MODULE – 2

TOPIC – SQL

- **TASK – 1**

1. **Table: Employee**

Step 1: Create Employee table.

Query:

```
CREATE TABLE Employee(EM_ID INT PRIMARY KEY AUTO_INCREMENT,
                      FIRST_NAME VARCHAR(50),
                      LAST_NAME VARCHAR(50),
                      SALARY FLOAT,
                      JOINING_DATE INT,
                      DEPARTMENT VARCHAR(50));
```

Output:

EM_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT

Step 2: Insert values into Employee table.

Query:

```
INSERT INTO employee VALUES('','JOHN','ABRAHAM',1000000,'2013-01-01','BANKING');
INSERT INTO employee VALUES('','MICHAEL','CLERK',800000,'2013-01-01','INSURANCE');
INSERT INTO employee VALUES('','ROY','THOMAS',700000,'2013-02-01','BANKING');
INSERT INTO employee VALUES('','TOM','JOSE',600000,'2013-02-01','INSURANCE');
INSERT INTO employee VALUES('','JERRY','PINTO',650000,'2013-01-01','INSURANCE');
INSERT INTO employee VALUES('','PHILIP','MATHEW',750000,'2013-01-01','SERVICES');
INSERT INTO employee VALUES('','TESTNAME1','123',650000,'2013-01-01','SERVICES');
INSERT INTO employee VALUES('','TESTNAME2','LNAME%',600000,'2013-02-01','INSURANCE');
```

Output:

EM_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	JOHN	ABRAHAM	1000000	2013-01-01	BANKING
2	MICHAEL	CLERK	800000	2013-01-01	INSURANCE
3	ROY	THOMAS	700000	2013-02-01	BANKING
4	TOM	JOSE	600000	2013-02-01	INSURANCE
5	JERRY	PINTO	650000	2013-01-01	INSURANCE
6	PHILIP	MATHEW	750000	2013-01-01	SERVICES
7	TESTNAME1	123	650000	2013-01-01	SERVICES
8	TESTNAME2	LNAME%	600000	2013-02-01	INSURANCE

2. **Table: Incentives**

Step 1: Create Incentives table.

Query:

```
CREATE TABLE Incentives(INCENTIVE_ID INT PRIMARY KEY AUTO_INCREMENT,
                        EMPLOYEE_REF_ID INT,
                        INCENTIVE_DATE DATE,
                        INCENTIVE_AMT FLOAT);
```

Output:

INCENTIVE_ID	EMPLOYEE_REF_ID	INCENTIVE_DATE	INCENTIVE_AMT

Step 2: Insert values into Incentives table.

Query:

```
INSERT INTO incentives VALUES('',1,'2013-02-01',5000);
INSERT INTO incentives VALUES('',2,'2013-02-01',3000);
INSERT INTO incentives VALUES('',3,'2013-02-01',4000);
INSERT INTO incentives VALUES('',1,'2013-01-01',4500);
INSERT INTO incentives VALUES('',2,'2013-01-01',3500);
```

Output:

INCENTIVE_ID	EMPLOYEE_REF_ID	INCENTIVE_DATE	INCENTIVE_AMT
1	1	2013-02-01	5000
2	2	2013-02-01	3000
3	3	2013-02-01	4000
4	1	2013-01-01	4500
5	2	2013-01-01	3500

a) Get FIRST_NAME from employee table using alias name “Employee Name”.

Query:

```
SELECT FIRST_NAME AS EmployeeName FROM employee;
```

Output:

EmployeeName
JOHN
MICHAEL
ROY
TOM
JERRY
PHILIP
TESTNAME1
TESTNAME2

b) Get FIRST_NAME, Joining year, Joining Month and Joining Date from employee table.

Query:

```
SELECT FIRST_NAME, EXTRACT(YEAR FROM JOINING_DATE) AS JOINING_YEAR,
EXTRACT(MONTH FROM JOINING_DATE) AS JOINING_MONTH, EXTRACT(DAY FROM
JOINING_DATE) AS JOINING_DAY FROM employee;
```

Output:

FIRST_NAME	JOINING_YEAR	JOINING_MONTH	JOINING_DAY
JOHN	2013	1	1
MICHAEL	2013	1	1
ROY	2013	2	1
TOM	2013	2	1
JERRY	2013	1	1
PHILIP	2013	1	1
TESTNAME1	2013	1	1
TESTNAME2	2013	2	1

c) Get all employee details from the employee table order by First Name Ascending And Salary descending?

Query:

```
SELECT * FROM `employee` ORDER BY FIRST_NAME ASC;
SELECT * FROM `employee` ORDER BY SALARY DESC;
```

Output:

EM_ID	FIRST_NAME ▲ 1	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
5	JERRY	PINTO	650000	2013-01-01	INSURANCE
1	JOHN	ABRAHAM	1000000	2013-01-01	BANKING
2	MICHAEL	CLERK	800000	2013-01-01	INSURANCE
6	PHILIP	MATHEW	750000	2013-01-01	SERVICES
3	ROY	THOMAS	700000	2013-02-01	BANKING
7	TESTNAME1	123	650000	2013-01-01	SERVICES
8	TESTNAME2	LNAME%	600000	2013-02-01	INSURANCE
4	TOM	JOSE	600000	2013-02-01	INSURANCE

EM_ID	FIRST_NAME	LAST_NAME	SALARY ▼ 1	JOINING_DATE	DEPARTMENT
1	JOHN	ABRAHAM	1000000	2013-01-01	BANKING
2	MICHAEL	CLERK	800000	2013-01-01	INSURANCE
6	PHILIP	MATHEW	750000	2013-01-01	SERVICES
3	ROY	THOMAS	700000	2013-02-01	BANKING
5	JERRY	PINTO	650000	2013-01-01	INSURANCE
7	TESTNAME1	123	650000	2013-01-01	SERVICES
4	TOM	JOSE	600000	2013-02-01	INSURANCE
8	TESTNAME2	LNAME%	600000	2013-02-01	INSURANCE

d) Get employee details from employee table whose first name contains “o”.

Query:

```
SELECT * FROM `employee` WHERE FIRST_NAME LIKE '%o%';
```

Output:

EM_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	JOHN	ABRAHAM	1000000	2013-01-01	BANKING
3	ROY	THOMAS	700000	2013-02-01	BANKING
4	TOM	JOSE	600000	2013-02-01	INSURANCE

e) Get employee details from employee table whose joining month is “January”.

Query:

```
SELECT * FROM employee WHERE EXTRACT(MONTH FROM JOINING_DATE)=1;
```

Output:

EM_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	JOHN	ABRAHAM	1000000	2013-01-01	BANKING
2	MICHAEL	CLERK	800000	2013-01-01	INSURANCE
5	JERRY	PINTO	650000	2013-01-01	INSURANCE
6	PHILIP	MATHEW	750000	2013-01-01	SERVICES
7	TESTNAME1	123	650000	2013-01-01	SERVICES

f) Get department, total salary with respect to a department from employee table order By total salary descending.

Query:

```
SELECT SUM(SALARY), DEPARTMENT FROM employee GROUP BY DEPARTMENT
ORDER BY SUM(SALARY) DESC;
```

Output:

SUM(SALARY) 1	DEPARTMENT
2650000	INSURANCE
1700000	BANKING
1400000	SERVICES

g) Get department wise maximum salary from employee table order by salary Ascending.

Query:

```
SELECT MAX(SALARY), DEPARTMENT FROM employee GROUP BY DEPARTMENT
ORDER BY MAX(SALARY) ASC;
```

Output:

MAX(SALARY) 1	DEPARTMENT
750000	SERVICES
800000	INSURANCE
1000000	BANKING

h) Select first_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000.

Query:

```
SELECT employee.FIRST_NAME, incentives.INCENTIVE_AMT FROM employee
INNER JOIN incentives ON employee.EM_ID=incentives.EMPLOYEE_REF_ID
AND INCENTIVE_AMT>3000;
```

Output:

FIRST_NAME	INCENTIVE_AMT
JOHN	5000
ROY	4000
JOHN	4500
MICHAEL	3500

i) Select 2nd Highest salary from employee table.

Query:

```
SELECT * FROM employee GROUP BY salary ORDER by salary DESC LIMIT 1,1;
```

Output:

EM_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
2	MICHAEL	CLERK	800000	2013-01-01	INSURANCE

j) Select first_name, incentive amount from employee and incentives table for all Employees who got incentives using left join.

Query:

```
SELECT employee.FIRST_NAME, incentives.INCENTIVE_AMT FROM employee LEFT JOIN incentives on
employee.EM_ID=incentives.EMPLOYEE_REF_ID;
```

Output:

FIRST_NAME	INCENTIVE_AMT
JOHN	5000
MICHAEL	3000
ROY	4000
JOHN	4500
MICHAEL	3500
TOM	*NULL*
JERRY	*NULL*
PHILIP	*NULL*
TESTNAME1	*NULL*
TESTNAME2	*NULL*

k) Create View OF Employee table in which store first name, last name and salary only.

Query:

```
CREATE view employees AS SELECT FIRST_NAME, LAST_NAME, SALARY FROM
employee;
```

Output:

FIRST_NAME	LAST_NAME	SALARY
JOHN	ABRAHAM	1000000
MICHAEL	CLERK	800000
ROY	THOMAS	700000
TOM	JOSE	600000
JERRY	PINTO	650000
PHILIP	MATHEW	750000
TESTNAME1	123	650000
TESTNAME2	LNAME%	600000

l) Create Procedure to find out department wise highest salary.

Query:

```
CREATE PROCEDURE `highSalary`() NOT DETERMINISTIC CONTAINS SQL SQL SECURITY
DEFINER SELECT department, max(salary) from employee GROUP BY department;
```

```
CALL `highSalary`();
```

Output:

department	max(salary)
BANKING	1000000
INSURANCE	800000
SERVICES	750000

m) Create After Insert trigger on Employee table which insert records in view table.

Query:

```
CREATE TRIGGER `after_insert` AFTER INSERT ON `employee` FOR EACH ROW INSERT INTO
view VALUES(NULL, NEW.EM_ID,'INSERTED',NOW());
```

Output:

Step 1: Now, insert values in employee table.

Table: employee

EM_ID	FIRST_NAME	LAST_NAME	SALARY	JOINING_DATE	DEPARTMENT
1	JOHN	ABRAHAM	1000000	2013-01-01	BANKING
2	MICHAEL	CLERK	800000	2013-01-01	INSURANCE
3	ROY	THOMAS	700000	2013-02-01	BANKING
4	TOM	JOSE	600000	2013-02-01	INSURANCE
5	JERRY	PINTO	650000	2013-01-01	INSURANCE
6	PHILIP	MATHEW	750000	2013-01-01	SERVICES
7	TESTNAME1	123	650000	2013-01-01	SERVICES
8	TESTNAME2	LNAME%	600000	2013-02-01	INSURANCE
9	JAGU	AHIRE	50000	2001-12-12	BANKING

Step 2: Then, values get automatically inserted into view table.

Table: view

id	EID	ACTION	CDATE
1	9	INSERTED	2022-07-22 15:24:01

- **TASK – 2**

1. **Table: Salesperson**

Step 1: Create Salesperson table.

Query:

```
CREATE TABLE Salesperson(SNO INT PRIMARY KEY,
                         SNAME VARCHAR(50),
                         CITY VARCHAR(50),
                         COMM FLOAT);
```

Output:

SNO	SNAME	CITY	COMM

Step 2: Insert values into Salesperson table.

Query:

```
INSERT INTO salesperson VALUES(1001,'PEEL','LONDON',0.12);
INSERT INTO salesperson VALUES(1002,'SERRES','SAN JOSE',0.13);
INSERT INTO salesperson VALUES(1003,'AXELROD','NEW YORK',0.1);
INSERT INTO salesperson VALUES(1004,'MOTIKA','LONDON',0.11);
INSERT INTO salesperson VALUES(1007,'RAFKIN','BARCELONA',0.15);
```

Output:

SNO	SNAME	CITY	COMM
1001	PEEL	LONDON	0.12
1002	SERRES	SAN JOSE	0.13
1003	AXELROD	NEW YORK	0.1
1004	MOTIKA	LONDON	0.11
1007	RAFKIN	BARCELONA	0.15

2. **Table: Customer**

Step 1: Create Customer table.

Query:

```
CREATE TABLE Customer(CNM INT PRIMARY KEY,
                      CNAME VARCHAR(50),
                      CITY VARCHAR(50),
                      RATING INT,
                      SNO INT);
```

Output:

CNM	CNAME	CITY	RATING	SNO

Step 2: Insert values into Customer table.

Query:

```
INSERT INTO customer VALUES(201,'HOFFMAN','LONDON',100,1001);
INSERT INTO customer VALUES(202,'GIOVANNE','ROME',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,'PERRIRA','ROME',100,1004);
```

Output:

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	PERRIRA	ROME	100	1004

3. Table: Order

Step 1: Create Orders table.

Query:

```
CREATE TABLE Orders(ONM INT PRIMARY KEY,
                    AMT FLOAT,
                    ODE DATE,
                    CNM INT,
                    SNO INT);
```

Output:

ONM	AMT	ODE	CNM	SNO

Step 2: Insert values into Order table.

Query:

```
INSERT INTO orders VALUES(3001,18.69,'1994-10-03',201,1007);
INSERT INTO orders VALUES(3002,1900.1,'1994-10-03',207,1004);
INSERT INTO orders VALUES(3003,767.19,'1994-10-03',201,1001);
INSERT INTO orders VALUES(3005,3005,'1994-10-03',203,1002);
INSERT INTO orders VALUES(3006,3006,'1994-10-04',201,1007);
INSERT INTO orders VALUES(3007,3007,'1994-10-05',204,1002);
INSERT INTO orders VALUES(3008,3008,'1994-10-05',206,1001);
INSERT INTO orders VALUES(3009,3009,'1994-10-04',202,1003);
INSERT INTO orders VALUES(3010,3010,'1994-10-06',204,1002);
INSERT INTO orders VALUES(3011,3011,'1994-10-06',206,10071);
```

Output:

ONM	AMT	ODE	CNM	SNO
3001	18.69	1994-10-03	201	1007
3002	1900.1	1994-10-03	207	1004
3003	767.19	1994-10-03	201	1001
3005	3005	1994-10-03	203	1002
3006	3006	1994-10-04	201	1007
3007	3007	1994-10-05	204	1002
3008	3008	1994-10-05	206	1001
3009	3009	1994-10-04	202	1003
3010	3010	1994-10-06	204	1002
3011	3011	1994-10-06	206	10071

a) All orders for more than $1000.

Query:

```
SELECT * FROM orders WHERE amt>1000;
```

Output:

ONM	AMT	ODE	CNM	SNO
3002	1900.1	1994-10-03	207	1004
3005	3005	1994-10-03	203	1002
3006	3006	1994-10-04	201	1007
3007	3007	1994-10-05	204	1002
3008	3008	1994-10-05	206	1001
3009	3009	1994-10-04	202	1003
3010	3010	1994-10-06	204	1002
3011	3011	1994-10-06	206	10071

b) Names and cities of all salespeople in London with commission above 0.10.

Query:

```
SELECT * FROM salesperson WHERE city="london" AND comm>0.10;
```

Output:

SNO	SNAME	CITY	COMM
1001	PEEL	LONDON	0.12
1004	MOTIKA	LONDON	0.11

c) All salespeople either in Barcelona or in London.

Query:

```
SELECT * FROM salesperson WHERE CITY IN('LONDON','BARCELONA');
```

Output:

SNO	SNAME	CITY	COMM
1001	PEEL	LONDON	0.12
1004	MOTIKA	LONDON	0.11
1007	RAFKIN	BARCELONA	0.15

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

Query:

```
SELECT * FROM salesperson WHERE comm BETWEEN 0.10 AND 0.12;
```

Output:

SNO	SNAME	CITY	COMM
1001	PEEL	LONDON	0.12
1003	AXELROD	NEW YORK	0.1
1004	MOTIKA	LONDON	0.11

e) All customers with NULL values in city column.

Query:

```
SELECT * FROM customer WHERE CITY IS NULL;
```

Output:

CNM	CNAME	CITY	RATING	SNO

f) All orders taken on Oct 3Rd and Oct 4th 1994.

Query:

```
SELECT * FROM orders WHERE ODE IN('1994-10-03','1994-10-04');
```

Output:

ONM	AMT	ODE	CNM	SNO
3001	18.69	1994-10-03	201	1007
3002	1900.1	1994-10-03	207	1004
3003	767.19	1994-10-03	201	1001
3005	3005	1994-10-03	203	1002
3006	3006	1994-10-04	201	1007
3009	3009	1994-10-04	202	1003

g) All customers serviced by peel or Motika.

Query:

```
SELECT customer.CNAME FROM customer INNER JOIN salesperson ON
customer.SNO=salesperson.SNO WHERE salesperson.SNAME IN('PEEL','MOTIKA');
```

Output:

CNAME
HOFFMAN
PERRIRA

h) All customers whose names begin with a letter from A to B.

Query:

```
SELECT * FROM customer WHERE CNAME LIKE 'A%' OR 'B%';
```

Output:

CNM	CNAME	CITY	RATING	SNO

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

Query:

```
SELECT * FROM customer WHERE RATING>=100 OR CITY="ROME";
```

Output:

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	PERRIRA	ROME	100	1004

j) All orders except those with 0 or NULL value in amt field.

Query:

```
SELECT * FROM orders WHERE AMT!=0 OR AMT!=NULL;
```

Output:

ONM	AMT	ODE	CNM	SNO
3001	18.69	1994-10-03	201	1007
3002	1900.1	1994-10-03	207	1004
3003	767.19	1994-10-03	201	1001
3005	3005	1994-10-03	203	1002
3006	3006	1994-10-04	201	1007
3007	3007	1994-10-05	204	1002
3008	3008	1994-10-05	206	1001
3009	3009	1994-10-04	202	1003
3010	3010	1994-10-06	204	1002
3011	3011	1994-10-06	206	10071

k) Count the number of salespeople currently listing orders in the order table.

Query:

```
SELECT COUNT(DISTINCT SNO) FROM orders;
```

Output:

COUNT(DISTINCT SNO)
6