**SQL**

**TASK-1**

CREATE DATABASE company;

CREATE TABLE EMPLOYEE(

EM\_ID INT PRIMARY KEY AUTO\_INCREMENT,

FIRST\_NAME VARCHAR(255),

LAST\_NAME VARCHAR(255),

SALARY INT(6),

JOINING\_DATE DATE,

DEPARTMENT VARCHAR(255) );

CREATE TABLE Incentives(

incentive\_id INT(11) PRIMARY KEY AUTO\_INCREMENT,

employee\_ref\_id INT(11), FOREIGN KEY (employee\_ref\_id) REFERENCES employee(EM\_ID),

incentive\_DATE DATETIME,

incentive\_amt INT(6) );

INSERT INTO Employee (FIRST\_NAME, LAST\_NAME,SALARY,JOINING\_DATE,DEPARTMENT )

VALUES

('John','Abraham', 1000000,'2013-01-01','Banking'),

('Michael','Clarke', 8000000,'2013-01-01','Insurance'),

('Roy', 'Thomas', 7000000,'2013-02-01', 'Banking'),

('Tom', 'Jose', 6000000,'2013-02-01', 'Insurance'),

('Jerry','Pinto',650000, '2013-01-01', 'Insurance'),

('Philip', 'Mathew', 750000, '2013-01-01', 'Service'),

('TestName1', '123', 650000, '2013-01-01', 'Serivice'),

('Testname2', 'Lname%', 6000000, '2013-01-01', 'Insurance'),

('N\_John','N\_Abraham', 1000000,'2013-01-01','Banking'),

('N\_Michael','N\_Clarke', 8000000,'2013-02-01','Insurance');

**a) Get First\_Name from employee table using alias name “Employee Name”.**

SELECT FIRST\_NAME AS Employee\_Name FROM employee;

**b) Get FIRST\_NAME, Joining year, Joining Month and Joining Date from employee**

**table.**

SELECT

FIRST\_NAME,

YEAR(JOINING\_DATE) AS joining\_Year,

MONTH(JOINING\_DATE) AS joining\_Month,

DAY(JOINING\_DATE) AS joining\_Date

FROM employee;

**c) Get all employee details from the employee table order by First Name Ascending**

**And Salary descending?**

SELECT \* FROM employee ORDER BY FIRST\_NAME;

SELECT \* FROM employee ORDER BY SALARY DESC;

**d) Get employee details from employee table whose first name contains „o‟.**

SELECT \* FROM employee

WHERE FIRST\_NAME LIKE 'o%';

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

SELECT \* FROM employee

WHERE MONTH(JOINING\_DATE) = 1;

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

SELECT DEPARTMENT, SUM(SALARY) AS total\_salary

FROM employee

GROUP BY DEPARTMENT

ORDER BY total\_salary DESC;

**g) Get department-wise maximum salary from employee table ordered by salary ascending**

SELECT DEPARTMENT, MAX(SALARY) AS max\_salary

FROM employee

GROUP BY DEPARTMENT

ORDER BY max\_salary ;

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

SELECT e.FIRST\_NAME, i.incentive\_amt

FROM employee e

JOIN incentives i ON e.EM\_ID = i.employee\_ref\_id

WHERE i.incentive\_amt > 3000;

**i) Select 2nd highest salary from employee table**

SELECT MAX(SALARY) AS second\_highest\_salary

FROM employee

WHERE SALARY < (SELECT MAX(SALARY) FROM employee);

**j) Select first\_name, incentive amount from employee and incentives table for all employees who got incentives using LEFT JOIN**

SELECT e.FIRST\_NAME, i.incentive\_amt

FROM employee e

LEFT JOIN incentives i ON e.EM\_ID = i.employee\_ref\_id;

**k) Create a view of Employee table which stores first name, last name, and salary only**

CREATE VIEW EmployeeView AS

SELECT FIRST\_NAME, LAST\_NAME, SALARY

FROM employee;

**l) Create a procedure to find out department-wise highest salary**

**m) Create an AFTER INSERT trigger on Employee table which inserts records into the view table**

CREATE TABLE EmployeeViewTable (

FIRST\_NAME VARCHAR(100),

LAST\_NAME VARCHAR(100),

SALARY BIGINT

);

**TASK-2**

CREATE DATABASE sales;

CREATE TABLE salesperson(

SNO INT PRIMARY KEY AUTO\_INCREMENT,

SNAME VARCHAR(255),

CITY VARCHAR(255),

COMM BIGINT );

CREATE TABLE customer(

CNM INT PRIMARY KEY AUTO\_INCREMENT,

CNAME VARCHAR(255),

CITY VARCHAR(255),

RATING INT(6),

SNO INT, FOREIGN KEY (SNO) REFERENCES salesperson(SNO));

CREATE TABLE orders(

ONM INT PRIMARY KEY AUTO\_INCREMENT,

AMT BIGINT,

ODE DATETIME,

CNM INT(6),FOREIGN KEY (CNM) REFERENCES customer(CNM),

SNO INT, FOREIGN KEY (SNO) REFERENCES salesperson(SNO));

INSERT INTO salesperson (SNO, SNAME, CITY, COMM) VALUES

(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);

INSERT INTO customer (CNM, CNAME, CITY, RATING, SNO) VALUES

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

(202, 'Givoanne', 'Rome', 200, 1003),

(203, 'Leu', 'San Jose', 300, 1002),

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

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

(207, 'Pereira', 'Rome', 100, 1004);

INSERT INTO orders (ONM, AMT, ODE, CNM, SNO) VALUES

(3001, 18.69, '1994-10-3', 201, 1007),

(3002, 1900.1, '1994-10-3', 207, 1004),

(3003, 767.19, '1994-10-3', 201, 1001),

(3005, 3005, '1994-10-3', 203, 1002),

(3006, 3006, '1994-10-4', 201, 1007),

(3007, 3007, '1994-10-5', 204, 1002),

(3008, 3008, '1994-10-5', 206, 1001),

(3009, 3009, '1994-10-4', 202, 1003),

(3010, 3010, '1994-10-6', 204, 1002),

(3011, 3011, '1994-10-6', 206, 1001);

**a) All orders for more than $1000.**

SELECT \*FROM orders

WHERE AMT > 1000;

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

SELECT SNAME, CITY

FROM salesperson

WHERE CITY = 'London' AND COMM > 0.10;

**c) All salespeople either in Barcelona or in London.**

SELECT \*

FROM salesperson

WHERE CITY IN ('Barcelona', 'London');

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

**be excluded).**

SELECT \*

FROM salesperson

WHERE COMM > 0.10 AND COMM < 0.12;

**e) All customers with NULL values in city column.**

SELECT \*

FROM customer

WHERE CITY IS NULL;

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

SELECT \*

FROM orders

WHERE ODE BETWEEN '1994-10-03' AND '1994-10-04';

**g) All customers serviced by peel or Motika**

SELECT \*

FROM customer

WHERE SNO IN (

SELECT SNO FROM salesperson WHERE SNAME IN ('Peel', 'Motika')

);

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

SELECT \*

FROM customer

WHERE CNAME LIKE 'A%' OR CNAME LIKE 'B%';

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

SELECT \*

FROM customer

WHERE (RATING <= 100 OR CITY = 'Rome') AND RATING IS NOT NULL;

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

SELECT \*

FROM orders

WHERE AMT IS NOT NULL AND AMT != 0;

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

SELECT COUNT(DISTINCT SNO) AS num\_salespeople

FROM orders;