQL packages.
8. It also supports for Object-oriented programming.
9. It provides support for developing web applications and server pages.

(b) Write the advantages of Functions

Functions: Function can be used as a part of SQL expression i.e. we can use them with select/update/merge commands. One most important characteristic of a function is that, unlike procedures, it must return a value.

Problem solving: Functions act like new expressions and operators. You can use top-down design and the stepwise refinement approach to problem solving.

Use them in multiple locations: The main purpose of functions is to replicate the common task easily. We can build functions one time and can use them in multiple locations based on our needs.

Used to perform data operations: Functions are methods used to perform data operations. SQL has many in-built functions used to perform string concatenations, mathematical calculations etc.

Early output: Functions can very easily manipulate output for group of rows.

Use of date formats: Functions can alter date formats for display.

Handle variety of data: Functions can manipulate character as well as numeric type of data.

(c) Discuss about types of Triggers.

Trigger: A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

Types of triggers:

1. **Before Trigger:** BEFORE trigger execute before the triggering DML statement (INSERT, UPDATE, DELETE) execute. Triggering SQL statement is may or may not execute, depending on the BEFORE trigger conditions block.
2. **After Trigger:** AFTER trigger execute after the triggering DML statement (INSERT,UPDATE,DELETE) executed. Triggering SQL statement is execute as soon as followed by the code of trigger before performing Database operation.
3. **Row Level Trigger:** ROW trigger fire for each and every record which are performing INSERT, UPDATE, DELETE from the database table. If row deleting is define as trigger event, when trigger file, deletes the five rows each times from the table.
4. **Statement Level Trigger:** Statement trigger fire only once for each statement. If row deleting is define as trigger event, when trigger file, deletes the five rows at once from the table.

5.Instead-of Trigger

This is a type of trigger which enables you to stop and redirect the performance of a DML statement. Often this type of trigger helps you in managing the way you write to non-Adaptable views.