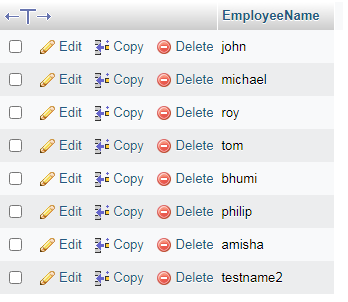
**Topic – SQL Task -1**

1. Get First Name from employee table using alias name “Employee Name”.

* SELECT first\_name AS EmployeeName FROM employee;



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

employee table.

* SELECT first\_name,joining\_year,joining\_month,joining\_date FROM employee;



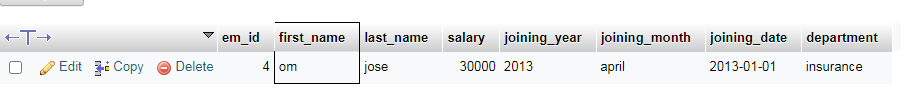
c) Get all employee details from the employee table order by First Name Ascending And Salary descending?

* 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\_month = 'january';



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

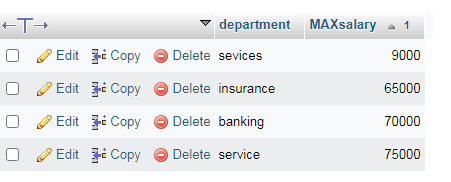
->SELECT \* FROM employee ORDER BY department, salary DESC;



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

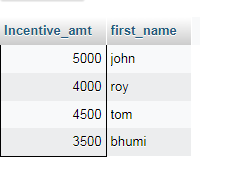
Ascending.

-> SELECT department, MAX(salary) MAXsalary FROM employee GROUP BY department 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

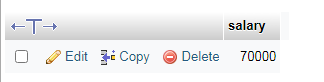
-> SELECT incentives.Incentive\_amt, employee.first\_name FROM incentives INNER JOIN employee ON incentives.em\_id = employee.em\_id WHERE Incentive\_amt > 3000;



i) Select 2nd Highest salary from employee table.

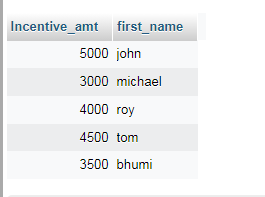
->

select salary from employee order by salary desc limit 1,1;



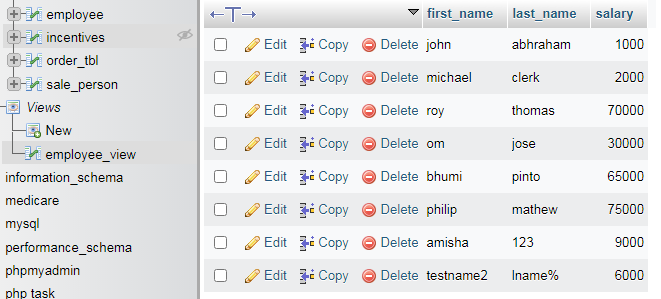
j) Select first\_name, incentive amount from employee and incentives table for all Employees who got incentives using left join.

-> SELECT incentives.Incentive\_amt, employee.first\_name FROM incentives LEFT JOIN employee ON incentives.em\_id = employee.em\_id;



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

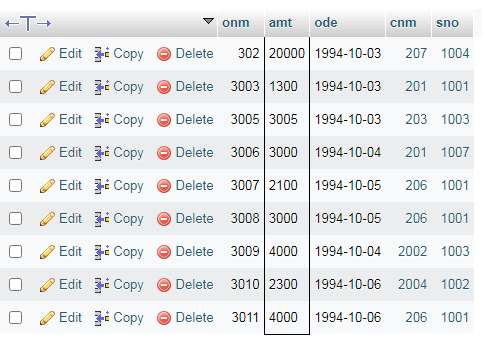
-> CREATE VIEW employee\_view as SELECT first\_name,last\_name,salary FROM employee;



**Task – 2**

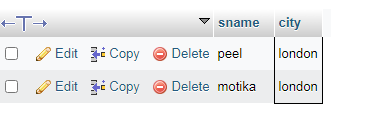
1. All orders for more than $1000.

-> select \* from orde\_tbl WHERE amt > 1000;



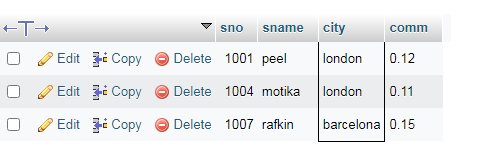
1. Names and cities of all salespeople in London with commission above 0.10.

-> Select sname, city from sale\_person where comm > 0.10 and city = 'london';



1. All salespeople either in Barcelona or in London.

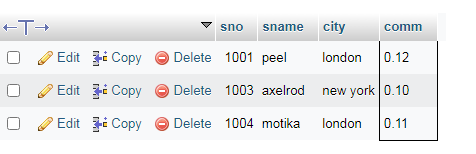
-> SELECT \* from sale\_person WHERE city='barcelona' OR city='london';



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

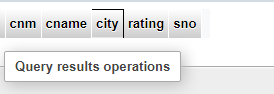
->

SELECT \* from sale\_person WHERE comm BETWEEN 0.10 AND 0.12;



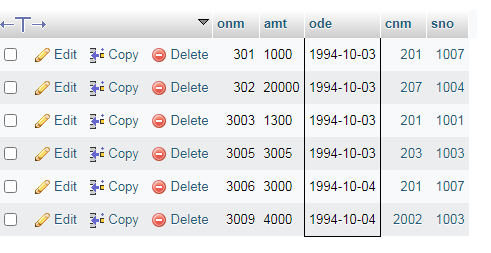
1. All customers with NULL values in city column.

-> Select \* from customer where city is null;



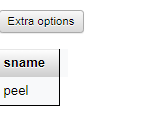
1. All orders taken on Oct 3Rd and Oct 4th 1994.

-> Select \* from order\_tbl WHERE ode IN ('1994-10-03','1994-10-04');



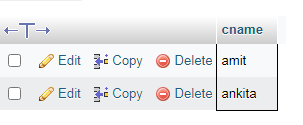
1. All customers serviced by peel or Motika.

-> SELECT sale\_person.sname FROM customer INNER JOIN sale\_person ON customer.sno = sale\_person.sno WHERE sname = 'peel' or 'motika';



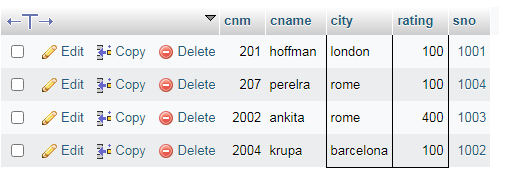
1. All customers whose names begin with a letter from A to B

-> Select cname from customer WHERE cname like 'a%' or cname like 'b%';



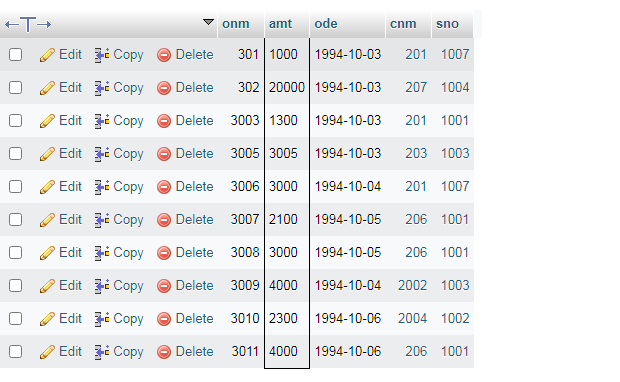
1. All customers excluding those with rating <= 100 unless they are located in Rome.

-> Select \* from customer WHERE rating <= 100 or city = 'rome';



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

-> Select \* from order\_tbl WHERE amt != 0 or amt is not null;



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

-> Select count(DISTINCT sno) from order\_tbl;

