MODULE 5

DATABASE

Ques 1: What do you understand By Database?
Ans: A database is a organized collection of information or data stored in a computer. A database is usually control by data management system called DBMS.

Ques 2. What is Normalization?
Ans: database normalization is the process of organizing the attributes of the database to reduce or eliminates data redundancy (having the same data but at different places

Ques 3. What is Difference between DBMS and RDBMS?
Ans:

DBMS	RDBMS
Dbms stored data in the form of files.	Rdbms stored data in the form of tables.
It deals with the small quantity of data.	It deals with the large amount of data.
Data element needs to be access individually.	Multiple data element can be access at same time.
Normalization is not present.	Normalization is present.

Ques 4: What is MF Cod Rule of RDBMS Systems?
Ans: These rules are made to ensure data intrigity, consistency,and usability. This set of rules basically signifies the characteristics and requirement database management system.

Ques 5: What do you understand By Data Redundancy?
Ans: Data redundancy means the occurrence of duplicate copies of similar data. It is done intentionally to keep the same piece of data at different places, or it occurs accidentally.

Ques 6: What is DDL Interpreter?
Ans: Data Definition Language (DDL) is used to create and modify the structure of objects in a database using predefined commands and a specific syntax. These

database objects include tables, sequences, locations, aliases, schemas and indexes.

Ques 7: What is DML Compiler in SQL?
Ans: full form of DML is 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.

Ques 8: What is SQL Key Constraints writing an Example of SQL Key Constraints?
Ans: In a database table, we can add rules to a column known as constraints. These rules control the data that can be stored in a column.

- NOT NULL - Ensures that a column cannot have a NULL value

  Example : CREATE TABLE studinfo (

  Student_id INT NOT NULL,

  Student_name VARCHAR(20) NOT NULL);

- UNIQUE - Ensures that all values in a column are different

  Example : CREATE TABLE studinfo (
  Student_id INT NOT NULL UNIQUE,
  Student_name VARCHAR(20) UNIQUE);

- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table

  Example : CREATE TABLE studinfo (
  Student_id INT PRIMARY KEY,
  Student_name VARCHAR(20) NOT NULL);

- FOREIGN KEY - Prevents actions that would destroy links between tables

Example : CREATE TABLE studinfo(
Student_id INT PRIMARY KEY,
College_id INT REFRENCE student(id);

- CHECK - Ensures that the values in a column satisfies a specific condition

  Example : CREATE TABLE orders (
  order_id INT PRIMARY KEY,
  amount INT CHECK (AMOUNT >=100)
  );
- DEFAULT - Sets a default value for a column if no value is specified

  Example : CREATE TABLE studinfo (
  Student_id INT PRIMARY KEY,
  Student_name VARCHAR(20) NOT NULL,
  Student_country VARCAR(20) DEFAULT 'INDIA');

- CREATE INDEX - Used to create and retrieve data from the database very quickly

  Example : CREATE TABLE studinfo (
  Student_id INT PRIMARY KEY,
  Student_name VARCHAR(20) NOT NULL,
  Student_country VARCAR(20));

  CREATE INDEX stud_index

  On studinfo(student_name);

Ques 9: What is save Point? How to create a save Point write a Query?
Ans : A savepoint is essentially a marker within a transaction. It allows you to rollback part of a transaction without aborting the entire thing. This is incredibly useful in large transactions where you want the ability to undo some changes without starting over from scratch.
Example: Using Savepoint and Rollback
Following is the table **class**,

Id	name
101	Priya
102	Hasti
103	Sweta

Lets use some SQL queries on the above table and see the results.

INSERT INTO class VALUES(104, ‘Aarti’);

UPDATE class SET name = ‘kinjal’ WHERE ID = ‘104’;

SAVEPOINT A;

INSERT INTO class VALUES(105, ‘ANKITA’);

SAVEPOINT B;

INSERT INTO class VALUES(106, ‘PRANJAL’);

SAVEPOINT C;

SELECT * FROM class;

Resultant data will look like these.

ID	NAME
101	Priya
102	Hasti
103	sweta
104	Aarti
105	Ankita
106	Pranjal

Now let's use the ROLLBACK command to roll back the state of data to the **savepoint B**.

ROLLBACK TO B;

SELECT * FROM class;

Now our **class** table will look like,

ID	NAME
101	Priya
102	Hasti
103	sweta
104	Aarti
105	Ankita

Now let's again use the ROLLBACK command to roll back the state of data to the **savepoint A**

Now the table will look like,

ID	NAME
101	Priya
102	Hasti
103	sweta
104	Aarti

So now you know how the ROLLBACK and SAVEPOINT works.

Ques 10:What is trigger and how to create a Trigger in SQL ?

Ans: A trigger is a set of SQL statements that reside in system memory with unique names. It is a specialized category of stored procedure that is called automatically when a database server event occurs. Each trigger is always associated with a table.

Syntax of Trigger

We can create a trigger in SQL Server by using the **CREATE TRIGGER** statement as follows:

CREATE **TRIGGER** .trigger_name
ON table_name
AFTER {**INSERT**, **UPDATE**, **DELETE**}
AS

{SQL_Statements}