Module-5

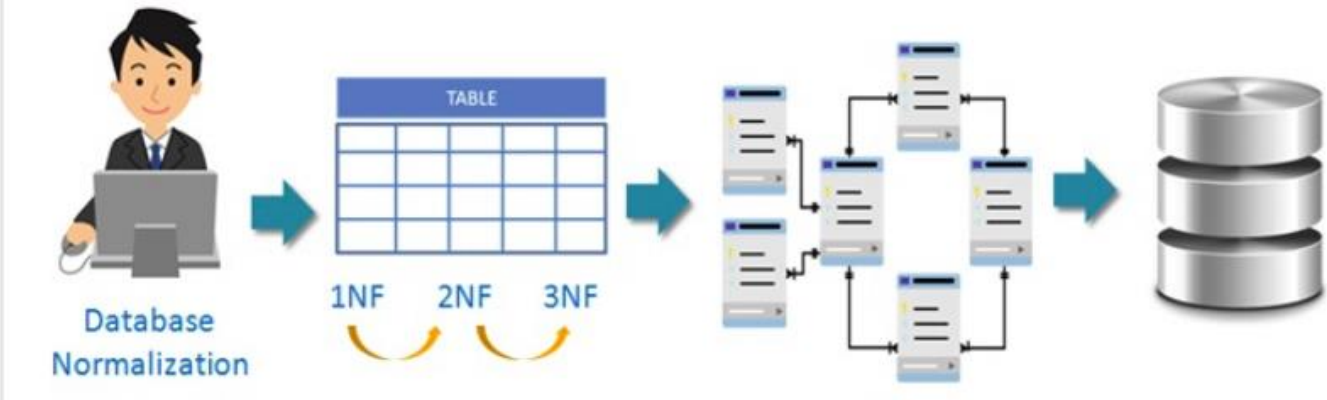
SE – Database

1)) what do you understand By Database?

A **database** is an organized collection of data stored in a computer system and usually controlled by a database management system (DBMS). The data in common databases is modelled in tables, making querying and processing efficient. Structured query language (SQL) is commonly used for data querying and writing.

2)) what is Normalization?

Normalization is the process to eliminate data redundancy and enhance data integrity in the table. Normalization also helps to organize the data in the database. It is a multi-step process that sets the data into tabular form and removes the duplicated data from the relational tables.



3)) what is Difference between DBMS and RDBMS?

RDBMS	DBMS
Data stored is in table format	Data stored is in the file format
Multiple data elements are accessible together	Individual access of data elements
Data in the form of a table are linked together	No connection between data
Support distributed database	No support for distributed database
Data is stored in a large amount	Data stored is a small quantity
RDBMS supports multiple users	DBMS supports a single user
The software and hardware requirements are higher	The software and hardware requirements are low
Example: Oracle, SQL Server.	Example: XML, Microsoft Access.

5)) what do you understand By Data Redundancy?

Data redundancy is when multiple copies of the same information are stored in more than one place at a time. This challenge plagues organizations of all sizes in all industries and leads to elevated storage costs, errors, and compromised analytics.

6)) what is DDL Interpreter?

It interprets the DDL (Data Definition Language) Instructions and stores the record in a data dictionary (in a table containing meta-data) Query Optimizer: It executes the DML Instructions and picks the lowest cost evaluation plan out of all the alternatives present.

7)) what is DML Compiler in SQL?

A DML (data manipulation language) refers to a computer programming language that allows you to add (insert), delete (delete), and alter (update) data in a database. A DML is typically a sublanguage of a larger database language like SQL, with the DML containing some of the language's operators.

8)) What is SQL Key Constraints writing an Example of SQL Key Constraints.

- SQL constraints are used to specify rules for the data in a table. Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.
- Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

  The following constraints are commonly used in SQL:

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Prevents actions that would destroy links between tables
- CHECK - Ensures that the values in a column satisfies a specific condition

- DEFAULT - Sets a default value for a column if no value is specified
- CREATE INDEX - Used to create and retrieve data from the database very quickly

NOT NULL Constraint:

```
CREATE TABLE Persons
(
ID int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255) NOT NULL,
Age int
);
```

Unique Constraint:

```
CREATE TABLE Persons
(
ID int NOT NULL UNIQUE,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Age int
);
```

Primary Key Constraint:

```
CREATE TABLE Persons
(
ID int NOT NULL PRIMARY KEY,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Age int
);
```

Foreign Key Constraint :

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES
Persons(PersonID)
);
```

Check Constraint :

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int CHECK (Age>=18)
);
```

Default Constraint :

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    City varchar(255) DEFAULT 'Sandnes'
);
```

9)) What is save Point? How to create a save Point writes a Query?

A SAVEPOINT is a point in a transaction in which you can roll the transaction back to a certain point without rolling back the entire transaction.

Syntax for Savepoint command:

SAVEPOINT SAVEPOINT_NAME;

This command is used only in the creation of SAVEPOINT among all the transactions.

10)) what is trigger and how to create a Trigger in SQL?

A trigger is a stored procedure in a database that 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 specific table columns are updated in simple words a trigger is a collection of SQL statements with particular names that are stored in system memory. It belongs to a specific class of stored procedures that are automatically invoked in response to database server events. Every trigger has a table attached to it.

QUERIES

3.	Get First_Name from employee table using Tom name “Employee Name”. SELECT First_Name FROM Employee WHERE "Employee Name" LIKE '%Tom%';
4.	Get FIRST_NAME, Joining Date, and Salary from employee table. SELECT FIRST_NAME, "Joining Date", Salary FROM Employee;
5.	Get all employee details from the employee table order by First_Name Ascending and Salary descending? SELECT * FROM Employee ORDER BY First_Name ASC, Salary DESC;
6.	Get employee details from employee table whose first name contains ‘J’. SELECT * FROM Employee WHERE First_Name LIKE '%J%';
7.	Get department wise maximum salary from employee table order by 8. Salary ascending? SELECT Department, MAX(Salary) AS MaxSalary FROM Employee GROUP BY Department ORDER BY MaxSalary ASC;
9.	Select first_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than 3000. SELECT Department, MAX(Salary) AS MaxSalary FROM Employee GROUP BY Department ORDER BY MaxSalary ASC;

10.	Create After Insert trigger on Employee table which insert records in viewtable
	CREATE TRIGGER after_employee_insert
	AFTER INSERT ON Employee
	FOR EACH ROW
	BEGIN
	INSERT INTO viewtable (employee_id, first_name, last_name, ...)
	VALUES (NEW.employee_id, NEW.first_name, NEW.last_name, ...);
	END;

12.	Retrieve the data from sales table. SELECT *FROM SALSESPERSON; SELECT * FROM Customer;
13.	All Customer name whose rating is more than 100. SELECT * FROM orders WHERE order_amount > 100;
14.	Names and cities of all salespeople in London with commission above 0.12. SELECT salesperson_name, city FROM salespeople WHERE city = 'London' AND commission > 0.12;
15.	All salespeople either in Barcelona or in London. SELECT salesperson_name, city FROM salespeople WHERE city = 'London' AND commission > 0.12;
16.	All salespeople with commission between 0.10 and 0.12. (Boundary 13.All Customer name whose rating is more than 100. Values should be excluded). SELECT * FROM salespeople WHERE commission > 0.10 AND commission < 0.12;
17.	All customers excluding those with rating <= 100 unless they are located in Rome. SELECT * FROM customers WHERE rating > 100 OR (rating <= 100 AND city = 'Rome');
18.	Write a SQL statement that displays all the information about all salespeople. SELECT salesman_id, name, city, commission FROM salespeople;
19.	From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord_no, ord_date, purch_amt. SELECT ord_no, ord_date, purch_amt FROM orders WHERE salesman_id = 5001;
20.	From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro_id, pro_name, pro_price, and pro_com. SELECT PRO_ID, PRO_NAME, PRO_PRICE, PRO_COM FROM item_mast WHERE PRO_PRICE BETWEEN 200.00 AND 600.00;
21.	From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg. SELECT AVG(PRO_PRICE) AS avg_price FROM item_mast WHERE PRO_COM = 16;
22.	From the following table, write a SQL query to display the pro_name as 'Item Name' and pro_priceas 'Price in Rs.' SELECT PRO_NAME AS 'Item Name', PRO_PRICE AS 'Price in Rs.' FROM item_mast;