**Topic – SQL Task-1**

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) JoinYear , month(Joining\_Date), day (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’ 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 Joining\_Date LIKE 'January';

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

=> SELECT Department, SUM(Salary) AS TotalSalary FROM employee GROUP BY Department ORDER BY Salary DESC;

g) Get department wise maximum salary from employee table order by salary ascending? =>SELECT Department, MAX(Salary) AS MaxSalary FROM Employee GROUP BY Depart ment ORDER BY MaxSalary 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,Incentive\_Amt FROM employee emp INNER JOIN Incentives I NC ON emp.Employee\_Id = Inc.Employee\_Ref\_Id AND Incentive\_Amt> 3000;

i) Select 2nd Highest salary from employee table. =>Select Max(Salary) as Salary from employee where Salary <(select MAX(Sala ry) 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,NVL(Incentive\_Amt,0) FROM employee A RIGHT JOIN Incenti ves B ON A.Employee\_Id=B.Employee\_Ref\_Id;

k) Create View OF Employee table in which store first name, last name and salary only. =>CREATE VIEW Employee\_View AS SELECT Employee\_Id, First\_Name, Last\_Name , Salary FROM Employee;

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

=> CREATE PROCEDURE Find\_Department\_Highest\_Salary() BEGIN DECLARE done INT DEFAULT FALSE; DECLARE Department\_id INT; DECLARE Max\_Salary DECIMAL(10, 2);

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

=> CREATE TRIGGER AfterEmployeeInsert AFTER INSERT ON Employee FOR EACH ROW BEGIN -- Inserting into the EmployeeView table INSERT INTO EmployeeView (Employee\_ID, First\_Name, Last\_Name, Department) VALUES (NEW.EmployeeID, NEW.FirstName, NEW.LastName, NEW.Department); END;

**Topic – SQL Task-2**

a) All orders for more than $1000.

=> SELECT \* FROM `order` WHERE O\_Amt>1000;

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

=>SELECT S\_Name , City FROM `sales\_person` WHERE Comm > 0.10 AND City ='LON DON';

c) All salespeople either in Barcelona or in London. =>SELECT S\_Name , City FROM `sales\_person` WHERE City IN ('BARCELONA','LOND ON');

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

=>SELECT S\_Name , Comm FROM `sales\_person` WHERE Comm > 0.10 AND Comm < 0.1 2;

e) All customers with NULL values in city column.

=> SELECT C\_Name FROM `customer` WHERE City IS NULL;

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

=> SELECT \* FROM `order` WHERE O\_Date IN ('03-10-1994','04-10-1994');

g) All customers serviced by peel or Motika.

=> SELECT \* FROM customer WHERE S\_Num = 1001 OR S\_Num = 1004;

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

=>SELECT C\_Name FROM `customer` WHERE C\_Name LIKE 'A%' OR C\_Name LIKE 'B%';

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

=> SELECT C\_Name FROM `customer` WHERE Rating <= 100 OR City = 'ROME';

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

=> SELECT O\_Num FROM `order` WHERE O\_Amt != 0 OR O\_Amt IS NOT NULL;

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

=> SELECT COUNT(DISTINCT S\_Num) AS num\_active\_sales\_person FROM `order` WHERE order\_status = 'Active';