**MODULE: 5 (Database)**

**(5.1)Basics of Database**

1. What do you understand By Database?

->A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS).

2. What is Normalization?

->**Normalization**is the process of minimizing **redundancy** from a relation or set of relations. Redundancy in relation may cause insertion, deletion, and update anomalies. So, it helps to minimize the redundancy in relations.

3. What is Difference between DBMS and RDBMS?

->The main difference between a DBMS and an RDBMS is that a DBMS is a software application used to store, retrieve, and manage data in a database, while an RDBMS is a type of DBMS that stores data in a relational database.

4. What is MF Cod Rule of RDBMS Systems?

-> Edgar F. Codd, a pioneer of the relational model for databases, designed to define what is required from a database management system in order for it to be considered relational, i.e., a relational database management system (RDBMS).

Rule 1:The information rule

->The data stored in a database, may it be a user data or metadata, must be a value of some table cell. Everything in a database must be stored in a table format.

Rule 2:Guaranteed access rule

->Every single data element(value) is guaranteed to be accessible logically with a combination of table name, primary-key and attribute-name.

Rule 3:Systematic treatement of Null values

->The NULL values in a database must be given a systematic and uniform treatment. This is very important rule became a NULL can be interpreted as one the following-data is missing . Data is not known, or data is not applicable.

Rule 4:Active online catalog

->The structure description of the database must be stored in an online catalog, known as data dictionary, which can be accessed by authorized users. Users can use the same query language to access the catalog which they use to access the database itself.

Rule 5:Comprehensive data sub-language-rule

->A database can only be accessed using a language having linear syntax that supports data definition, data manipulation, and transaction management operation. This language can be used directly or by means of some application . If the database allows access to data without any help of this language, then it is considered as a violation.

Rue 6:View updating Rule

->All the views of a database, which can theoretically be updated, must also be updatable by the systems.

Rule 7: High-level insert, update and delete rule

-> A database must support high-level insertion ,updation and deletion.This must not be limited to a single row. That is, must also support union,insertion and minus operation to yield sets of data record.

Rule 8:Physical data independence

->The data stored in a database must be independent of the applications that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.

Rule 9:Logical data independence

->The logical data in a database must be independent of its user’s view. Any change in logical data must not affect the application using it.

Rule 10:Integrity independence

->A database must be independent of the application that uses it. All its intregrity constraints can be independently modified without the need of any change in the application. This rule makes a database independent of the front-end application and its interface.

Rule 11:Distribution independence

->The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.

Rule 12:Non-subversion rule

->If as system has an interface that provides access to low-level records, then the interface must not be able to subvert the system and bypass security and integrity constraints.

5. What do you understand By Data Redundancy?

->Data redundancy is when multiple copies of the same information are stored in more than one place at a time.

6. What is DDL Interpreter?

->DDL Interpreter interprets the DDL (Data Definition Language) Instructions and stores the record in a data dictionary (in a table containing meta-data).

7. What is DML Compiler in SQL?

->A DML (data manipulation language) refers to a computer programming language that allows you to add (insert), delete (delete), and alter (update) data in a database. A DML is typically a sublanguage of a larger database language like SQL, with the DML containing some of the language's operators.

8. What is SQL Key Constraints writing an Example of SQL Key Constraints?

->In a database table, we can add rules to a column known as **constraints.** These rules control the data that can be stored in a column.

For example, if a column has “NOT NULL” constraint, it means the column cannot store “NULL” values.

9. What is save Point? How to create a save Point write a Query?

->A SAVEPOINT is a point in a transaction in which you can roll the transaction back to a certain point without rolling back the entire transaction.

Syntax for Savepoint command: SAVEPOINT SAVEPOINT\_NAME; This command is used only in the creation of SAVEPOINT among all the transactions.

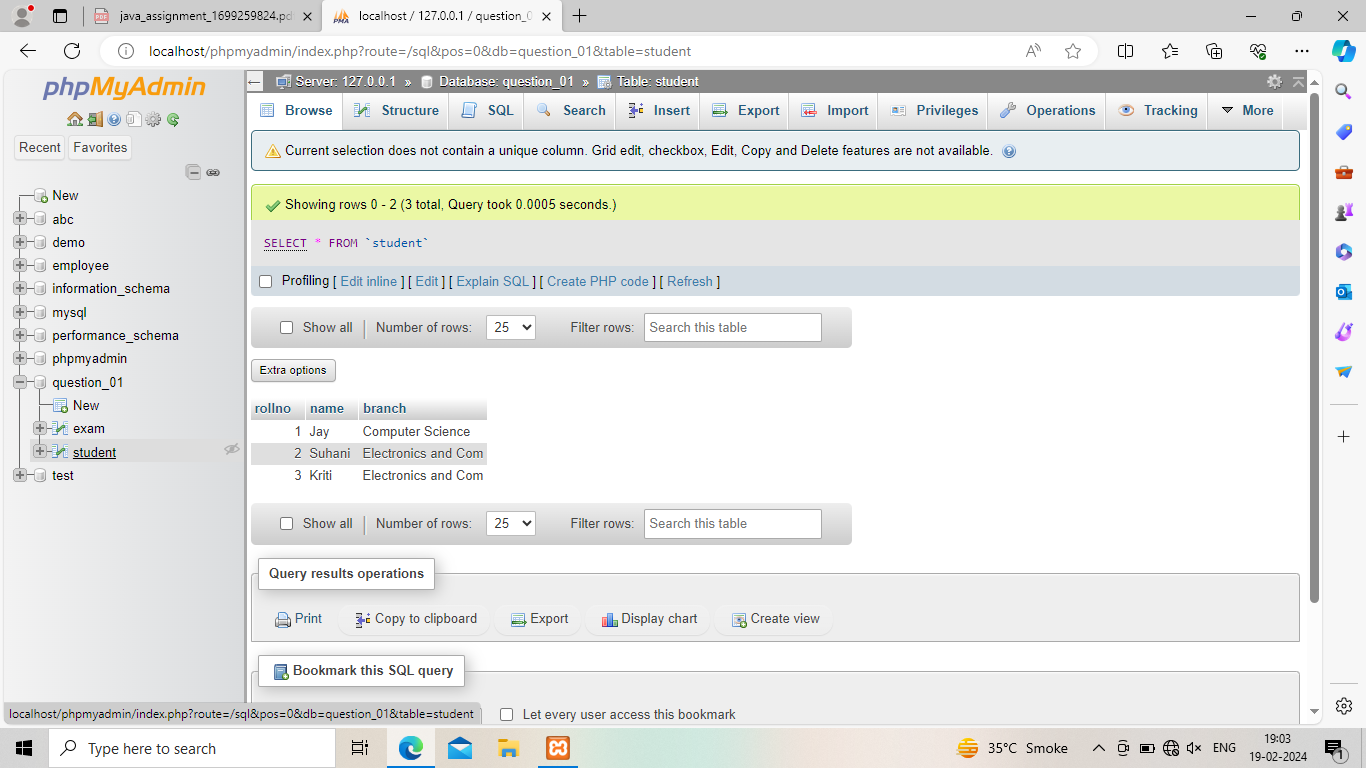
10.What is trigger and how to create a Trigger in SQL?

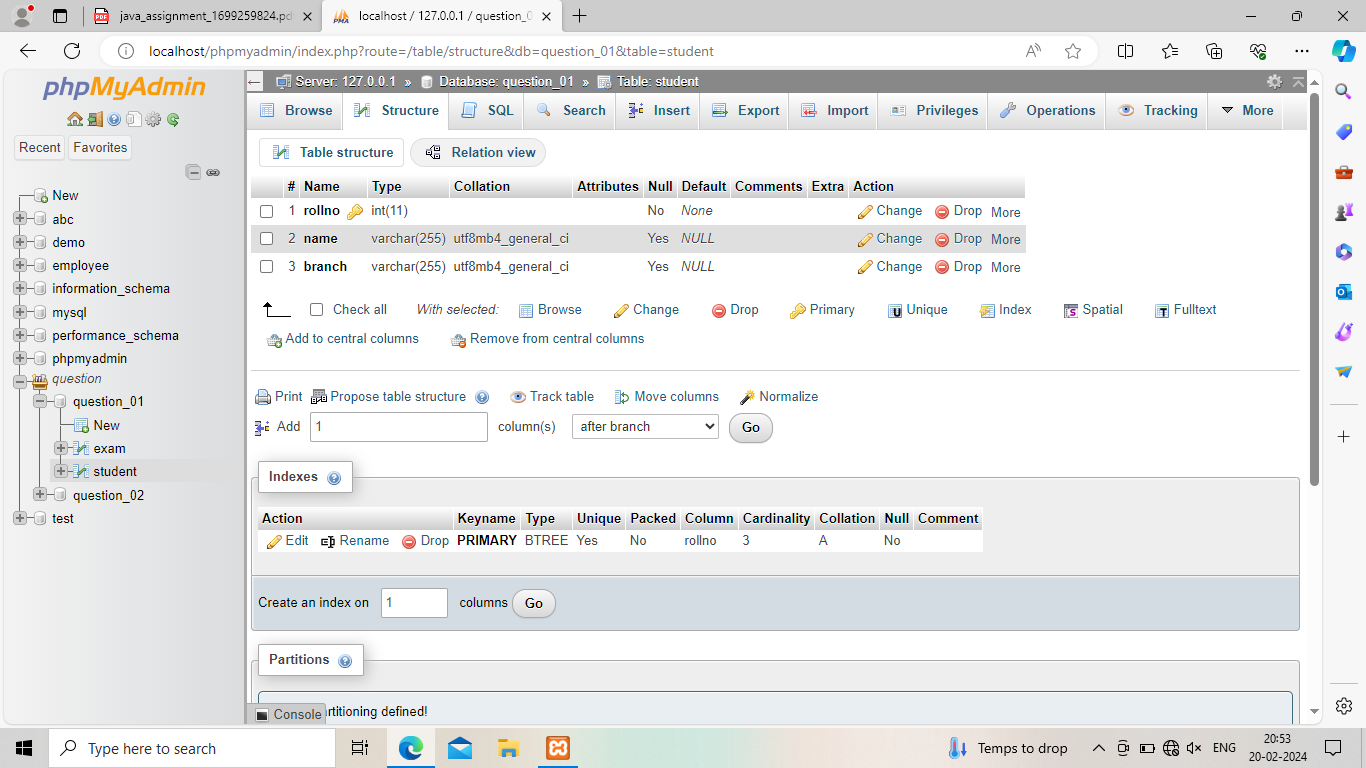
->An SQL trigger allows you to specify SQL actions that should be executed automatically when a specific event occurs in the database. For example, you can use a trigger to automatically update a record in one table whenever a record is inserted into another table.

(5.2) SQL QUERIES

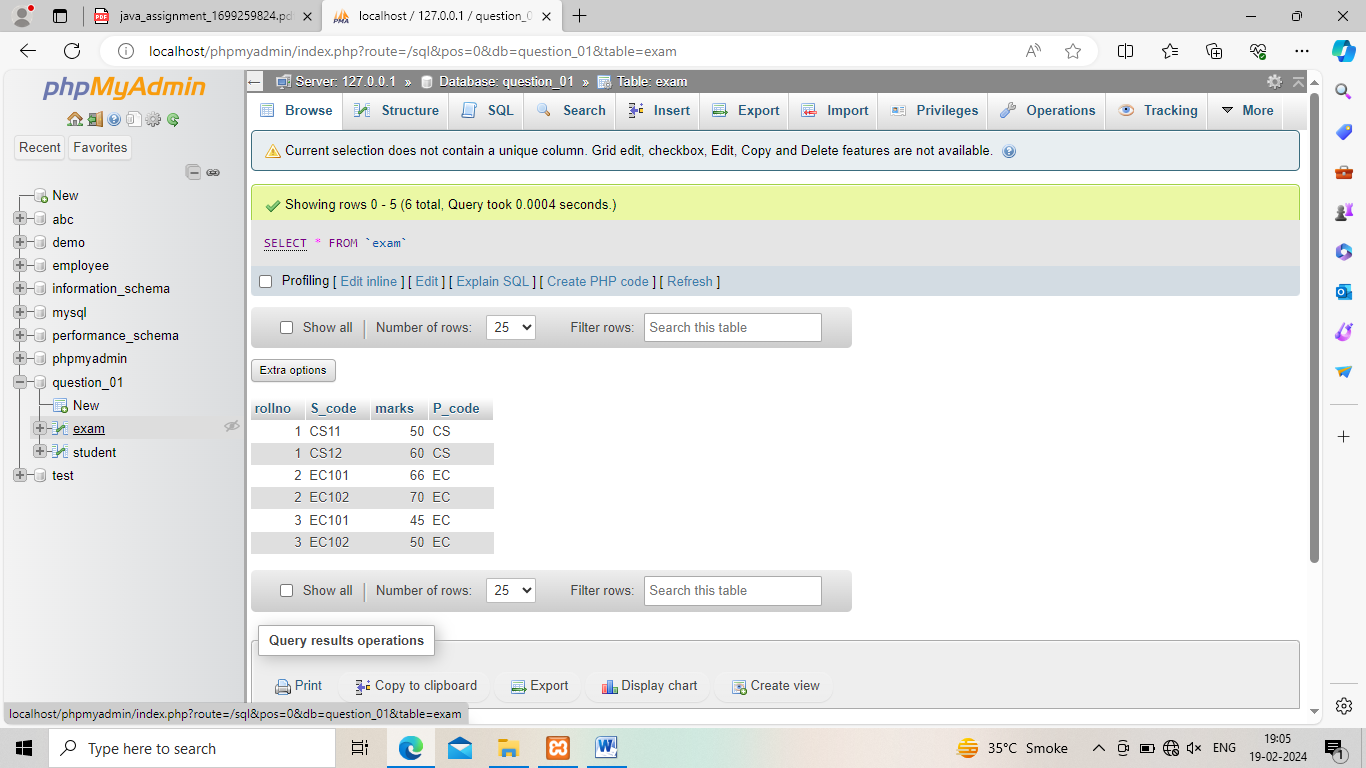
Create Table Name : Student and Exam

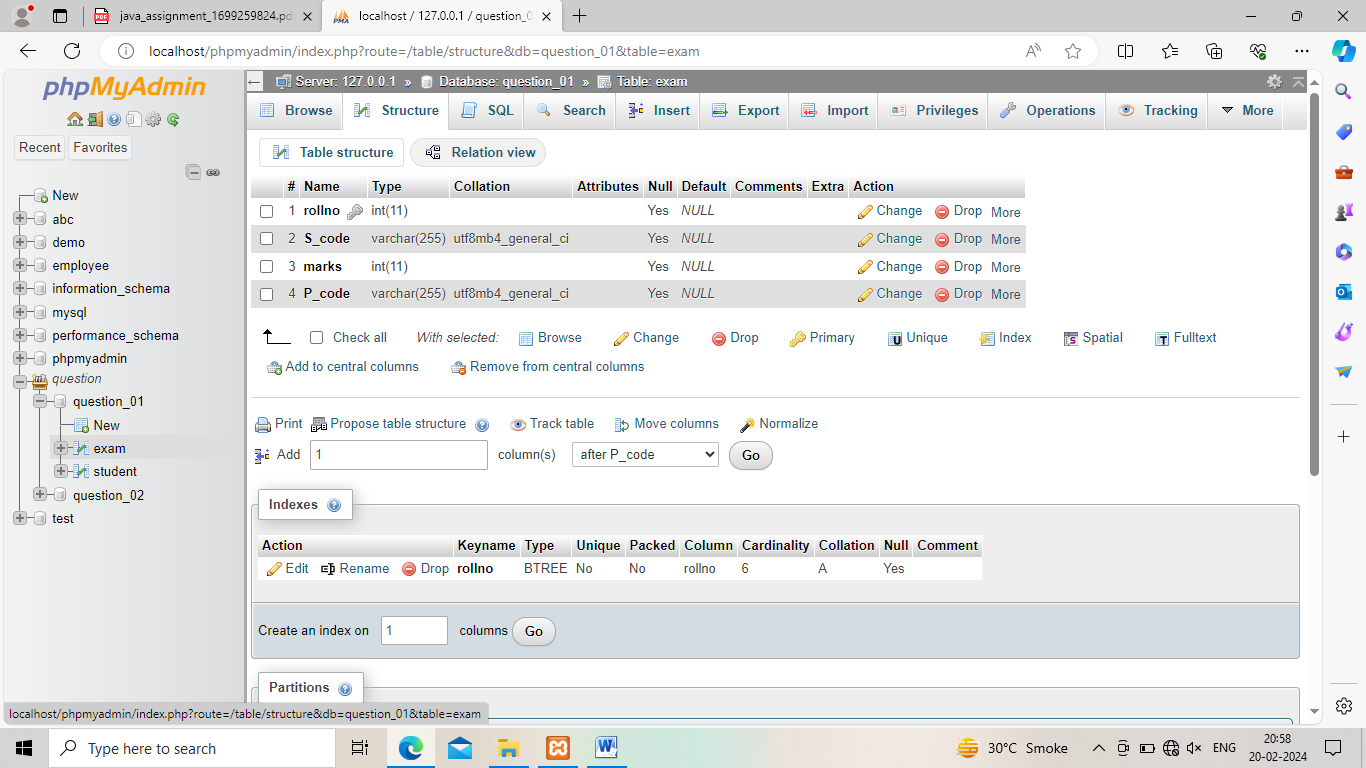
student





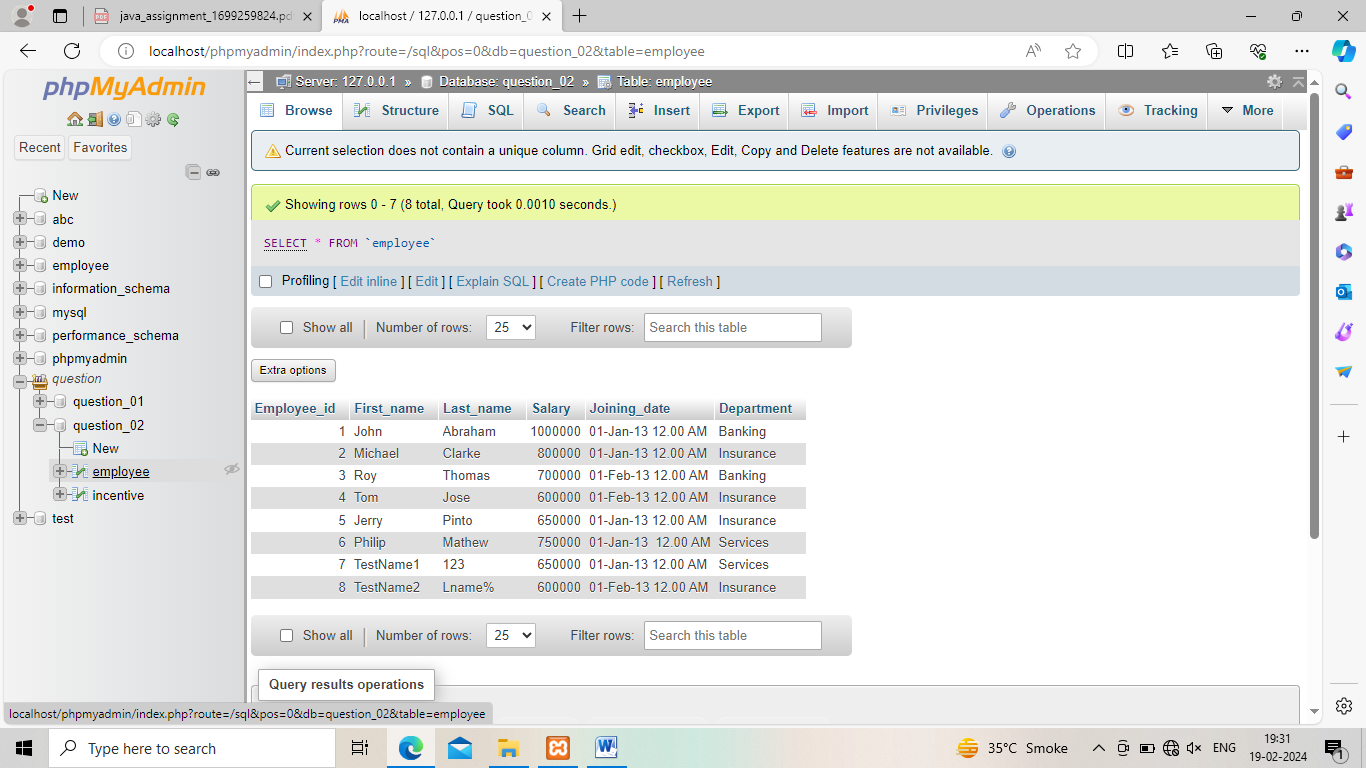
**Exam**



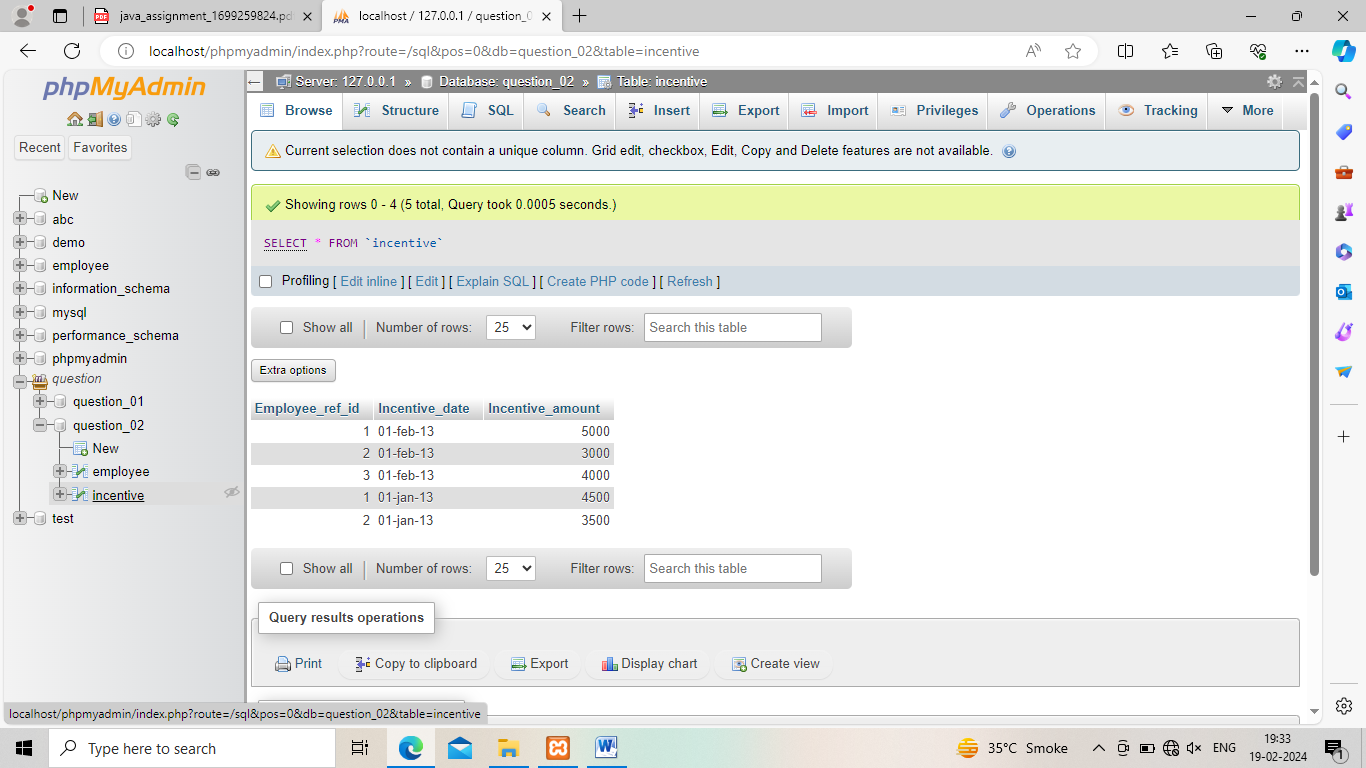


1. Create table given below: Employee and IncentiveTable

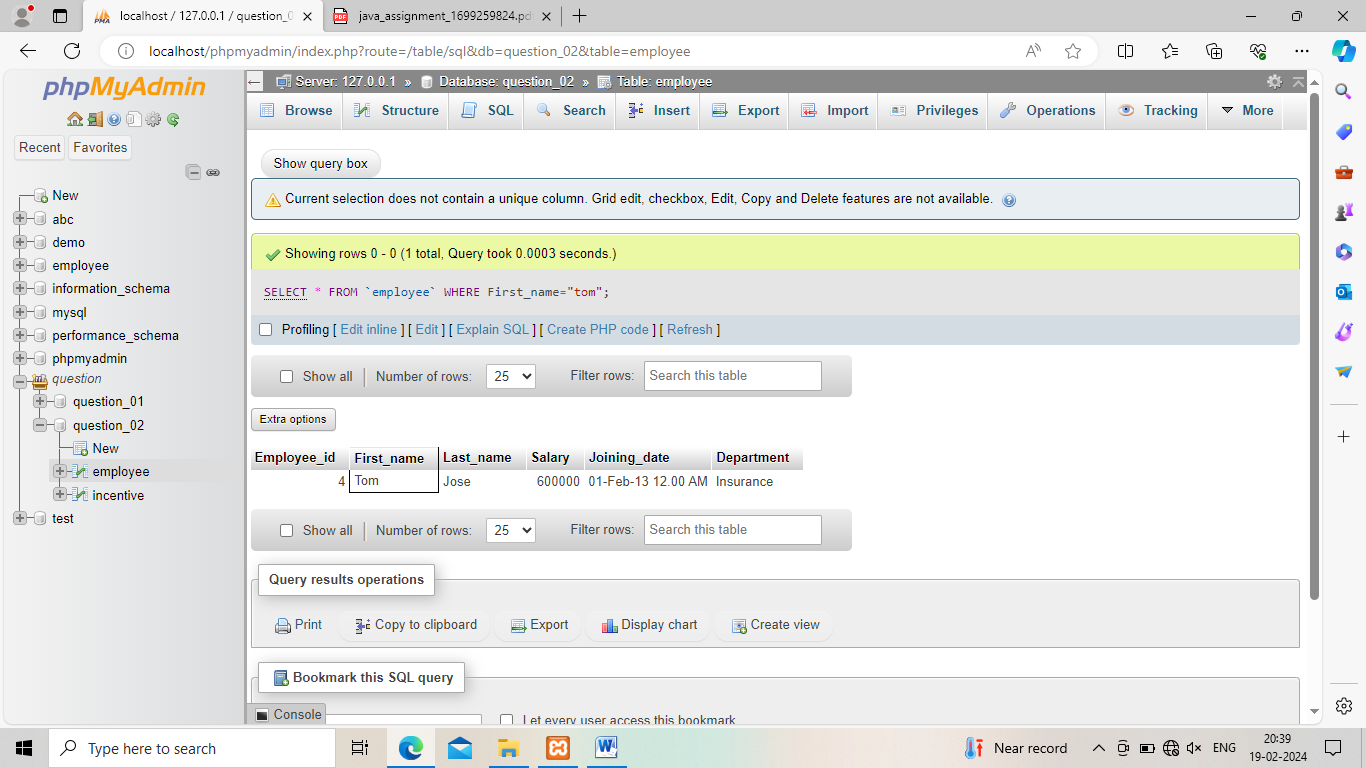
**Employee**



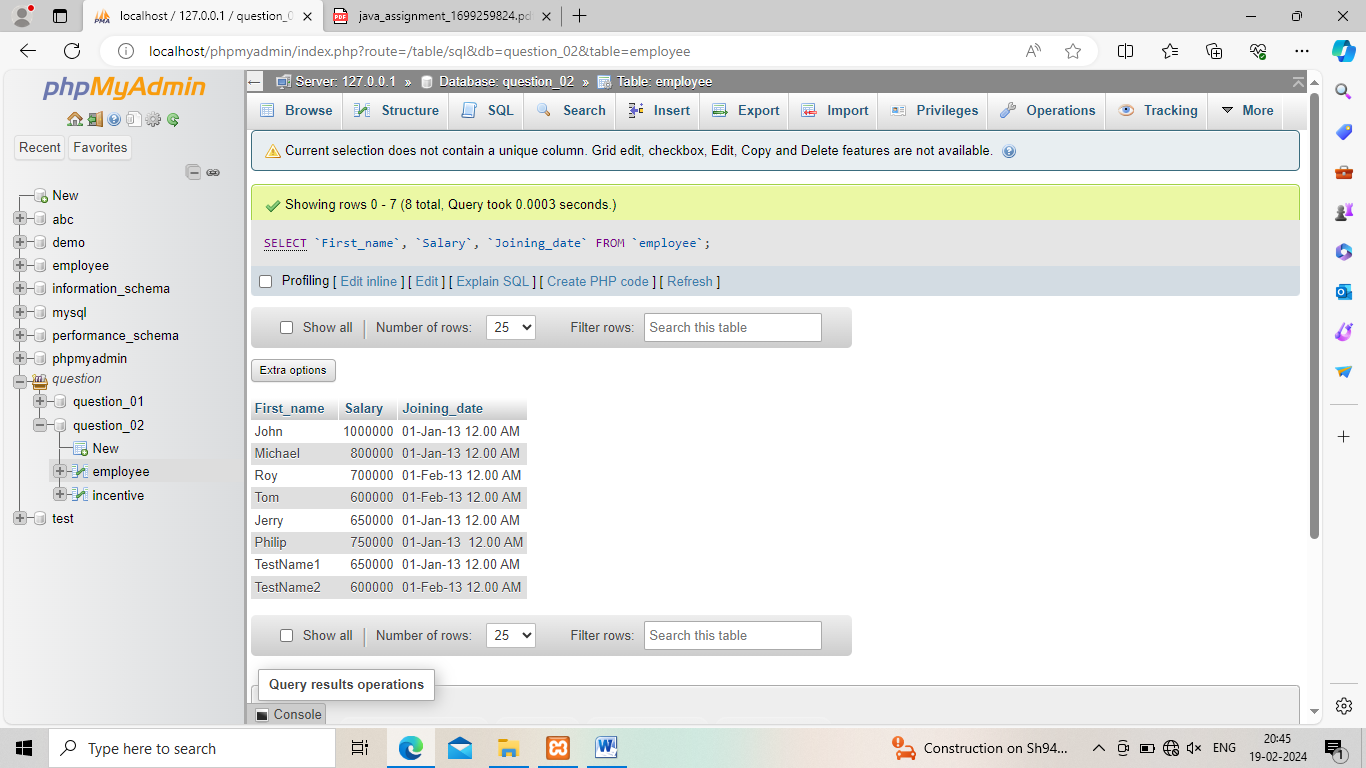
**Incentive table**



3.Get First\_Name from employee table using Tom name “Employee Name”.



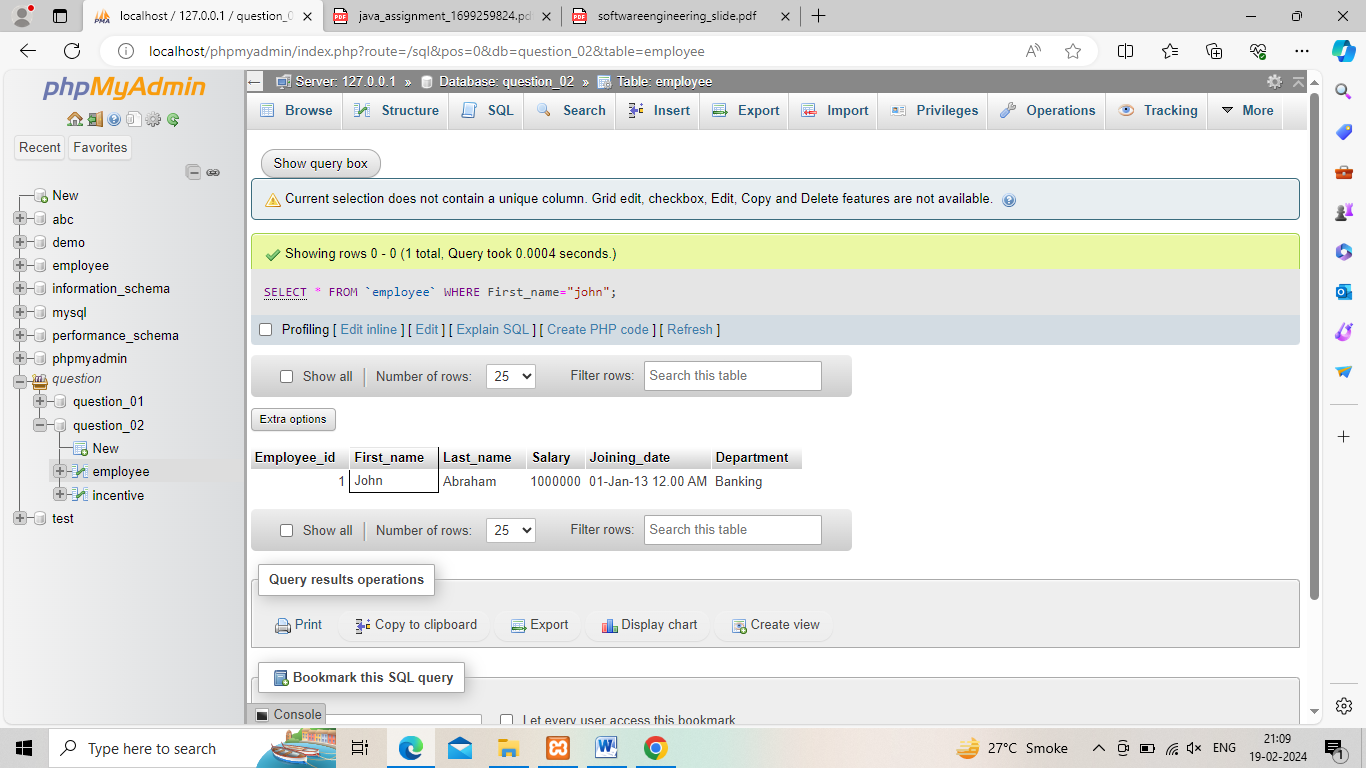
4.Get FIRST\_NAME, Joining Date, and Salary from employee table.



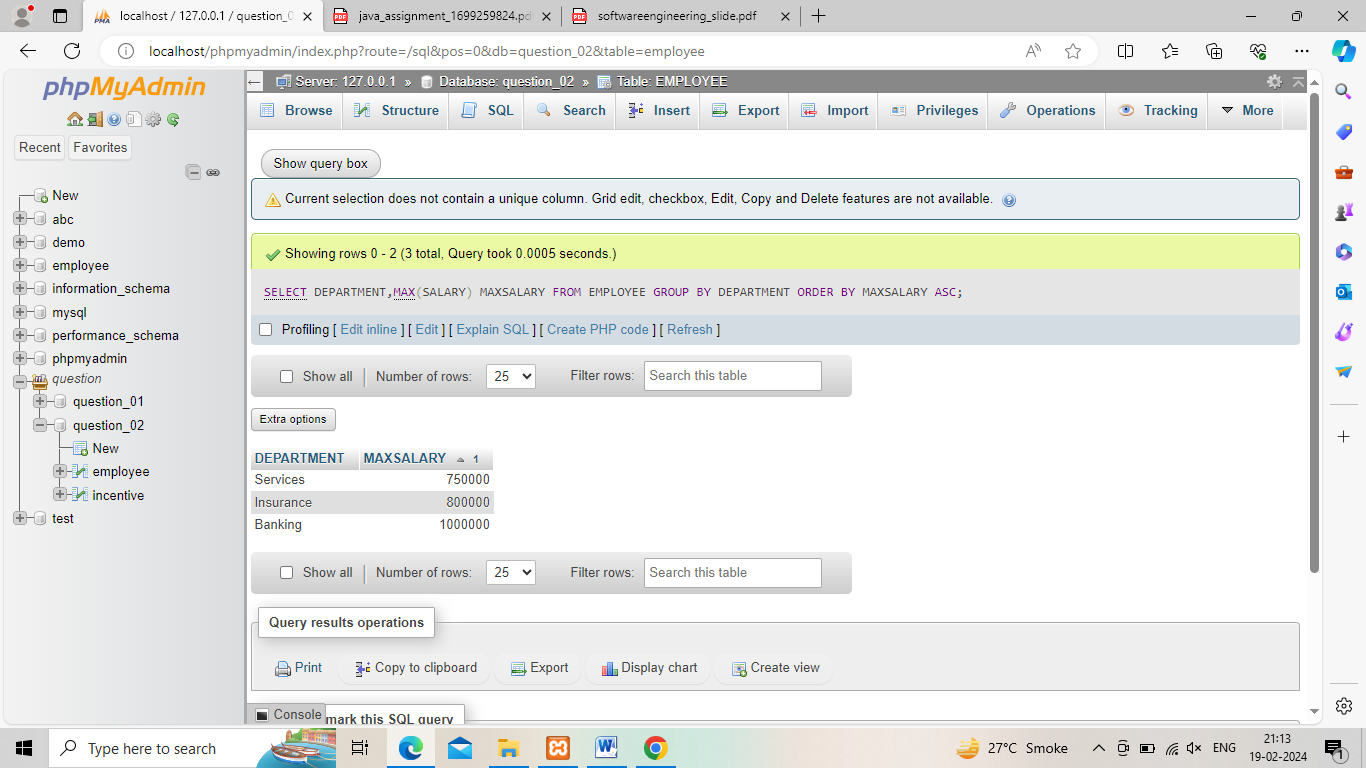
5. Get all employee details from the employee table order by First\_Name Ascending and Salary descending?



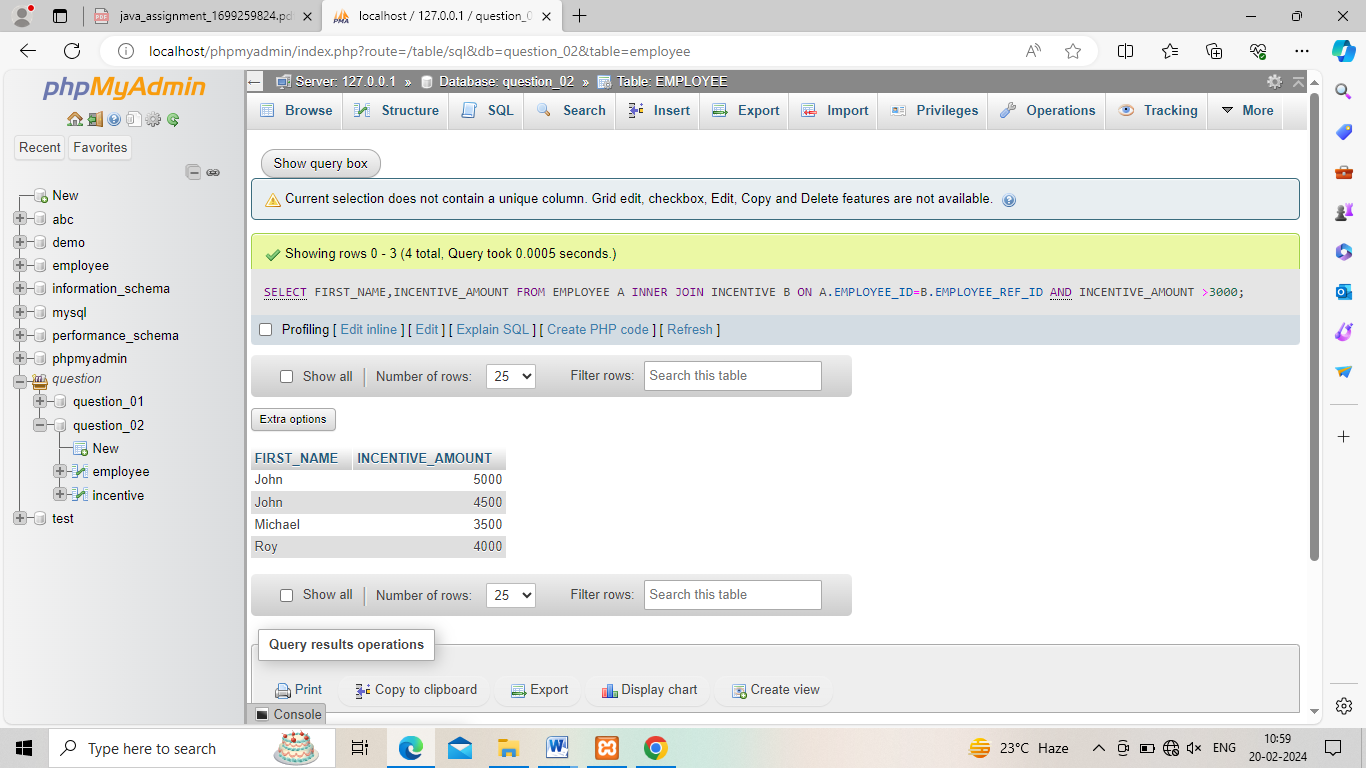
6. Get employee details from employee table whose first name contains ‘J’.



7. Get department wise maximum salary from employee table order by salary ascending?



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



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

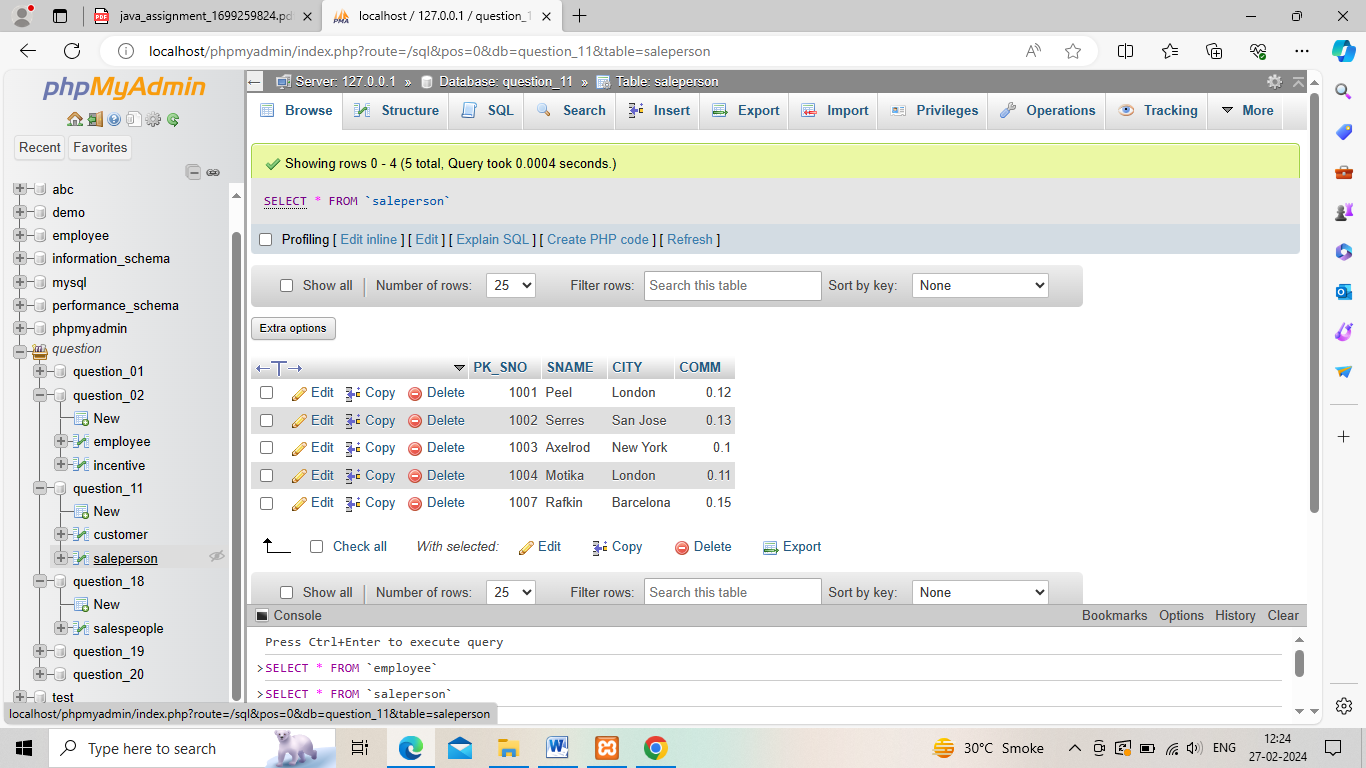
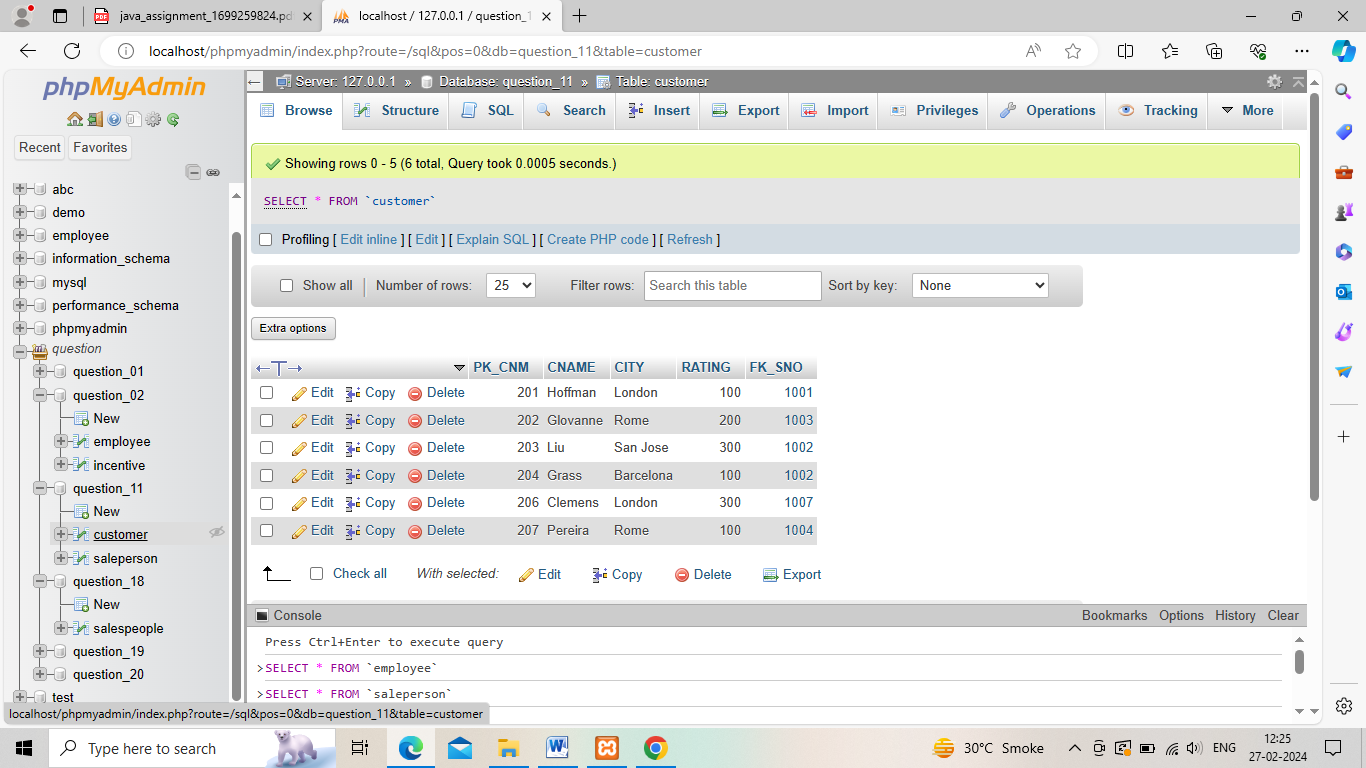
->Query:

CREATE TRIGGER 'after insert' AFTER INSERT ON 'employee' FOR EACH ROW INSERT INTO VIEWS

(id,First\_name,Last\_name,Salary,Joining\_date,Department)

VALUES(new.employee\_id,new.first\_name,new.last\_name, new.salary,new.joining\_date,new.department);

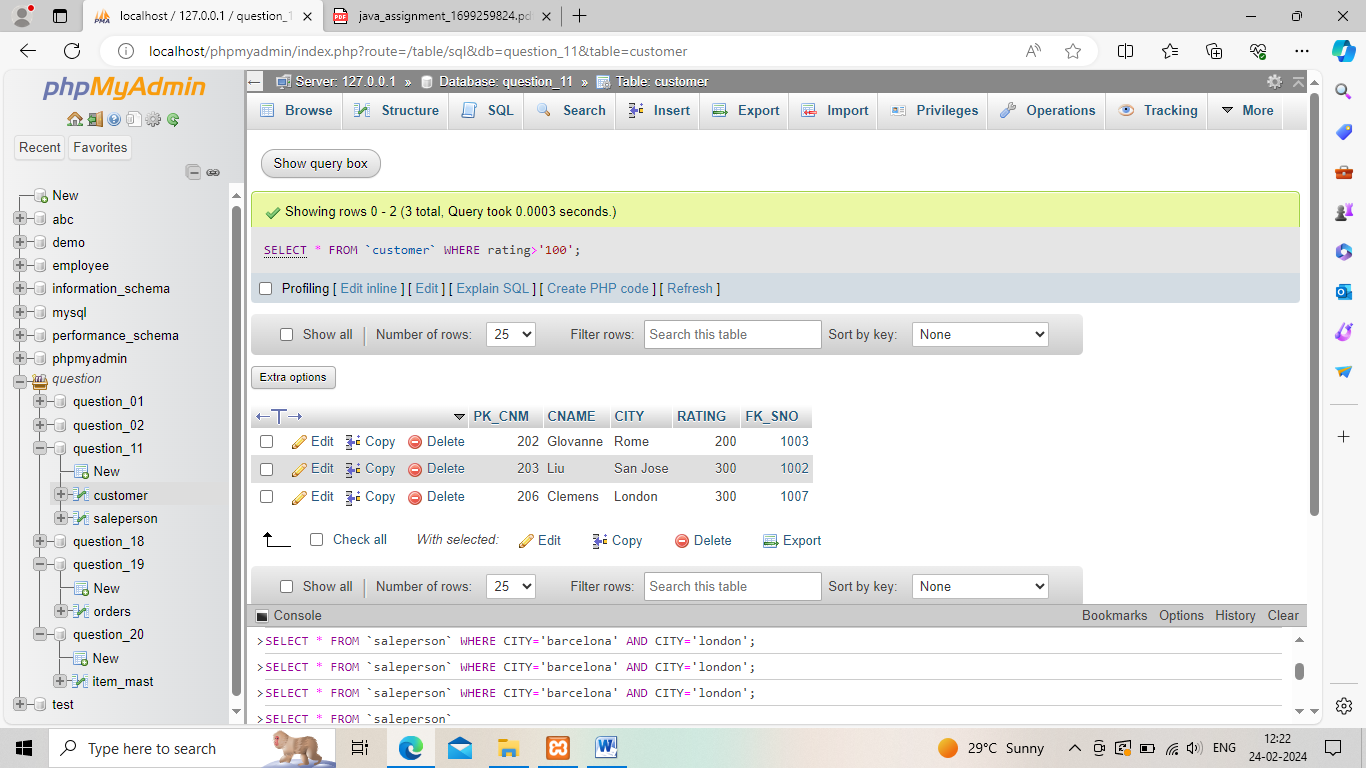
11.Create table given below: Salesperson and Customer

 12.Retrieve the below data from above table

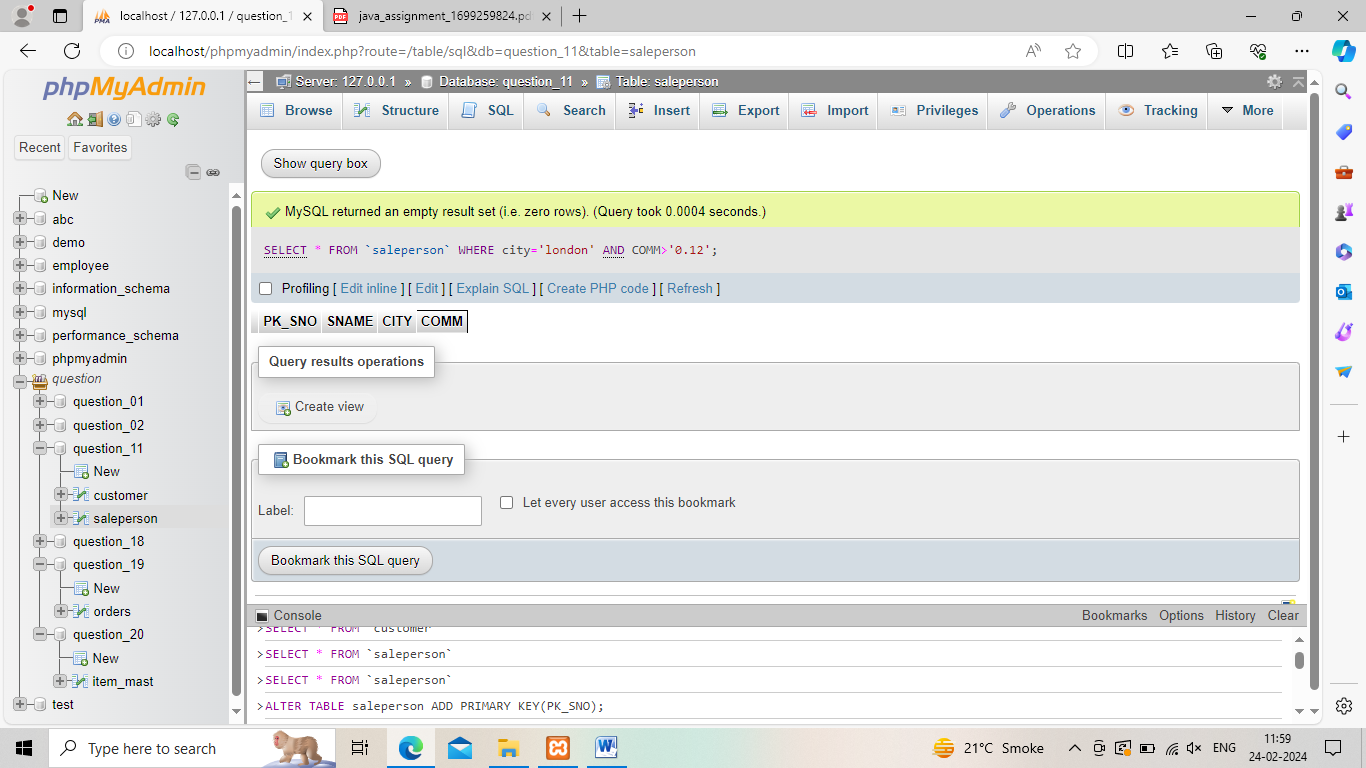
->Use the SELECT statement to retrieve a specific row or retrieve data in a specific way.

Select column1,column2 from table1,table2 WHERE column2=’value’;

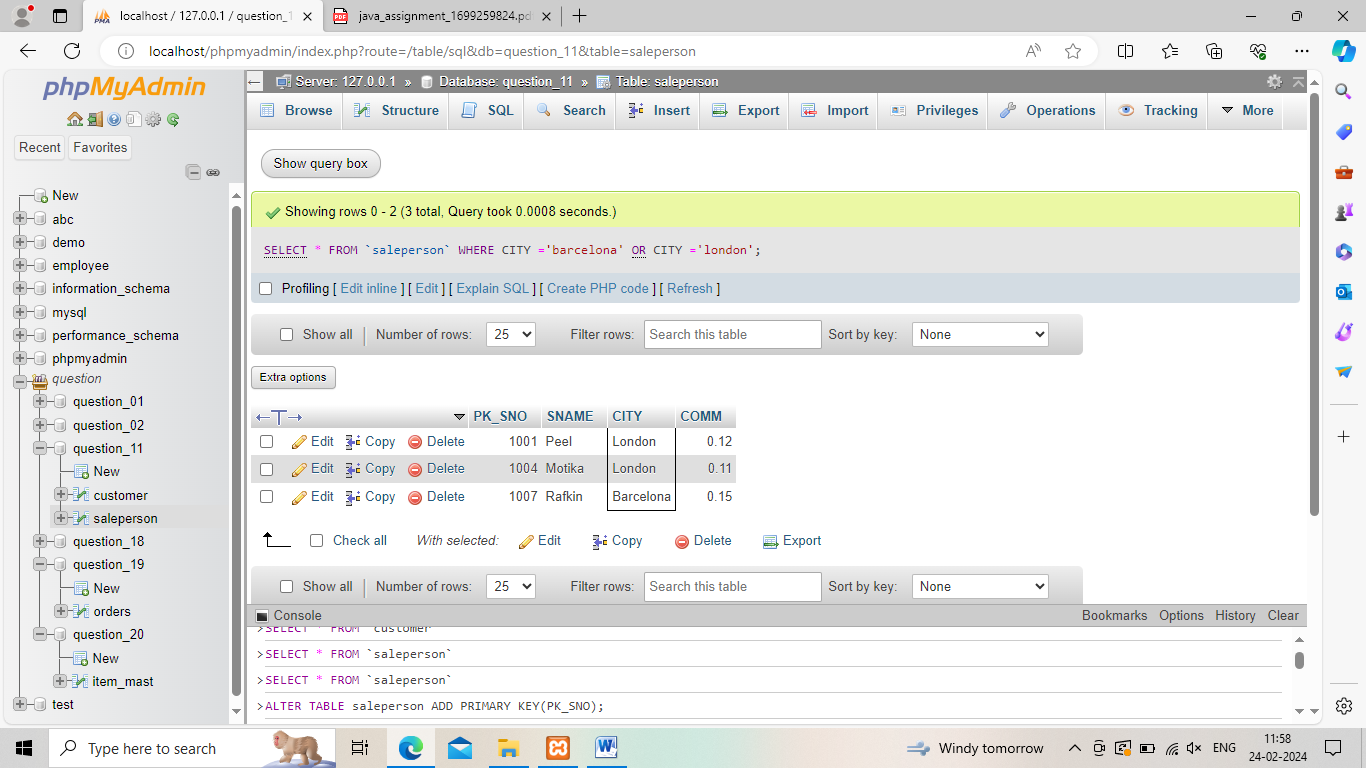
13.All customers name whose ratings is more than 100.



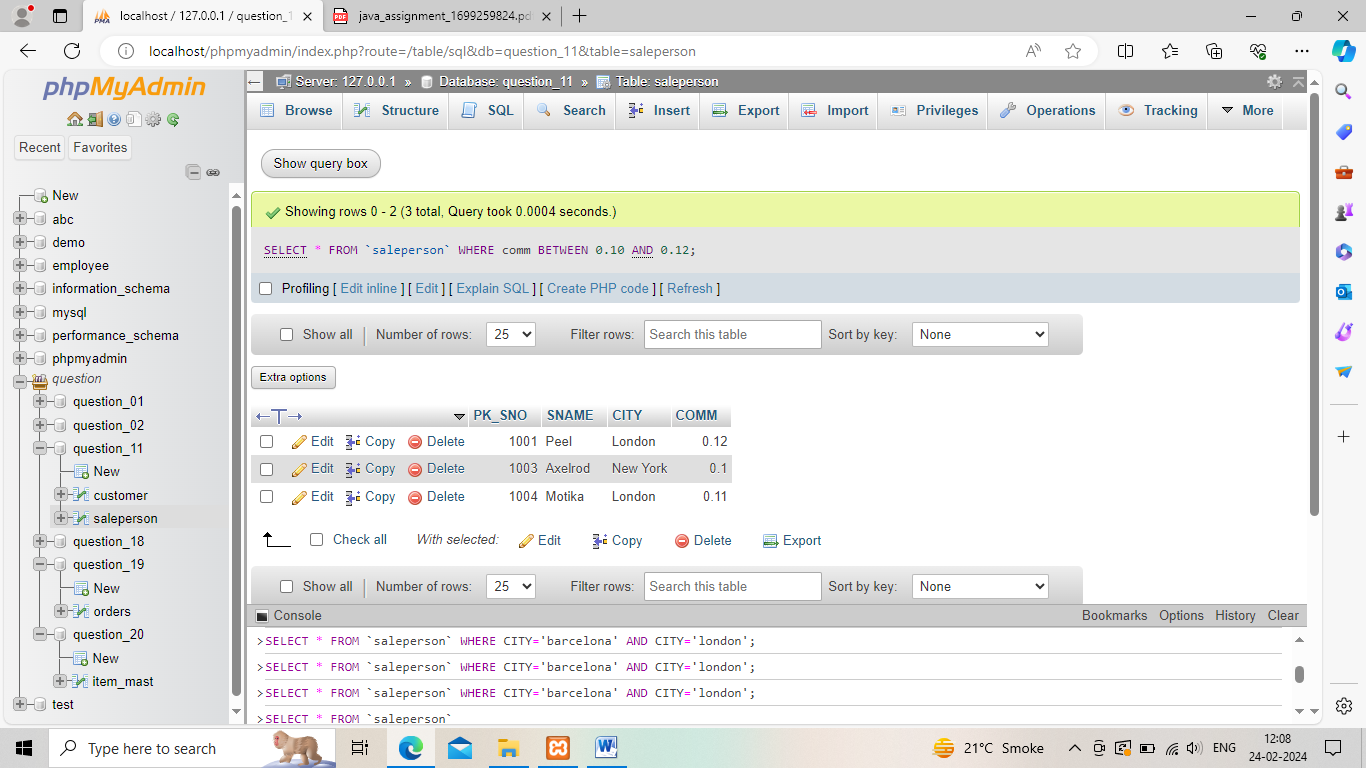
14.Names and cities of all salespeople in London with commission above 0.12



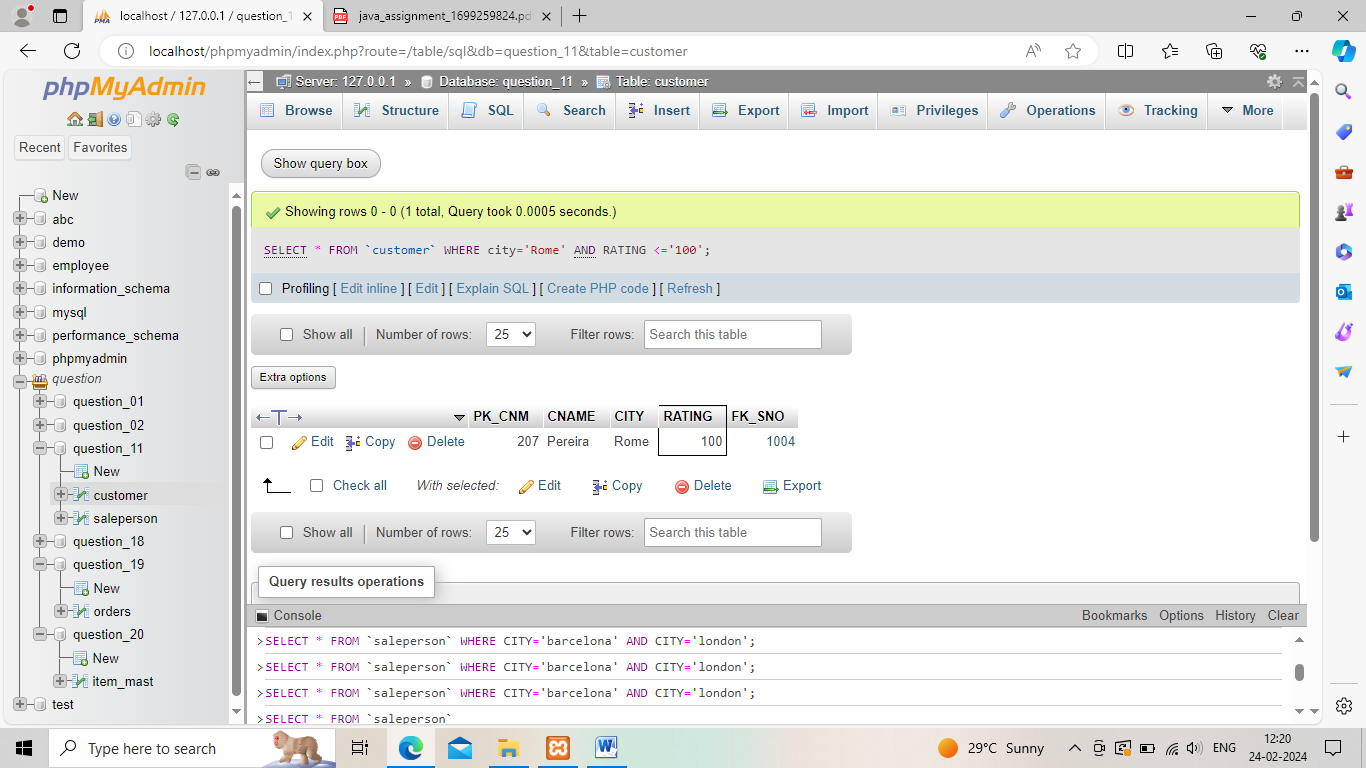
15.All salespeople either in Barcelona or in London



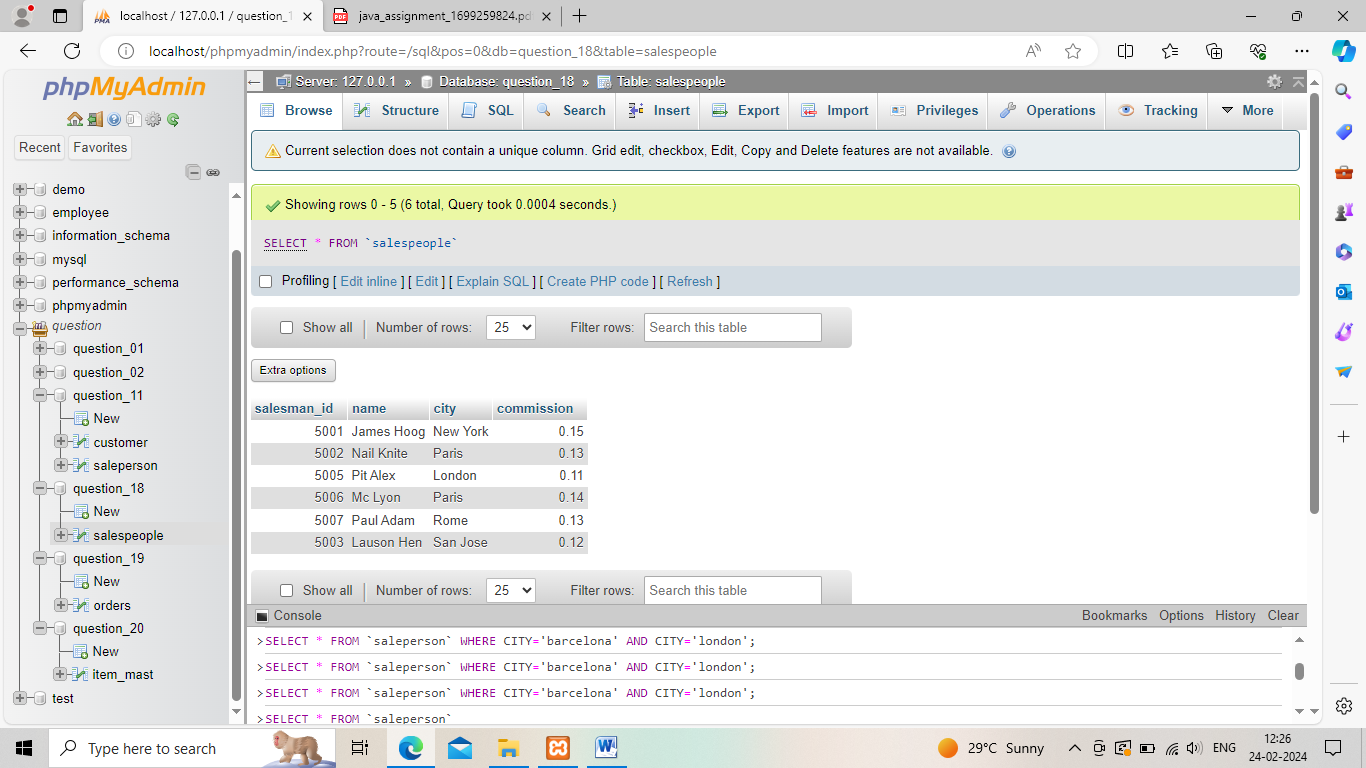
16.All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).



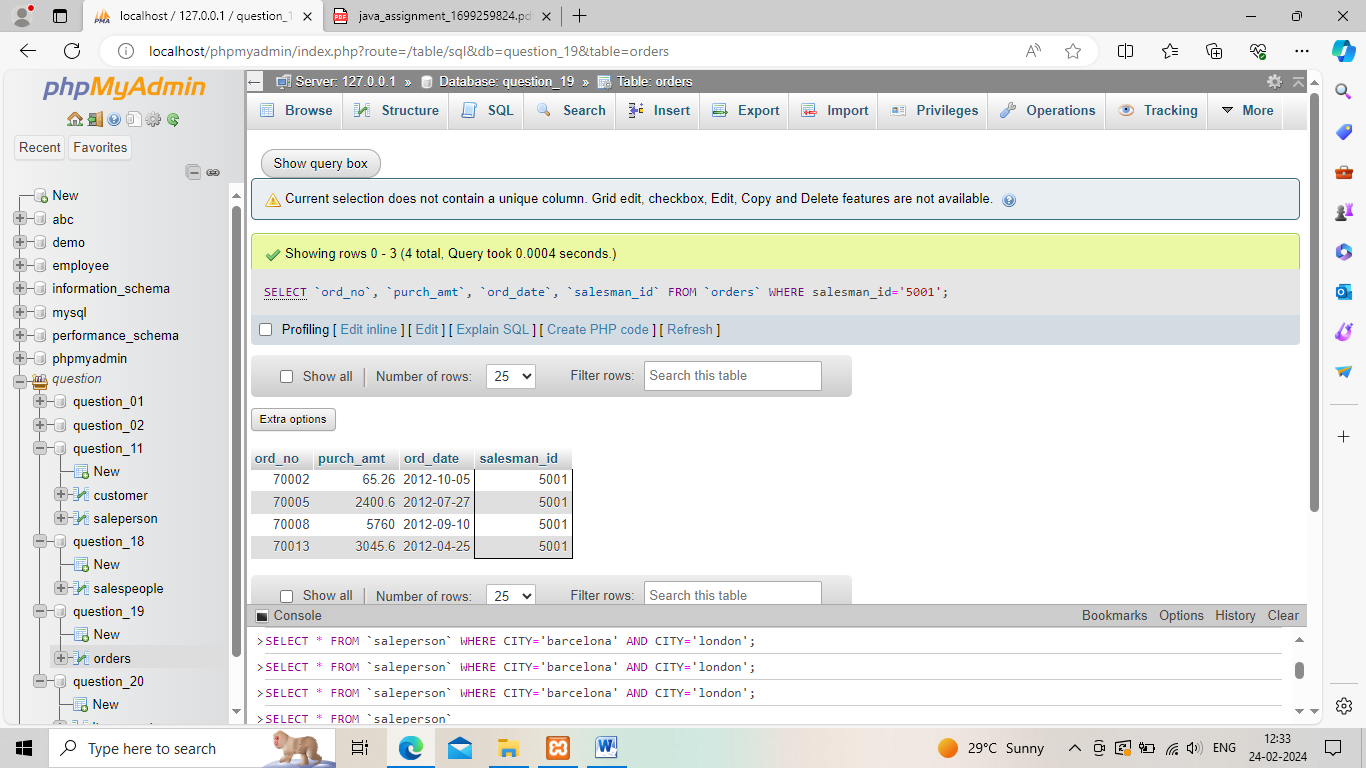
17.All customers excluding those with rating <= 100 unless they are located in Rome.



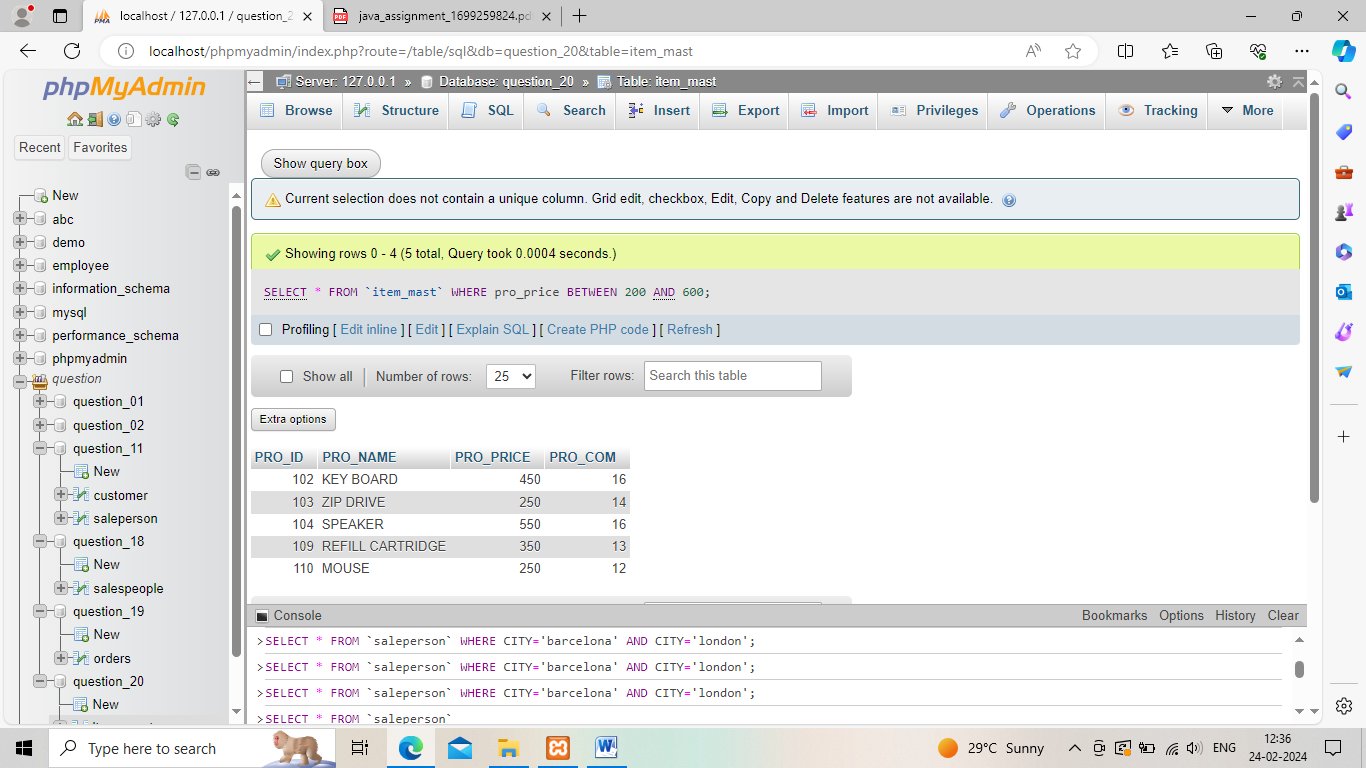
18. Write a SQL statement that displays all the information about all salespeople



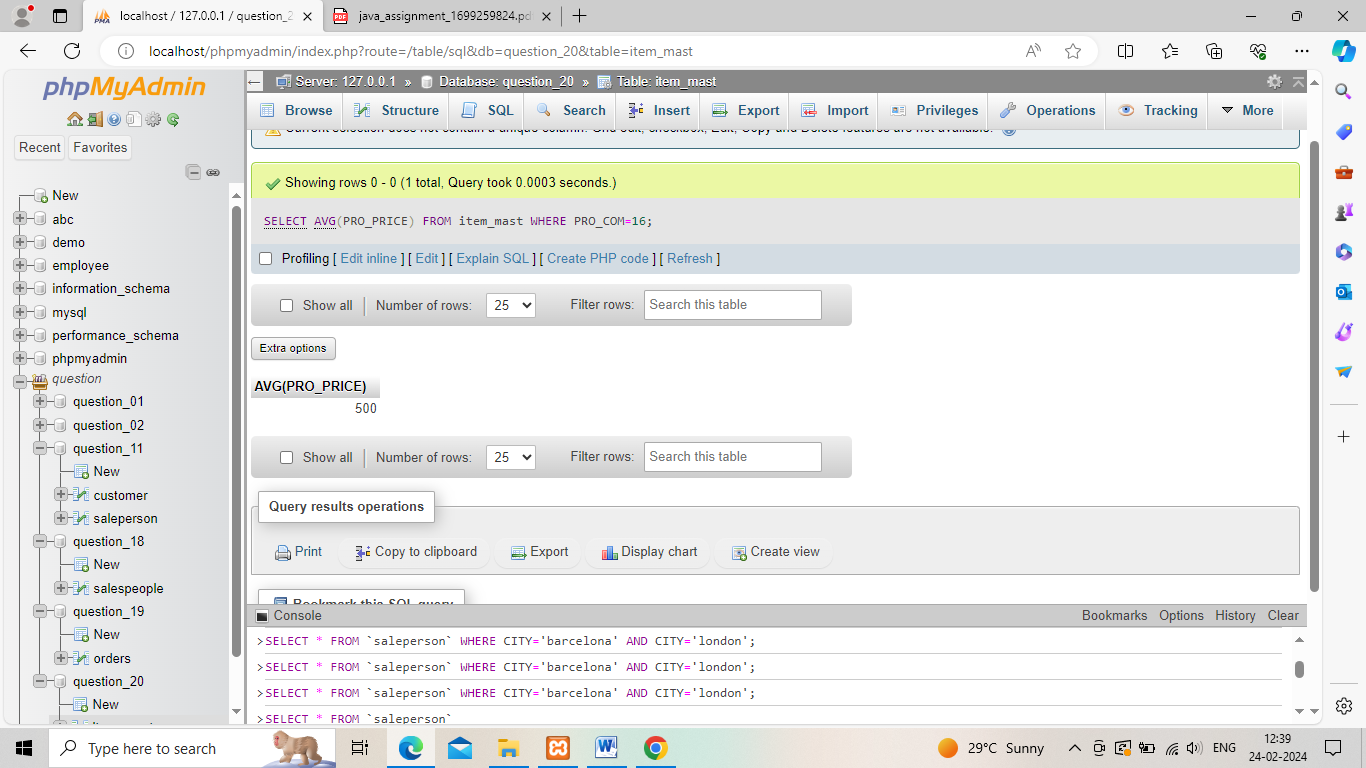
19. From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord\_no, ord\_date, purch\_amt.



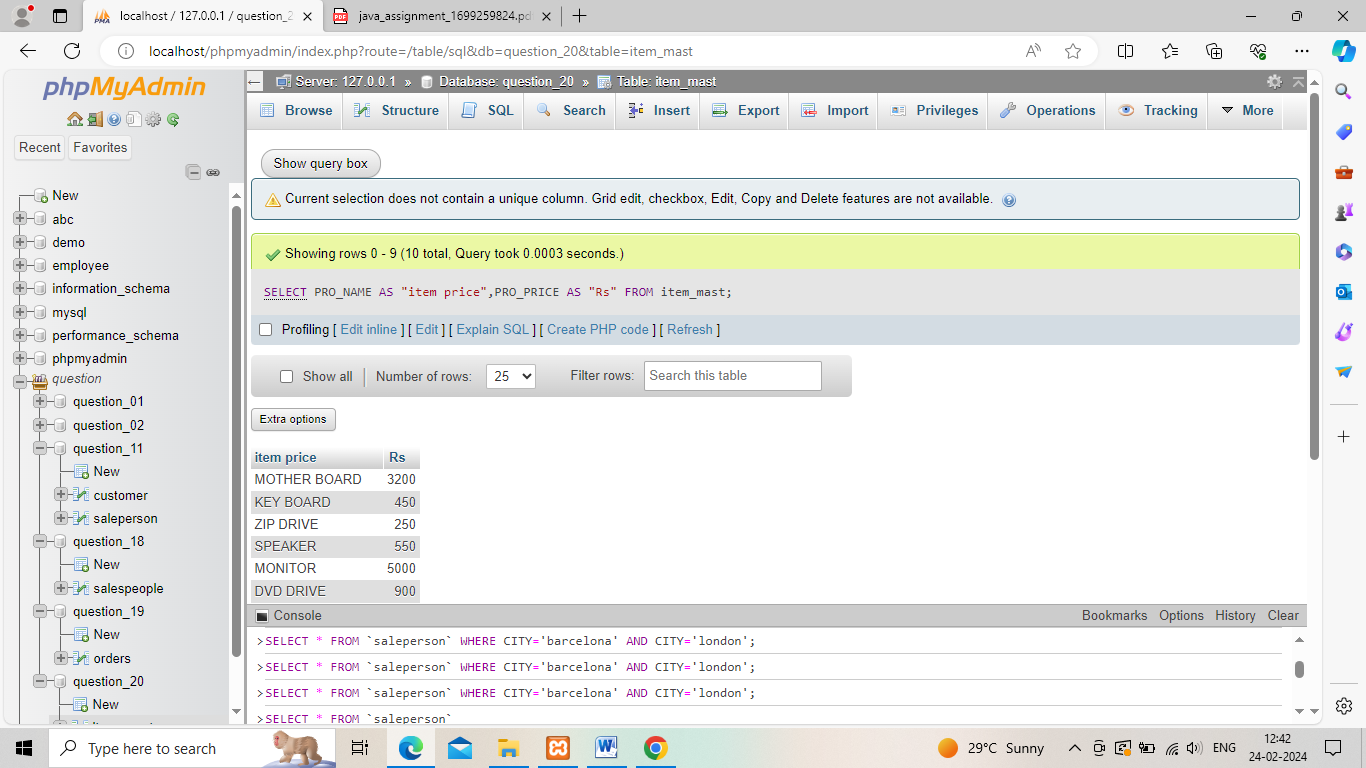
20. From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro\_id, pro\_name, pro\_price, and pro\_com.



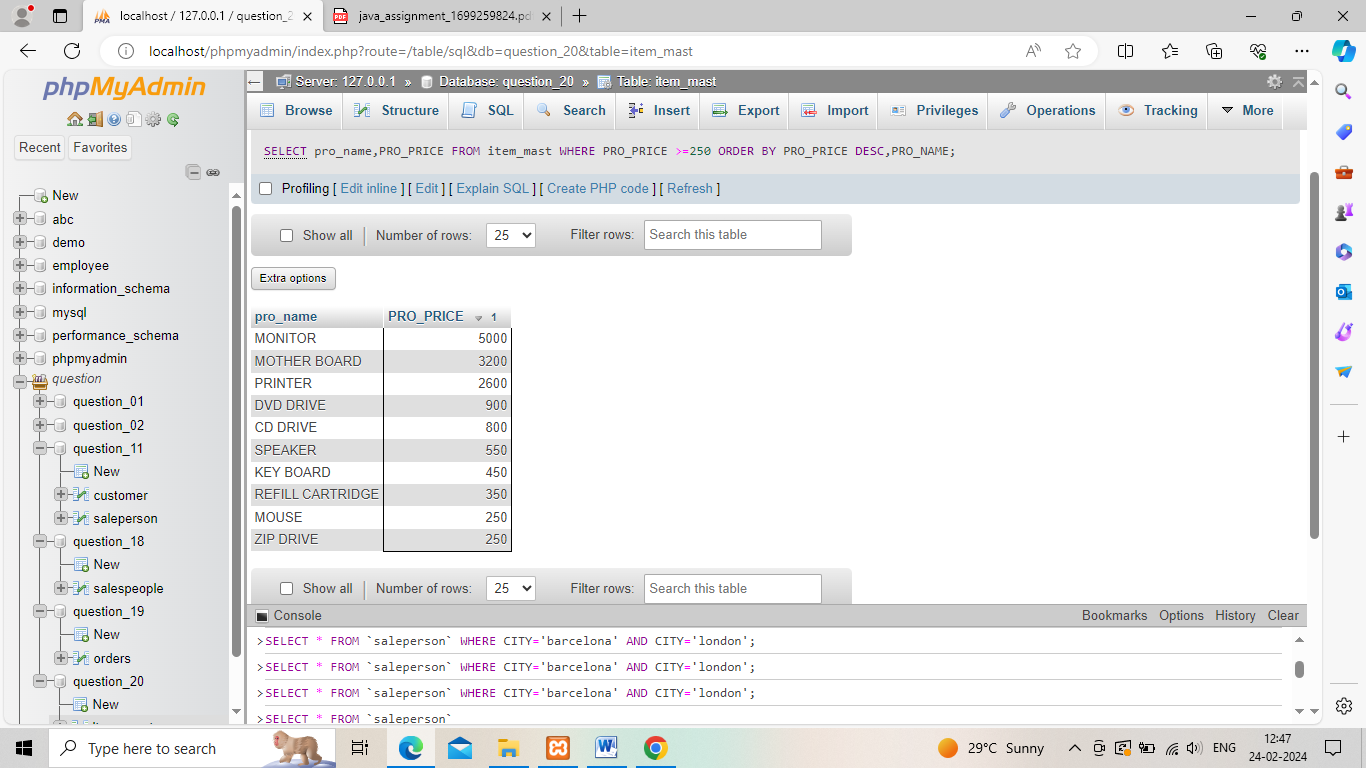
21. From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg.



22. From the following table, write a SQL query to display the pro\_name as 'Item Name' and pro\_priceas 'Price in Rs.'



23. From the following table, write a SQL query to find the items whose prices are higher than or equal to $250. Order the result by product price in descending, then product name in ascending. Return pro\_name and pro\_price.



24. From the following table, write a SQL query to calculate average price of the items for each company. Return average price and company code.

