**ASSIGNMENT SQL :**

**TABLE : Employee**

CREATE TABLE Employee(EM\_ID INT PRIMARY KEY AUTO\_INCREMENT,FIRST\_NAME VARCHAR(255),LAST\_NAME VARCHAR(255),SALARY INT(10),JOINING\_DATE DATE,DEPARTMENT VARCHAR(255));

INSERT INTO employee(FIRST\_NAME,LAST\_NAME,SALARY,JOINING\_DATE,DEPARTMENT)VALUES("JOHN","ABRAHAM","1000000","2013-01-01","BANKING");

INSERT INTO employee(FIRST\_NAME,LAST\_NAME,SALARY,JOINING\_DATE,DEPARTMENT)VALUES("MICHAEL","CLERK","800000","2013-01-01","INSURANCE");

INSERT INTO employee(FIRST\_NAME,LAST\_NAME,SALARY,JOINING\_DATE,DEPARTMENT)VALUES("ROY","THOMAS","700000","2013-02-01","BANKING");

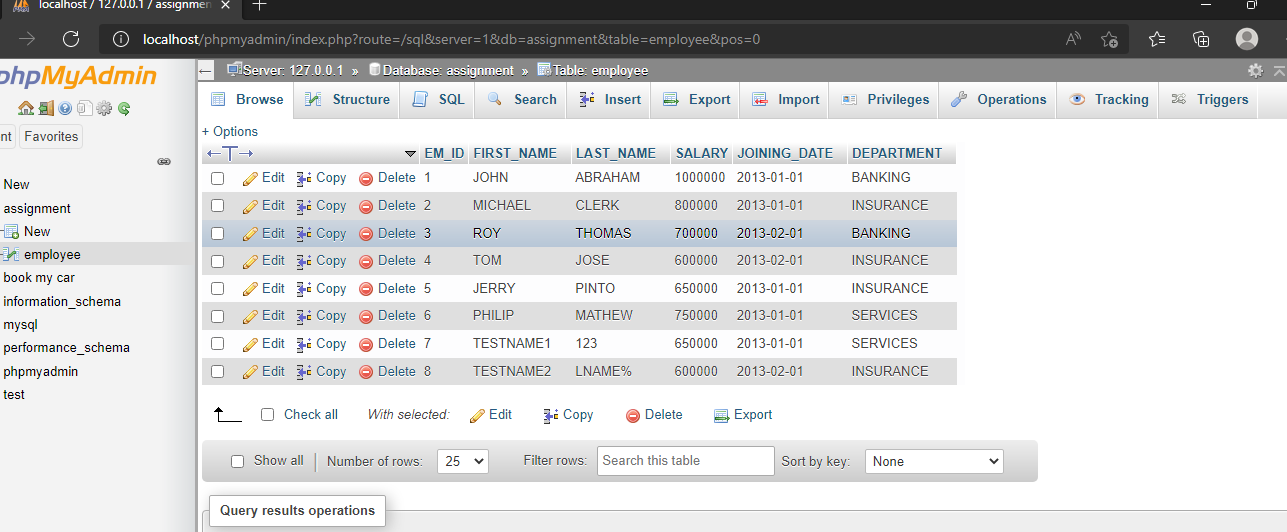
INSERT INTO employee(FIRST\_NAME,LAST\_NAME,SALARY,JOINING\_DATE,DEPARTMENT)VALUES("TOM","JOSE","600000","2013-02-01","INSURANCE");

INSERT INTO employee(FIRST\_NAME,LAST\_NAME,SALARY,JOINING\_DATE,DEPARTMENT)VALUES("JERRY","PINTO","650000","2013-01-01","INSURANCE");

INSERT INTO employee(FIRST\_NAME,LAST\_NAME,SALARY,JOINING\_DATE,DEPARTMENT)VALUES("PHILIP","MATHEW","750000","2013-01-01","SERVICES");

INSERT INTO employee(FIRST\_NAME,LAST\_NAME,SALARY,JOINING\_DATE,DEPARTMENT)VALUES("TESTNAME1","123","650000","2013-01-01","SERVICES");

INSERT INTO employee(FIRST\_NAME,LAST\_NAME,SALARY,JOINING\_DATE,DEPARTMENT)VALUES("TESTNAME2","LNAME%","600000","2013-02-01","INSURANCE");



**TABLE : INCENTIVE**

CREATE TABLE incentives(insentive\_id INT PRIMARY KEY AUTO\_INCREMENT,EMPLOYEE\_REF\_ID INT,INCENTIVE\_DATE DATE,INCENTIVE\_AMT INT,FOREIGN KEY(EMPLOYEE\_REF\_ID)REFERENCES employee(EM\_ID));

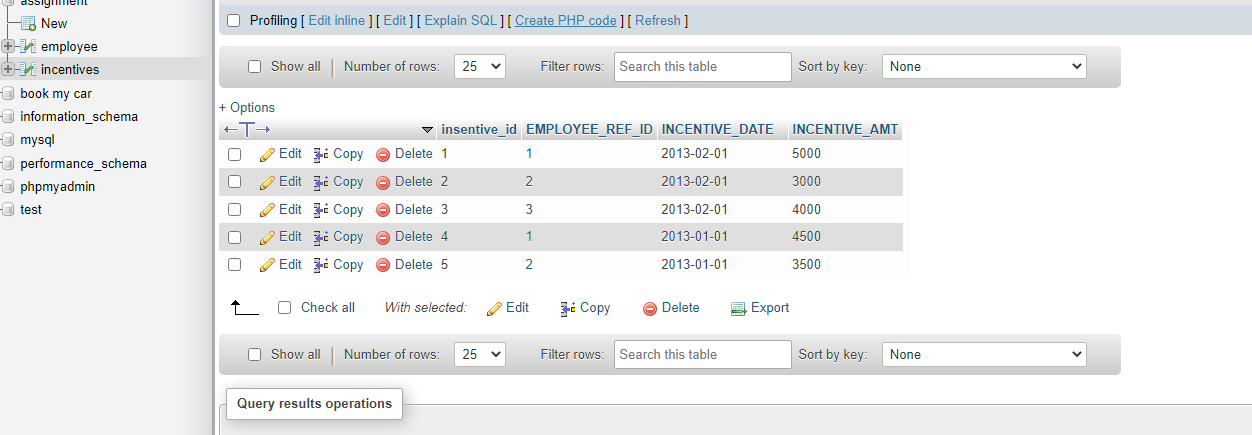
INSERT INTO incentives(EMPLOYEE\_REF\_ID,INCENTIVE\_DATE,INCENTIVE\_AMT)VALUES("1","2013-02-01","5000");

INSERT INTO incentives(EMPLOYEE\_REF\_ID,INCENTIVE\_DATE,INCENTIVE\_AMT)VALUES("2","2013-02-01","3000");

INSERT INTO incentives(EMPLOYEE\_REF\_ID,INCENTIVE\_DATE,INCENTIVE\_AMT)VALUES("3","2013-02-01","4000");

INSERT INTO incentives(EMPLOYEE\_REF\_ID,INCENTIVE\_DATE,INCENTIVE\_AMT)VALUES("1","2013-01-01","4500");

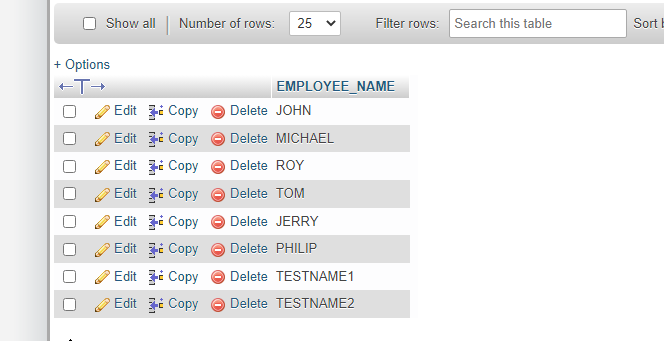
INSERT INTO incentives(EMPLOYEE\_REF\_ID,INCENTIVE\_DATE,INCENTIVE\_AMT)VALUES("2","2013-01-01","3500");



**B1).Get First\_Name from employee table using alias name “Employee Name”**.

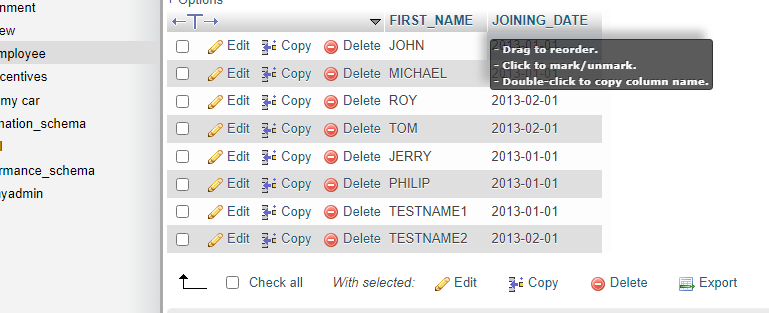
SELECT FIRST\_NAME AS EMPLOYEE\_NAME

FROM employee;



**B2).Get FIRST\_NAME, Joining year, Joining Month and Joining Date from employee table.**

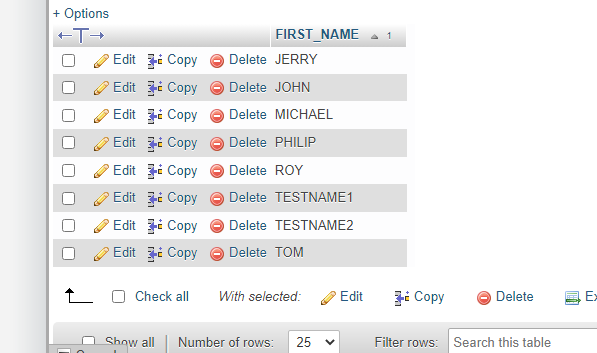
SELECT FIRST\_NAME,JOINING\_DATE FROM employee;



**B3).Get all employee details from the employee table order by First\_Name Ascending and Salary descending.**

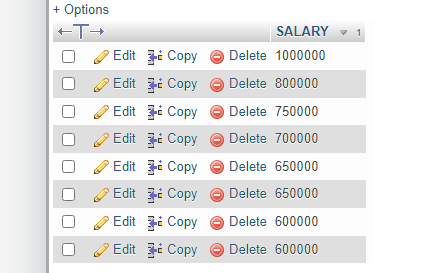
SELECT FIRST\_NAME FROM employee

ORDER BY FIRST\_NAME ASC;



SELECT SALARY FROM employee

ORDER BY SALARY DESC;



**B4).Get employee details from employee table whose first name contains ‘o’.**

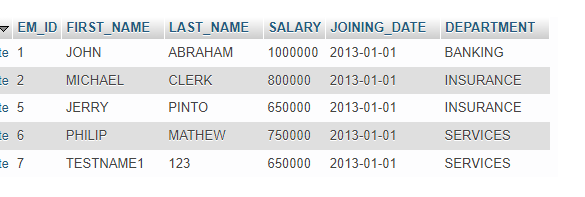
SELECT \* FROM employee

WHERE FIRST\_NAME LIKE '%O%';



**I1).Get employee details from employee table whose joining month is “January”.**

Select \* from EMPLOYEE where month(joining\_date)='01';

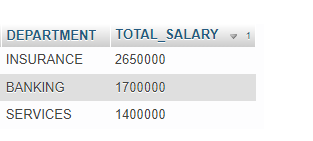


**I2).Get department, total salary with respect to a department from employee table order by total salary descending.**

SELECT DEPARTMENT,SUM(SALARY) TOTAL\_SALARY

FROM EMPLOYEE GROUP BY DEPARTMENT

ORDER BY TOTAL\_SALARY DESC;

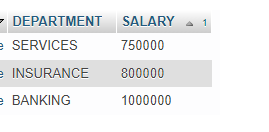


**I3).Get department wise maximum salary from employee table order by salary ascending.**

SELECT DEPARTMENT,MAX(SALARY) SALARY

FROM EMPLOYEE GROUP BY DEPARTMENT

ORDER BY SALARY ASC;



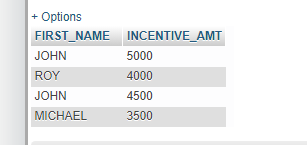
**I4).Select first\_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000**

SELECT FIRST\_NAME,INCENTIVE\_AMT

FROM EMPLOYEE A INNER JOIN INCENTIVES B

ON A.EM\_ID=B.EMPLOYEE\_REF\_ID

AND INCENTIVE\_AMT >3000;

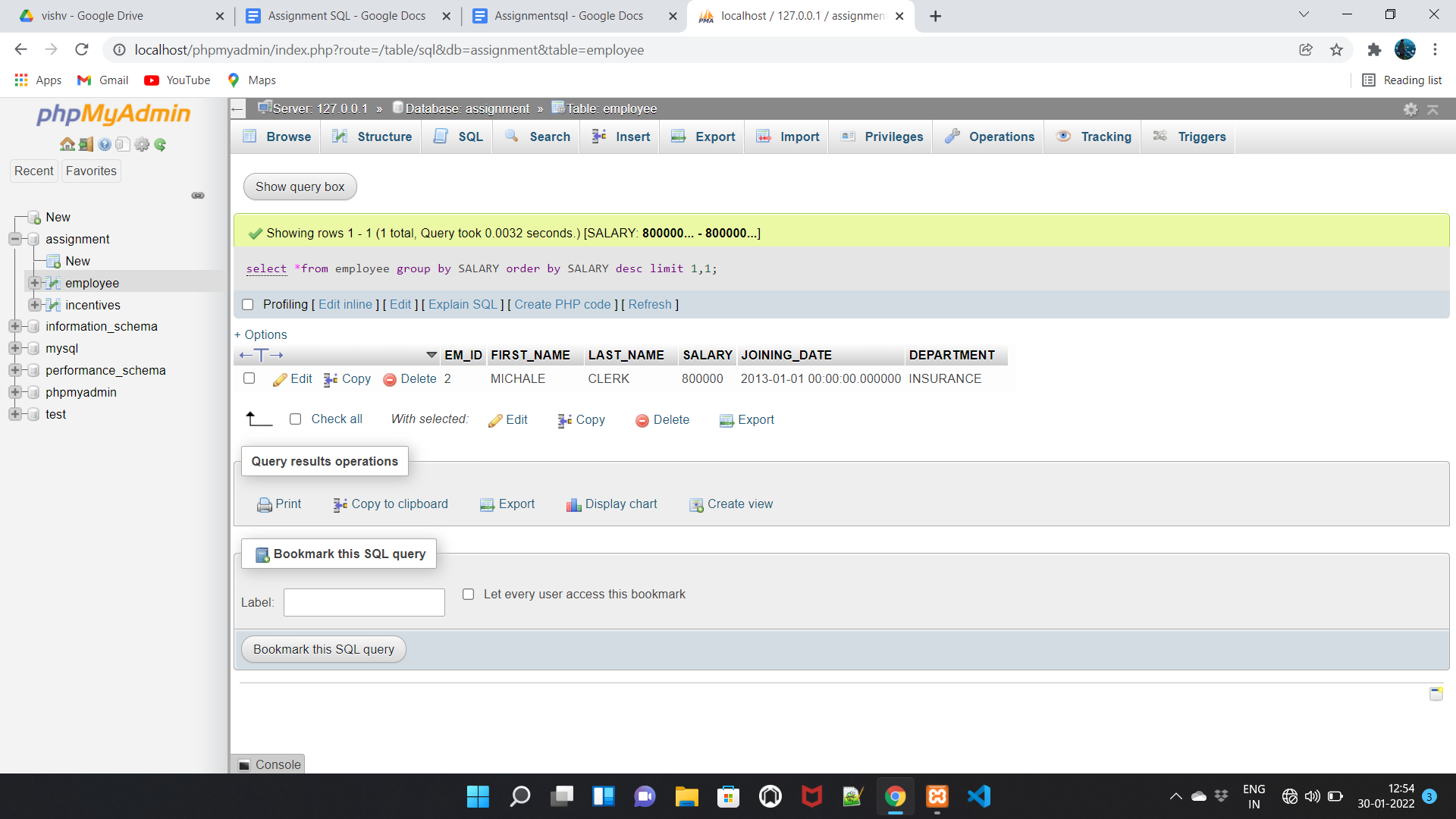


**A1).Select 2nd Highest salary from employee table.**

select \*from employee

group by SALARY

order by SALARY desc limit 1,1;

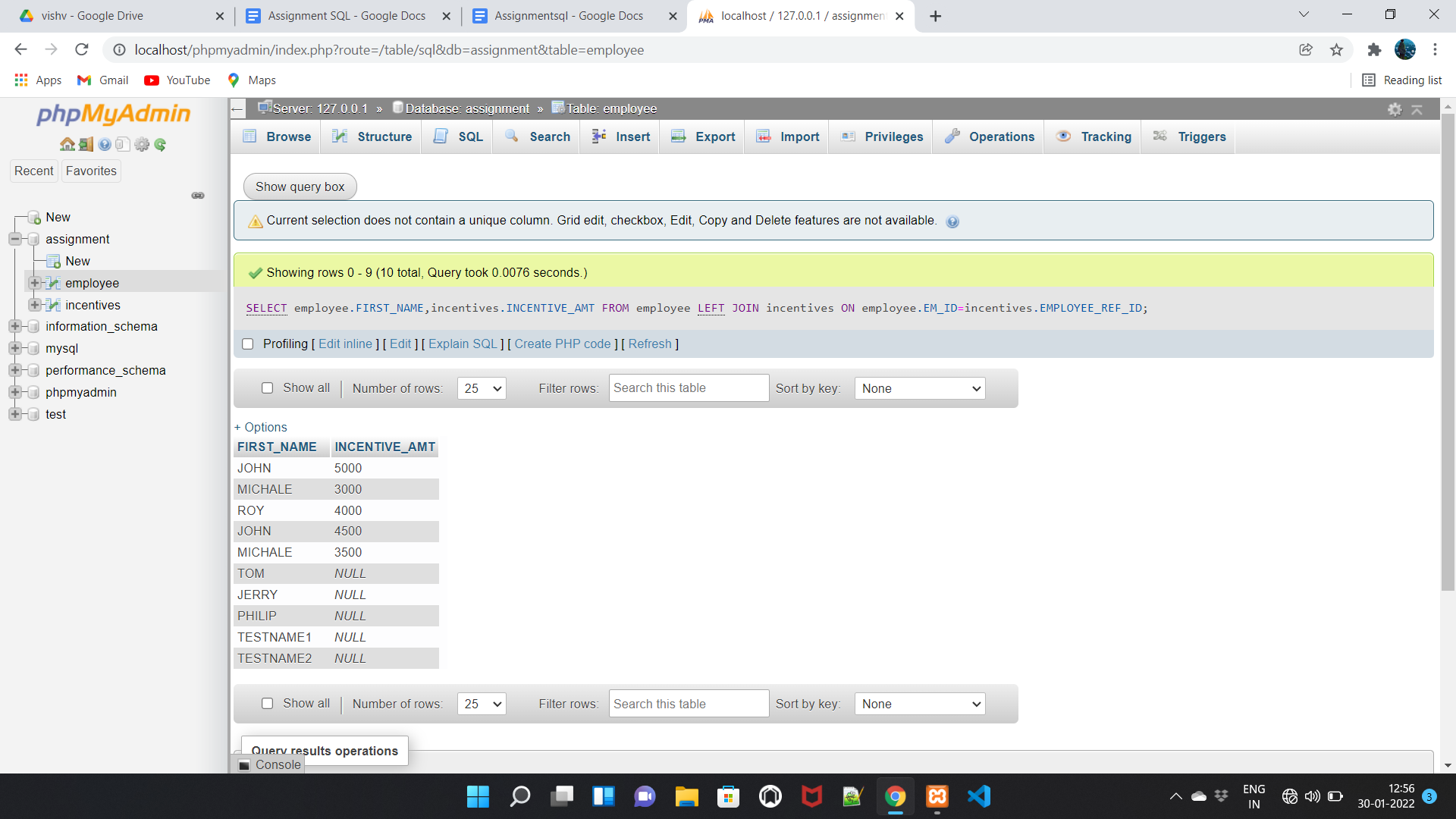


**A2. Select first\_name, incentive amount from employee and incentives table for all employees who got incentives using left join**

SELECT employee.FIRST\_NAME,incentives.INCENTIVE\_AMT

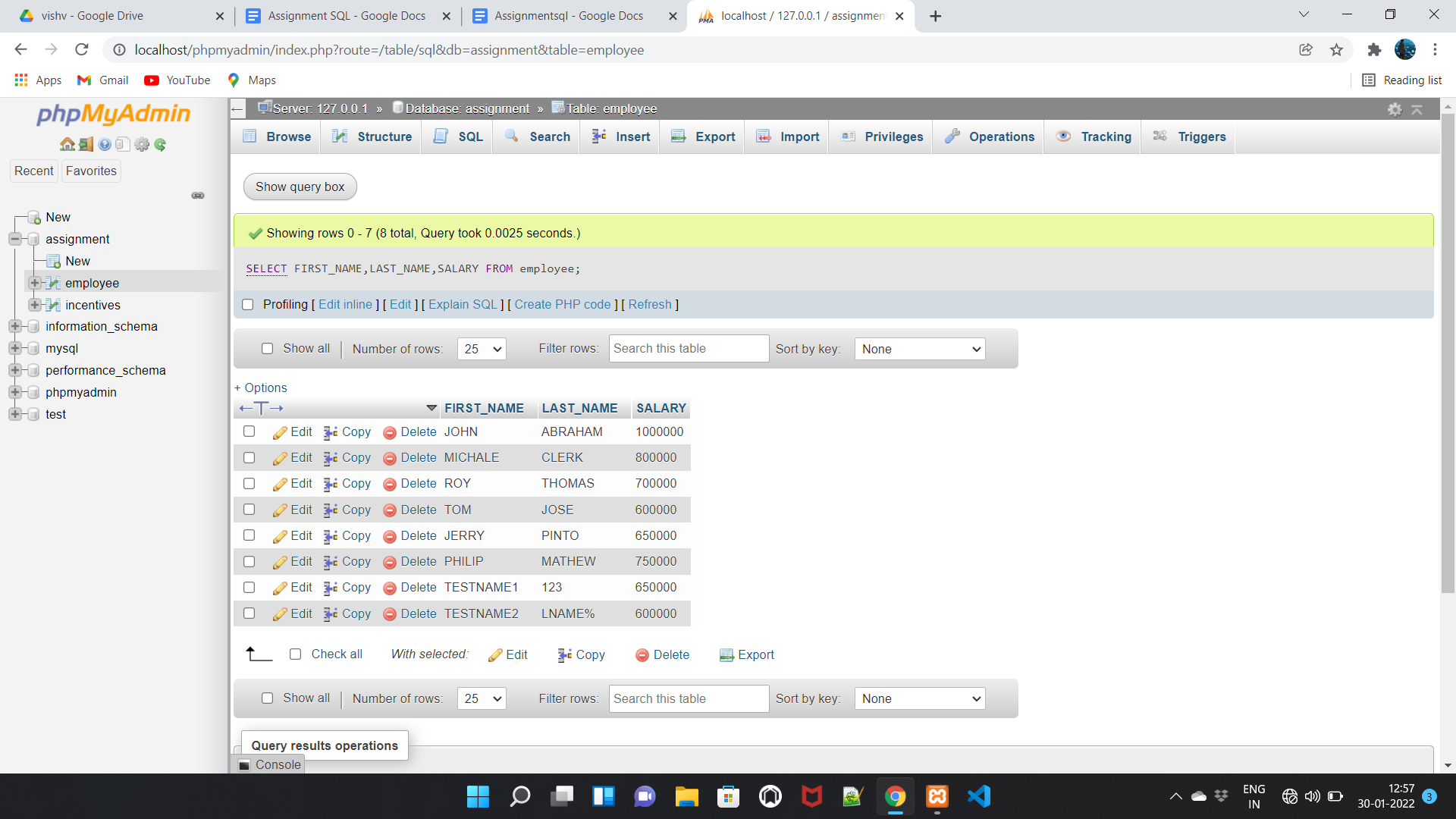
FROM employee LEFT JOIN incentives

ON employee.EM\_ID=incentives.EMPLOYEE\_REF\_ID;



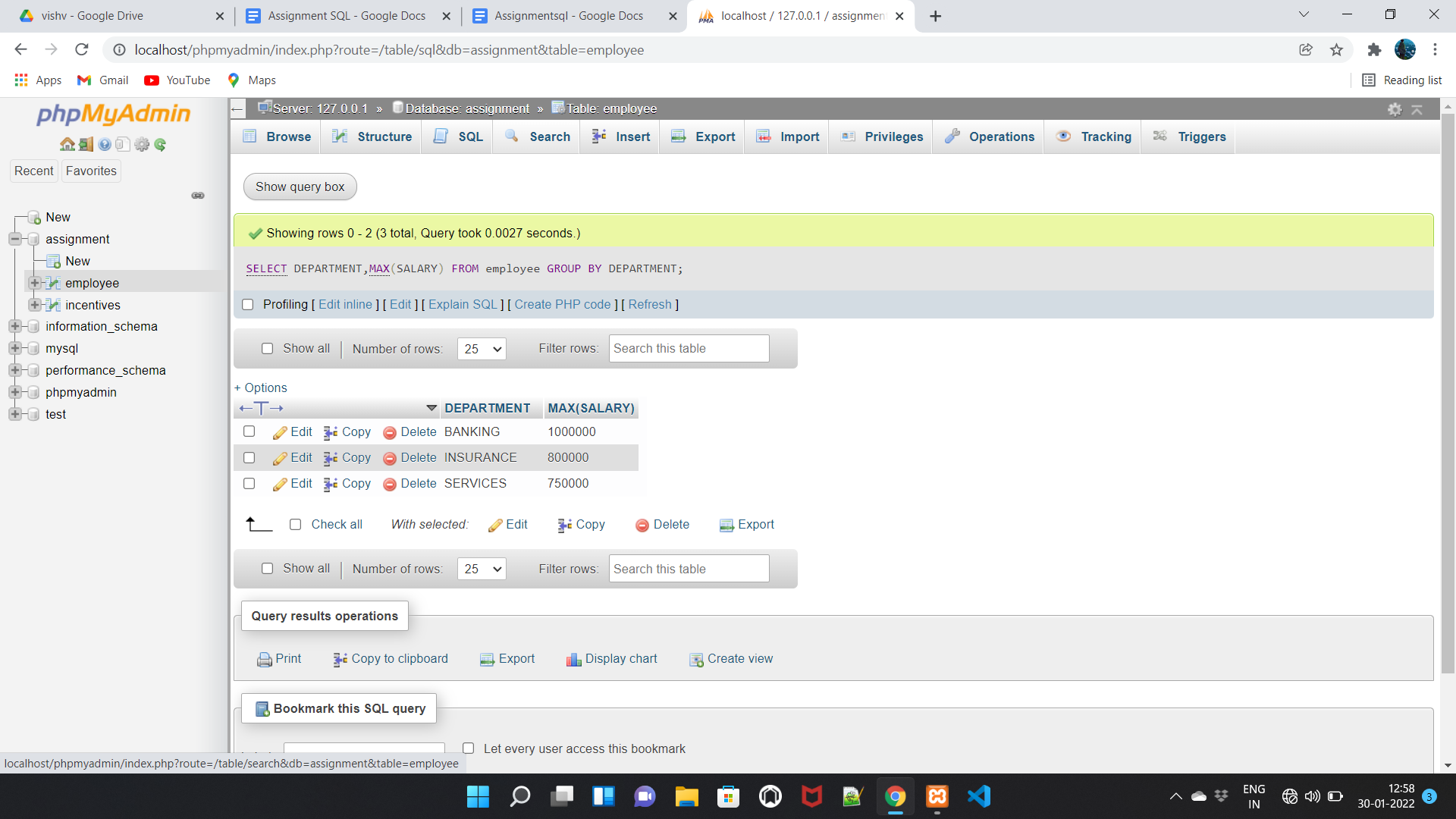
**A3. Create View OF Employee table in which store first name ,last name and salary only.**

SELECT FIRST\_NAME,LAST\_NAME,SALARY FROM employee;



**A4. Create Procedure to find out department wise highest salary.**

SELECT DEPARTMENT,MAX(SALARY) FROM employee GROUP BY DEPARTMENT;



**A5. Create After Insert trigger on Employee table which insert records in view table.**

**TASK:-2**

**TABLE : SALES PERSON**

CREATE TABLE SALES\_PERSON(SNO INT PRIMARY KEY AUTO\_INCREMENT,SNAME VARCHAR(255),CITY VARCHAR(255),COMM FLOAT);

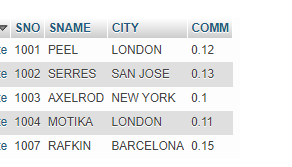
INSERT INTO sales\_person(SNO,SNAME,CITY,COMM)VALUES(“1001”,”PEEL”,”LONDON”,”0.12”);

INSERT INTO sales\_person(SNO,SNAME,CITY,COMM)VALUES("1002","SERRES","SAN JOSE","0.13");

INSERT INTO sales\_person(SNO,SNAME,CITY,COMM)VALUES("1003","AXELROD","NEW YORK","0.1");

INSERT INTO sales\_person(SNO,SNAME,CITY,COMM)VALUES("1004","MOTIKA","LONDON","0.11");

INSERT INTO sales\_person(SNO,SNAME,CITY,COMM)VALUES("1007","RAFKIN","BARCELONA","0.15");



**TABLE : CUSTOMER**

CREATE TABLE CUSTOMER(CNM INT PRIMARY KEY AUTO\_INCREMENT,CNAME VARCHAR(255),CITY VARCHAR(255),RATING INT,SNO INT, FOREIGN KEY(SNO)REFERENCES sales\_person(SNO));

INSERT INTO customer(CNM,CNAME,CITY,RATING,SNO)VALUES("201","HOFFMAN","LONDON","100","1001");

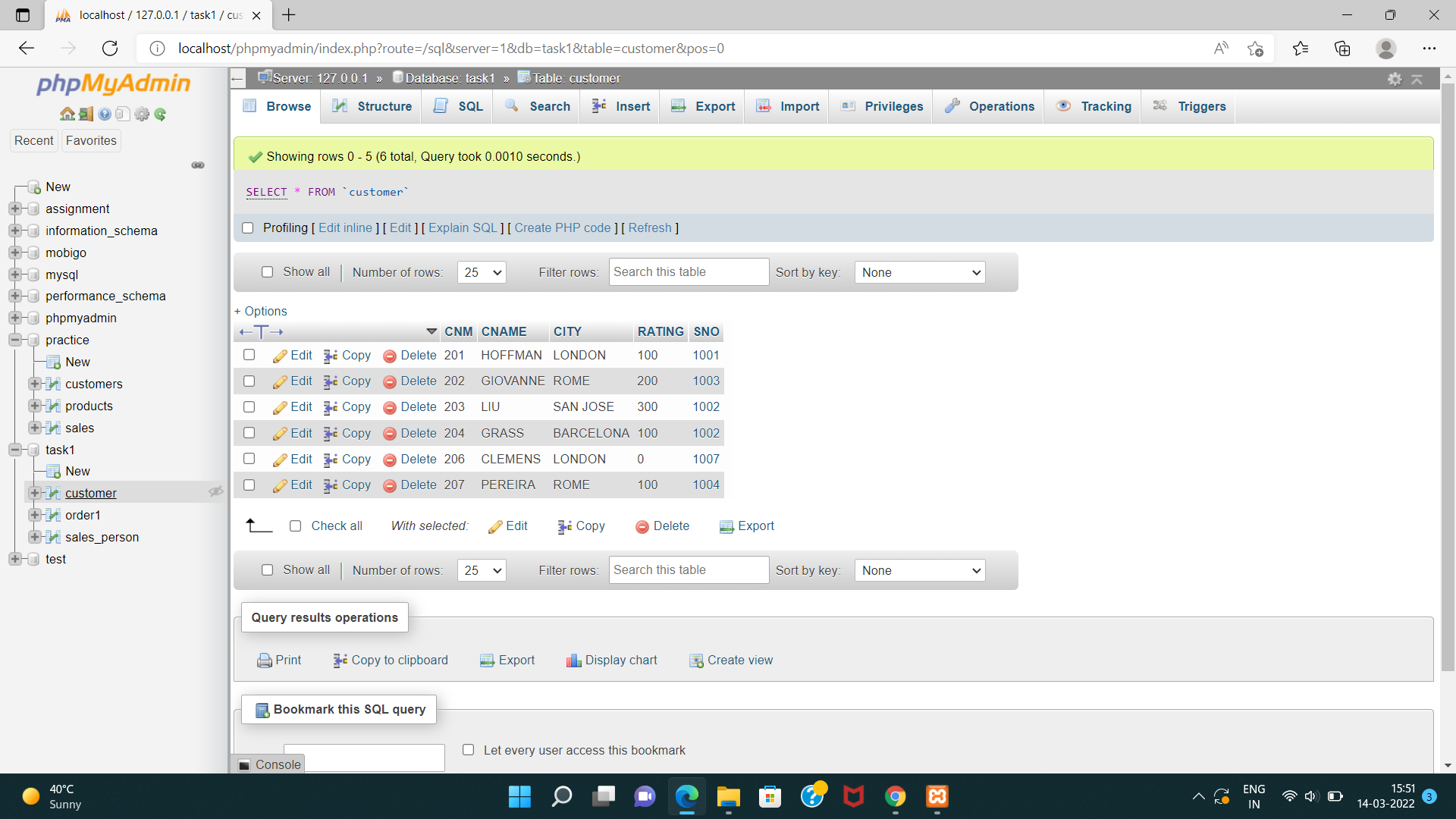
INSERT INTO customer(CNM,CNAME,CITY,RATING,SNO)VALUES("202","GIOVANNE","ROME","200","1003");

INSERT INTO customer(CNM,CNAME,CITY,RATING,SNO)VALUES("203","LIU","SAN JOSE","300","1002");

INSERT INTO customer(CNM,CNAME,CITY,RATING,SNO)VALUES("204","GRASS","BARCELONA","100","1002");

INSERT INTO customer(CNM,CNAME,CITY,RATING,SNO)VALUES("206","CLEMENS","LONDON","00","1007");

INSERT INTO customer(CNM,CNAME,CITY,RATING,SNO)VALUES("207","PEREIRA","ROME","100","1004");



**TABLE : ORDER1**

CREATE TABLE ORDER1(ONM INT PRIMARY KEY AUTO\_INCREMENT,AMT INT,ODE DATETIME,CNM INT,SNO INT,FOREIGN KEY(CNM)REFERENCES customer(CNM),FOREIGN KEY(SNO)REFERENCES sales\_person(SNO));

INSERT INTO order1(ONM,AMT,ODE,CNM,SNO)VALUES("3001","18.69","1994-10-03","201","1007");

INSERT INTO order1(ONM,AMT,ODE,CNM,SNO)VALUES("3002","1900.1","1994-10-03","207","1004");

INSERT INTO order1(ONM,AMT,ODE,CNM,SNO)VALUES("3003","767.19","1994-10-03","201","1001");

INSERT INTO order1(ONM,AMT,ODE,CNM,SNO)VALUES("3005","3005","1994-10-03","203","1002");

INSERT INTO order1(ONM,AMT,ODE,CNM,SNO)VALUES("3006","3006","1994-10-04","201","1007");

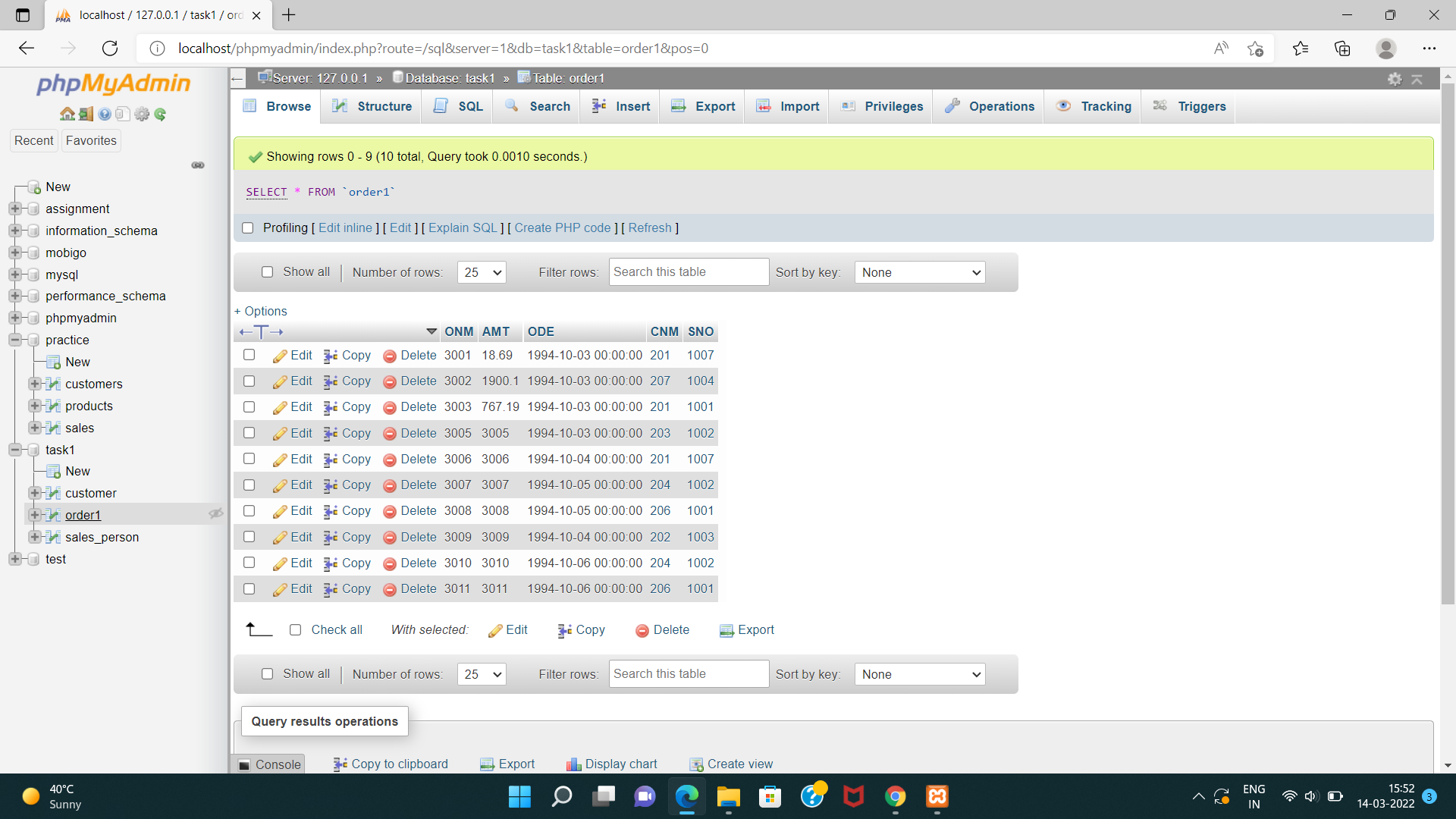
INSERT INTO order1(ONM,AMT,ODE,CNM,SNO)VALUES("3007","3007","1994-10-05","204","1002");

INSERT INTO order1(ONM,AMT,ODE,CNM,SNO)VALUES("3008","3008","1994-10-05","206","1001");

INSERT INTO order1(ONM,AMT,ODE,CNM,SNO)VALUES("3009","3009","1994-10-04","202","1003");

INSERT INTO order1(ONM,AMT,ODE,CNM,SNO)VALUES("3010","3010","1994-10-06","204","1002");

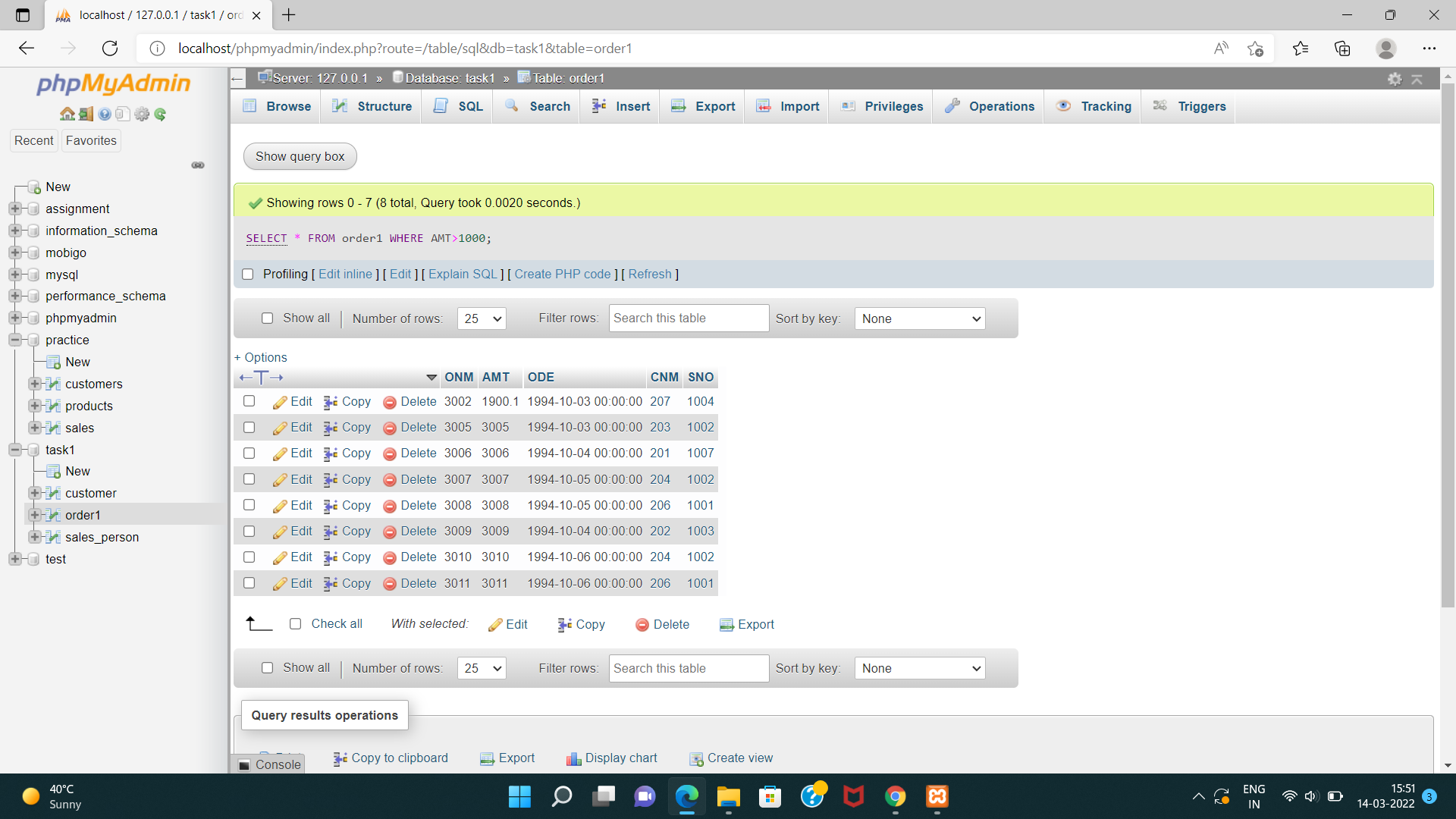
INSERT INTO order1(ONM,AMT,ODE,CNM,SNO)VALUES("3011","3011","1994-10-06","206","1001");



**QUERY**

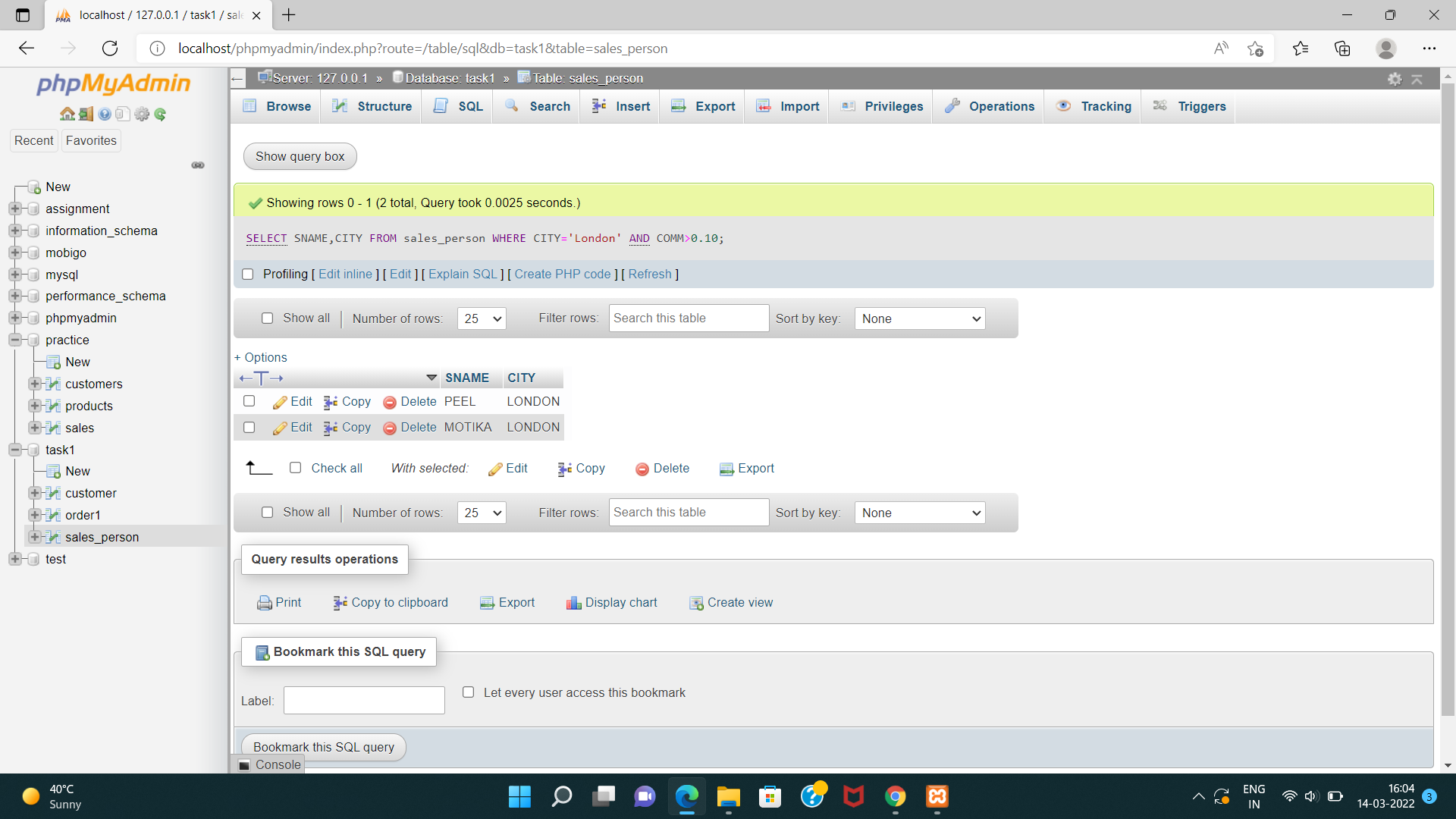
**B1) All orders for more than $1000**

SELECT \* FROM order1 WHERE AMT>1000;



**B2)Names and cities of all salespeople in London with commission above 0.10.**

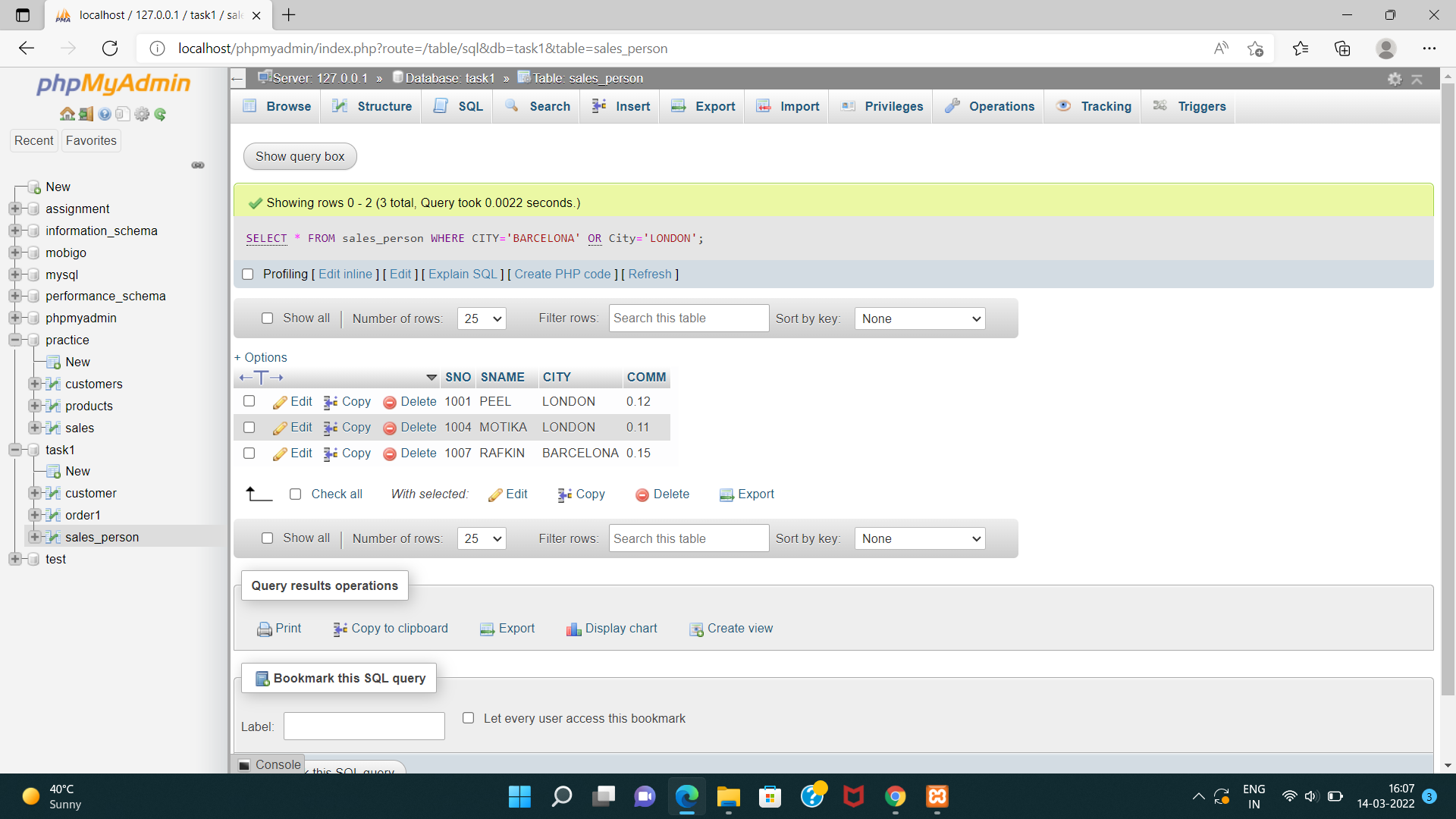
SELECT SNAME,CITY FROM sales\_person WHERE CITY='London' AND COMM>0.10;



**B3). All salespeople either in Barcelona or in London.**

SELECT \* FROM sales\_person

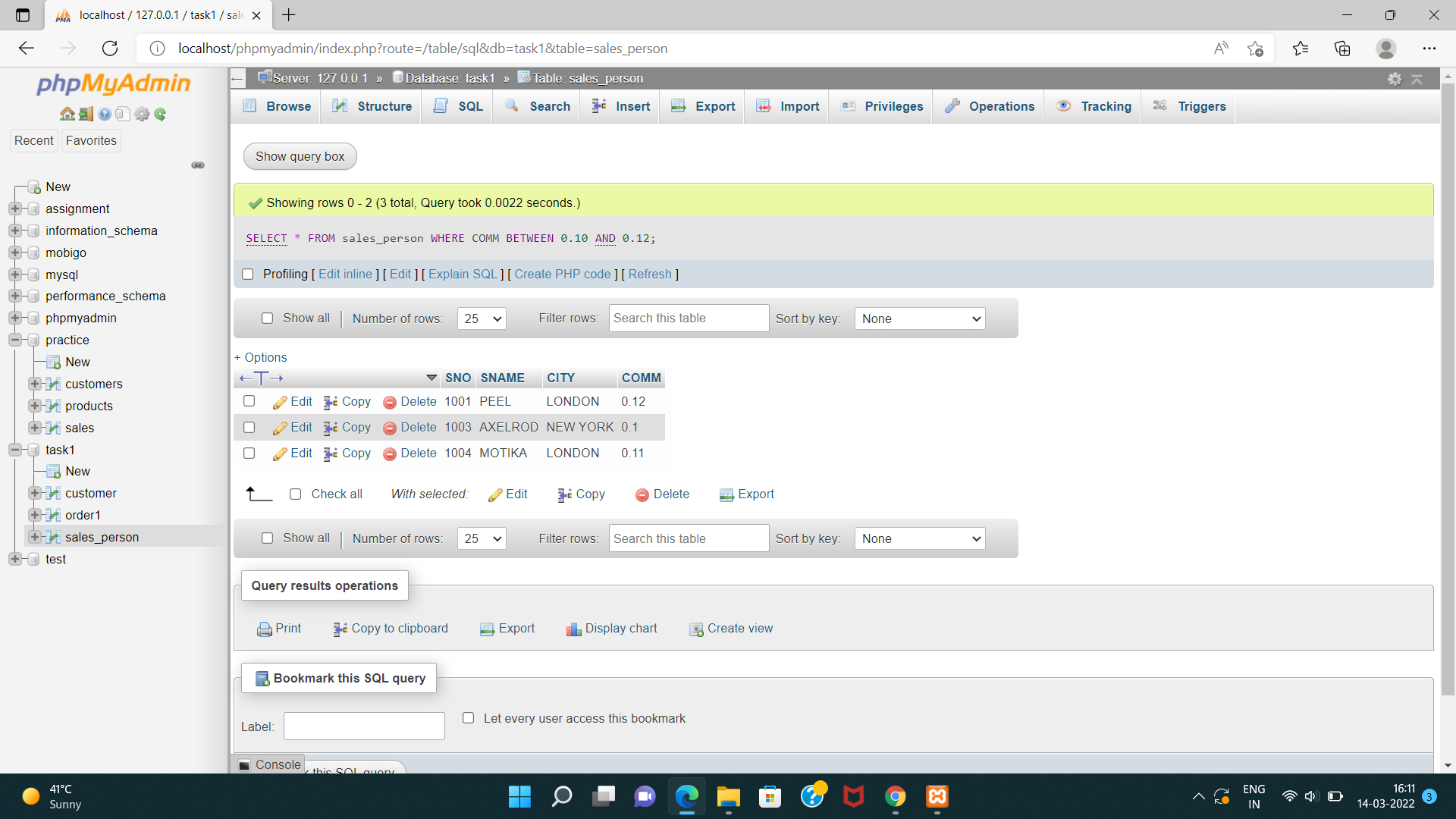
WHERE CITY='BARCELONA' OR City='LONDON';



**B4). 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;



**B5). All customers with NULL values in city column.**

**I1). All orders taken on Oct 3Rd and Oct 4th 1994.**

**I2). All customers serviced by peel or Motika.**

**I3). All customers whose names begin with a letter from A to B .**

**I4). All customers excluding those with rating <= 100 unless they are located in Rome.**

**A1). All orders except those with 0 or NULL value in amt field.**

**A2). Count the number of salespeople currently listing orders in the order table.**

**TASK-3**

**CREATE DATABASE** : PRACTICE

CREATE DATABASE PRACTICE;

**TABLE**:PRODUCTS

CREATE TABLE PRODUCTS(tID int PRIMARY key AUTO\_INCREMENT,ProductName varchar(20),RecommendedPrice int,Categoey varchar(10));

**INSERT DATA**

INSERT INTO products(tID,ProductName,RecommendedPrice,Categoey)VALUES("1","DVD","105.00","LivingRoom");

INSERT INTO products(tID,ProductName,RecommendedPrice,Categoey)VALUES("2","Microwave","98.00","Kitchen");

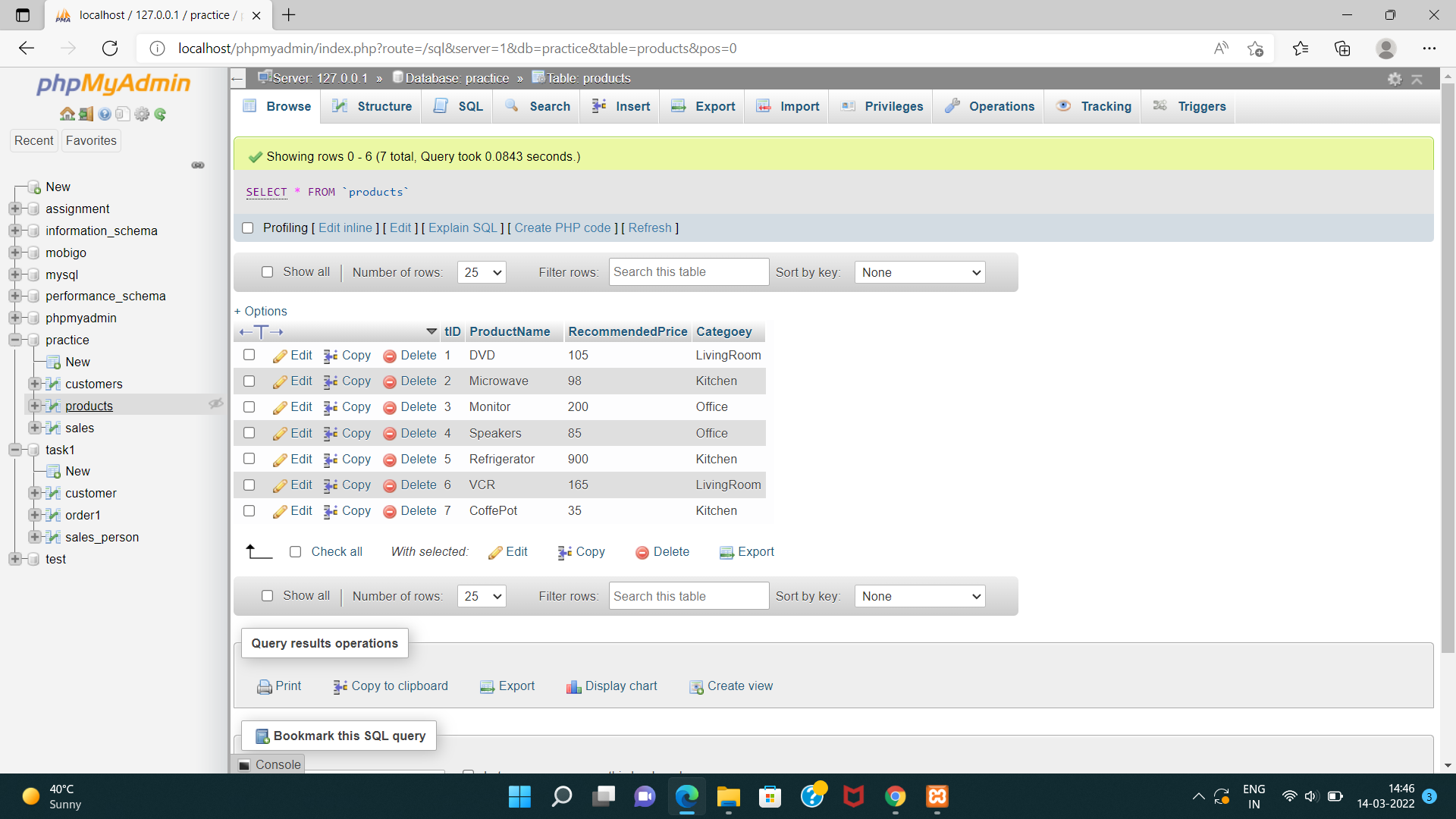
INSERT INTO products(tID,ProductName,RecommendedPrice,Categoey)VALUES("3","Monitor","200.00","Office");

INSERT INTO products(tID,ProductName,RecommendedPrice,Categoey)VALUES("4","Speakers","85.00","Office");

INSERT INTO products(tID,ProductName,RecommendedPrice,Categoey)VALUES("5","Refrigerator","900.00","Kitchen");

INSERT INTO products(tID,ProductName,RecommendedPrice,Categoey)VALUES("6","VCR","165.00","LivingRoom");

INSERT INTO products(tID,ProductName,RecommendedPrice,Categoey)VALUES("7","CoffePot","35.00","Kitchen");



**TABLE**: CUSTOMERS

CREATE TABLE CUSTOMERS(CustomerId int PRIMARY KEY AUTO\_INCREMENT,FirstName varchar(50),LastName varchar(50),City varchar(50));

**INSERT DATA**

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Chintan","Patel","Anand","GJ","388001");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Paresh","Prajapti","Nadiad","GJ","387001");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Pragnesh","Patel","Surat","GJ","395008");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","NIlesh","Dharsandia","Mumbai","MH","400002");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Sonal","Patel","Mumbai","MH","400002");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Harshal","Patel","Mogri","GJ","388345");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Prakash","Rathod","Mogri","GJ","388345");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Aarzoo","Dodhiya","Rajkot","GJ","360006");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Heta","Dave","Varanasi","UP","221002");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","NIkita","Dave","Varanasi","UP","221002");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Vaibhav","Dave","Varanasi","UP","221002");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Paresh","Patel","Pune","MH","411001");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Prakash","Patel","Pune","MH","411001");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Sandhya","Patel","Hyedrabad","AP","500031");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Divesh","Patel","Banglore","KA","560002");

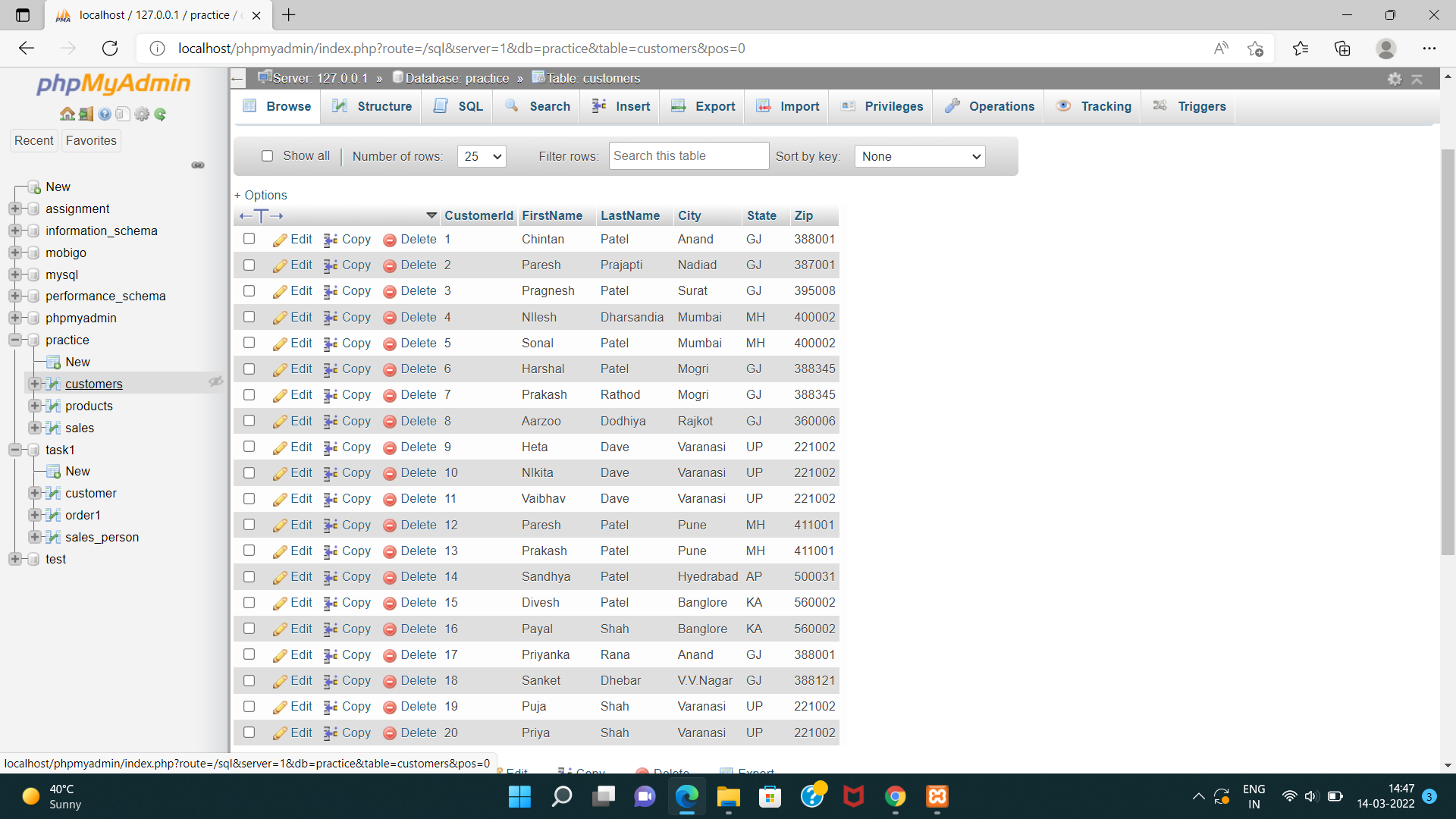
INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Payal","Shah","Banglore","KA","560002");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Priyanka","Rana","Anand","GJ","388001");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Sanket","Dhebar","V.V.Nagar","GJ","388121");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Puja","Shah","Varanasi","UP","221002");

INSERT INTO customers(CustomerId,FirstName,LastName,City,State,Zip)VALUES("","Priya","Shah","Varanasi","UP","221002");



**TABLE**: SALES

CREATE TABLE sales(SalesId int PRIMARY KEY AUTO\_INCREMENT,ProductId int,CustomerId int,SalesPrice float(6),SalesDate datetime,FOREIGN KEY(ProductId)REFERENCES products(tID),FOREIGN KEY(CustomerId)REFERENCES customers(CustomerId));

**INSERT DATA**

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","1","1","130.00","2005-06-14");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","2","2","97.00","2005-06-19");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","3","3","200.00","2005-09-20");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","4","4","80.00","2005-03-22");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","5","5","899.00","2005-01-23");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","6","6","150.00","2005-03-24");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","3","7","209.00","2005-03-10");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","4","8","90.00","2005-08-11");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","6","9","130.00","2005-08-12");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","2","14","85.00","2005-12-13");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","3","15","240.00","2005-05-14");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","1","17","87.00","2005-07-19");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","2","18","99.00","2005-09-20");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","6","19","150.00","2005-07-22");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","5","5","900.00","2005-03-06");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","4","6","86.00","2005-04-07");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","2","7","88.00","2005-11-08");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","3","8","198.00","2005-05-09");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","1","9","150.00","2005-10-10");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","6","14","99.00","2005-05-09");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","6","15","104.00","2005-09-20");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","4","14","90.00","2005-07-22");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","1","1","130.00","2005-03-06");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","2","2","102.00","2005-04-07");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","1","3","114.00","2005-11-08");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","5","4","1000.00","2005-05-09");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","5","5","1100.00","2005-10-10");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","3","6","285.00","2005-06-11");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","2","7","87.00","2005-10-12");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","3","8","300.00","2005-07-31");

INSERT INTO sales(SalesId,ProductId,CustomerId,SalesPrice,SalesDate)VALUES("","3","20","205.00","2005-12-31");

**QUERY**

**B1) Return the FirstName, LastName, ProductName, and SalePrice for all products sold in the month of October2005.**

SELECT FirstName,

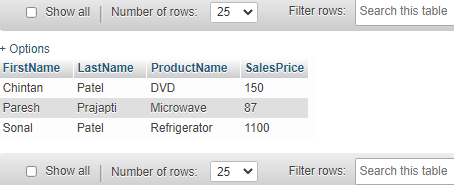
LastName,

ProductName,

SalesPrice

FROM Customers INNER JOIN Products ON Products.tID=Customers.CustomerID INNER JOIN Sales ON tID=ProductID

WHERE SalesDate BETWEEN '2005-10-01' AND '2005-10-31'

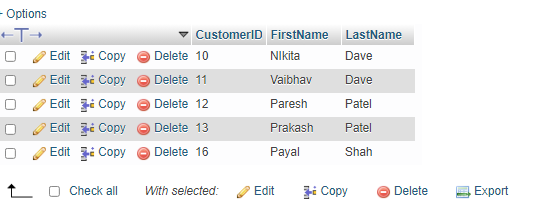


**B2) Return the CustomerID, FirstName, and LastName of those individuals in the Customer table who have made no Sales purchases.**

SELECT CustomerID,FirstName,LastName

FROM Customers

WHERE CustomerID not in (select distinct CustomerID from Sales)



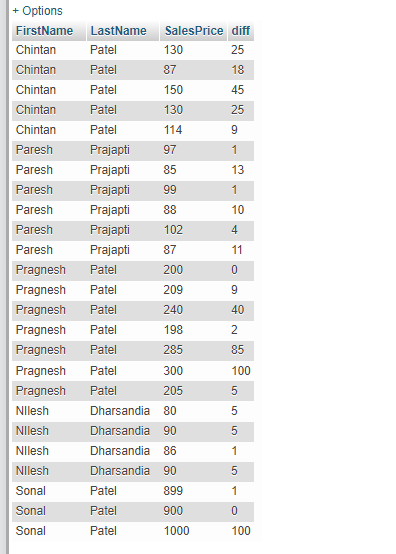
**B3) Return the FirstName, LastName, SalePrice, Recommended SalePrice, and the difference between the SalePrice and Recommended SalePrice for all Sales. The difference must be returned as a positive number.**

SELECT FirstName,LastName,SalesPrice,abs(SalesPrice-RecommendedPrice) as diff

FROM Customers

INNER JOIN Products ON Products.tID= Customers.CustomerID

INNER JOIN Sales ON tID=ProductID

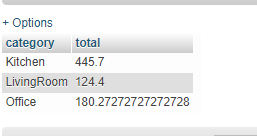


**B4) Return the average SalePrice by Product Category.**

SELECT category,AVG(salesprice) as total

FROM Products inner join Sales on ProductID=tID

GROUP BY category



**I1) Add the following Customer and Sale information to the database.** (using store procedure)

FirstName : Priyanka LastName : Chopra City:Mumbai State:MH Zip:400001

ProductID:3 SalePrice:205 SaleDate:12/31/2005

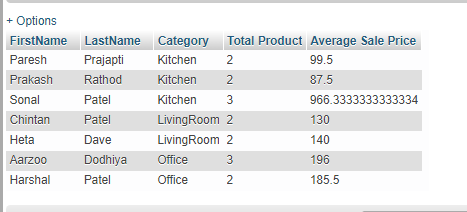
**I2) Return the Product Category and the average Sale Price for those customers who have purchased two or more products**

SELECT distinct FirstName,LastName,Category, COUNT(\*) AS 'Total Product',AVG(salesprice) as 'Average Sale Price'

FROM Products INNER JOIN Sales ON Products.tID=Sales.ProductID INNER JOIN Customers ON Sales.CustomerID=Customers.CustomerID

GROUP BY category,FirstName,LastName

HAVING COUNT(\*)>=2



**A1) Update the Sale Price to the Recommended Sale Price of those Sales occurring between 6/10/2005and6/20/2005.**

UPDATE Sales SET SalesPrice= RecommendedPrice

FROM Sales

INNER JOIN Products ON Sales.ProductID = Products.tID

WHERE SalesDate BETWEEN '2005-06-10' AND '2005-06-20'

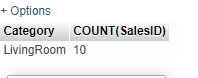
**A2) Number of Sales by Product Category where the average Recommended Price is 10 or more dollars greater than the average Sale Price**

SELECT Category,COUNT(SalesID)

FROM Products INNER JOIN Sales ON tID=ProductID

GROUP BY category

HAVING AVG(RecommendedPrice) >= 10 AND AVG(RecommendedPrice)>AVG(SalesPrice)



**A3) Without using a declared iterative construct, return Sale Date and the running total for all sales, ordered by the Sale Date in Ascending Order**

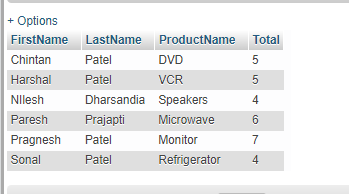
SELECT distinct FirstName,LastName,ProductName, COUNT(\*) as 'Total'

FROM Customers

INNER JOIN Sales ON Customers.CustomerID=Sales.ProductID

INNER JOIN Products ON Products.tID = Customers.CustomerID

GROUP BY FirstName,LastName,ProductName



**TASK-4**

**tblemp(eno,ename,bdate,title,salary, dno), tblproj(pno,pname,budget,dno), tbldept(dno,dname,mgreno), tblworkson(eno,pno,resp,hours).**

**B1) Write an SQL query that returns the project number and name for projects with a budget greater than $100,000.**

SELECT pno, pname

FROM proj

WHERE budget > 100000

**B2) Write an SQL query that returns all works on records where hours worked is less than 10 and the responsibility is “Manager”**

SELECT \*

FROM workson

WHERE hours < 10 AND resp = 'Manager'

**B3) Write an SQL query that returns the employees (number and name only) who have a title of “EEE‟ or “SA‟ and make more than $35,000**.

SELECT eno, ename

FROM emp

WHERE (title = 'EE' OR title = 'SA') AND salary > 35000

**B4) Write an SQL query that returns the employees (name only) in department “D1‟ordered by decreasing salary.**

SELECT ename

FROM emp

WHERE dno = 'D1'

ORDER BY salary DESC

**I1) Write an SQL query that returns the departments (all fields) ordered by ascending department name**.

SELECT \*

FROM dept

ORDER BY dname ASC

**I2) Write an SQL query that returns the employee name, department name, and employee title.**

SELECT ename, dname, title

FROM emp, dept

WHERE emp.dno = dept.dno

**I3) Write SQL query that returns the project name, hours worked, and project number for all works on records where hours > 10**.

SELECT pname, hours, proj.pno

FROM workson, proj

WHERE hours > 10 AND proj.pno = workson.pno

**A1) Write an SQL query that returns the project name, department name, and budget for all projects with a budget < $50,000.**

SELECT pname, dname, budget

FROM proj, dept

WHERE budget < 50000 AND proj.dno = dept.dno

**A2) Write an SQL query that returns the employee numbers and salaries of all employees in the “Consulting” department ordered by descending salary**

SELECT eno, salary

FROM emp, dept

WHERE dname = 'Consulting'

ORDER BY salary DESC

**A3) Write an SQL query that returns the employee name, project name,**

**employee title and hours for all works on records**

SELECT ename, pname, title, hours

FROM emp, proj, workson

WHERE emp.eno = workson.eno and proj.pno = workson.pno