MODULE: 5 (Database)

1. What do you understand By Database

=> A database is a collection of arranged data that can be easily accessed,updated/modified or controlled.Information within the database can be easily placed into rows and columns,or tables.

2. What is Normalization?

=> Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion, and update anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

3. What is Difference between DBMS and RDBMS?

=> In RDBMS,
-Data stored is in table format.
-Multiple data elements are accesible together.
-Support distributed database.
-Ex:-Oracle,SQL server.

=> In DBMS,
-Data stored in file format.
-Individual access of data elements.
-No support for distributed database.
-Ex:-XML,Microsoft Access.

4. What is MF Cod Rule of RDBMS Systems?

=> The MF Cod Rule, part of RDBMS systems, refers to "Maintenance of Forthcoming Codes." It ensures that databases accommodate future data changes without requiring extensive alterations to existing structures. This principle aids in system scalability and adaptability, crucial for evolving data requirements in various applications.

5. What do you understand By Data Redundancy?

=> Data redundancy means having the same information stored in more than one place. It's like having duplicates of the same thing, which can waste space and cause confusion. Keeping redundancy low means having just one copy of each piece of information, which is more efficient and organized.

6. What is DDL Interpreter?

=> A DDL (Data Definition Language) interpreter is a component of a database management system (DBMS) responsible for processing and executing DDL commands. These commands are used to define and manage the structure of the database, such as creating, modifying, or deleting tables, indexes, views, and other database objects. The interpreter translates DDL statements into internal instructions that manipulate the database schema.

7. What is DML Compiler in SQL?

=>DML is an abbreviation for Data Manipulation Language.Represents a collection of programming languages explicitly used to make changes to the database ,such as :CRUD operations to create,read,update and delete data.Using INSERT,SELECT,UPDATE and DELETE commands.

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

=>SQL key constraints are rules applied to columns in a database table to enforce data integrity and define relationships between tables. Common key constraints include:

1)Primary Key Constraint: Ensures uniqueness and identifies each record uniquely in a table. Example:

```
CREATE TABLE Employees (
EmployeeID INT PRIMARY KEY,
Name VARCHAR(50),
Department VARCHAR(50)
);
```

2)Foreign Key Constraint: Establishes a relationship between two tables by referencing the primary key of one table in another. Example:

```
CREATE TABLE Orders (
OrderID INT PRIMARY KEY,
CustomerID INT,
OrderDate DATE,
FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

3)Unique Constraint: Ensures that all values in a column (or combination of columns) are unique. Example:

```
CREATE TABLE Students (
StudentID INT PRIMARY KEY,
StudentName VARCHAR(50),
Email VARCHAR(50) UNIQUE
);
```

9. What is save Point? How to create a save Point write 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.

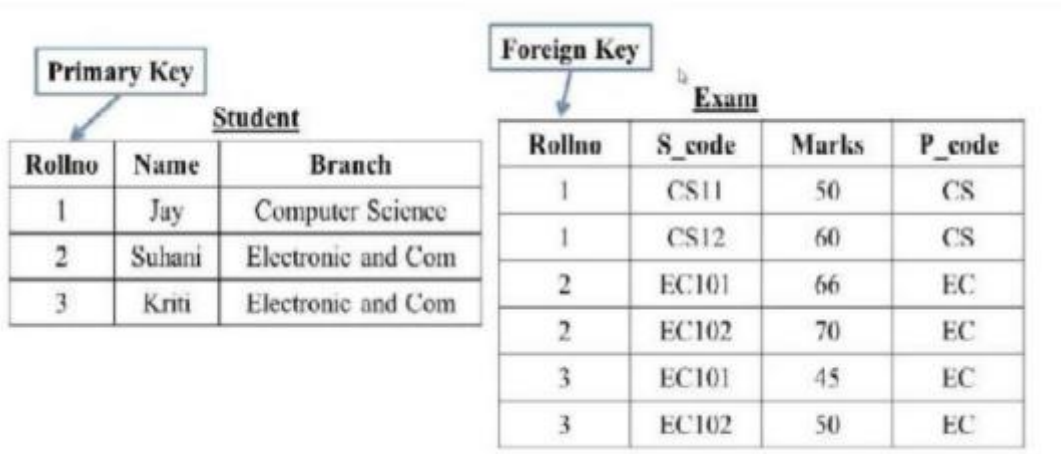
10.What is trigger and how to create a Trigger in SQL?

=>A trigger in SQL is like a watchful guardian for a database. It automatically reacts to specific events, such as adding, changing, or deleting data in a table. When these events occur, the trigger springs into action, executing predefined actions or commands. For example, it can log changes, enforce rules, or update other tables. Creating a trigger involves specifying the event that triggers it, when it should happen (before or after the event), and what actions it should take. Triggers help maintain data integrity, enforce business rules, and streamline database management by automating tasks based on certain conditions.

EXAMPLE:-

```
CREATE TRIGGER trg_after_insert
AFTER INSERT ON Employees
FOR EACH ROW
BEGIN
INSERT INTO AuditLog (Event, EmployeeID, Timestamp)
VALUES ('New Employee Added', NEW.EmployeeID, NOW());
END;
```

Q-1. Create Table Name : Student and Exam



Student

Rollno	Name	Branch
1	Jay	Computer Science
2	Suhani	Electronic and Com
3	Kriti	Electronic and Com

Exam

Rollno	S_code	Marks	P_code
1	CS11	50	CS
1	CS12	60	CS
2	EC101	66	EC
2	EC102	70	EC
3	EC101	45	EC
3	EC102	50	EC

ANS.

```
CREATE TABLE Student
(
    Rollno int PRIMARY KEY,
    name varchar(30),
    branch varchar(30)

);

CREATE TABLE exam

(
    Rollno int ,
    S_code varchar(20),
    Marks int,
    p_code varchar(20),
    FOREIGN KEY (Rollno) REFERENCES Student(Rollno)

);
INSERT INTO  students VALUES
(1,'JAY','COMPUTER SCIENCE'),
(2,'SUHANI','ELECTRONICS AND COMMUNICATION'),
(3,'KRITI','ELECTRONICS AND COMMUNICATION');

INSERT into exams VALUES
(1,'CS11',50,'CS'),
(1,'CS12',60,'CS'),
(2,'EC101',66,'EC'),
(2,'EC102',70,'EC'),
(3,'EC101',45,'EC'),
(3,'EC102',50,'EC');
```

output

Rollno	S_code	Marks	p_code
1	CS11	50	CS
1	CS12	60	CS
2	EC101	66	EC
2	EC102	70	EC
3	EC101	45	EC
3	EC102	50	EC

--EXAM TABLE

Rollno	name	branch
1	JAY	COMPUTER SCIENCE
2	SUHANI	ELECTRONICS AND COMMUNICATION
3	KRITI	ELECTRONICS AND COMMUNICATION

--STUDENT TABLE

Q-2.Create table given below: Employee and IncentiveTable

Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	John	Abraham	1000000	01-JAN-13 12.00.00 AM	Banking
2	Michael	Clarke	800000	01-JAN-13 12.00.00 AM	Insurance
3	Roy	Thomas	700000	01-FEB-13 12.00.00 AM	Banking
4	Tom	Jose	600000	01-FEB-13 12.00.00 AM	Insurance
5	Jerry	Pinto	650000	01-FEB-13 12.00.00 AM	Insurance
6	Philip	Mathew	750000	01-JAN-13 12.00.00 AM	Services
7	TestName1	123	650000	01-JAN-13 12.00.00 AM	Services
8	TestName2	Lname%	600000	01-FEB-13 12.00.00 AM	Insurance

----EMPLOYEE TABLE

Employee_ref_id	Incentive_date	Incentive_amount
1	01-FEB-13	5000
2	01-FEB-13	3000
3	01-FEB-13	4000
1	01-JAN-13	4500
2	01-JAN-13	3500

-----INCENTIVE TABLE

ANS.

```
create TABLE Employee

(
  Employee_id int PRIMARY KEY,
  First_name varchar(30),
  Last_name varchar(30),
  Salary int ,
  Joining_date datetime ,
  Department varchar(30)
);

create TABLE Incentive ( Employee_ref_id int, Incentive_date datetime,
Incentive_amount int, FOREIGN key (Employee_ref_id) REFERENCES
employee(Employee_id) );



INSERT INTO employee VALUES
(1,'JOHN','ABRAHAM',1000000,'2013-01-01 12:00:00','BANKING'),
(2,'MICHAEL','CLARKE',800000,'2013-01-02 12:00:00','INSURANCE'),
(3,'ROY','THOMAS',700000,'2013-01-02 12:00:00','BANKING'),
(4,'TOM','JOSE',600000,'2013-01-01 12:00:00','INSURANCE'),
(5,'JERRY','PINTO',650000,'2013-01-01 12:00:00','INSURANCE'),
(6,'PHILIP','MATHEW',750000,'2013-02-01 12:00:00','SERVICES'),
(7,'TESTNAME1','123',650000,'2013-01-01 12:00:00','SERVICES'),
(8,'JOHN','Lname%',600000,'2013-01-03 12:00:00','INSURANCE');
```

3. Get First_Name from employee table using Tom name “Employee Name”.

ANS. `SELECT First_name from employee where First_name = "Tom";`

OUTPUT

First_name
TOM

Q-4. 4. Get FIRST_NAME, Joining Date, and Salary from employee table.

ANS. SELECT First_name,Joining_date,Salary FROM employee;

OUTPUT

First_name	Joining_date	Salary
JOHN	2013-01-01 12:00:00	1000000
MICHAEL	2013-01-02 12:00:00	800000
ROY	2013-01-02 12:00:00	700000
TOM	2013-01-01 12:00:00	600000
JERRY	2013-01-01 12:00:00	650000
PHILIP	2013-02-01 12:00:00	750000
TESTNAME1	2013-01-01 12:00:00	650000
JOHN	2013-01-03 12:00:00	600000

Q-5. 5. Get all employee details from the employee table order by First_Name Ascending and Salary descending?

ANS. SELECT * from employee order BY First_name ASC ;

SELECT * FROM employee order BY Salary DESC;

OUTPUT.

Employee_id	First_name ▲ 1	Last_name	Salary	Joining_date	Departmen
5	JERRY	PINTO	650000	2013-01-01 12:00:00	INSURANC
1	JOHN	ABRAHAM	1000000	2013-01-01 12:00:00	BANKING
8	JOHN	Lname%	600000	2013-01-03 12:00:00	INSURANC
2	MICHAEL	CLARKE	800000	2013-01-02 12:00:00	INSURANC
6	PHILIP	MATHEW	750000	2013-02-01 12:00:00	SERVICES
3	ROY	THOMAS	700000	2013-01-02 12:00:00	BANKING
7	TESTNAME1	123	650000	2013-01-01 12:00:00	SERVICES
4	TOM	JOSE	600000	2013-01-01 12:00:00	INSURANC

FIRST NAME in ascending order

Employee_id	First_name	Last_name	Salary ▼ 1	Joining_date	Department
1	JOHN	ABRAHAM	1000000	2013-01-01 12:00:00	BANKING
2	MICHAEL	CLARKE	800000	2013-01-02 12:00:00	INSURANCE
6	PHILIP	MATHEW	750000	2013-02-01 12:00:00	SERVICES
3	ROY	THOMAS	700000	2013-01-02 12:00:00	BANKING
5	JERRY	PINTO	650000	2013-01-01 12:00:00	INSURANCE
7	TESTNAME1	123	650000	2013-01-01 12:00:00	SERVICES
4	TOM	JOSE	600000	2013-01-01 12:00:00	INSURANCE
8	JOHN	Lname%	600000	2013-01-03 12:00:00	INSURANCE

Salary in descending order.

6. Get employee details from employee table whose first name contains ‘J’

ANS. SELECT * FROM employee WHERE First_name LIKE 'J%';

OUTPUT.

Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	JOHN	ABRAHAM	1000000	2013-01-01 12:00:00	BANKING
5	JERRY	PINTO	650000	2013-01-01 12:00:00	INSURANCE
8	JOHN	Lname%	600000	2013-01-03 12:00:00	INSURANCE

7-8. Get department wise maximum salary from employee table order by salaryascending?

ANS. SELECT Department, MAX(Salary) AS max_salary FROM employee GROUP BY Department ORDER BY max_salary ASC;

OUTPUT.

Department	max_salary ▲ 1
SERVICES	750000
INSURANCE	800000
BANKING	1000000

9. Select first_name, incentive amount from employee and incentives table forthose employees who have incentives and incentive amount greater than 3000.

ANS.
```
SELECT employee.First_name,incentive.Incentive_amount
FROM employee e INNER join incentive i on
e.Employee_id=i.Employee_ref_id
WHERE i.Incentive_amount > 3000;
```

OUTPUT.

First_name	Incentive_amount
JOHN	5000
ROY	4000
JOHN	4500
MICHAEL	3500

10. Create After Insert trigger on Employee table which insert records in viewtab

ANS.

```
-- Step 1: Create the viewtable
CREATE TABLE viewtable (
   Employee_id int,
   First_name varchar(30),
   Last_name varchar(30),
   Salary int,
   Joining_date datetime,
   Department varchar(30)
);

-- Step 2: Create the after insert trigger
DELIMITER $$
CREATE TRIGGER after_employee_insert
AFTER INSERT ON Employee
FOR EACH ROW
BEGIN
   INSERT INTO viewtable (Employee_id, First_name, Last_name, Salary,
Joining_date, Department)
   VALUES (NEW.Employee_id, NEW.First_name, NEW.Last_name,
NEW.Salary, NEW.Joining_date, NEW.Department);
END$$
DELIMITER ;
```

OUTPUT.

Employee_id	First_name	Last_name	Salary	Joining_date	Department
1	JOHN	ABRAHAM	1000000	2013-01-01 12:00:00	BANKING
2	MICHAEL	CLARKE	800000	2013-01-02 12:00:00	INSURANCE
3	ROY	THOMAS	700000	2013-01-02 12:00:00	BANKING
4	TOM	JOSE	600000	2013-01-01 12:00:00	INSURANCE
5	JERRY	PINTO	650000	2013-01-01 12:00:00	INSURANCE
6	PHILIP	MATHEW	750000	2013-02-01 12:00:00	SERVICES
7	TESTNAME1	123	650000	2013-01-01 12:00:00	SERVICES
8	JOHN	Lname%	600000	2013-01-03 12:00:00	INSURANCE

11. Create table given below: Salesperson and Customer

ANS.

```
CREATE TABLE salesperson (
  SNO INT PRIMARY KEY,
  SName VARCHAR(30),
  CITY VARCHAR(20),
  COMM FLOAT
);

INSERT INTO salesperson VALUES (1001,'Peel','London',0.12);
INSERT INTO salesperson VALUES (1002,'Serres','San Jose',0.13);
INSERT INTO salesperson VALUES (1004,'Motika','London',0.11);
INSERT INTO salesperson VALUES (1007,'Rafkin','Barcelona',0.15);
INSERT INTO salesperson VALUES (1003,'Axelord','New York',0.1);

CREATE TABLE customer (
  CNM INT PRIMARY KEY,
  CNAME VARCHAR(30),
  CITY VARCHAR(30),
  RATING INT,
  SNO INT,
  FOREIGN KEY (SNO) REFERENCES salesperson(SNO)
);

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);
```

OUTPUT. SAME AS 12TH QUES OUTPUT.

12. Retrieve the below data from above table

ANS. SELECT * from salesperson;

SELECT * from customer;

OUTPUT.

SNO	SName	CITY	COMM
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1003	Axelord	New York	0.1
1004	Motika	London	0.11
1007	Rafkin	Barcelona	0.15

----SALESPERSON TABLE

CNM	CNAME	CITY	RATING	SNO
201	Hoffman	London	100	1001
202	Giovanne	Rome	200	1003
203	Liu	San Jose	300	1002
204	Grass	Barcelona	100	1002
206	Clemens	London	300	1007
207	Pereira	Rome	100	1004

----CUSTOMER TABLE

13. All orders for more than $1000.

ANS. SELECT * FROM salesperson WHERE SNO > 1000;

OUTPUT.

SNO	SName	CITY	COMM
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1003	Axelord	New York	0.1
1004	Motika	London	0.11
1007	Rafkin	Barcelona	0.15

14. Names and cities of all salespeople in London with commission above 0.12

ANS. SELECT SNAME,CITY FROM salesperson WHERE COMM>0.12;

OUTPUT.

SNAME	CITY	COMM
Serres	San Jose	0.13
Rafkin	Barcelona	0.15

15. All salespeople either in Barcelona or in London

ANS. SELECT *FROM salesperson WHERE CITY IN ('Barcelona', 'London');

OUTPUT.

SNO	SName	CITY	COMM
1001	Peel	London	0.12
1004	Motika	London	0.11
1007	Rafkin	Barcelona	0.15

16. 16. All salespeople with commission between 0.10 and 0.12. (Boundary valu should be excluded).

ANS. Select * from salesperson where COMM > 0.10 AND COMM < 0.12;

OUTPUT.

SNO	SName	CITY	COMM
1004	Motika	London	0.11

17. All customers excluding those with rating <= 100 unless they are located in Rome

ANS. SELECT * FROM Customer WHERE RATING > 100 OR CITY = 'Rome';

OUTPUT.

CNM	CNAME	CITY	RATING	SNO
202	Giovanne	Rome	200	1003
203	Liu	San Jose	300	1002
206	Clemens	London	300	1007
207	Pereira	Rome	100	1004

18. Write a SQL statement that displays all the information about all salespeople

ANS. create table salesman(salesman_id int,name varchar(30),city varchar(20),commission float);

INSERT INTO salesman VALUES
(5001, 'James Hoog', 'New York', 0.15),
(5002, 'Nail Knite', 'Paris', 0.13),
(5005, 'Pit Alex', 'London', 0.11),
(5006, 'Mc Lyon', 'Paris', 0.14),
(5007, 'Paul Adam', 'Rome', 0.13),
(5003, 'Lauson Hen', 'San Jose', 0.12);

select * from salesman;--->>display whole table

OUTPUT.

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12

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.

ANS.
```
CREATE TABLE orders (
    ord_no INT,
    purch_amt INT,
    ord_date INT,
    customer_id INT,
    salesman_id INT
);

insert into orders values(70001,150.5,2012-10-05,3005,5002),
(70009,270.65,2012-09-10,3001,5005),
(70002,65.26,2012-10-05,3002,5001),
(70004,110.5,2012-08-17,3009,5003),
(70007,948.5,2012-09-10,3005,5002),
(70005,2400.6,2012-07-27,3007,5001),
(70008,5760,2012-09-10,3002,5001),
(70010,1983.43,2012-10-10,3004,5006),
(70003,2480.4,2012-10-10,3009,5003),
(70012,250.45,2012-06-27,3008,5002),
(70011,75.29,2012-08-17,3003,5007),
(70013,3045.6,2012-04-25,3002,5001);

select ord_no,ord_date,purch_amt from orders where salesman_id = 5001;
```

OUTPUT.

ord_no	ord_date	purch_amt
70002	1997	65
70005	1978	2401
70008	1993	5760
70013	1983	3046

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.

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
101	Mother Board	3200.00	15
102	Key Board	450.00	16
103	ZIP drive	250.00	14
104	Speaker	550.00	16
105	Monitor	5000.00	11
106	DVD drive	900.00	12
107	CD drive	800.00	12
108	Printer	2600.00	13
109	Refill cartridge	350.00	13
110	Mouse	250.00	12

ANS.

```
INSERT INTO items VALUES
(101,'Mother Board',3200.00,15),
(102,'Key Board',450.00,16),
(103,'ZIP drive',250.00,14),
(104,'Speaker',550.00,16),
(105,'Monitor',5000.00,11),
(106,'DVD drive',900.00,12),
(107,'CD drive',800.00,12),
(108,'Printer',2600.00,13),
(109,'Refill cartridge',350.00,13),
(110,'Mouse',250.00,12);

CREATE TABLE items (PRO_ID int,PRO_NAME varchar(30), PRO_PRIC
int,PRO_COM int);


SELECT * FROM `items` WHERE PRO_PRICE>=200 &&
PRO_PRICE<=600;
```

OUTPUT.

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
102	Key Board	450	16
103	ZIP drive	250	14
104	Speaker	550	16
109	Refill cartridge	350	13
110	Mouse	250	12

21. From the following table, write a SQL query to calculate the averageprice for manufacturer code of 16. Return avg.

ANS. select avg(PRO_PRICE) AVERAGE_PRICE_IS from items where PRO_COM = 16;//21

OUTPUT.

AVERAGE_PRICE_IS
500.0000

22. From the following table, write a SQL query to display the pro_name as 'Item Name' and pro_price as 'Price in Rs.'

ANS. SELECT PRO_NAME AS items_name, PRO_PRICE AS Price_in_Rs FROM items;

OUTPUT.

items_name	Price_in_Rs
Mother Board	3200
Key Board	450
ZIP drive	250
Speaker	550
Monitor	5000
DVD drive	900
CD drive	800
Printer	2600
Refill cartridge	350
Mouse	250

23. From the following table, write a SQL query to find the items whose prices a higher than or equal to $250. Order the result by product price in descending, then product name in ascending. Return pro_name and pro_price.

ANS. SELECT PRO_NAME, PRO_PRICE FROM items WHERE PRO_PRICE >= 250 ORDER BY PRO_PRICE DESC,PRO_NAME ASC;

OUTPUT.

PRO_NAME ▲ 2	PRO_PRICE ▼ 1
Monitor	5000
Mother Board	3200
Printer	2600
DVD drive	900
CD drive	800
Speaker	550
Key Board	450
Refill cartridge	350
Mouse	250
ZIP drive	250

24. From the following table, write a SQL query to calculate average price ofthe items for each company. Return average price and companycode.

ANS. SELECT PRO_COM, AVG(PRO_PRICE) AS average_price FROM items GROUP BY PRO_COM;

OUTPUT.

PRO_COM	average_price
11	5000.0000
12	650.0000
13	1475.0000
14	250.0000
15	3200.0000
16	500.0000