**SQL Task - 1**

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

=> SELECT FIRST\_NAME FROM Employee

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

=> SELECT FIRST\_NAME, JOINING\_DATE, DATE\_FORMAT(JOINING\_DATE, ‘%Y’), DATE\_FORMAT(JOINING\_DATE, ‘%m’), DATE\_FORMAT(JOINING\_DATE, ‘%d’) 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 ASC, 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 DATE\_FORMAT(JOINING\_DATE, ‘%M’) = ‘January’

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

=> SELECT DEPARTMENT, SUM (SALARY) Total\_Salary FROM Employee GROUP BY DEPARTMENT ORDER BY Total\_Salary DESC

**g) Get department wise maximum salary from employee table order by salary ascending?**

=> SELECT DEPARTMENT, MAX(SALARY) Max\_Salary FROM Employee GROUP BY DEPARTMENT ORDER BY Max\_Salary ASC

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

=> SELECT FIRST\_NAME, INENTIVE\_AMT FROM Employee A INNER JOIN Incentives B ON A.EM\_ID = B.EMPLOYEE\_REF\_ID AND INENTIVE\_AMT > 3000

**i) Select 2nd Highest salary from employee table.**

=>SELECT MAX(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 FIRST\_NAME, INENTIVE\_AMT FROM Employee LEFT JOIN Incentives ON Employee.EM\_ID = Incentives.EMPLOYEE\_REF\_ID

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

=> CREATE VIEW emp\_view AS SELECT FIRST\_NAME, LAST\_NAME, SALARY FROM Employee

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

=> DELIMITER &&

CREATE PROCEDURE get\_dep\_highest\_sal()

BEGIN

SELECT DEPARTMENT, MAX(SALARY) FROM Employee GROUP BY DEPARTMENT;

END &&

DELIMITER;

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

=> DELIMITER $$

CREATE TRIGGER employee\_insert\_view

AFTER INSERT

ON Employee FOR EACH ROW

BEGIN

INSERT INTO view\_table (EM\_ID, NAME, CREATEAD\_AT)

VALUES (new.EM\_ID, CONCAT(FIRST\_NAME, ' ', LAST\_NAME), NOW());

END$$

DELIMITER;

**SQL Task - 2**

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

=> SELECT \* FROM ORDER WHERE AMT>1000

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

=> SELECT\* FROM SALESPERSON WHERE CITY=’LONDON’ AND COMM > 0.10

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

=> SELECT \* FROM SALESPERSON WHERE CITY=’LONDON’ OR CITY=’BARCELONA’

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

=> SELECT \* FROM SALESPERSON WHERE COMM BETWEEN 0.10 AND 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 ORDER WHERE ODE BETWEEN ‘1994-10-03’ AND ‘1994-10-04’

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

=> SELECT \* FROM SALESPERSON WHERE SNAME=’PEEL’ OR SNAME=’MOTIKA’

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

=> Select CNAME 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 CITY=’ROME’ AND RATING<=100

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

=> SELECT \* FROM ORDER WHERE AMT IS NULL OR 0

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

=> SELECT SNAME, COUNT(\*) AS Number\_Of\_Orders FROM SALESPERSON INNER JOIN ORDER ON SALESPERSON.SNO = ORDER.SNO GROUP BY ORDER.SNO