

## **Topic – SQL Task-1**

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

```
SELECT firstname from employee as Employee_Name;
```

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

```
SELECT firstname,  
    YEAR(joiningdate) AS JOINING_YEAR,  
    MONTH(joiningdate) AS JOINING_MONTH,  
    DAY(joiningdate) AS JOINING_DAY  
FROM employee;
```

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

```
SELECT * FROM employee ORDER BY firstname;  
SELECT * FROM employee ORDER BY salary DESC;
```

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

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

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

```
SELECT * from employee where joiningdate like '%-01-%';
```

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

```
SELECT sum(salary), department FROM employee GROUP BY department DESC;
```

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

**SELECT max(salary), department FROM employee GROUP BY department;**

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

**SELECT employee.firstname FROM employee JOIN incentives ON employee.employee\_id=incentives.employee\_ref\_id WHERE incentive\_amt > 3000;**

i) Select 2nd Highest salary from employee table.

**SELECT max(salary), firstname FROM employee AS 2nd\_Highest\_Salary 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 employee.firstname, incentives.incentive\_amt FROM employee LEFT JOIN incentives ON employee.employee\_id=incentives.employee\_ref\_id**

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

**SELECT firstname,lastname, salary from employee;**

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

**SELECT max(salary), department FROM employee GROUP BY department;**

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

**NA**

## **Topic – SQL Task-2**

- a) All orders for more than \$1000.

```
SELECT AMT FROM order_ WHERE AMT>1000;
```

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

```
SELECT sname,city,comm FROM sales_person WHERE city="London" AND  
comm>0.10;
```

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

```
SELECT sname,city FROM sales_person WHERE city="London" OR city="Barcelona";
```

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

```
SELECT * FROM sales_person WHERE comm > 0.10 AND comm <= 0.12;
```

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

```
SELECT cname,city FROM customer WHERE city IS NULL;
```

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

```
SELECT * FROM tbl_order WHERE ODE BETWEEN '1994-10-03' AND '1994-10-04';
```

g) All customers serviced by peel or Motika.

**SELECT cname,sno FROM customer WHERE sno=1001 OR sno=1004;**

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  $\leq 100$  unless they are located in Rome.

**SELECT cname,city,rating FROM customer WHERE rating $\leq$ 100 AND city="rome";**

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

**SELECT ONM FROM tbl\_order WHERE AMT IS NOT NULL;**

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

**SELECT COUNT (DISTINCT SalesPerson) AS Total\_Sales\_Person FROM tbl\_order;**