**Topc-SQL Task-1**

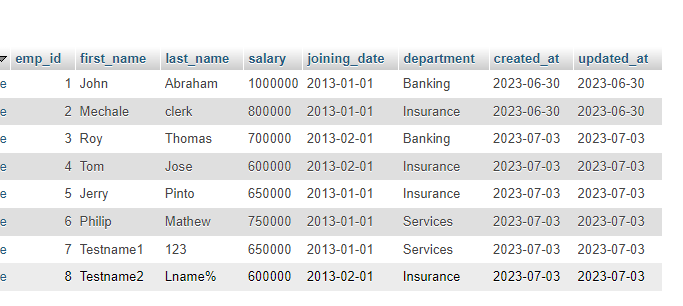
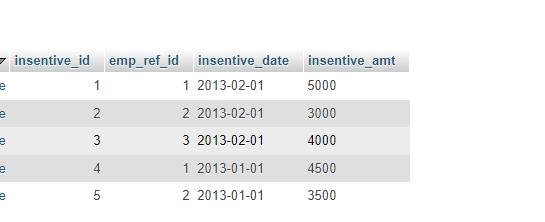
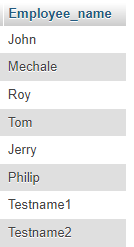
TABLE-Employee 

TABLE-Incentives



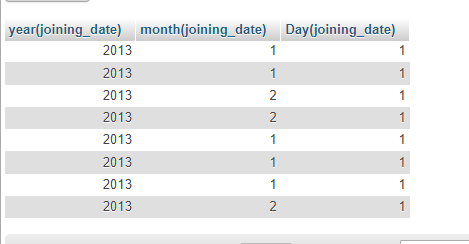
1. Get First\_Name from employee table using alias name “Employee Name”.

* select first\_name as Employee\_name from employee;



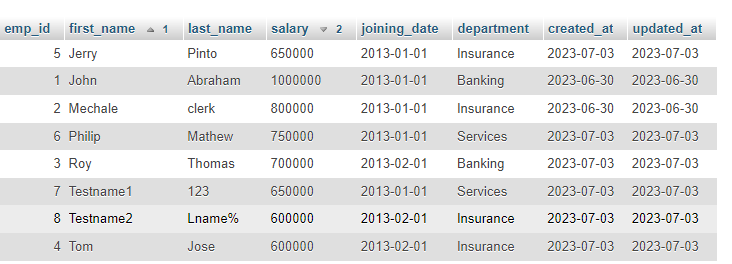
1. Get FIRST\_NAME, Joining year, Joining Month and Joining Date from employee table.

* Select year(joining\_date),month(joining\_date),Day(joining\_date) from employee;



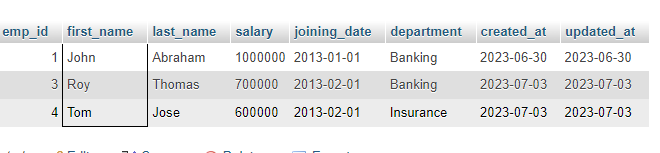
1. Get all employee details from the employee table order by First Name Ascending And Salary descending?

* Select \*from employee order by first\_name asc, salary desc;



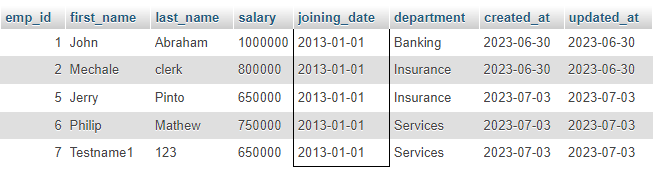
1. Get employee details from employee table whose first name contains „o‟.

* Select \* from employee where first\_name like’%o%’;



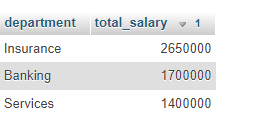
e) Get employee details from employee table whose joining month is “January”.

=> Select \*from employee where month(joining-date)=”01”;



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

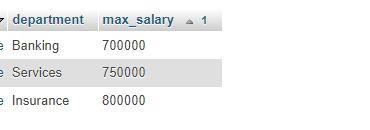
=> Select \* department ,sum(salary) total\_salary from employee group by department order by total\_salary desc;



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

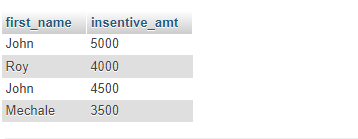
=> Select department max(salary) max\_salary from employee group by department order by max\_

Salary ASC;



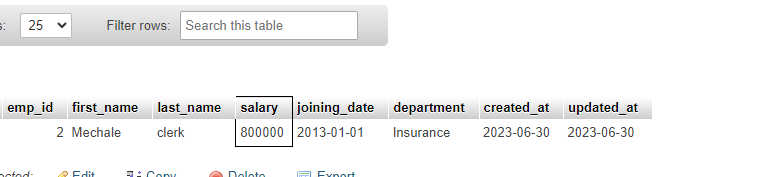
h) Select first\_name, incentive amount from employee and incentives table for those Employees who have incentives and incentive amount greater than 3000

=> select first\_name ,incentive\_amt from employee A inner join incentive B on A.emp\_id=B.emp\_ref\_id and incentive\_amt>3000;



i) Select 2nd Highest salary from employee table.

=> select \* 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,insentive amount from employee A left join incentives B on A.emp\_id=B.emp\_ref\_id;



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

=>Create view employeeview as

Select first\_name,last\_name,salary from employee;



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

=>create procedure department\_sp

as

begin

select Department, max(salary) Highest\_salary from emp group by Department;

end

exec department\_sp;



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

=>CREATE TRIGGER EmployeeInsertTrigger

AFTER INSERT ON employee

FOR EACH ROW

BEGIN

INSERT INTO EmployeeView (first\_name, last\_name, salary)

VALUES (NEW.first\_name, NEW.last\_name, NEW.salary);

END

Task-2:

TABLE- SALES PERSON:

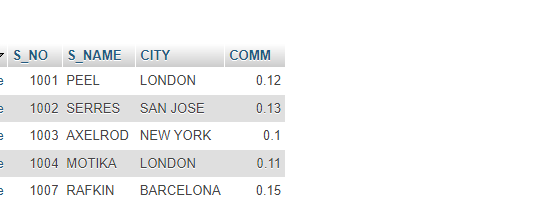


TABLE-CUSTOMER:

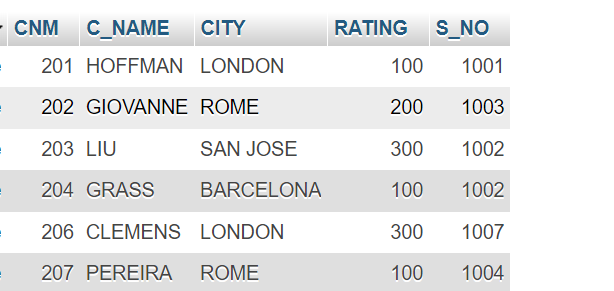
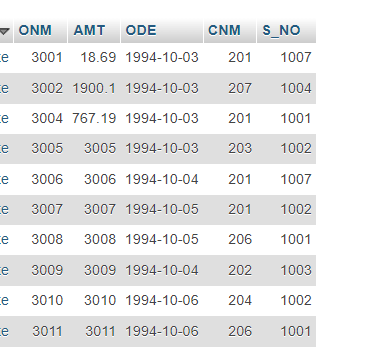
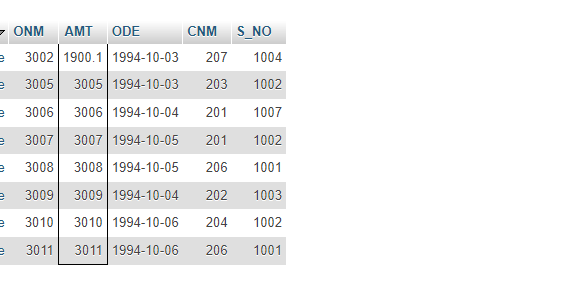


TABLE-ORDER:



1. 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 SALES PERSON WHERE COMM>0.10 AND CITY=”LONDON”;



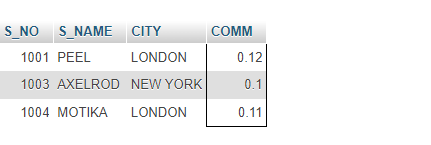
C)All salespeople either in Barcelona or in London.

=> SELECT \* 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 BETWEEN 0.10 AND 0.12 ;



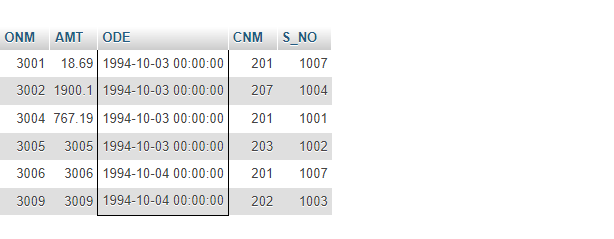
e)All customers with NULL values in city column.

=> SELECT \* FROM `customer` WHERE CITY IS NULL;



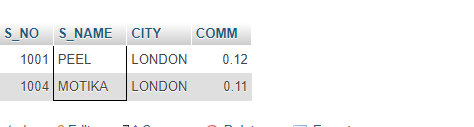
f) All orders taken on Oct 3Rd and Oct 4th 1994.

=>SELECT \* FROM `order` WHERE ODE in('1994-10-3','1994-10-4');



g)All customers serviced by peel or Motika.

=> SELECT \* FROM `sales person` WHERE S\_NAME="PEEL" or S\_NAME="MOTIKA";



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";

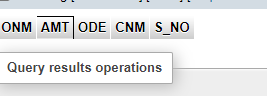


j) All orders except those with 0 or NULL value in amt field.

=> SELECT \* FROM `order` WHERE AMT is null or 0;

or

=>select \*from order where AMT <> 0 AND AMT is NULL;



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

=> SELECT COUNT(DISTINCT S\_NO) FROM `order`;

