**MODULE:4**

**DATABASE**

1. **What do you understand by Database?**

A database is a structured collection of data organized in a way that a computer program can quickly select and retrieve specific pieces of data. It can be anything from a simple list in a text file to a complex system with multiple tables, relationships, and data integrity constraints.

1. **What is Normalization?**

Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves breaking down large tables into smaller, related tables and defining relationships between them. The main goal is to eliminate data redundancy and dependency, ensuring that data is stored in the most efficient and organized manner.

1. **What is the difference between DBMS and RDBMS?**

**DBMS (Database Management System):** It is a software that manages databases. It provides an interface for interacting with the database, managing data, and ensuring data integrity.

**RDBMS (Relational Database Management System):** It is a type of DBMS that stores data in tables and establishes relationships between them. It enforces the relational model, ensuring consistency, integrity, and reliability of the data.

1. **What is the MF Cod Rule of RDBMS Systems?**

Codd's rule in DBMS also known as Codd's 12 rules/commandments is a set of thirteen rules (numbered 0 to 12) that define a database to be a correct Relational Database Management System (RDBMS). If a database follows Codd's 12 rules, it is called a True relational database management system.

1. **What do you understand by Data Redundancy?**

Data redundancy occurs when the same piece of data is stored in multiple places within a database. This redundancy can lead to inefficiencies, data inconsistency, and increased storage requirements.

1. **What is DDL Interpreter?**

DDL (Data Definition Language) is a subset of SQL used to define and manage database structures. A DDL interpreter is a component of a database management system that processes DDL statements. It performs operations such as creating, altering, or deleting database objects like tables, indexes, and views.

1. **What is DML Compiler in SQL?**

DML (Data Manipulation Language) is a subset of SQL used to manipulate data stored in a database. A DML compiler is a component that processes DML statements like SELECT, INSERT, UPDATE, and DELETE. It translates these statements into low-level instructions that the database engine can execute to perform the requested operations.

1. **What is SQL Key constraints? Provide an example.**

SQL Key Constraints are rules applied to columns in a table to enforce the integrity of the data. There are several types of key constraints:

**Primary Key Constraint:** Ensures that each row in a table is uniquely identified. Example:

CREATE TABLE Students (

StudentID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50)

);

**Foreign Key Constraint:** Establishes a link between two tables, enforcing referential integrity. Example:

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

ProductID INT,

FOREIGN KEY (StudentID) REFERENCES Students(StudentID)

);

1. **What is a Save Point? How to create a Save Point? Write a Query.**

A savepoint is a point within a transaction to which you can later roll back. It allows you to partially roll back a transaction to a specific point without affecting the entire transaction.

SAVEPOINT my\_savepoint;

1. **What is a Trigger, how to create a trigger in SQL?**

A trigger is a set of instructions that are automatically executed ("triggered") in response to certain events on a particular table or view. These events can be INSERT, UPDATE, DELETE, etc.

CREATE TRIGGER my\_trigger

AFTER INSERT ON MyTable

FOR EACH ROW

BEGIN

TRIGGER CODE

END;

**Task:**

1. Create Table Name : Student and Exam

TABLE QUERY:

CREATE TABLE STUDENT(

ROLL\_NO INT PRIMARY KEY AUTO\_INCREMENT,

NAME VARCHAR(50),

BRANCH VARCHAR(20)

);

CREATE TABLE EXAM (

ROLL\_NO INT,

S\_CODE VARCHAR(6),

MARKS INT,

P\_CODE VARCHAR(5),

FOREIGN KEY (ROLL\_NO) REFERENCES STUDENT(ROLL\_NO)

);

1. Create table given below

TABLE QUERY:

CREATE TABLE PERSONAL\_INFO(

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR(50),

ADDRESS VARCHAR(100),

CITY VARCHAR(40),

AGE INT

);

INSERT QUERY:

INSERT INTO `personal\_info` (`FIRST\_NAME`, `LAST\_NAME`, `ADDRESS`, `CITY`, `AGE`) VALUES ('MICKEY', 'MOUSE', '123 FANTASY WAY', 'ANAHEIM', '73'), ('BAT', 'MAN', '321 CAVERN AVE', 'GOTHAM', '54'), ('WONDER', 'WOMAN', '987 TRUTH WAY', 'PARADISE', '39'), ('DONALD', 'DUCK', '555 QUACK STREET', 'MALLARD', '65'), ('BUGS', 'BUNNY', '567 CARROT STREET', 'RASCAL', '58'), ('WILEY', 'COYOTE', '999 ACME WAY', 'CANYON', '61'), ('CAT', 'WOMAN', '234 PURRFECT STREET', 'HAIRBALL', '32'), ('TWEETY', 'BIRD', '543 NESTED STREET', 'ITOOTLAW', '28');

1. Create table given below: Employee and Incentive

TABLE:1

CREATE TABLE EMPLOYEE(

employee\_id INT PRIMARY KEY AUTO\_INCREMENT,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

salary BIGINT,

joining\_date DATETIME,

department VARCHAR(40)

);

INSERT QUERY:

INSERT INTO employee(EMPLOYEE\_ID,FIRST\_NAME,LAST\_NAME,SALARY,JOINING\_DATE,DEPARTMENT)

VALUES (1,'JOHN','ABRAHAM',1000000,01-01-13,'BANKING'),

(2,'MICHAEL','CLARKE',800000,01-01-13,'INSURANCE'),

(3,'ROY','THOMAS',700000,01-02-13,'BANKING'),

(4,'TOM','JOSE',600000,01-02-13,'INSURANCE'),

(5,'JERRY','PINTO',650000,01-02-13,'INSURANCE'),

(6,'PHILIP','MATTHEW',750000,01-01-13,'SERVICES'),

(7,'TESTNAME','123',650000,01-01-13,'SERVICES'),

(8,'TESTNAME2','LNAME',60000,01-02-13,'INSURANCE');

TABLE:2

CREATE TABLE INCENTIVE(

EMPLOYEE\_REF\_ID INT,

INCENTIVE\_DATE DATETIME,

INCENTIVE\_AMOUNT BIGINT

);

INSERT DATA:

INSERT INTO incentive(EMPLOYEE\_REF\_ID,INCENTIVE\_DATE,INCENTIVE\_AMOUNT)

VALUES(1,01-02-13,5000),

(2,01-02-13,3000),

(3,01-02-13,4000),

(1,01-01-13,4500),

(2,01-01-13,3500);

a) QUERY:

SELECT FIRST\_NAME AS EMPLOYEE\_NAME FROM EMPLOYEE;

b) QUERY:

SELECT FIRST\_NAME,JOINING\_DATE,SALARY

FROM employee;

c) QUERY:

SELECT FIRST\_NAME,SALARY

FROM employee

ORDER BY FIRST\_NAME ASC ,SALARY DESC;

d) QUERY:

SELECT \* FROM employee

WHERE FIRST\_NAME LIKE 'J%';

e) QUERY:

SELECT department, MAX(salary) AS max\_salary

FROM employee

ORDER BY max\_salary ASC;

f) QUERY:

SELECT E.FIRST\_NAME, I.INCENTIVE\_AMOUNT

FROM employee E

INNER JOIN incentive I

ON E.EMPLOYEE\_ID = I.EMPLOYEE\_REF\_ID

WHERE I.INCENTIVE\_AMOUNT > 3000;

1. Create table given below: Salesperson and Customer

Table:1

CREATE TABLE salesperson(

S\_no INT PRIMARY KEY AUTO\_INCREMENT,

s\_name VARCHAR(50),

city VARCHAR(20),

comm DOUBLE

);

Insert Query:

INSERT INTO customer

VALUES(1001,'Peel','London',.12);

INSERT INTO customer

VALUES(1002,'Serres','San Jose',.13);

INSERT INTO customer

VALUES(1004,'Motika','London',.11);

INSERT INTO customer

VALUES(1001,'Rafkin','Barcelona',.15);

INSERT INTO customer

VALUES(1003,'Axelrod','New York',.1);

Table:2

CREATE TABLE customer(

cnm INT PRIMARY KEY AUTO\_INCREMENT,

cname VARCHAR(50),

city VARCHAR(20),

rating INT,

S\_no INT,

FOREIGN KEY (S\_no) REFERENCES salesperson(S\_no)

);

Insert Query:

INSERT INTO customer

VALUES(201,'Hoffman','London',100,1001);

INSERT INTO customer

VALUES(202,'Giovanne','Roe',200,1003);

INSERT INTO customer

VALUES(203,'Liu','San Jose',300,1002);

INSERT INTO customer

VALUES(204,'Grass','Barcelona',100,1002);

INSERT INTO customer

VALUES(206,'Clemens','London',300,1007);

INSERT INTO customer

VALUES(207,'Pereira','Roe',100,1004);

b)SELECT s\_name,city

FROM salesperson

WHERE city='London' AND comm > 0.12;

c)SELECT s\_name FROM salesperson

WHERE city='Barcelona' OR city='London'

d)SELECT s\_name FROM salesperson

WHERE comm BETWEEN 0.10 and 0.12;

e)SELECT cname FROM customer

WHERE (rating<=100) OR (city='Rome');