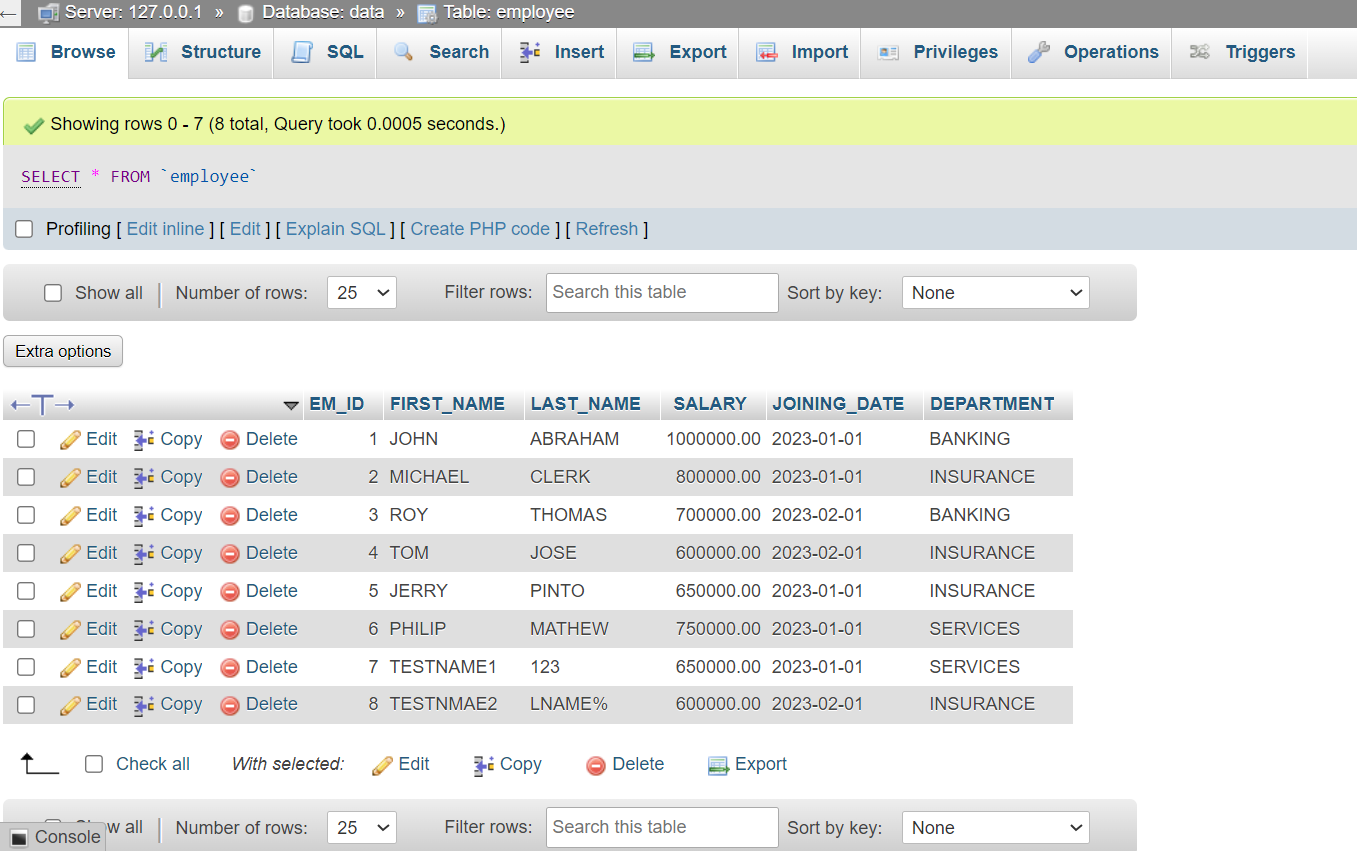
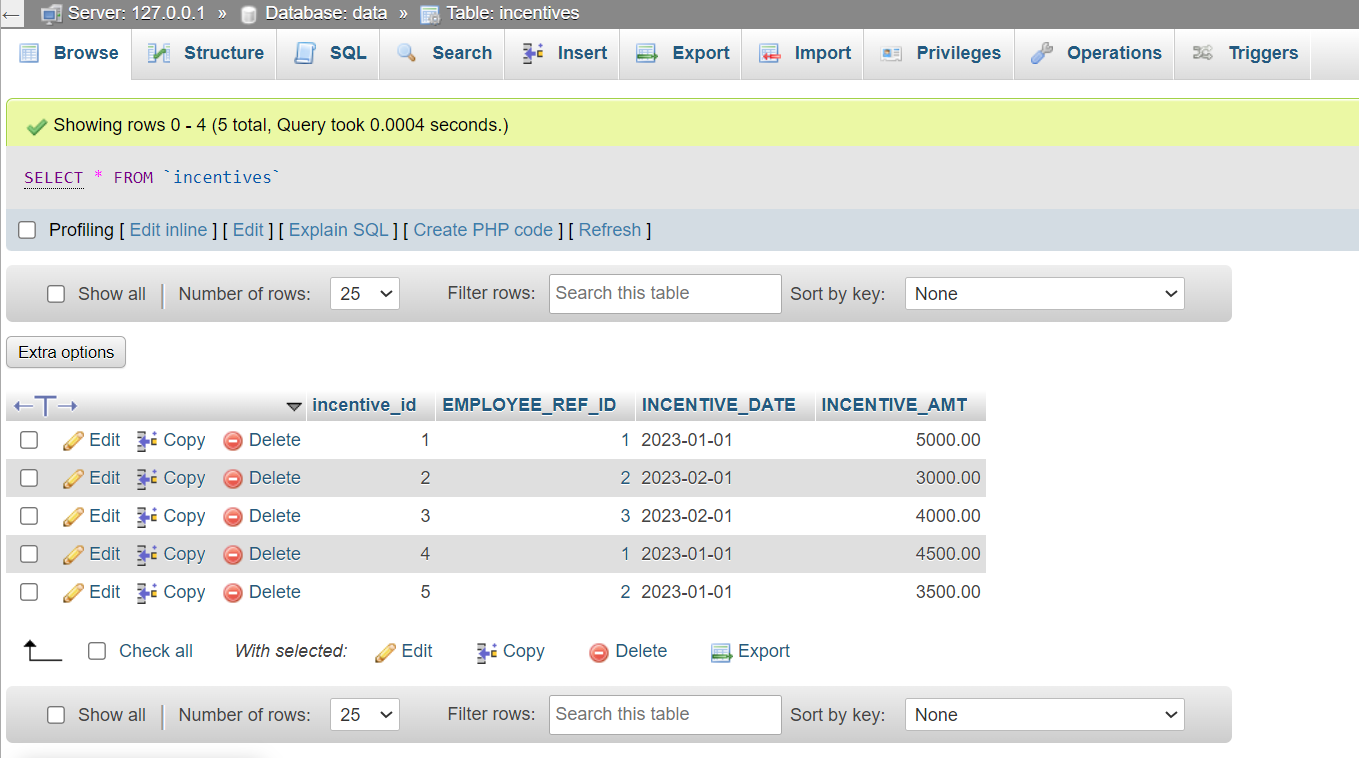
**TOPIC – SQL**

Task – 1

**TABLE 1 : - Employee**



**TABLE 2 :- Incentives**



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

Ans.SELECT First\_Name AS “Employee Name” FROM employee;

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

Ans. SELECT FIRST\_NAME, YEAR(Joining\_Date) AS Joining\_Year, MONTH(Joining\_Date) AS Joining\_Month, DAY(Joining\_Date) AS Joining\_Date FROM employee;

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

Ans. SELECT \* FROM employee ORDER BY First\_Name ASC, Salary DESC;

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

Ans. SELECT \* FROM employee WHERE First\_Name LIKE '%o%';

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

Ans. SELECT \* FROM employee WHERE MONTH(Joining\_Date) = 1;

f) Get department, total salary with respect to a department from employee table

Order By total salary descending.

Ans. SELECT Department, SUM(Salary) AS 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?

Ans. SELECT Department, MAX(Salary) AS 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

Ans. SELECT e.first\_name, i.incentive\_amount

FROM employee e

JOIN incentives i ON e.employee\_id = i.employee\_id

WHERE i.incentive\_amount > 3000;

i) Select 2nd Highest salary from employee table.

Ans. SELECT DISTINCT Salary FROM employee ORDER BY Salary DESC LIMIT 1 OFFSET 1;

j) Select first\_name, incentive amount from employee and incentives table for all

Employees who got incentives using left join.

Ans. SELECT e.first\_name, i.incentive\_amount

FROM employee e

LEFT JOIN incentives i ON e.employee\_id = i.employee\_id;

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

Ans. CREATE VIEW Employee\_View AS

SELECT First\_Name, Last\_Name, Salary FROM employee;

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

Ans. DELIMITER //

CREATE PROCEDURE DepartmentWiseHighestSalary()

BEGIN

SELECT Department, MAX(Salary) AS Highest\_Salary

FROM employee

GROUP BY Department;

END //

DELIMITER ;

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

Ans. CREATE TRIGGER InsertEmployeeView AFTER INSERT ON Employee

FOR EACH ROW

INSERT INTO Employee\_View (First\_Name, Last\_Name, Salary)

VALUES (NEW.First\_Name, NEW.Last\_Name, NEW.Salary);

Task – 2

**TABLE 1 : - SALES PERSON**

****

**TABLE 2 : - CUSTOMER**

****

**TABLE 3 : - ORDER**

****

a) All orders for more than $1000.

Ans. SELECT \* FROM orders WHERE amount > 1000;

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

Ans. SELECT name, city FROM salespeople WHERE city = 'London' AND commission > 0.10;

c) All salespeople either in Barcelona or in London.

Ans. SELECT \* FROM salespeople WHERE city IN ('Barcelona', 'London');

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

be excluded).

Ans. SELECT \* FROM salespeople WHERE commission > 0.10 AND commission < 0.12;

e) All customers with NULL values in city column.

Ans. SELECT \* FROM customers WHERE city IS NULL;

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

Ans. SELECT \* FROM orders WHERE order\_date >= '1994-10-03' AND order\_date <= '1994-10-04';

g) All customers serviced by peel or Motika.

Ans. SELECT \* FROM customers WHERE serviced\_by IN ('Peel', 'Motika');

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

Ans. SELECT \* FROM customers WHERE name LIKE 'A%' OR name LIKE 'B%';

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

Ans. SELECT \* FROM customers WHERE rating > 100 OR city = 'Rome';

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

Ans. SELECT \* FROM orders WHERE amt IS NOT NULL AND amt != 0;

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

Ans. SELECT COUNT(DISTINCT salesperson\_id) AS num\_salespeople FROM orders;