ASSIGNMENT - 01

UNIT - 1

Overwiew of Database Management System.

1. Explain different levels of data abstractions with diagram and example.

Ans Data abstraction is one of the most important concept in DBMS. Data abstraction is the process of hiding unwanted and irrelevant details from the end user.

→ It help to store information in such a way that the end user can access data which is necessary, the user will not be able to see what data is stored or how it is stored in a database.

There are mainly 3 levels of data abstraction:-

1. Physical level.
2. Logical level
3. View level

1. Physical level.

This is the lowest level of data abstraction.

→ It tells us how the data is actually stored in memory. Access method like sequential or random access and file organization methods are used for the same.

→ Developers or database application programmers decide how to store data in the database. It is comple x to understand.

Example of physical level abstraction would be sequential file organization due to the continuous storage of records. While in indexed file organization, we can access the records with the help of indexes.

2. Logical Level.

The logical level is the next higher level or intermediate level.

→ It explains what data is stored in the data base and how those data is stored are related.

→ It seeks to explain the complete or entire data by describing what tables should be constructed and what the linkages between those tables should be

→ It is less complex than the physical level.

Example :- The logical level in DBMS is used for representing entities and relationships among the data stored. Defining tables and their attributes and specifying relationship between them. A table named 'class' may have different attributes like student_name, Roll_no, student. ID and Marks.

3. View Level.

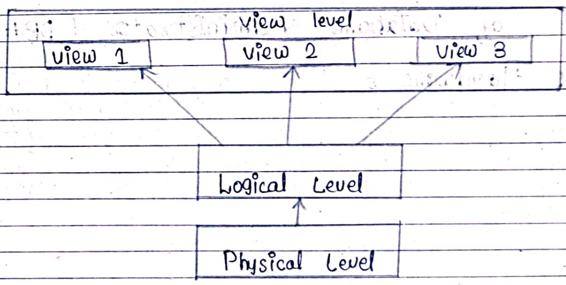
This is the top level. There are various views at the view level, with each view defining only a portion of the total data.

→ It also facilitates user engagement by providing a variety of views or numerous views of a single database.

→ All user have access to the view level.

→ This is the easiest & most simple level.

Example :- Interacting with a system using a graphical user interface (GUI) to access an applications features. Here GUI is the view level and the user does not know how and what data is exactly stored i.e hiding the details from the users.



2. What is DBA ? what are the roles and responsibilities of DBA.

Ans A Database Administrator (DBA) is an individual or person responsible for controlling, maintaining, coordinating and operating a database management system.

→ Managing, securing and taking care of the database systems is a prime responsibility.

→ They are responsible and in charge of authorizing access to the database, coordinating, capacity, planning, installation and monitoring uses, and acquiring and gathering software and hardware resources as and when needed.

→ They are overall commanders of the Database system.

Role of Database Administrator (DBA)

1) Decides Hardware :- They decide on economical hardware based on cost, performance and efficiency of hardware, and best suits the organization. It is hardware that is an interface between end users and the database.

2) Manages Data Integrity and Security :-

Data integrity needs to be checked and managed accurately as it protects and restricts data from unauthorized use. DBA eyes on relationships within data to maintain data integrity.

3) Database Accessibility :-

Database Administrator is solely responsible for giving permission to access data available in the database. It also makes sure who has the right to change the content.

4) Database Design :-

DBA is held responsible and accountable for logical, physical design, external model design, and integrity and security control.

5) Database Implementation :-

DBA implements database Management System and checks database loading at the time of its implementation.

6) Query Processing Performance :-

DBA enhances query processing by improving speed, performance and accuracy.

7) Tuning Database Performance :-

If the user is not able to get data speedily and accurately then it may lose organizations business. So by tuning SQL commands DBA can enhance the performance of the database.

8) Data Handling

9) Software Installation & Maintenance. (10) Backup Recovery.

Responsibilities of DBA:

- Responsible for designing overall database scheme (tables & fields)

- To select and install database software & Hardware.

- Responsible for deciding on access methods, and data storage.

- DBA selects appropriate DBMS software like oracle, SQL server or MYSQL

- Used in designing recovery ~~produce~~ procedures

- DBA decides the user access level and ~~seq~~ security checks for accessing, modifying or manipulating data.

- DBA responsible for specifying various techniques for monitoring the database performance.

- DBA responsible for operation managements.

Q3. What is Database Management System? What is the need of Database Management System. Advantages & Disadvantages of DBMS.

Ans Database management system is a software which is used to manage the database. for example : MySQL, 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, creating a table in the database and many more.

→ It provides protection and security to the database. In the case of multiple users, it also maintains data consistency.

→ It organizes data into tables, views, schemas and reports providing a structured approach to data management.

Need of DBMS

1) Data Organization.

DBMS helps to organize data in a structured way, which makes it easier to store, retrieve, and manage.

2) Data Integrity

DBMS ensures that the data stored in the database is accurate and consistent.

3) Data Security

DBMS helps to protect data from unauthorized access, manipulation and theft.

4) Data Backup

DBMS protects data against loss by creating copies of files stored in a database.

5) Data Sharing

DBMS enables users to access the database securely from any location.

6) Concurrency control

DBMS ensures multiple users can access the database simultaneously without conflicts.

7) Scalability

DBMS allows users to insert a lot of data into a database easily and access it quickly.

~~8)~~

ADVANTAGES of DBMS

1) Data Sharing

2) Backup & Recovery

3) Data Security

4) concurrent access

5) Data Integrity

6) Data Organization.

DISADVANTAGES of DBMS

1) Complexity :- It can be complex to set up and maintain requiring specialized knowledge and skills.

2) Performance Overhead :-

The use of a DBMS can add overhead to the performance of an application, especially in cases where high levels of concurrency are required.

3) Cost

4) Limited Use Cases :- Not all cases are suitable for a DBMS some solutions don't need high reliability or security and may be better served by other types of data storage.

Applications of DBMS

1. Enterprise Information. :-

2. Banking & finance Sector. :-

3. University :- Student course, student grade, staff

4. Airlines :- Reservations & Schedules

5. Telecommunications : Prepaid, postpaid bills maintenance

6. E- Commerce
7. Social Media.

Q4. Differentiate between file system and DBMS

Ans

Basis	DBMS	File System
Meaning	DBMS is a collection of data. In DBMS, the user is not required to write the procedure.	The file system is a collection of data. In this system the user has to write the procedure for managing database
Sharing of Data	Due to centralized, data sharing is easy.	Data is distributed in many files and it may be of different format so it is not easy.
Data Abstraction	DBMS gives an abstract view of data that hides the details	The file system provides the detail of the data representation and storage of data.
Security & Protection	DBMS provides a good protection.	It is not easy to protect a file under file system.
Recovery Mechanism.	DBMS provides a crash recovery mechanism i.e DBMS protects the user from system failure.	It doesn't have a crash mechanism, i.e if the system crashes, then the content of the file will be lost.

Cost	DBMS is expensive to design	The file system is cheaper to design.
Structure	complex	Simple.
Example	Oracle, SQL Server	Cobol, C++ etc
Redundancy	The redundancy of data is low	The redundancy of data is greater.
Data Inconsistency	DBMS has low data inconsistency	File system has higher data inconsistancy
Concurrency	It doesn't offer concurrency	It provide a concurrency facility
Use Cases	Suitable for simple data storage and retrieval tasks	Suitable for complex data management tasks such as e-commerce, finance

Q5. Draw and explain 3 levels architectures of DBMS

Ans A database stores a lot of critical information to access data quickly and securely. Hence it is important to select the correct architecture for efficient data management

→ DBMS architecture is crucial for efficient data management and system performance. It helps users to get their requests done while connecting to the database.

There are several types of DBMS Architecture that we use according to the usage requirements. They are given below:-

1) 1-Tier Architecture

2) 2- Tier Architecture

3) 3- Tier Architecture

1-Tier Architecture.

In 1-Tier Architecture the database is directly available to the user, the user can directly sit on the DBMS and use it that is, the client, server, and Database are all present on the same machine.

⇒ This setup is simple and is often used in personal or standalone applications where the user interacts directly with the database.

Ex- A microsoft Excel spreadsheet is a great example of one-tier architecture.

The user directly interacts with the applications performs operations like calculations or data entry and stores data locally on the same machine.

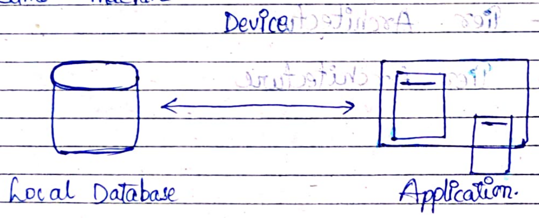


fig 1-Tier Architecture.

ADVANTAGES

1) Simple Architecture
2) Cost - Effective
3) Easy to Implement.
4

2- Tier Architecture

The 2-tier architecture is similar to a basic client-server model. The application at the client and directly communicates with the database on the server side.

→ This side is responsible for providing query processing and transaction management functionalities

→ On the client side, the user interfaces and application programs are run. The application on the client side establishes a connection with the server side to communicate with the DBMS

Ex- A library Management system used in schools or small organizations is a classic example of two tier architecture.

⇒ This intermediate layer acts as a medium for the exchange of partially processed data b/w the server and the client.

⇒ This type of architecture is used in the case of large web applications.

Ex - E - commerce store

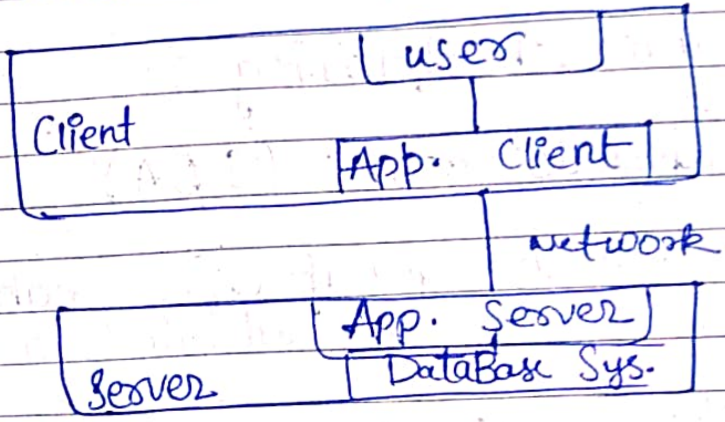


fig :- 3-tier Architecture

ADVANTAGES

1. Enhanched scalability
2. Data Integrity
3. Security

DISADVANTAGES
1. More Complex
2. Difficult to Interact.

Q6. Write four different types of database system users.

→ Every person who uses a database and takes advantages of it is called a DBMS user. There are many users in a DBMS who have their own specific tasks such as some users manage and create the database, some define the database.

DBMS users are the following :-

1.) Database Administrator (DBA)

DBA is a person or group who makes any changes to the database such as update, delete and create.

→ These are the users who are very familiar with the database. They have complete control over the database. DBA is responsible for any type of action in the DATABASE

The following are the functions of a DBA :-

i) It defines the schema and also controls all three levels of the database.

ii) It provides security to the database and allows only authorized users to access and modify the database.

iii) It has an account. which is called super user account.

iv) It repairs damage caused by software and hardware failure.

v) It also provides technical support to other users.

2) Database Designers.

Database designers are users who design the structure of the database. This includes tables, indexes, views, constraints, triggers and stored procedures.

→ They identify the data stored in the database and choose the right structure to represent this data. Database designers have to communicate with database users to understand their requirements.

3) End Users / Naive Users.

These users are unsophisticated users and they do not have any knowledge about databases but these users use database applications every day in their. daily life.

→ These are the end users who interact with the database with the help of application interface. They do not know about the designing access mechanism and working of the database and they just use the system to complete the task.

for Example- Users who book railway tickets are naive users. A bank clerk is also a naive user, he does not have knowledge of DBMS but he uses the database and completes his work.

4. Application Programs

These users are programmers who write code for application programs in back end.

→ These are the users who write programs for the database. These programs are usually written in C, Java, C++, Javascript, PHP and other general purpose programming language. And these applications programs are used to perform different types of tasks.

UNIT - 2

1. Explain different data models in detail with diagram.

→ Data model gives us an idea that how the final system will look like after its complete implementation.

→ It defines the data elements and the relationship b/w the data elements.

→ Data model are used to show how data is stored, connected, accessed and updated in the DBMS.

Some of the data models in DBMS are :-

1) Hierarchical Model.

Hierarchical model was the first DBMS model. This model ~~is~~ organises the data in the hierarchical tree structure.

→ The hierarchy starts from the root which has root data and then it expands in the form of a tree adding child node to the parent node.

→ This model easily represents some of the real-world relationships like food recipes, sitemap of a website etc.

Ex- We can represent the relationship b/w the shoes present on a shopping website.

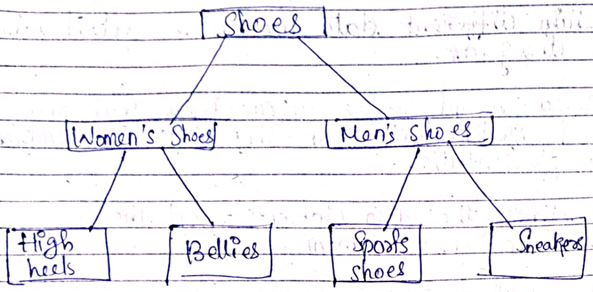


fig. Hierarchical model.

Advantags :-

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

Dis advantages :-

- Complex relationships are not supported.
- If the parent node is deleted then the child node is automatically deleted.

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 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 :- The node student has two parents i.e CSE Department and Library. This was earlier not possible in the hierarchical model.

child node also having multiple parent

He There is only one parent and multiple child.

Net Multiple child have multiple parent

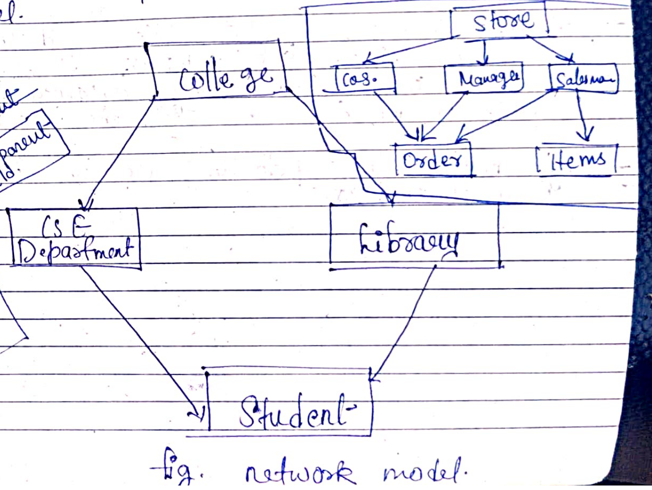


fig. network model.

Advantages.

- The data can be accessed faster as compared to the hierarchical model. This is because the data is more related in the network model; and there can be more than one path to reach a particular node. So the data can be accessed in many ways.

- As there is a parent-child relationship so data integrity is present. Any change in parent record is reflected in the child record.

Disadvantages

- As more and more relationships need to be handled the system might get complex. So, a user ~~must~~ ~~might~~ ~~got~~ must be having detailed knowledge of the model to work with the model.

- Any change like updation, deletion, insertion is very complex.

8. Relational Model

Relational model is the most widely used model. In this model, the data is maintained in the form of a two-dimensional table. All the information is stored in the form of row and columns. The basic structure of a relational model is tables.

So, the tables are also called relations in the relational model.

Example :- Employee table.

Name	Address	Salary	Id
Ram	Dipka	10,000	99
Raju	Bilaspur	15,000	100
Rahul	korba	20,000	62
Ritik	Raipur	25,000	43

Advantages :-

- This model is more simple as compared to the network and hierarchical model.

- This model can be easily Scaled as we can add as many rows and columns we want.

- Easy to design implement, maintain & Uses.

Q2. Define Specialization and generalization with example.

→ Generalization.

In Enhanced Entity - Relationship (EER) diagrams, generalization is a bottom-up method used to combine lower - level entities into a higher - level object.

→ In generalization, an entity of a higher level can also combine with the entities of the lower level to form a further higher level entity.

→ Generalization is more like subclass and superclass system, but the only difference is the approach. Generalization uses the bottom- up approach.

Advantages

- Cut down on Redundancy : Cuts down on data duplication by combining related entities into a single entity.
- Simplifies Schema : Combines many things into a single, clearer schema.
- Enhances Data Organization : By cohesively presenting related entities, it make better organization possible.

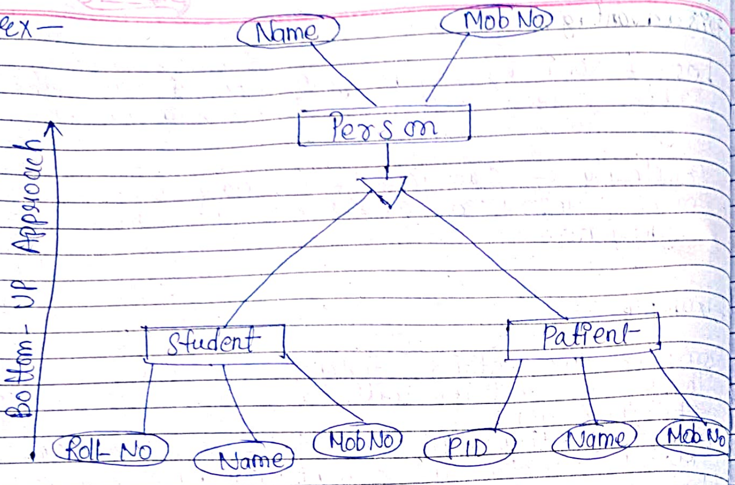
Disadvantages.

- Loss of Specificity : The generic entity may take center stage over the distinctive qualities of lower - level entities.
- Complexity of Querying : As data becomes more abstracted, queries may get more complicated.

Example :-

Consider two entities Student and Patient. These two entities will have some characteristics of their own.
Student entity will have Roll- No, Name and Mob-No while the patient will have PId, Name and Mob- No.
Now In this example the name and Mob- No of both patient and Student can be combined as a person to form one higher - level entity And this process is called generalization process.

ex—



Specialization.

- Specialization is a top - down approach, and it is opposite to Generalization. In specialization, one higher- level entity can be broken down into two lower level entities.

- Specialization is used to identify the Subset of an entity set that shares some distinguishing Characteristics.

- Normally the super class is defined first, the subclass and its related attributes are defined next, and relationship set are then added.

Advantages.

- Enhances Specificity :- By forming specialized subgroup, it is possible to depict things in more depth.

- Encourages Inheritance :- Relationships and characteristics from higher level entities are passed down to lower- level entities.

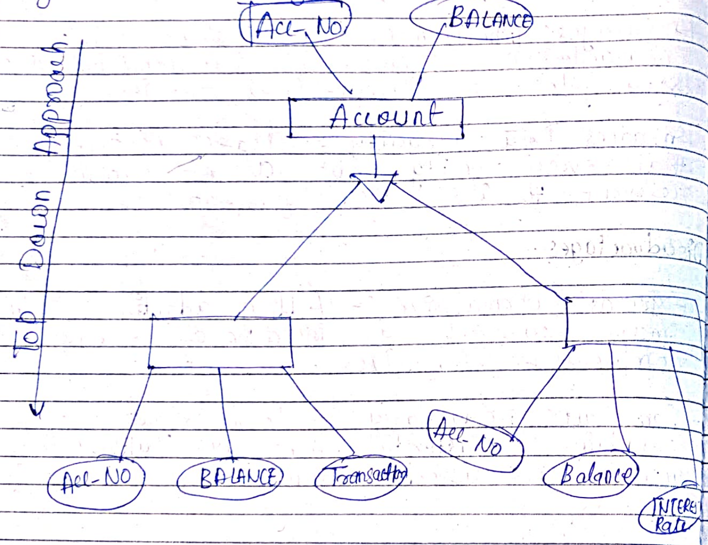
- Enhances Data Integrity :- Makes certain that every entity have distint qualities relevent to its area of expertise.

Disadvantages.

- Expands Schema Size :- Adding additional entities may lead to an increase of Schema's complexity and size.

- Can Cause Redundancy :- There might be certain characterstics that are duplicated across specialized entities.

Ex- Consider an entity Account. This will have some attributes consider them Acc-No and Balance. Account entity may have some other attributes like Current-Acc and Savings-Acc. Now Current-Acc may have Acc-No, Balance and Transactions while Saving-Acc may have Acc-No, Balance and Interest-Rate ∴ we can say that specialized entities inherits characteristics of higher level entity.



GENERALIZATION	SPECIALIZATION
Generalization works in Bottom - up approach.	Specialization works in Top - down approach.
In Generalization, size of schema gets reduced.	In specialization, size of schema gets increased.
Generalization is normally applied to group of entities.	We can apply Specialization to a single entity.
It can be defined as a process of creating groupings from various entity sets.	It can be defined as process of creating subgrouping within an entity set.
In this process, what actually happens is that it takes the union of two or more lower level entity sets to produce a higher-level entity sets.	It is a reverse of generalization. Specialization is a process of taking a subset of a higher level entity set to form a lower-level entity set.

Q3. Explain different types of relationship in DBMS.

→ Relationship is used to describe the relation b/w entities. It is represented by a diamond.

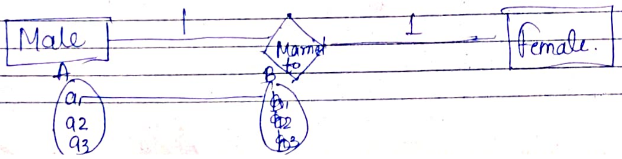
There are different types of relationship given below :-

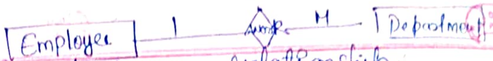
1. One to One
2. One to Many
3. Many to One
4. Many to Many.

1) One to one Relationship

When an instance of entity A is associated with an instance of entity B. Then it is called one to one relationship

For Ex - A male can marry only one female and a female can marry only one male. So we can call it one to one.

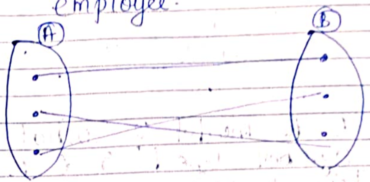




2) One to many relationship

This relationship occurs when one instance of entity A is associated with many instances of entity B.

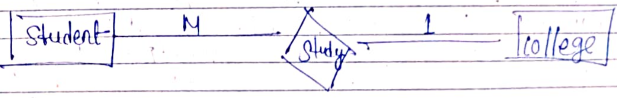
For example - An employee works in a department ; a department may have may employee.



3) Many to one relationship

This relationship occurs when many instances of entity A are associated with only one instance of entity B.

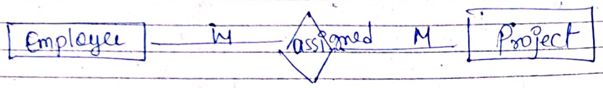
For ex - In real world, a student can study in one college and he cannot study in any other college simultaneously. Where as many students study in one college. So we will call it many to one.



4) Many to Many relationship

This relationship occurs when many instances of entity A are associated with many instances of entity B.

Ex - Many employees can work on many projects.



Q4. What is E-R model? What are the different symbol used in E-R model? Write with their meanings.

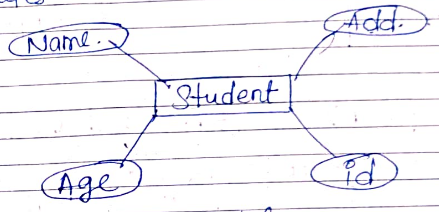
→ The full form of ER model is entity relationship model. It is a high level data model. This model is used to define data elements and relationships for a system.

→ In DBMS, ER model is a data model that describes the structure of the database with the help of a diagram.

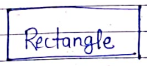
→ ER model is also called ER diagram because it presents entities in the form of a diagram and shows the relationship b/w entities.

→ This model was developed by Peter chen in 1976. The ER model is used to present the conceptual design for the database.

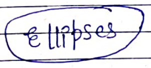
→ Ex - Suppose we have designed a database of a school. In this database student is an entity and address, name, id and age are its attributes.



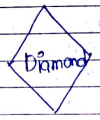
Symbols used in ER model.



It represent Entities in E-R model.

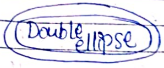


It represent attributes in E R model.

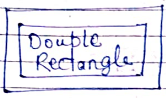


It represent relationships among Entities.

lines — It represent attributes to entities and entity sets with other relationship types.



It represent Multi-valued Attributes.



It represent a weak Entity.

Components of ER model

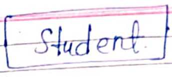
There are three main components of E-R model which are as follows :-

1. Entity
2. Relationship
3. Attributes

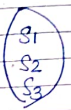
1) Entity :-

An Entity can be any person, place, car and real word object. In ER diagram an entity is represented by a rectangle. Entity must have an attributes and a unique key.

Eg :-



Entity Set :- An entity set is a group of entities of the same type that share similar properties.



Ex- In a school database, student, Teacher, Class and Course can be considered as entities. If a student is an entity then the data set of all students is called an entity set.

Types of Entity

1. Weak Entity

An Entity that is dependent on another entity is called a weak entity. It is represented by a double rectangle. A weak entity cannot be uniquely identified by its attributes.

2. Strong Entity.

An Entity that has a primary key is called a strong entity. A strong entity can be uniquely identified by its attributes.

2) Relationship

Relationship is used to describe the relation b/w entities. It is represented by a diamond.

for ex - Teacher teaches at school and soldier enrolls in a military.

Types of relationship.

1 One to One.

2 One to Many

3 Many to One

4. Many to Many

3) Attributes

Attribute is used to describe the property of an entity. It is represented by ellipse.

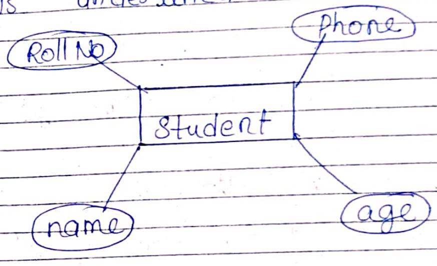
ex- Student in an entity and its subject name, subject code and gender are its attributes.

Types of attributes

i) key attributes.

The attribute that uniquely identifies each entity in the entity set is called the key attribute.

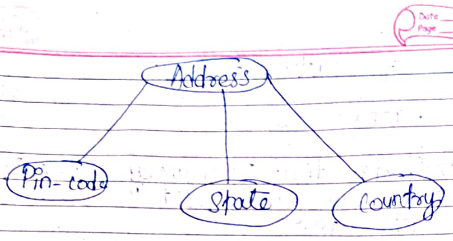
For ex - Roll-no is unique for each student. In the ER diagram, the text of the key attribute is underlined.



ii) Composite attribute

The attribute which is made up of a combination of other attributes is called a composite attribute. ~~for~~

For example - the address of a student is a composite attribute because it is made up of other attributes like pin-code, city, state and country.



iii) Single value attribute

The attribute which has only one value for a particular entity is called single value attribute.

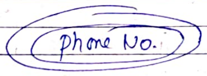
Ex - age of a person. is a single value attribute.

iv) Multivalued attribute.

An attribute that has multiple values for an entity is called a multivalued attribute

It is represented by a double ~~ovr~~ ellipse.

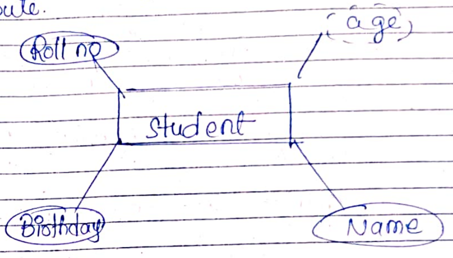
Ex - Colors for a car & phone number of employees.



v) Derived attribute.

The value of this type of attribute is derived from the value of another related attribute. It is represented by dashed ellipse.

Ex- The age of a person is derived from his birth date. So age is a derived attribute.



Advantages of E-R Model

- It is very easy to create this model. If you know the relationship b/w entities and attributes then you can easily create an ER diagram.

- In this model the structure of the database is presented in a diagram so it is easy to understand the relationship in it.

- This is a very effective communication tool for database designers.

- It is very easy to convert this model into other models. We can easily convert it into other models.

Disadvantages of E-R models.

- There is no industry standard for developing this model :- one develops notation cannot be understood by another developer.

- In this model some information or data gets lost or hidden.

- It is difficult to show data manipulation in this.

Q6. Explain Database Schema & Instances.

Schema

→ Database Schema is a logical representation of a database that shows how data is stored in the database.

→ Database schema is a design of the database that provides information about the entire database.

→ Database schema tells us how the data is stored in the database and what are the relationship b/w them.

→ Database schema consists of tables, views, fields, keys and relationships.

Purpose of Schema :-

1) The main purpose of the database schema is to identify the different rows and columns of each table.

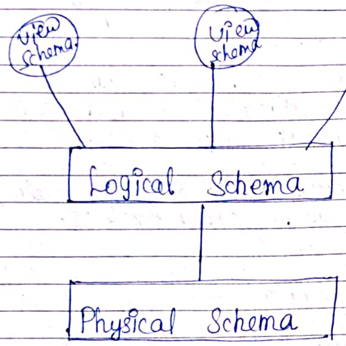
2) Along with this, the database schema also tells what is the relationship b/w the tables.

Types of Database Schema.

1. Physical Database Schema.

2. Logical Database Schema

3. View Database Schema



1. Physical Database Schema.

The physical database schema tells the user ~~How~~ how data is stored on a storage device.

The physical database schema defines how data is represented in the database and how the data is stored.

2. Logical Database Schema.

A logical database schema defines all the logical constraints that are applied to store data. A logical database schema is a conceptual model of data.

3. View Database Schema.

This is the view level design of the database. It provides information about the interaction b/w the end user and the database.

Instance

- Storing data or information in the database at a particular point in time is called an instance.

- Instance is also called "current state" or "database state".

- In other words "A group of data stored in a database at a particular point in time is called an instance.

Schema	Instance.
1. This is the complete description of the database.	It is a collection of data stored in a database at a particular point of time.
2. The schema is the same for the entire database.	Data in an instance can be updated, added and deleted.
3. It does not change constantly	It keeps changing constantly.
4. It provides information about the basic structure of the database	It is a group of data or information that is stored at a particular point in time.
5. Affects the entire database structure.	Affect only current state of data.

Q6. Draw the E-R diagram for library management system.

→ ER diagram is known as Entity-Relationship Diagram, it is used to analyze the structure of the database.

It shows relationships b/w entities and their attributes. An ER model provides a means of communication.

UNIT – 3

Query Language & Data Integrity.

SQL

→ The full name of SQL is structured Query language.

→ SQL is a programming language used to insert, delete, search and update data in a database.

→ SQL is a computer language used to store, monipulate, and retrieve data in a relational database.

→ The SQL language was created by IBM (International Business Machines Corporation) in 1970.

→ SQL is not a database system but it is a query language.

→ It supports various data types such as integer, text and dates.

→ It use popular database systems like MySQL, PostgreSQL, Oracle, SQL server, and SQLite

Advantages –

Data types –

1. CHAR (size) – It stores fix-length string of letter, number & special characters

2. VARCHAR – variable length string

3. BOOL – Stores TRUE or FALSE values.

4. INT (size) – Stores whole numbers.

5. FLOAT (size, d) – stores decimal foating point number.

6. DATE – stores a calerder date in the format YYYY – MM-DD

7. TIME – Stores time of day in HH : MM : SS

8. DATETIME – stores both date and time in one field. YYYY – MM- DD HH : MM :SS

9. YEAR – stores a year value in 4-digit format. YYYY.

Commands of SQL.

1. Data Defination Language (DDL)

→ DDL is called Data Defination Language. It is used to define the conceptual schema in the database.

→ DDL is used to change the structure of tables in the database. Such as creating a table, deleting it and altering a table.

Commonds of DDL :–

1. CREATE

→ It creates a database, tables, indexes, view.

Syntax –

```
CREATE TABLE
  table-name ( column 1 data-type,
               column 2 data-type, ...);
```

Ex–

```
CREATE TABLE Employees (
  ID INT PRIMARY KEY,
  NAME VARCHAR (100),
  Salary DECIMAL (10,2)
);
```

2. DROP :–

→ Deletes a database object permanently

Syntax – `DROP TABLE table-name;`

Ex – `DROP TABLE Employees;`

3. ALTER –

→ Modifies the structure of an existing table.

Syntax –

```
ALTER TABLE table-name
  ADD COLUMN column-name
  data-types ;
```

Ex –

```
ALTER TABLE Employees ADD Department
 VARCHAR (50);
```

4. TRUNCATE –

→ Removes all rows from a table without logging individual deletions.

Syntax – `TRUNCATE TABLE table-name ;`

Ex – `TRUNCATE TABLE Employees;`

Q) Data Manipulation Language (DML)

DML is called Data manipulation language. It is used to manipulation the database.

⇒ Using DML language the user can change the data present in the database.

⇒ DML language is used extensively in SQL database.

Commands of DML —

1. INSERT

Adds new data to a table.

Syntax —
```
INSERT INTO table_name
(column 1, column2, ...)
VALUES (value1, value2, ..);
```

Ex —
```
INSERT INTO Employees
( ID, Name, Salary)
VALUES ( 1, 'Tia', 5000);
```

2. UPDATE

Modifies existing data in a table.

Syntax —
```
UPDATE table-name
SET column 1 = value 1, column 2 = value 2
WHERE condition ;
```

Ex —
```
UPDATE Employees
SET Salary = 5500
WHERE ID = 1 ;
```

3. DELETE

Removes specific data from a table.

Syntax —
```
DELETE FROM table-name
WHERE condition ;
```

Ex —
```
DELETE FROM Employees
WHERE ID = 1 ;
```

4. CALL

Executes a stored procedure.

Syntax —
```
CALL procedure-name (arguments);
```

Ex —
```
CALL IncreaseSalaries ( ); // asuming a
                 stored procedure named increasedSalaries
```

5. EXPLAIN CALL

Display the execution plan of a stored procedure.

Syntax —
```
EXPLAIN CALL procedure-name (arguments);
```

Ex —
```
EXPLAIN CALL IncreaseSalaries ( );
```

6. LOCK

Locks a table or rows to prevent concurrent changes.

Syntax - LOCK table_name IN
lock_mode ;

Ex - LOCK TABLE Employees IN
EXCLUSIVE MODE ;

3) Transaction Control Language (TCL)

It manages all the transactions that are made in the database.

1. COMMIT - Save all changes made during the current transaction.

syntax - COMMIT ;

Ex - Changes को permanently save करता |
UPDATE Student set age - 22
WHERE ID = 1;
COMMIT ;

2. SAVEPOINT — Create a point in a transactions you can roll back tater.

syntax — SAVEPOINT savepoint_name ;

Ex - SAVEPOINT SAVE 1 ;

3. ROLLBACK — This command is used to undo changes that have been made in the current transaction.

Syntax - ROLLBACK To savepoint_~~save 1~~; name ;

Ex - ROLLBACK TO SAVE 1 ;

4. SET TRANSACTION - Set the properties of a transaction.

Syntax - SET TRANSACTION READ, WRITE ;
(characterstics);

5. SET CONSTRAINTS - This command is use for activate and decactivate the constraints

example - SET CONSTRAINTS ALL INMDIATE ;

4. Data Control Language (DCL)

This commands used to control access to data in a database. DCL is crucial for ensuring security and proper data management, especially in multi-user database environments.

1. GRANT - This command is used to give users access privileges to the database. This privelages can include the ability to select, insert, update, delete, ...

Syntax - GRANT privilege_name ON
object-name. TO user-name ;

Ex -
```
GRANT SELET, ~~INSERT~~
ON employees TO user 123;
```

2. REVOKE - This command is used to remove previously granted access privilages from a user.

Syntax -
```
REVOKE
privalege-name ON
object-name FROM
user-name;
```

Ex -
```
REVOKE SELECT
ON employees FROM user 123;
```

5. Data Query Language (DQL)

This command used primarily to query and retrieve data from existing database tables.

1. SELECT - The main command used in DQL SELECT retrives data from one or more tables.

Syntax -
```
SELECT
column1, column2, ... FROM
Table-name;
```
To select all column SELECT *:

Ex -
```
SELECT Name, Age, FROM
student WHERE Age = 18;
```

SQL Operator.

SQL operators are symbols or keywords that is used to perform operations on the given values, column, or conditions. SQL statements.

1. Arithmatic Operators.

This operator is used for calculations.

Operator	Meaning	Example.
+	Addition.	Salary + bonus
-	Subtraction	price - discount
*	Multiplication	quantity * rate
/	Division	total / count
%	Remainder	marks % 5

Ex -
```
SELECT Product-name, price, discount,
price - discount AS final - price
FROM products;
```

2. Comparison Operator.

It compares two values and return TRUE / FALSE

Operator	Meaning	Example
=	Equal to	age = 25
!= or <>	Not equal to	name != 'Rahul'
>	Greater than	marks > 60
>=	Greater than or equal to	salary >= 50000
<	Less than	prise < 100
<=	Less than or equal to	age <= 30

Ex-
```
SELECT * FROM employees WHERE
    department = 'sales';

SELECT * FROM products WHERE
    price > 100;
```

3. Logical Operators —

This operators are used to manipulate the conditions.

Operator	Meaning	Example.
AND	If both condition TRUE	age>18 AND age<60
OR	Any one condition TRUE	city = "Delhi" OR city = "Mumbai".
NOT	opposite the condition	NOT status = 'inactive'

Ex-
```
SELECT *
FROM employees
 WHERE department = 'Sales'.
   AND salary > 50000
   OR job-title = 'Manager';
   NOT city = 'New York';
```

4. Bitwise Operators.

ये integers पर bit level operation perform करता है।

Operator	Meaning	Example.
&	AND operation	a & b
|	OR	a · b
^	XOR	a ^ b
~	NOT	~ a
<<	left shift	a << 2
>>	right shift	a >> 1

5. Compound operators.

ये assignment के साथ साथ operation भी operform करता है

Operator	Meaning	Example.
+=	add and assign	a+ = 5
-=	subtract and assign	b- = 3
*=	multiply and assign	c* = 2
/=	run and assign.	d/ = 4
%=	mod and assign	e% = 2

6. Special operators —

Special operator are used to perform operations that go beyond the standard comparison operators (=, <, >, etc) These operators enhance the ability to filter, search, and compare data in flexiable and powerful ways.

1. ~~IN~~ BETWEEN

BETWEENoperator used to check if a value lies within a range.

Syntax -

```
SELECT * FROM table-name
WHERE column_name BETWEEN value1
    AND value2;
```

Example -

```
SELECT * FROM students
WHERE marks BETWEEN 60 AND 80;
```

2. IN

IN operator used to check if a value matches any value in a list.

Syntax -

```
SELECT * FROM table-name
WHERE column_name IN (value1, value2, ..);
```

Example -

```
SELECT * FROM employees
WHERE department IN ('HR', 'sales', 'Finance');
```

3. LIKE

LIKE operator used for pattern matching in strings

% matches any sequence of characters.

_ matches a single character.

Example 1 -

```
SELECT * FROM customers
WHERE name LIKE 'A%';
```

Returns all names starting with A

Example 2 -

```
SELECT * FROM products
WHERE name LIKE '_a%';
```

finds products where the second letter is a.

4. ANY and ALL

ANY operator, TRUE if any value in the subquery satisfies the condition.

Example -

```
SELECT * FROM products
WHERE price > ANY (SELECT price FROM
products WHERE category = 'Books');
```

ALL operator, TRUE if all values in the subquery satisfy the condition.

Example -

```
SELECT * FROM products
WHERE price > ALL (SELECT price FROM
products WHERE category = 'Toys');
```

5. EXISTS

EXISTS operator checks If a subquery returns any result

Example –
```
SELECT * FROM customers
WHERE EXISTS (SELECT * FROM orders
WHERE customer-id = orders.customer_id);
```

This returns all customers who have placed at least one order

6. UNIQUE

UNIQUE operator checks if a subquery returns only unique values. It is rarely used, and mostly replaced by constraints like DISTINCT or GROUP BY.

7. UNION

The UNION operator in SQL is used to combine the result of two or more SELECT queries into a single result set without duplicate rows.

- ALL SELECT quries must have same number of columns.
- The columns must have similar data type.
- The order of columns must be the same.
- By default, UNION removes duplicates.
- To include duplicate, use UNION ALL.

Syntax. –
```
SELECT column 1, column 2, ...
FROM table 1
UNION
SELECT column 1, column 2, ...
FROM table 2;
```

Example –
```
SELECT city FROM customers
UNION
SELECT city FROM suppliers;
```

This query returns a list of all unique cities from both customers & suppliers tables.

8. DISTINCT

DISTINCT operator is used in SQL to remove duplicate values from the result of a SELECT query. It reeturns only unique values.

Syntax –
```
SELECT DISTINCT column1, column2, ..
FROM table - name.
```

Id	name	city
1	Raj	Bihar
2	Priya	Mumbai
3	Raj	Mumbai
4	Neha	Delhi
5	Anita	Delhi

```
SELECT DISTINCT city
FROM students;
```

city
Delhi
Mumbai
Bihar.

* all duplicates are removed.

Constraints

यह rules होते हैं जो database table में data को सही तरीके से store करने के लिए लगाए जाते हैं।

Constraints help करते हैं —
- गलत data insert होने से रोकता है।
- duplicate data को avoid करने में।
- दो tables के बीच relation maintain करने में।

Types of Constraints.

1. PRIMARY KEY CONSTRAINTS.

A primary key is a column or a set of columns in a table that uniquely identifies each row in that table.

- It must be unique
- It cannot be NULL
- Each table can have only one primary key

EX-
```
CREATE TABLE students (
  student_id INT PRIMARY KEY,
   name VARCHAR (50),
   age INT
  );
```

2. FOREIGN KEY CONSTRAINTS.

A foreign key is a column in one table that refers to the primary key in another table.

It is used to link two related tables.

- To maintain referential integrity (i.e, valid relationships b/w tables)

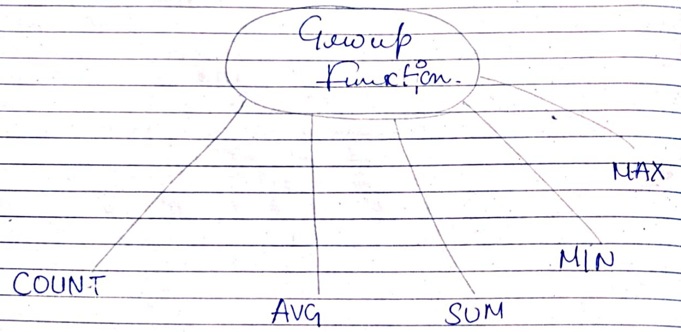
UNIT-4
SQL FUNCTION

SQL Group function

SQL मे group function (जिसे हम Aggregate functions भी कहते है) का use multiple rows पर calculation करने के लिए किया जाता है। और यह एक single value return करता है।

Uses – जब हमे किसी table के बहुत सारे records पर calculation करना होता है। जैसे count करना, average निकालना, maximum या maximum value, तब हम group functions का use करते है।

Types of Group function –



1. COUNT ( ) – count करता है.

→ यह function database table मे number of row को count करने मे use होता है। यह numeric or or non-numeric datatype मे work कर सकता है।

→ returns the number of rows in a set.

count (*) यह table सभी rows को count करता है और यह table मे duplicate और null value को भी count करता है।

Syntax – SELECT COUNT (*) FROM table-name.

Exampie –

id	name	age	marks
1	Riya	18	75
2	Amit	20	88
3	Neha	19	NULL
4	Mohit	20	65
5	Rina	18	

a) COUNT (*) – Count all rows, Including NULL's

```
SELECT COUNT (*) FROM students;
```

Output – 5

2. SUM function —

Sum function सभी selected column के sum value को calculate करने के लिए use होता है, यह केवल numeric filed (columns) पर काम करता है। ↓(INT, DECIMAL, FLOAT)

Syntax - SELECT SUM (entity) FROM table-name

Ex—

id	name	marks
1	Riya	75
2	Amil	88
3	Neha	NULL
4	Mohit	65
5	Rina	90

→ SELECT SUM(marks) FROM students ;
output - 318

SELECT SUM (marks) FROM students WHERE marks > 70 ;

Output 75 + 88+90 = 253.

3. AVG function.

Avg function सभी selected columns को numeric data types को AVG value calculate करके देता है।

syntax - SELECT AVG (column_name) FROM table_name ;

EX

id	name	marks
1	Riya	75
2	Anil	88
3	Neha	Null
4	Mohit	65
5	Rina	90

SELECT AVG (marks) FROM students ;

Output = 79.5

4. MAX function.

It return the largest value within the selected column.

- MAX function किसी column का maximum value find करने के लिए use होता है।
- यह function column की सभी selected value का largest value determine करता है।

Syntax — `SELECT MAX (column_name) FROM table_name`

Ex-

id	name	marks
1	Riya	75
2	Amit	88
3	Neha	null
4	Mohit	65
5	Rina	90

```
SELECT MAX (marks) FROM students ;
```

Output - 90

5. MIN function.

MIN function किसी column का minimum value find करने के लिए use होता है।

→ यह function column की सभी selected value का smallest value determine करता है।

Syntax — `SELECT MIN (column_name) FROM table_name;`

Ex

id	name	marks
1	Riya	75
2	Amit	88
3	Neha	NULL
4	Mohit	65
5	Rina	90

```
SELECT MIN(marks) FROM students ;
```

Output - 65

GROUP BY

The GROUP BY statement in SQL is used to arrange identical data into groups based on specified columns. If a particular column has the same values in multiple rows, the GROUP BY clause will group these rows together.

Syntax -
```
SELECT column_name,
AGGREGATE_FUNCTION ( column_name)
FROM table_name
GROUP BY column_name ;
```

id	product	quantity	price
1	Pen	10	5
2	Pencil	20	2
3	Pen	15	5
4	Eraser	30	3
5	Pencil	25	

SUM

```
SELECT product , SUM (quantity) AS
total_quantity
FROM sales
GROUP BY product ;
```

Output

product	total_quantity
Pen	25
Pencil	45
Eraser	30

AVG

```
SELECT product AVG (quantity) AS
average_quantity
FROM sales
GROUP BY product ;
```

Output

product	avg_quantity
pen	12.5
pencil	22.5
Eraser	

COUNT

```
SELECT product, COUNT (*) AS number_of_sales
FROM sales
GROUP BY product ;
```

Output

product	number_of_sales
pen	2
pencil	2
Eraser	1

MAX/MIN

```
SELECT product, MAX (quantity) AS
max_quantity, MIN (quantity) As min_quantity
FROM sales
GROUP BY product ;
```

product	max_quantity	min_quantity
Pen	15	10
pencil	25	20
Eraser.	30	

Having Clause.

The HAVING clause is used to filter the results after grouping them using the GROUP BY clause.

WHERE	filter before grouping
HAVING	filter after grouping

Syntax ~~id~~ ~~product~~ ~~quantity~~ ~~price~~.

```
SELECT column_name,
AGGREGATE_FUNCTION (column_name)
FROM table_name
GROUP BY column_name.
HAVING condition ;
```

Ex -

id	product	quantity	price
1	pen	10	5
2	pencil	20	2
3	pen	15	5
4	eraser	30	3
5	pencil	25	2

```
SELECT product, SUM (quantity) AS
  total_quantity
FROM sales
GROUP BY product
HAVING SUM (quantity) > 30 ;
```

Output

produt	total quantity
pencil	45

ORDER BY clause.

ORDER BY, clause का use SQL में rows को Sort करने के लिए किया जाता है ascending या decending order में।

जब हमें किसी Query के result को alphabetically numerically या logically sort करना होता है increasing या decreasing order में।

Syntax –
```
SELECT column1, column2...
FROM table_name
ORDER BY column1 [ASC|DESC], colum2
      (ASC|DESC) ...;
```

Ex –

id	name	marks
1	Rohan	75
2	Raj	90
3	Priya	85
4	Mohit	60
5	Riya	90

```
SELECT * FROM students
ORDER BY marks;
```

Output

id	name	marks
4	Mohit	60
1	Rohan	45
3	Priya	85
2	Raj	40
5	Riya	90

Join Operations.

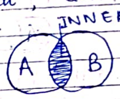
Join means to combine something.

A Join clause is used to combine data from two or more tables, based on a related column b/w them. It allows us to fetch data that is spread across multiple tables in a meaningful way.

Types of Join –

1. INNER JOIN.

- In inner join it returns only the matching rows b/w two tables.
- If there is no match, the row is not include in the result.



Syntax

```
SELECT table1.column1 . table2.column2
FROM table1
INNER JOIN table 2
ON table1.common_column = table2.common_column;
```

Student ID	Name
1001	A
1002	B
1003	C

Student ID	Department
1003	Mathematics
1004	CS
1001	ET

Student ID	Department Name
1003	Mathematics
1001	ET

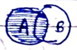
```
SELECT student. student-ID , Department. departmat.nam
fROM students As s
INNER JOIN department As d
ON s. student- id = d . student_id
```

2. OUTER JOIN

a) LEFT OUTER JOIN

In this it returns all rows from the left table, and matching rows from the right table.

- If no match NULL will be shown for the right table.



Syntax

```
SELECT table1. column 1 , table2. Column 2
FROM table 1
LEFT JOIN table 2
ON table1. common_column = table2. common_col
```

Ex –

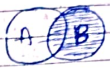
```
SELECT student. student_ID , department . department_name
FROM student AS s
LEFT ~~INNER~~ JOIN depasfment As d
ON s. student id = d. student id
```

Student ID	Department Name
1001	~~A~~ ET
1002	~~B~~
1003	Mathematics.

b.) RIGHT OUTER JOIN

It is the opposite of LEFT Join.

Returns all rows from the right table, & matching rows from the left table.



Syntax –

```
SELECT table 1 . column 1 , table 2 . column2
FROM table 1
RIGHT JOIN table 2
ON table1 . common- column = table2 . commn-col
```

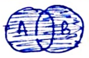
Ex -

```
SELECT student . student_id , depatmat . dep-
FROM student as s
RIGHT JOIN department as d
ON ~~students.~~ student_id = d. student_id.
```

student ID	department name
1003	Mathematics
NULL	CS
~~NULL~~ 1001	ET

c) FULL OUTER JOIN

- It combines the result of left join and right join.
- Returns all rows when there is a match in one of the tables.
- Missing matches will be filled with NULL.



Syntax -

```
SELECT table1. column 1 , table 2 . column 2
FROM table 1
OUTEER JOIN table 2
ON table 1. common_column = table2. common_colu
```

x -

```
SELECT student. student_id , department. departmt_name
FROM student AS s
OUTER JOIN department As d
ON s. student_id = d. student_id
```

Name	student id	department name
A	1001	ET
B	1002	NULL
C	1003	Mathematics
NULL	NULL	CS

NESTED QUERY

Nested query or also known as sub query is a query inside another query. The output of the inner query is used by the outer query. A nested query has two SELECT statements : one for the inner query & another for the outer query.

The inner query is called the subquery & the outer query is the main query.

- These query are written inside the WHERE , HAVING or FROM clauses.
- One query depends on the result of another.
- These are used when we want to filter data based on the result of another query.

~~Types of Nested Query~~

Syntax -

```
SELECT column 1 , column 2
FROM table 1
WHERE column 1 IN ( SELECT colum-1
                    FROM table 2
                    WHERE condition);
```