**MODULE: 5 (Database)**

**Qus.1…What do you understand By Database**

Ans 🡪 DBMS stands for Data Base Management System.

Data + Management System

🡪 Database is a collection of inter-related data and Management System is a set of programs to store and retrieve those data

**Qus.2… What is Normalization?**

Ans 🡪 Normalization is the process of minimizing redundancy (duplicity) from a relation or set of relations.

🡪 Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations.

**Qus.3… What is Difference between DBMS and RDBMS?**

Ans 🡪 DBMS:- Database Management System

RDBMS:- Relational Database Management System

🡪 Relational Database Management System is type of Database Management System

**Qus.4…** **What do you understand By Data Redundancy?**

Ans 🡪 Normalization is the process of minimizing redundancy (duplicity) from a relation or set of relations.

🡪 Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations.

**Qus.5… What is DDL Interpreter?**

Ans 🡪 DDL :- Data Definition Language

🡪 We can use DDL for Create table, view a table ,modifies data of table and we can also Delete database and table using DDL

**Qus.6… What is DML Compiler in SQL?**

Ans 🡪 DML :- Data Manipulation Language

🡪 We can use to DML for Create a record , Modifies a record, and also Delete a record.

**Qus.7… What is SQL Key Constraints writing an Example of SQL Key Constraints**

Ans 🡪 There are 3 type of key in SQL

(1) primary key:- A primary key is a column of table which uniquely identifies each tuple (row) in that table.

(2) unique key :- Unique key constraints also identifies an individual table uniquely in a relation or table.

(3) foreign key:- When, "one" table's primary key field is added to a related "many" table in order to create the common field which relates the two tables, it is called a foreign key in the "many" table.

**Qus.8… What is save Point? How to create a save Point write a Query?**

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

🡪 The syntax for a SAVEPOINT command is as shown below.

○ SAVEPOINT SAVEPOINT\_NAME;

**Qus.9… What is trigger and how to create a Trigger in SQL?**

Ans 🡪 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.

Syntax: create trigger [trigger\_name] [before | after] {insert | update | delete} on [table\_name] [for each row] [trigger\_body]

**TASK 1. Create Table Name : Student and Exam**

Ans. 🡪

Make a DATABASE:-

* Make a table 1 (STUDENT):-

🡪 code:-

CREATE TABLE STUDENT

(

ROLL\_NO INT PRIMARY KEY AUTO\_INCREMENT,

STUDENT\_NAME VARCHAR ( 50 ),

BRANCH VARCHAR ( 50 ),

)

INSERT VALUE:-

INSEAT TABLE STUDENT(ROLL\_NO,”NAME”,”BRANCH”)VALUE(1,”JAY”,”COMPUTER SCIENCE”);

INSEAT TABLE STUDENT(ROLL\_NO,”NAME”,”BRANCH”)VALUE(2,”SUHANI”,”ELECTRONIC AND COM.”);

INSEAT TABLE STUDENT(ROLL\_NO,”NAME”,”BRANCH”)VALUE(3,”KRITI”,”ELECTRONIC AND COM.”);

* Make a table 2 (EXAM):-

🡪 code:-

CREATE TABLE EXAM

(

ROLL\_NO INT PRIMARY KEY AUTO\_INCREMENT,

S\_CODE VARCHAR ( 50 ),

MARKS INT,

P\_CODE VARCHAR ( 50 ),

ROLL\_NO\_FK INT,

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

INSERT VALUE:-

INSEAT TABLE STUDENT(ROLL\_NO,”S\_CODE”,MARKS,”P\_CODE”)VALUE(1,”CS11”,50,”CS”);

INSEAT TABLE STUDENT(ROLL\_NO,”S\_CODE”,MARKS,”P\_CODE”)VALUE(1,”CS12”,60,”CS”);

INSEAT TABLE STUDENT(ROLL\_NO,”S\_CODE”,MARKS,”P\_CODE”)VALUE(2,”EC101”,66,”EC”);

INSEAT TABLE STUDENT(ROLL\_NO,”S\_CODE”,MARKS,”P\_CODE”)VALUE(2,”EC102”,70,”EC”);

INSEAT TABLE STUDENT(ROLL\_NO,”S\_CODE”,MARKS,”P\_CODE”)VALUE(3,”EC101”,45,”EC”);

INSEAT TABLE STUDENT(ROLL\_NO,”S\_CODE”,MARKS,”P\_CODE”)VALUE(3,”EC102”,50,”EC”);

**TASK.2 Create table given below :-**

ANS. 🡪

MAKE A DATABASE:-

* MAKE A TABLE ( INFORMATION ) :-

CODE:-

CREATE TABLE INFORMATION

(

FIRST\_NAME VARCHAR(50),

LAST\_NAME VARCHAR (50),

ADDRESS VARCHAR (50),

CITY VARCHAR (50),

AGE INT

)

INSERT VALUE:-

INSERT INTO information (`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', 'ITOTLTAW', '28');

**TASK. 3. Create table given below: Employee and Incentive**

ANS. 🡪

MAKE A DATABASE

* MAKE A TABLE ( EMPLOYEE )

CODE:-

CREATE TABLE EMPLOYEE

(

EMP\_ID INT PRIMARY KEY AUTO\_INCREMENT,

FIRST\_NAME VARCHAR (50),

LAST\_NAME VARCHAR (50),

SALARY BIGINT,

JOINING\_DATE VARCHAR (50),

DEPARTMENT VARCHAR (50)

)

INSERT VALUE:-

INSERT INTO `employee` (`EMP\_ID`, `FIRST\_NAME`, `LAST\_NAME`, `SALARY`, `JOINING\_DATE`, `DEPARTMENT`) VALUES (NULL, 'JOHN', 'ABRAHAM', '1000000', '01-JAN-13 12:00AM', 'BANKING'),;

INSERT INTO `employee` (NULL, 'MICHAEL', 'CLARKE', '800000', '01-JAN-13 12:00AM', 'INSURANCE'),

INSERT INTO `employee` (NULL, 'ROY', 'THOMAS', '700000', '01-FEB-13 12:00AM', 'BANKING'),

INSERT INTO `employee` (NULL, 'TOM', 'JOSE', '600000', '01-FEB-13 12:00AM', 'INSURANCE'),

INSERT INTO `employee` (NULL, 'JERRY', 'PINTO', '650000', '01-FEB-13 12:00AM', 'INSURANCE'),

INSERT INTO `employee` (NULL, 'PHILIP', 'MATHEW', '750000', '01-JAN-13 12:00AM', 'SERVICES'),

INSERT INTO `employee` (NULL, 'TESTNAME1', '123', '650000', '01-JAN-13 12:00AM', 'SERVICES'),

INSERT INTO `employee` (NULL, 'TESTNAME2', 'LNAME', '600000', '01-FEB-13 12:00AM', 'INSURANCE');

* MAKE A TABLE ( Incentive ) :-

CODE:-

CREATE TABLE INCENTIVE

(

EMP\_REF\_ID INT PRIMARY KEY AUTO\_INCREMENT,

INCENTIVE\_DATE VARCHAR(50),

INCENTIVE\_AMOUNT BIGINT,

EMP\_ID\_FK INT,

FOREIGN KEY(EMP\_ID\_FK)REFERENCES employee(EMP\_ID)

)

INSERT VALUE:-

INSERT INTO `incentive` (`EMP\_REF\_ID`, `INCENTIVE\_DATE`, `INCENTIVE\_AMOUNT`, `EMP\_ID\_FK`) VALUES (NULL, '01-FEB-13', '5000', '1'),

INSERT INTO `incentive` (NULL, '01-FEB-13', '3000', '2'),

INSERT INTO `incentive` (NULL, '01-FEB-13', '4000', '3'),

INSERT INTO `incentive` (NULL, '01-JAN-13', '4500', '1'),

INSERT INTO `incentive` (NULL, '01-JAN-13', '3500', '2');

**QUE.(a), Get First\_Name from employee table using Tom name “Employee Name”.**

ANS 🡪 CODE:-

SELECT\*FROM employee WHERE EMP\_ID=1

**QUE. b) Get FIRST\_NAME, Joining Date, and Salary from employee table.**

ANS 🡪 CODE:-

SELECT FIRST\_NAME,SALARY,JOINING\_DATE FROM employee

**QUE.( c ) Get all employee details from the employee table order by First\_Name Ascending and Salary descending?**

ANS. 🡪CODE:-

SELECT \*FROM employee ORDER BY FIRST\_NAME ASC;

SELECT \*FROM employee ORDER BY SALARY DESC

**QUE.( d ) Get employee details from employee table whose first name contains ‘J’**

ANS. 🡪 CODE :-

SELECT \* FROM employee WHERE FIRST\_NAME LIKE 'J%'

**QUE.( e ) Get department wise maximum salary from employee table order by salary ascending?**

ANS. 🡪 CODE :-

SELECT \* FROM employee ORDER BY SALARY ASC

**QUE.( f ) Select first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000**

ANS. 🡪 CODE :-

SELECT \* FROM incentive WHERE INCENTIVE\_AMOUNT>3000

**QUE.( g ) Create After Insert trigger on Employee table which insert records in view table**

* MAKE A TABLE (VIEW) :-

**🡪CODE**

CREATE TABLE VIEW1

( ID INT,

FNAME VARCHAR(50),

LNAME VARCHAR(50),

SALARY INT,

JOINING\_DATE VARCHAR(50),

DEPARTMENT VARCHAR(50),

DATE\_TIME TIMESTAMP,

ACTION\_PERFORM VARCHAR(50) )

INSERT TRIGGER :-

🡪 CODE:-

DELIMITER $$

CREATE TRIGGER INSERT\_TRIGGER AFTER INSERT ON employee FOR EACH ROW

BEGIN

INSERT INTO view1(ID,FNAME,LNAME,SALARY,JOINING\_DATE,DEPARTMENT,ACTION\_PERFORM)VALUES(NEW.EMP\_ID,NEW.FIRST\_NAME,NEW.LAST\_NAME,NEW.SALARY,NEW.JOINING\_DATE,NEW.DEPARTMENT,"RECORD INSERTED");

END

**TASK .4 Create table given below: Salesperson and Customer**

ANS. 🡪

MAKE A DATABASE

* MAKE A TABLE ( SELSEPERSON )

CODE :-

CREATE TABLE SALSEPORSON

(

PK\_SNO INT PRIMARY KEY,

SNAME VARCHAR (50),

CITY VARCHAR (50),

COMM VARCHAR (50)

)

INSERT VALUE :-

INSERT INTO `salseporson` (`PK\_SNO`, `SNAME`, `CITY`, `COMM`) VALUES ('1001', 'PEEL', 'LONDON', '.12'),

INSERT INTO `salseporson` ('1002', 'SERRES', 'SAN JOSE', '.13'),

INSERT INTO `salseporson` ('1004', 'MOTILKA', 'LONDON', '.11')

INSERT INTO `salseporson`, ('1007', 'RAFKIN', 'BARCELONA', '.15'),

INSERT INTO `salseporson` ('1003', 'AXELROD', 'NEW YORK', '.1');

* MAKE A TABLE ( CUSTOMER )

CODE:-

CREATE TABLE CUSTOMER

(CNM INT PRIMARY KEY,

CNAME VARCHAR (50),

CITY VARCHAR (50),

RATING INT ,

PK\_SNO\_KFNT,

FOREIGN KEY (PK\_SNO\_KF)REFERENCES salseporson(PK\_SNO)

)

INSERT VALUE :-

INSERT INTO `customer` (`CNM`, `CNAME`, `CITY`, `RATING`, `PK\_SNO\_KF`) VALUES ('201', 'HOFFMAN', 'LONDON', '100', '1001'), ('202', 'GIOVANNE', 'ROE', '200', '1003'), ('203', 'LIU', 'SAN JOSE', '300', '1002'), ('204', 'GRASS', 'BARCELOAN', '100', '1002'), ('206', 'CLEMENS', 'LONDON', '300', '1007'), ('207', 'PEREIRA', 'ROE', '100', '1004');

**QUE.(B) Names and cities of all salespeople in London with commission above 0.12**

ANS. 🡪 CODE :-

SELECT \* FROM salseporson WHERE COMM>.12

**QUE.(C) All salespeople either in Barcelona or in London**

ANS. 🡪 CODE :-

SELECT \* FROM salseporson WHERE CITY="BARCELONA"OR CITY="LONDON"

**QUE.(D) All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).**

ANS. 🡪 CODE :-

SELECT \* FROM salseporson WHERE COMM =.11

**QUE.(E) All customers excluding those with rating <= 100 unless they are located in Rome**

ANS. 🡪 CODE :-

SELECT \* FROM customer WHERE RATING<=100 AND CITY NOT IN('ROE')