**Module-4 Database**

**• What is RDBMS**

**Ans**. RDBMS stands for Relational Database Management System. It is a type of database management system that stores data in a structured format using tables, rows, and columns.

**Key Features of RDBMS:**

**1. Relational Model:** RDBMS is based on the relational model, which organizes data into tables with well-defined relationships between them.

**2. Tables**: Data is stored in tables, which are also known as relations. Each table has rows and columns, similar to an Excel spreadsheet.

**3. Primary Key:** Each table has a primary key, which is a unique identifier for each row in the table.

**4. Foreign Key:** Tables can have foreign keys, which are used to establish relationships between tables.

**5. SQL:** RDBMS uses Structured Query Language (SQL) to manage and manipulate data.

**• What is SQL**

**Ans.** SQL (Structured Query Language) is a programming language designed for managing and manipulating data stored in relational database management systems (RDBMS). SQL is used to perform various operations on data, including:

**1. Creating and modifying database structures:** SQL is used to create, modify, and delete database tables, indexes, views, and other objects.

**2. Inserting, updating, and deleting data:** SQL is used to add, modify, and delete data in database tables.

**3. Querying data:** SQL is used to retrieve specific data from database tables, using commands such as SELECT, FROM, WHERE, and JOIN.

**• Write SQL Commands**

**Ans. Here are some common SQL commands:**

**# Data Definition Language (DDL) Commands**

1. **CREATE TABLE:** Creates a new table.
2. **DROP TABLE:** Deletes a table
3. **ALTER TABLE:** Modifies a table.
4. **TRUNCATE TABLE:** Deletes all rows from a table.

**# Data Manipulation Language (DML) Commands**

1. **INSERT INTO:** Inserts new rows into a table.
2. **UPDATE:** Updates existing rows in a table
3. **DELETE:** Deletes rows from a table.

**# Data Query Language (DQL) Commands**

1. **SELECT:** Retrieves data from a table.
2. **SELECT DISTINCT**: Retrieves unique data from a table
3. **SELECT WHERE:** Retrieves data from a table based on conditions.
4. **SELECT ORDER BY**: Retrieves data from a table in a specific order.
5. **SELECT LIMIT:** Retrieves a limited number of rows from a table

**# Data Control Language (DCL) Commands**

1. **GRANT:** Grants privileges to a user.
2. **REVOKE:** Revokes privileges from a user.

**• What is join?**

**Ans.** In SQL, a JOIN is a clause used to combine rows from two or more tables based on a related column. The resulting table contains data from both tables.

**• Write type of joins.**

**Ans. There are several types of JOINs:**

**1. INNER JOIN:** Returns only the rows that have a match in both tables.

**2. LEFT JOIN (or LEFT OUTER JOIN):** Returns all the rows from the left table and the matching rows from the right table. If there is no match, the result will contain NULL values.

**3. RIGHT JOIN (or RIGHT OUTER JOIN):** Similar to LEFT JOIN, but returns all the rows from the right table and the matching rows from the left table.

**4. FULL JOIN (or FULL OUTER JOIN):** Returns all the rows from both tables, with NULL values in the columns where there are no matches.

**5. CROSS JOIN:** Returns the Cartesian product of both tables, with each row of one table combined with each row of the other table.

**• How Many constraint and describes itself.**

**Ans. In SQL, there are 6 main types of constraints:**

**1. Primary Key (PK) Constraint:** A primary key is a column or set of columns that uniquely identifies each row in a table. It ensures that no duplicate values are entered.

**2. Foreign Key (FK) Constraint**: A foreign key is a column or set of columns that references the primary key of another table. It ensures that data consistency is maintained between related tables.

**3. Unique Constraint:** A unique constraint ensures that all values in a column or set of columns are unique. It prevents duplicate values from being entered.

**4. Not Null Constraint:** A not null constraint ensures that a column cannot contain null values. It ensures that data is always present in that column.

**5. Check Constraint**: A check constraint ensures that all values in a column or set of columns satisfy a specific condition. It ensures that data is valid and consistent.

**6. Default Constraint:** A default constraint specifies a default value for a column when no value is specified. It ensures that a column always has a value.

**• Difference between RDBMS vs DBMS**

**Ans**. RDBMS (Relational Database Management System) and DBMS (Database Management System) are both used for managing and storing data, but they differ in their approach and functionality:

**# DBMS (Database Management System)**

1. A DBMS is a software system that allows you to define, create, maintain, and manipulate databases.

2. It provides a way to store and manage data in a structured and controlled manner.

3. DBMS supports a single-user or multi-user environment.

4. It provides data security, integrity, and recovery features.

5. DBMS does not support the concept of relations between tables.

**# RDBMS (Relational Database Management System)**

1. An RDBMS is a type of DBMS that uses the relational model to store and manage data.

2. It organizes data into tables (relations) with well-defined relationships between them.

3. RDBMS supports the concept of relations between tables using keys (primary key and foreign key).

4. It provides a structured query language (SQL) for managing and manipulating data.

5. RDBMS supports multi-user environments and provides features like data security, integrity, and recovery.

**• What is an SQL alias?**

**Ans.** In SQL, an alias is a temporary name given to a table, column, or expression within a query. It is used to make the query more readable and easier to understand.

**There are two types of aliases in SQL:**

**Column Alias:** A column alias is used to give a temporary name to a column in the output of a query.

**Table Alias**: A table alias is used to give a temporary name to a table in a query.

**• Write a query to create the table in Structured Query Language.**

**Ans.**

CREATE TABLE Product (

Product ID INT PRIMARY KEY,

Product Name VARCHAR(255) ,

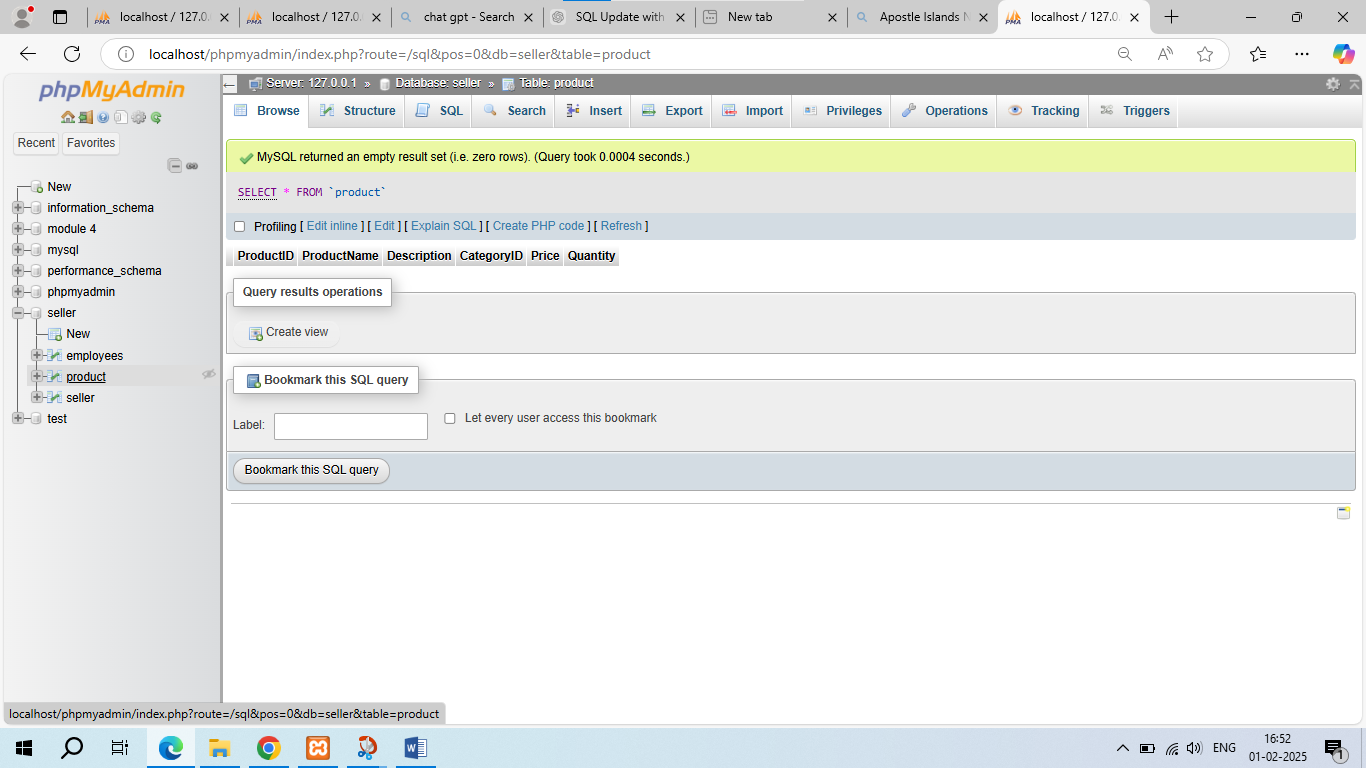
Description VARCHAR(500),

Category ID INT,

Price DECIMAL(10, 2) ,

Quantity INT

);

****

**• Write a query to insert data into table.**

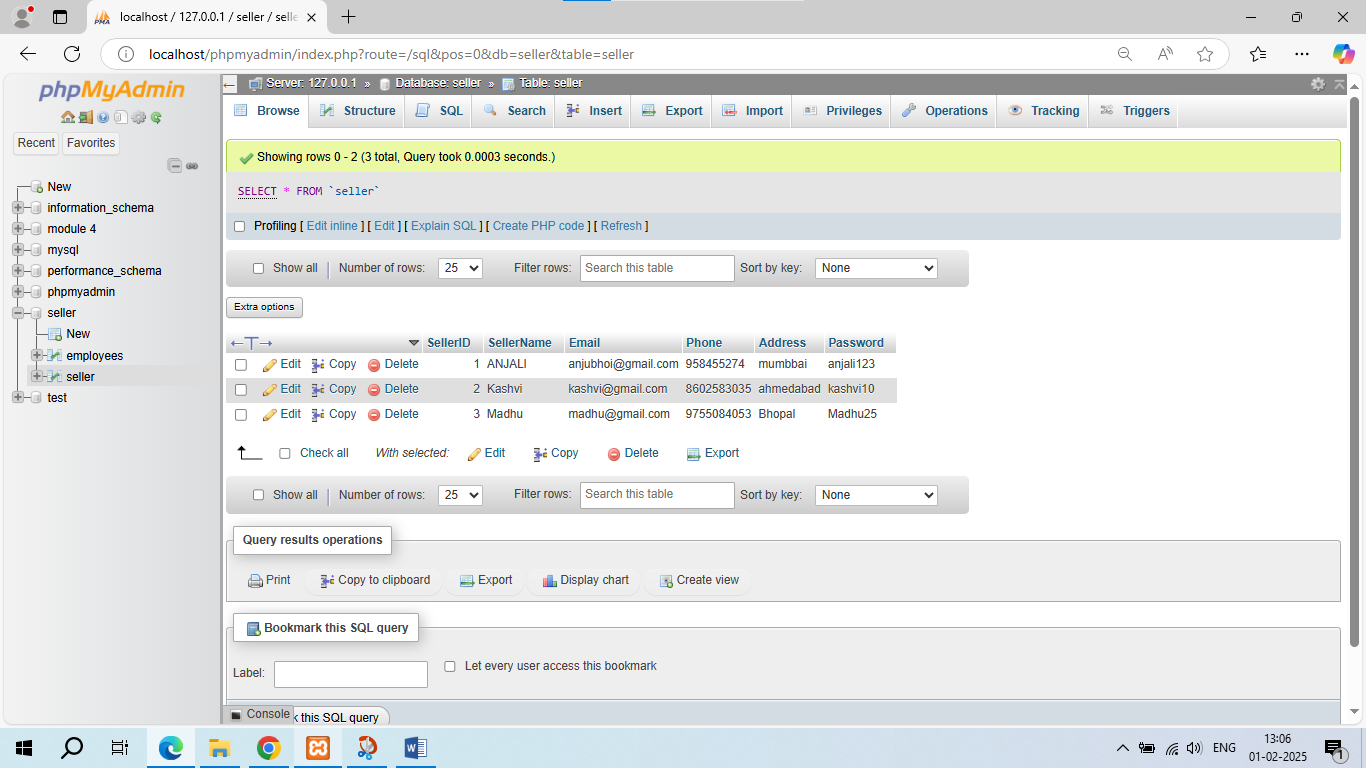
Ans. INSERT INTO `seller`( `SellerName`, `Email`, `Phone`, `Address`, `Password`)

VALUES

('ANJALI', 'anjubhoi@gmail.com',958455274,'mumbbai','anjali123'),

('Kashvi', 'kashvi@gmail.com',8602583035, 'ahmedabad','kashvi10'),

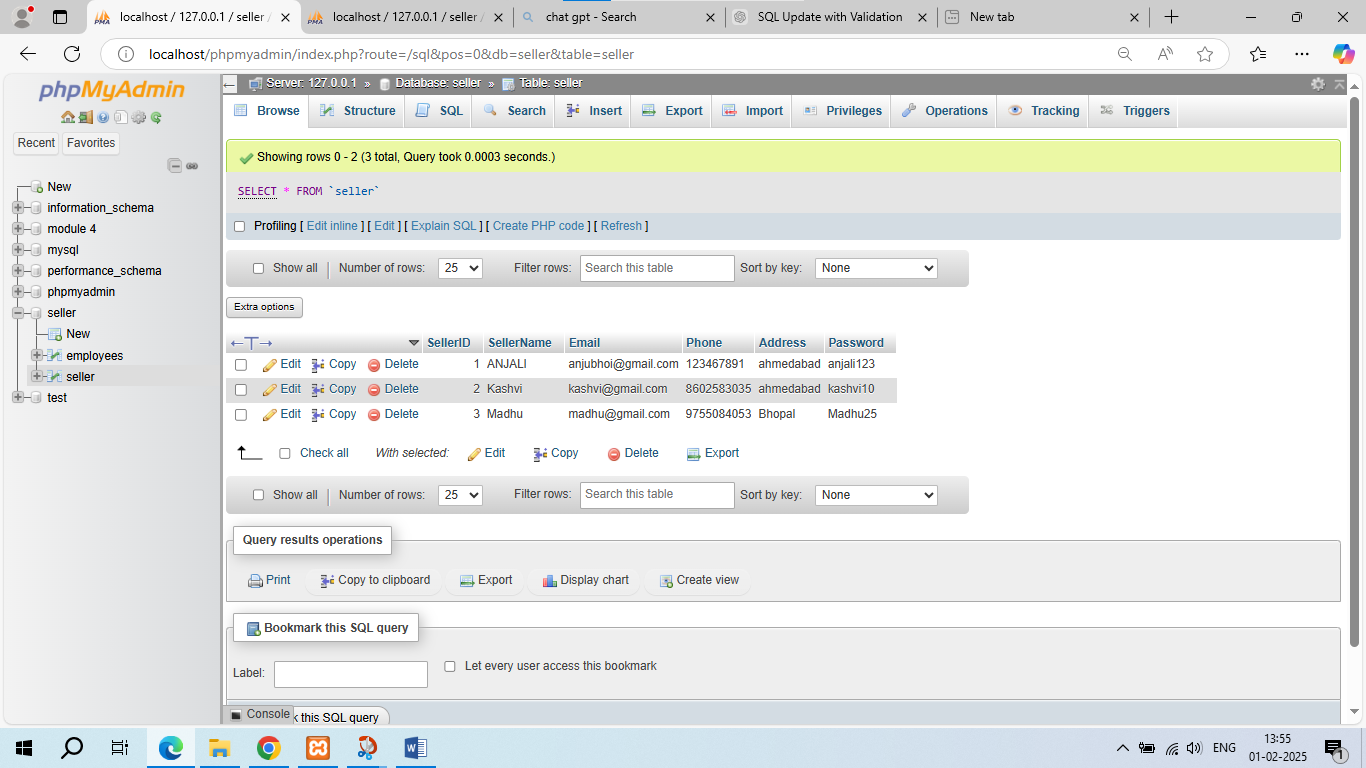
('Madhu', 'madhu@gmail.com',9755084053, 'Bhopal', 'Madhu25');

****

**• Write a query to update data into table with validations.**

**Ans.** UPDATE seller set Phone=123467891,

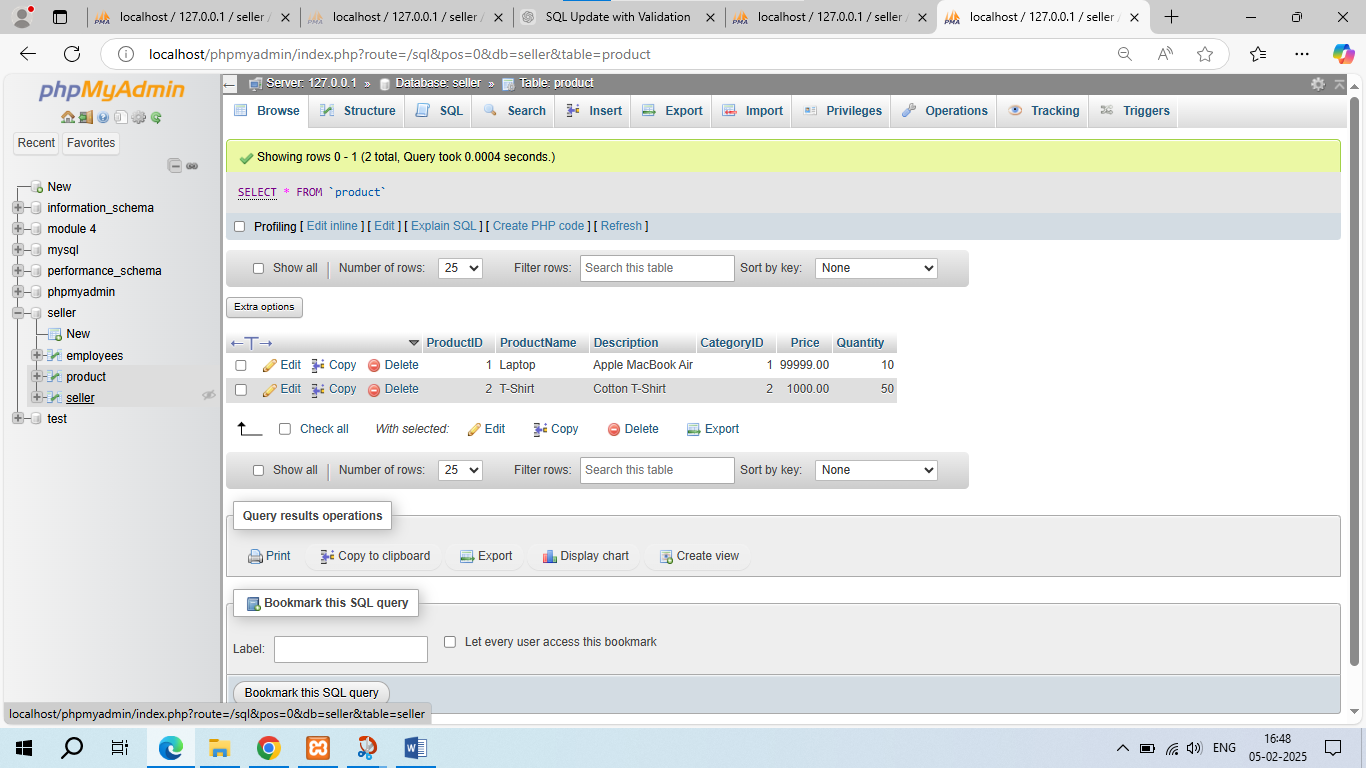
Address=’ahmedabad’ WHERE SellerID=1;

****

**• Write a query to delete data from table with validations.**

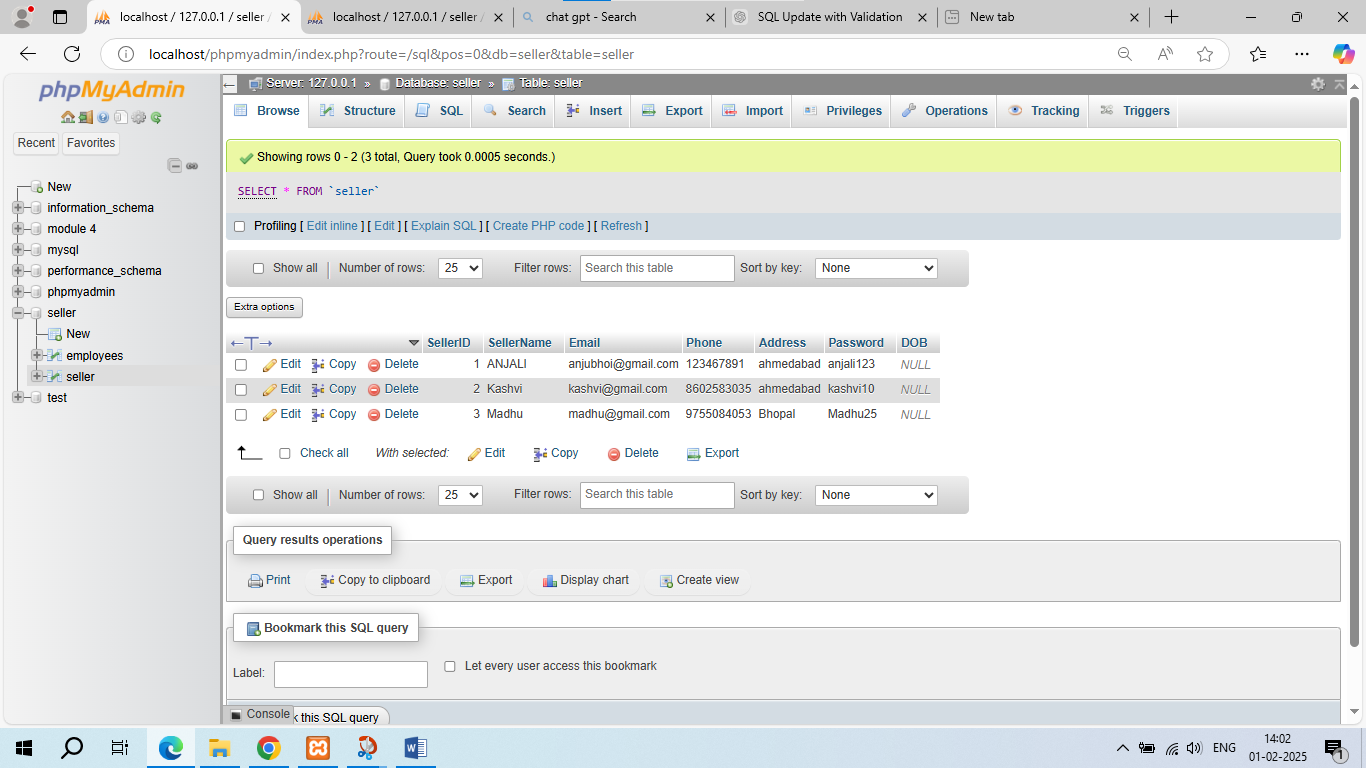
DELETE FROM Product

WHERE ProductID = 3;

****

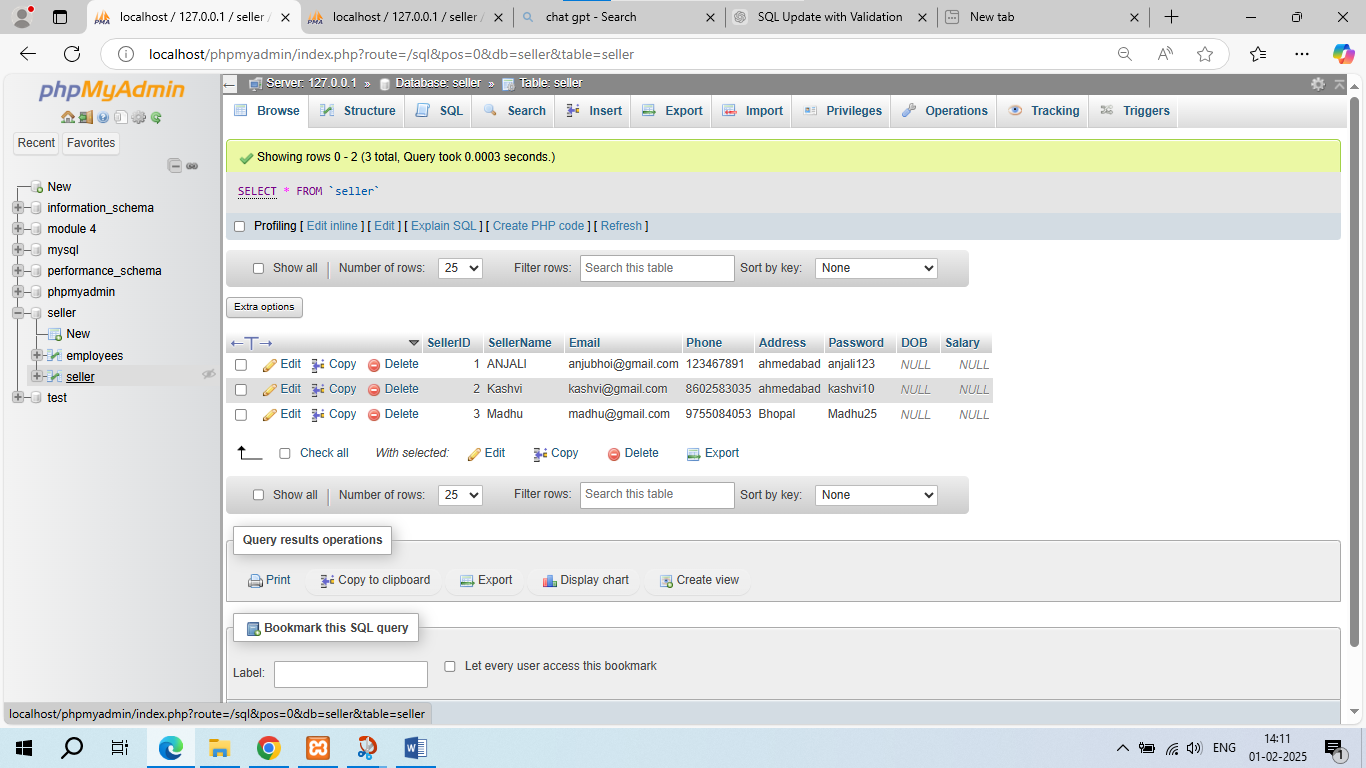
**• Write a query to insert new column in existing table.**

Ans. ALTER TABLE seller ADD COLUMN DOB date;

****

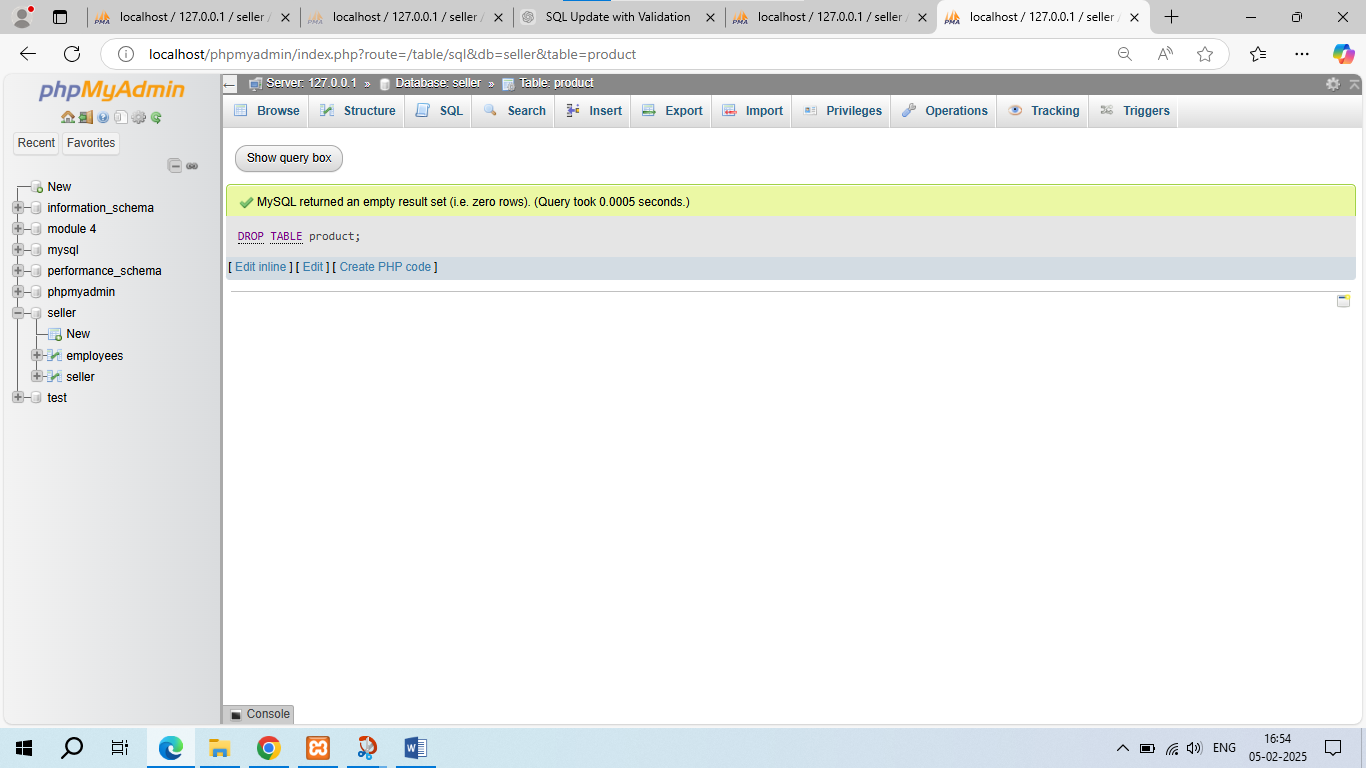
ALTER TABLE seller

ADD Salary DECIMAL (10,2);

****

**• Write a query to drop table and database.**

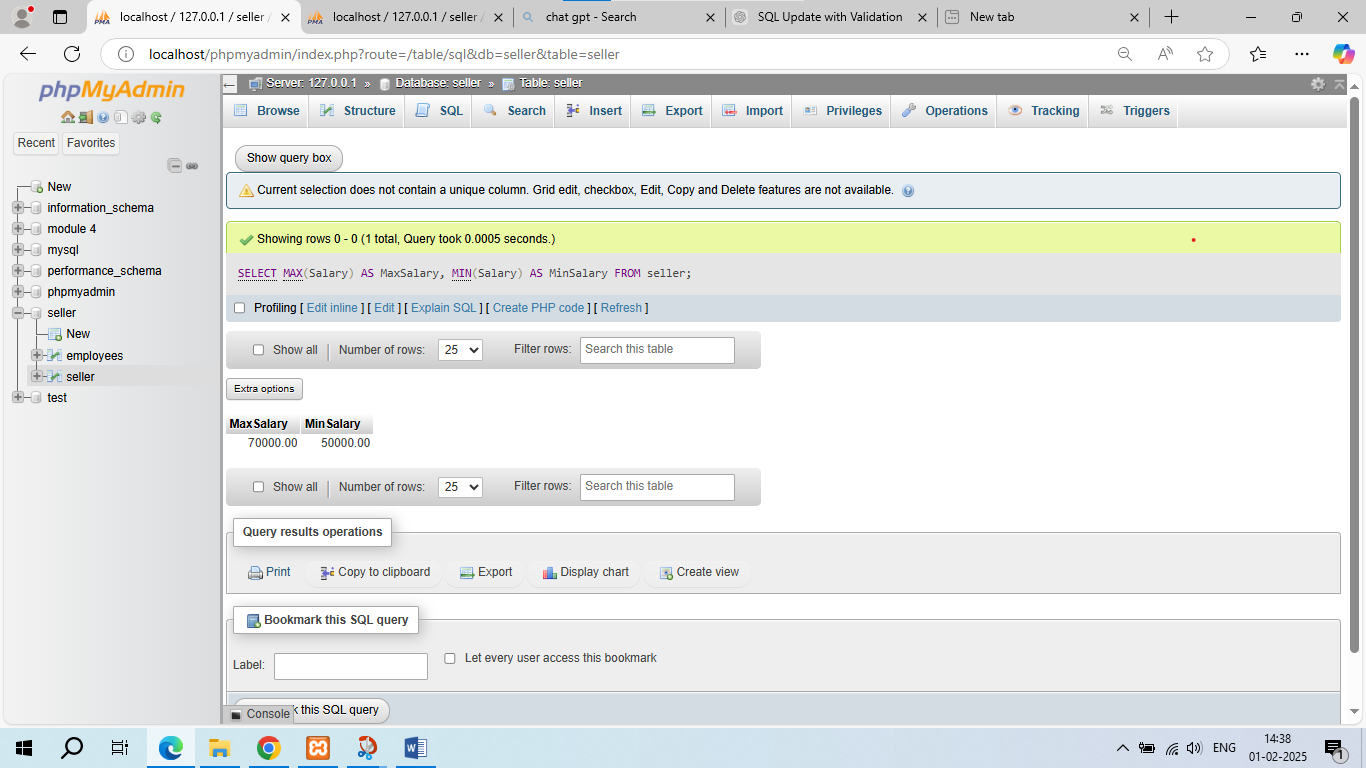
DROP TABLE Product;

****

**• Write a query to find max and min value from table.**

**Ans.** SELECT MAX(Salary) AS Max Salary, MIN(Salary) AS Min Salary

FROM seller;

****

**• Create two tables named Seller and Product apply foreign key in product table Fetch data from both table using different joins.**

CREATE TABLE Product (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(255) NOT NULL,

Description VARCHAR(500),

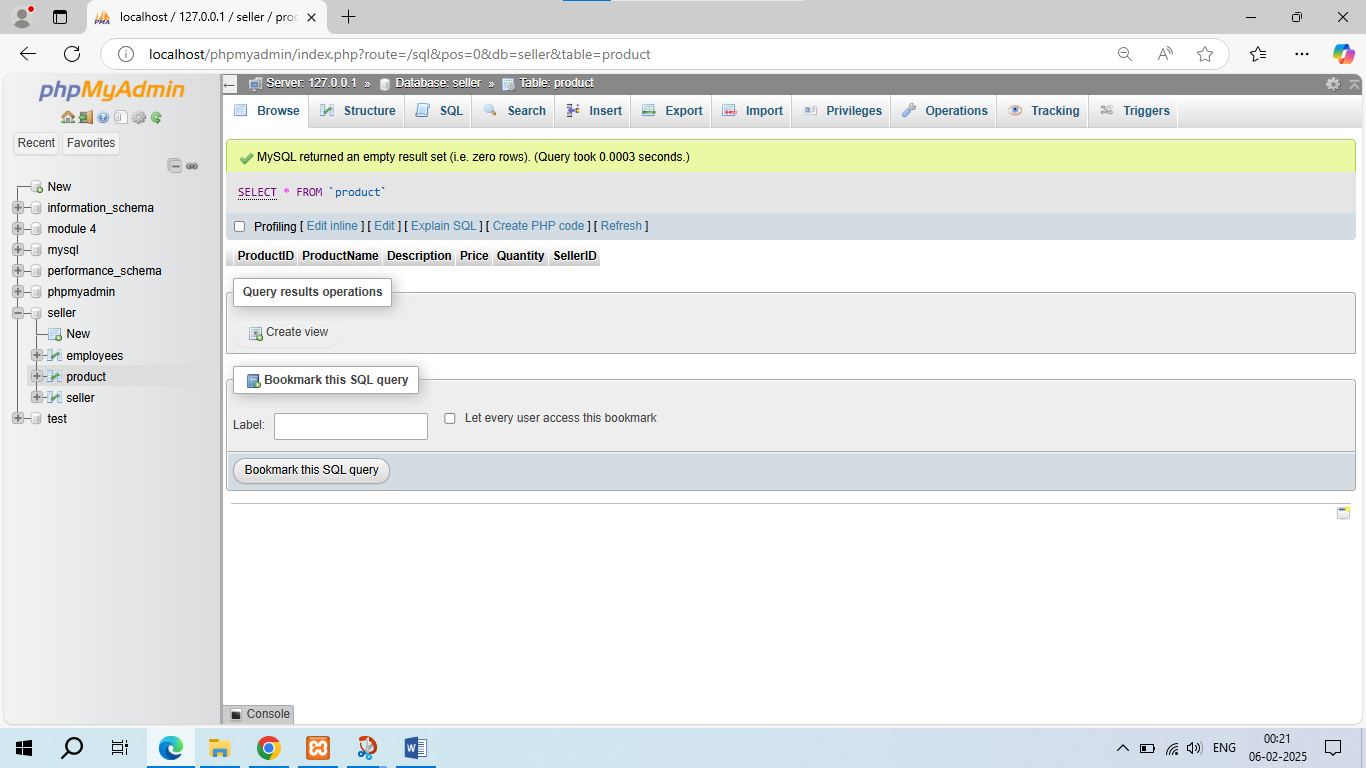
Price DECIMAL(10, 2) NOT NULL,

Quantity INT NOT NULL,

SellerID INT NOT NULL,

FOREIGN KEY (SellerID) REFERENCES Seller(SellerID)

);



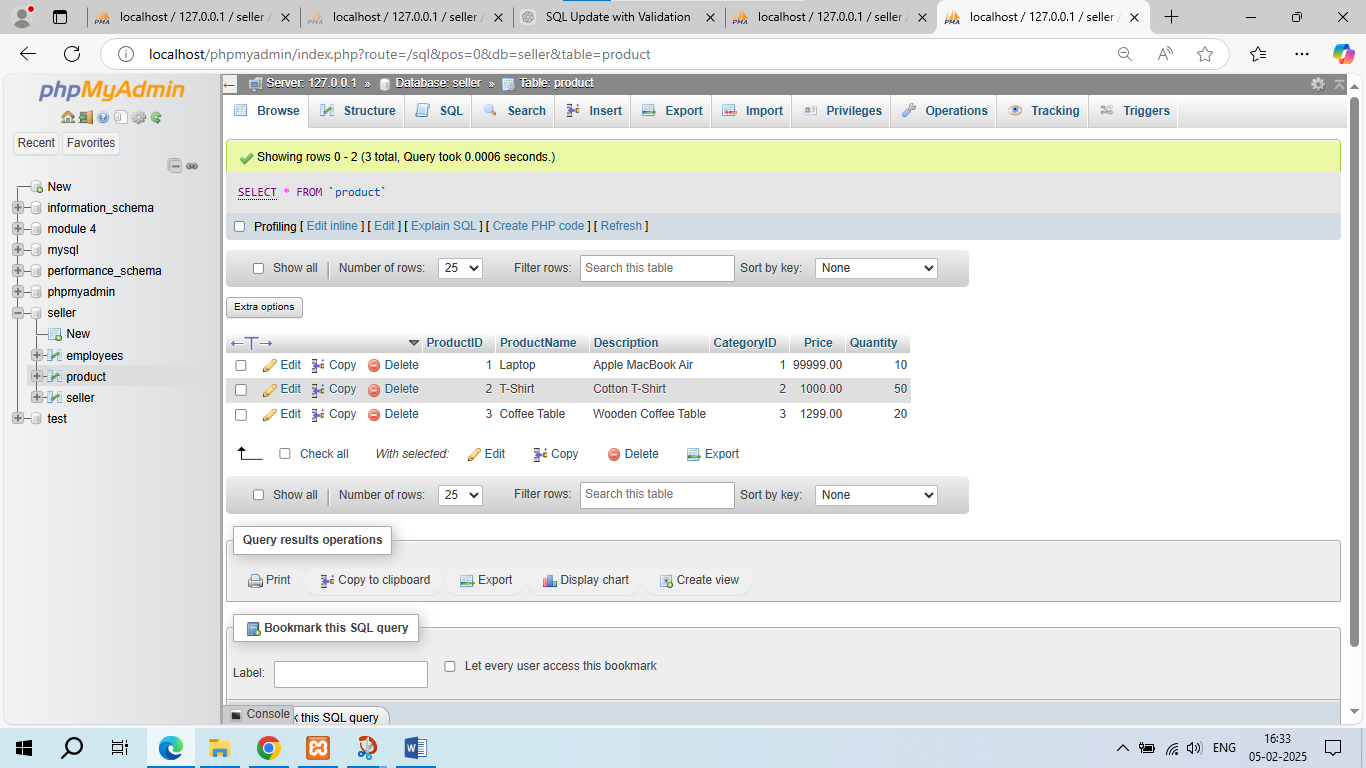
INSERT INTO Product (ProductID, ProductName, Description, CategoryID, Price, Quantity)

VALUES

(1, 'Laptop', 'Apple MacBook Air', 1, 99999, 10),

(2, 'T-Shirt', 'Cotton T-Shirt', 2, 1000, 50),

(3, 'Coffee Table', 'Wooden Coffee Table', 3, 1299, 20);

****

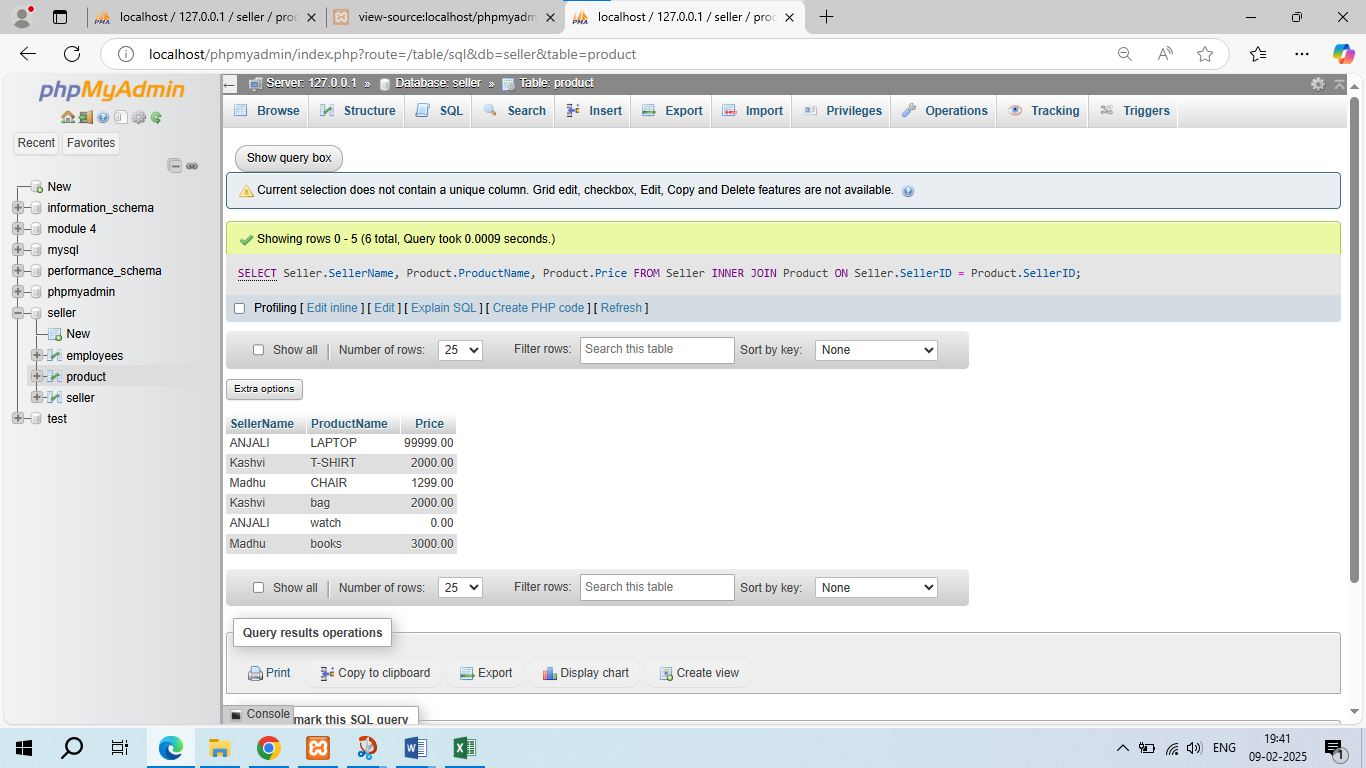
**# INNER JOIN**

**SELECT Seller.SellerName, Product.ProductName, Product.Price**

**FROM Seller**

**INNER JOIN Product**

**ON Seller.SellerID = Product.SellerID;**

****

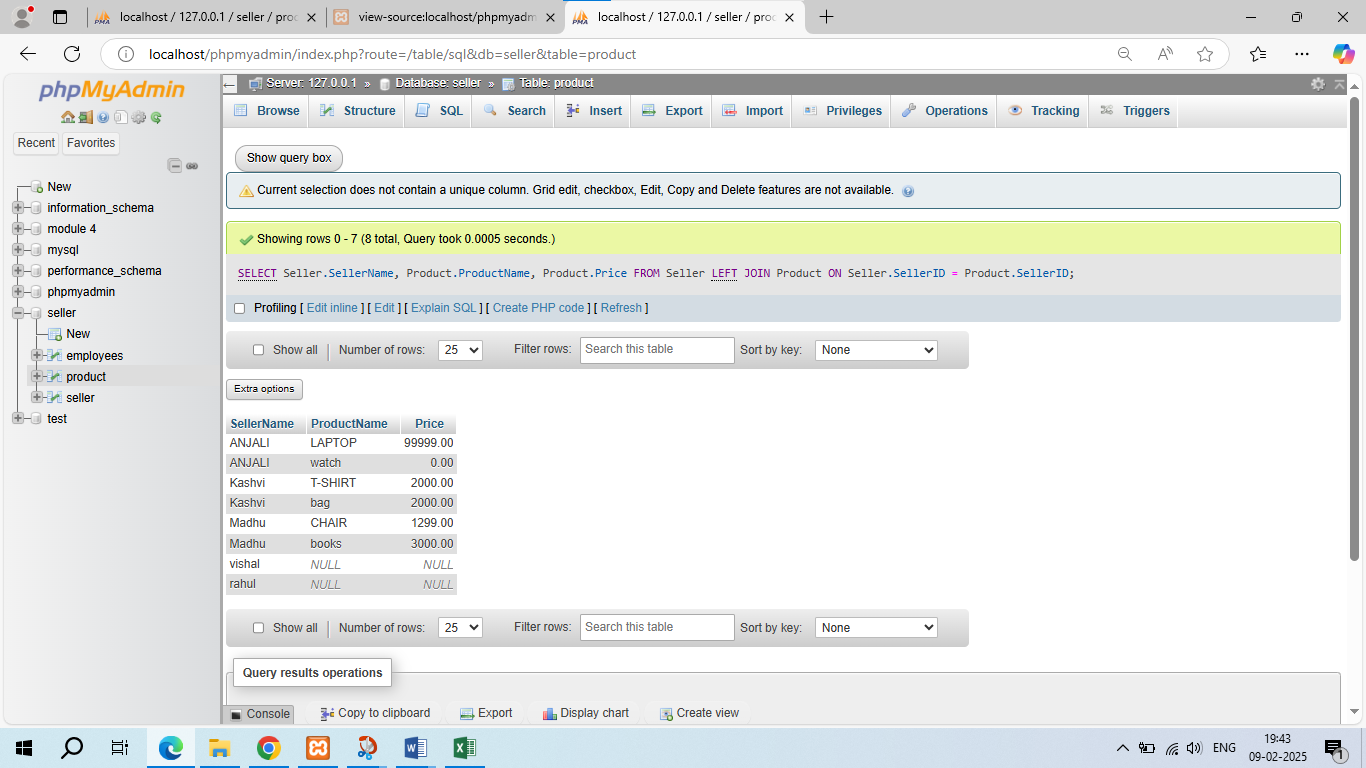
**# LEFT JOIN (or LEFT OUTER JOIN)**

**SELECT Seller.SellerName, Product.ProductName, Product.Price**

**FROM Seller**

**LEFT JOIN Product**

**ON Seller.SellerID = Product.SellerID;**

****

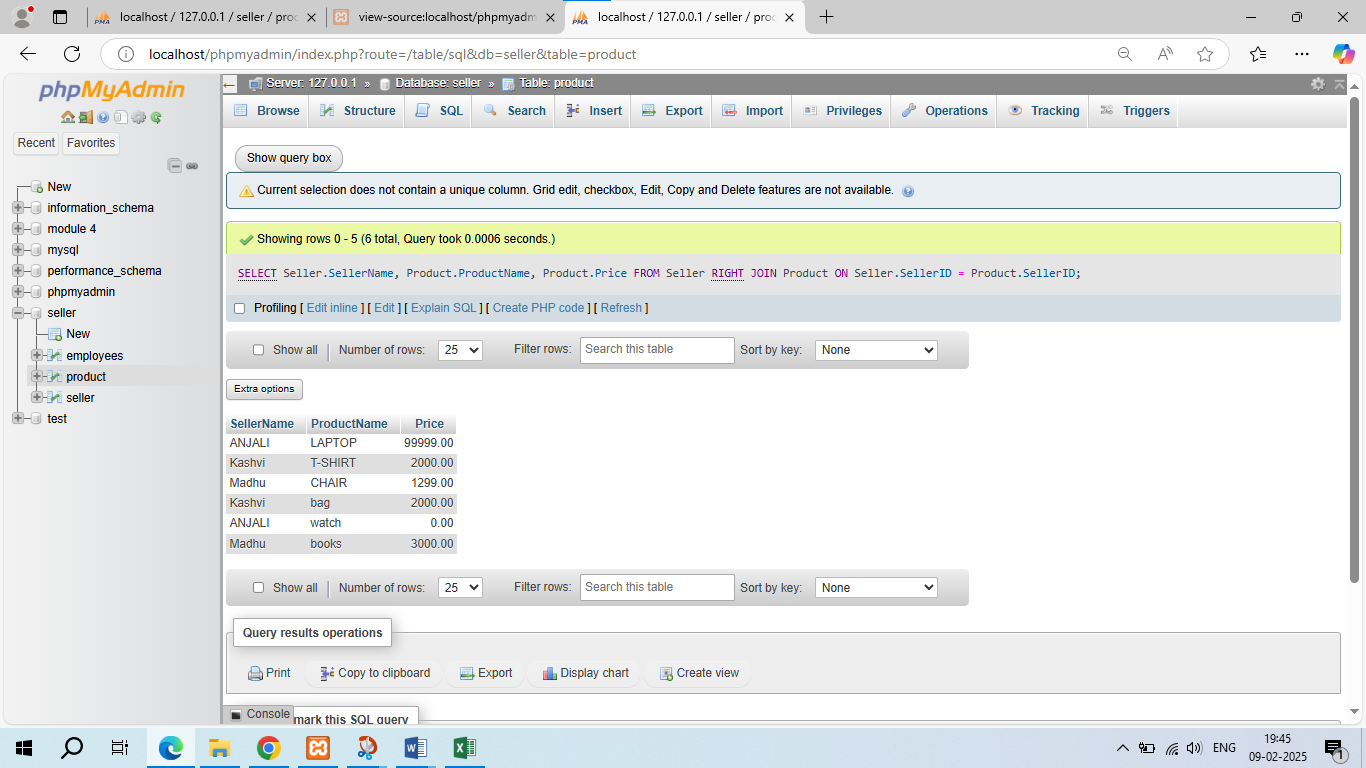
**# RIGHT JOIN (or RIGHT OUTER JOIN)**

**SELECT Seller.SellerName, Product.ProductName, Product.Price**

**FROM Seller**

**RIGHT JOIN Product**

**ON Seller.SellerID = Product.SellerID;**

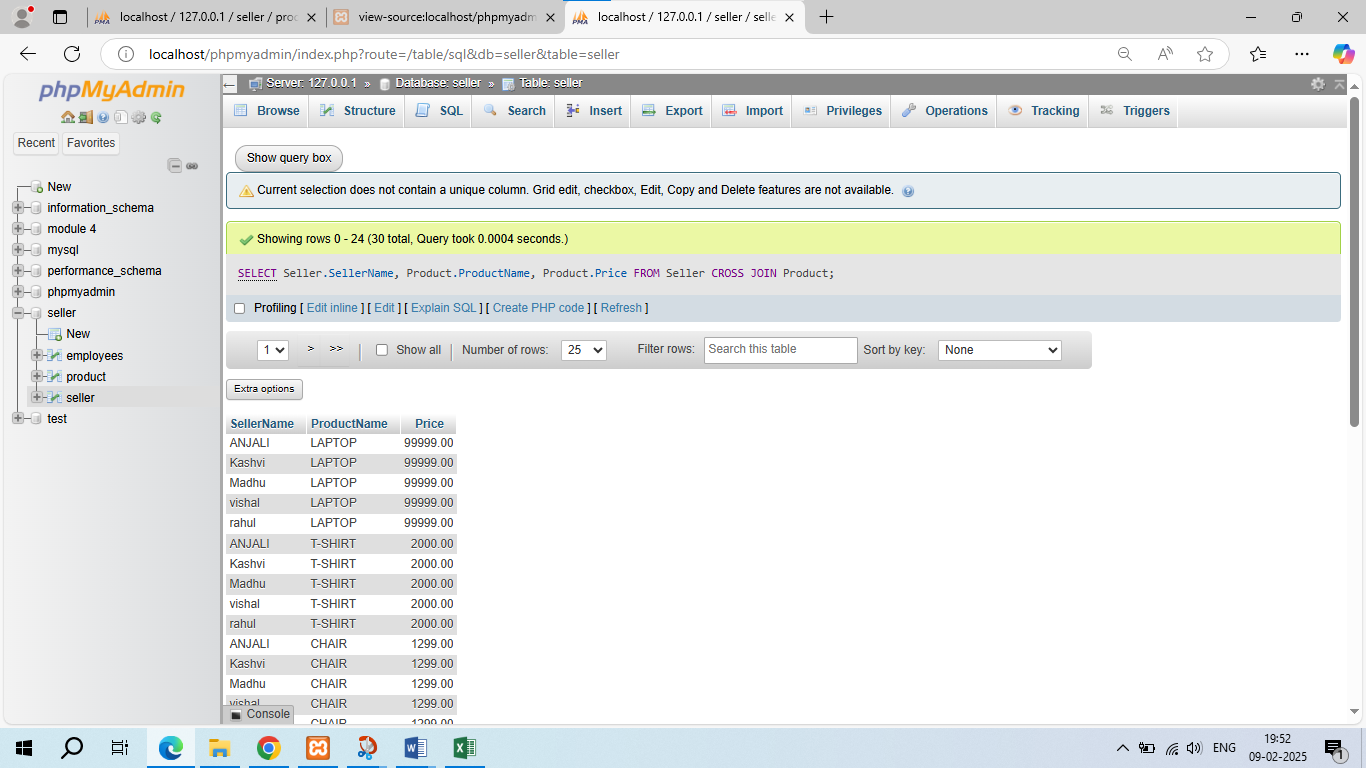
****

**# CROSS JOIN**

**SELECT Seller.SellerName, Product.ProductName, Product.Price**

**FROM Seller**

**CROSS JOIN Product;**

****

**• What is API Testing**

**Ans.** API testing is a type of software testing that involves testing the application programming interfaces (APIs) of a software application or system. APIs are sets of defined rules that enable different software systems to communicate with each other.

API testing involves verifying that the API functions as expected, returns the correct data, and behaves correctly under various conditions. The goal of API testing is to ensure that the API is reliable, stable, and performs as expected.

**• Types of API Testing**

**Ans. Types of API Testing:**

**1. Functional Testing**

Verifies that the API functions as expected and returns the correct data.

**2. Performance Testing**

Evaluates the API's performance under various loads and stress conditions.

**3. Security Testing**

Identifies vulnerabilities and weaknesses in the API's security.

**4. Interoperability Testing**

Verifies that the API works correctly with different software systems and platforms.

**5. Error Handling Testing**

Verifies that the API handles errors and exceptions correctly.

**6. Penetration Testing**

Simulates a cyber-attack to test the API's defences.

**7. Load Testing**

Tests the API's performance under a large number of concurrent requests.

**8. Stress Testing**

Tests the API's performance under extreme conditions.

**9. End-to-End Testing**

Tests the entire API workflow from start to finish.

**10. Integration Testing**

Tests how the API integrates with other components and systems.

**11. Unit Testing**

Tests individual API components and functions.

**12. Compatibility Testing**

Tests the API's compatibility with different browsers, devices, and platforms.

**13. Usability Testing**

Tests the API's usability and user experience.

**14. Scalability Testing**

Tests the API's ability to scale up or down to meet changing demands.

**15. Reliability Testing**

Tests the API's reliability and consistency over time**.**

**• What is Responsive Testing?**

**Ans.** Responsive testing, also known as responsive design testing, is a type of testing that ensures a website or application's user interface (UI) adapts correctly to different screen sizes, devices, and orientations. The goal of responsive testing is to verify that the UI is usable, readable, and visually appealing across various devices and screen sizes.

**Types of Responsive Testing:**

**1. Device Testing**: Testing on different devices, such as smartphones, tablets, laptops, and desktops.

**2. Screen Size Testing:** Testing on different screen sizes, such as 320px, 768px, 1024px, and 1920px**.**

**3. Orientation Testing:** Testing in different orientations, such as portrait and landscape.

**4. Browser Testing:** Testing on different browsers, such as Chrome, Firefox, Safari, and Edge.

**5. Operating System Testing:** Testing on different operating systems, such as Windows, macOS, iOS, and Android.

**• Which types of tools are available for Responsive Testing**

**Ans.** There are several types of tools available for responsive testing:

**# Manual Testing Tools**

1. **Browser Stack:** A cloud-based platform for manual testing on different browsers and devices.

2. **Cross Browser Testing:** A cloud-based platform for manual testing on different browsers and devices.

3. **Responsinator:** A free online tool for testing responsive designs on different devices and screen sizes.

**# Automated Testing Tools**

**1. Selenium:** An open-source tool for automating web browsers and testing responsive designs.

**2. Appium:** An open-source tool for automating mobile applications and testing responsive designs.

**3. Cypress:** A JavaScript-based tool for automating web applications and testing responsive designs.

**# Emulation Tools**

**1. Google Chrome Dev Tools**: A built-in tool for emulating different devices and screen sizes.

**2. Mozilla Responsive Design Mode:** A built-in tool for emulating different devices and screen sizes.

**3. Microsoft Edge Dev Tools**: A built-in tool for emulating different devices and screen sizes.

**# Virtual Machine Tools**

**1. Virtual Box:** A free and open-source tool for creating virtual machines and testing responsive designs.

**2. VMware**: A commercial tool for creating virtual machines and testing responsive designs.

**# Online Testing Tools**

**1. Test Object:** A cloud-based platform for testing responsive designs on different devices and browsers.

**2. Device Anywhere:** A cloud-based platform for testing responsive designs on different devices and browsers.

**3. Keynote**: A cloud-based platform for testing responsive designs on different devices and browsers.

**# Mobile Emulation Tools**

**1. Mobile Phone Emulator:** A free online tool for emulating different mobile devices and testing responsive designs.

**2. Screen fly:** A free online tool for emulating different devices and screen sizes.

**3. Responsivator:** A free online tool for emulating different devices and screen sizes.

**• What is the full form of .ipa , .apk**

**Ans.** -.ipa: iOS Application (file format used for iPhone, iPad, and iPod touch applications)

-.apk: Android Package (file format used for Android applications)

**• How to create step for to open the developer option mode ON?**

**Ans. Here are the steps to enable Developer Options on an Android device:**

**For Android 4.2 and above:**

1. Go to your device's Settings.

2. Scroll down and select About phone or About device.

3. Scroll down to the Build number section.

4. Tap the Build number 7 times. You will see a message that says "You are now a developer!"

5. Go back to the Settings menu and you will see Developer options.

**For Android 11 and above:**

1. Go to your device's Settings.

2. Scroll down and select About phone or About device.

3. Scroll down to the Build number section.

4. Tap the Build number 7 times. You will see a message that says "You are now a developer!"

5. Go back to the Settings menu and select System or Advanced.

6. Select Developer options.