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 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 =’2013-01-01’.

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

=>SELECT \* FROM EMPLOYEE ORDER BY SALARY DESC,

DEPARTMENT.

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

=>select department , 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 EMPLOYEE. FIRST\_NAME , INENTIVE\_AMT FROM

EMPLOYEE LEFT JOIN INCENTIVES ON EMPLOYEE.EM\_ID

= INCENTIVE\_ID ORDER BY EMPLOYEE.FIRST\_NAME.

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

=>select first\_name,last\_name, salary from employee.

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

=>

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

(2) SQL(TASK)

a) All orders for more than $1000.

=>SELECT \*FROM 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;

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

=>SELECT \* FROM SALESPERSON WHERE CITY IS NULL.

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

=> SELECT \*FROM ORDERS WHERE ODE BETWEEN ‘1994-10-3’ 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 ORDERS WHERE AMT IS NULL OR 0.

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

=> SELECT COUNT (DISTINCT SNUM) FROM ORDERS.

**ss**