Assignment

Assignment of Basic Database Concepts

Basic Database Concepts Assignment - 1

|  |  |
| --- | --- |
| Name: | Ali Ahasan |
| Position: | Intern |
| Tracking Number: | J2DFEEB |
| Team: | InnovX Team of Business Automation Limited |
| Session: | Basic Database Concepts with MySQL |

**Assignment Overview:**

In this assignment, you will design, create, and manage a simple e-commerce database. You will perform various tasks, including creating tables, inserting data, and writing SQL queries to retrieve and manipulate the data. This assignment will help you understand the practical aspects of using MySQL in an e-commerce context.

**Task:**

* Design a database schema for a simple e-commerce application, including tables for products, customers, orders with minimum column.
* Insert sample data into the e-commerce database created.
* Prepare a query to Retrieve customer information based on a specific order.
* Prepare a query to find the total number of orders placed by each customer.
* Prepare a query to get the list of top 10 customers who have ordered the most times.
* List of products that have been ordered the most number of times with the number of orders.

**Instructions:**

1. Complete all the query as outlined above and attach query & output screenshot in single file.
2. Name the file as "name\_yourId\_db\_assignment".
3. Submit the file through the provided Google Form.

Table of Contents

[Schema design 3](#_Toc173378402)

[Tables: 3](#_Toc173378403)

[Diagram: 4](#_Toc173378404)

[Table Making: 4](#_Toc173378405)

[Code: 4](#_Toc173378406)

[Input: 5](#_Toc173378407)

[Output: 6](#_Toc173378408)

[Table Data Insertion: 7](#_Toc173378409)

[Code: 7](#_Toc173378410)

[Input: 9](#_Toc173378411)

[Output: 12](#_Toc173378412)

[Table Data Query: 15](#_Toc173378413)

[Query code: 15](#_Toc173378414)

[Input: 16](#_Toc173378415)

[Output: 16](#_Toc173378416)

E-commerce database

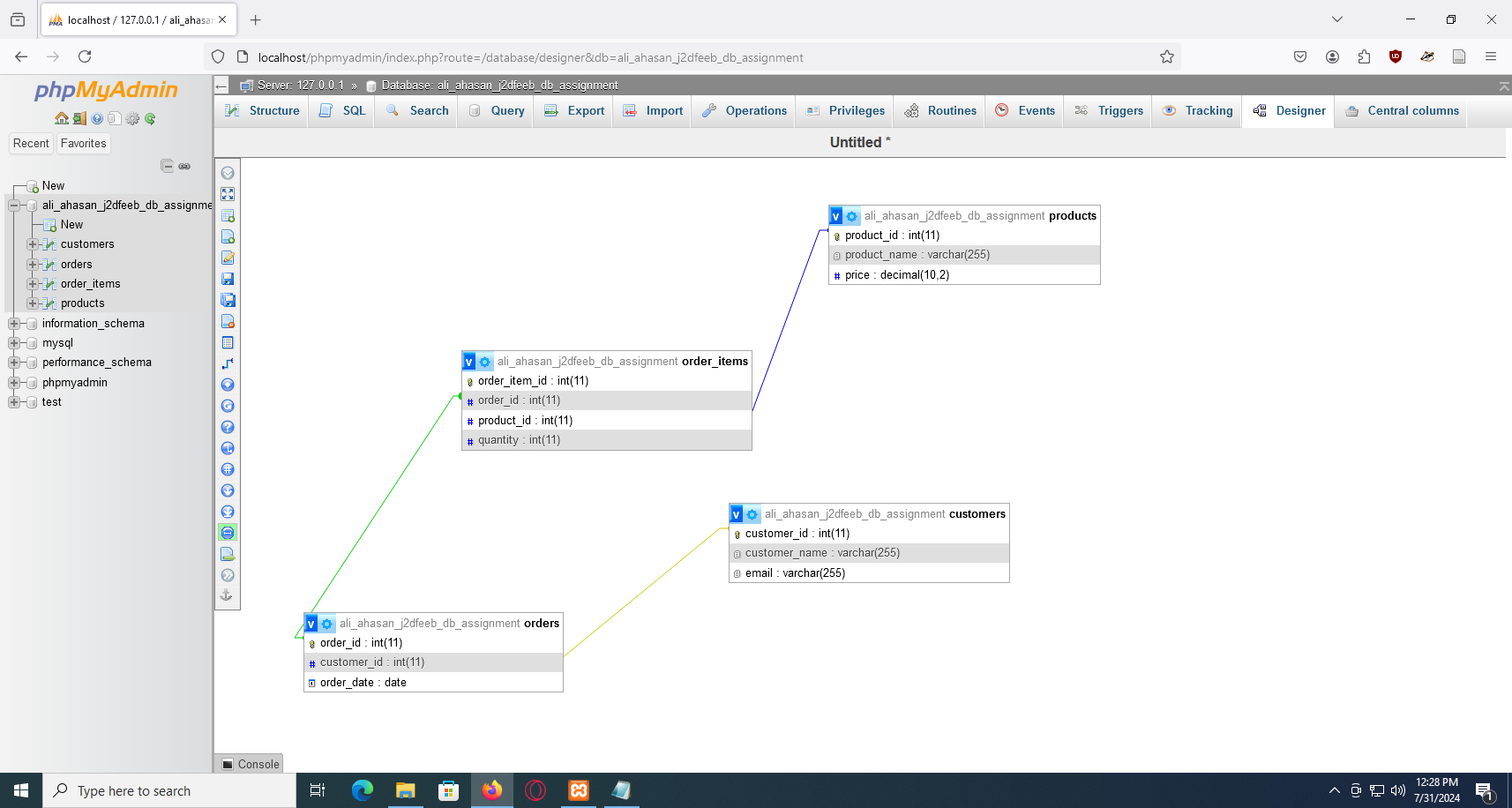
# Schema design

Database Schema

## Tables:

1. **Products**: This table holds all the information about the items available for purchase.
   * product\_id (INT, Primary Key)
   * product\_name (VARCHAR)
   * price (DECIMAL)
2. **Customers**: This Table Stores which are all the details about the customers in specific.
   * customer\_id (INT, Primary Key)
   * customer\_name (VARCHAR)
   * email (VARCHAR)
3. **Orders**: For this table the details that consists of the orders which are placed by the customers.
   * order\_id (INT, Primary Key)
   * customer\_id (INT, Foreign Key from Customers)
   * order\_date (DATE)
4. **Order\_Items**: This one table represents what kind of items that are in each of their order.
   * order\_item\_id (INT, Primary Key)
   * order\_id (INT, Foreign Key from Orders)
   * product\_id (INT, Foreign Key from Products)
   * quantity (INT)

## Diagram:



# Table Making:

Using DDL (Data Definition Language) to make the table.

## Code:

CREATE TABLE Products (

product\_id INT AUTO\_INCREMENT PRIMARY KEY,

product\_name VARCHAR(255),

price DECIMAL(10, 2)

);

CREATE TABLE Customers (

customer\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_name VARCHAR(255),

email VARCHAR(255)

);

CREATE TABLE Orders (

order\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

order\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

CREATE TABLE Order\_Items (

order\_item\_id INT AUTO\_INCREMENT PRIMARY KEY,

order\_id INT,

product\_id INT,

quantity INT,

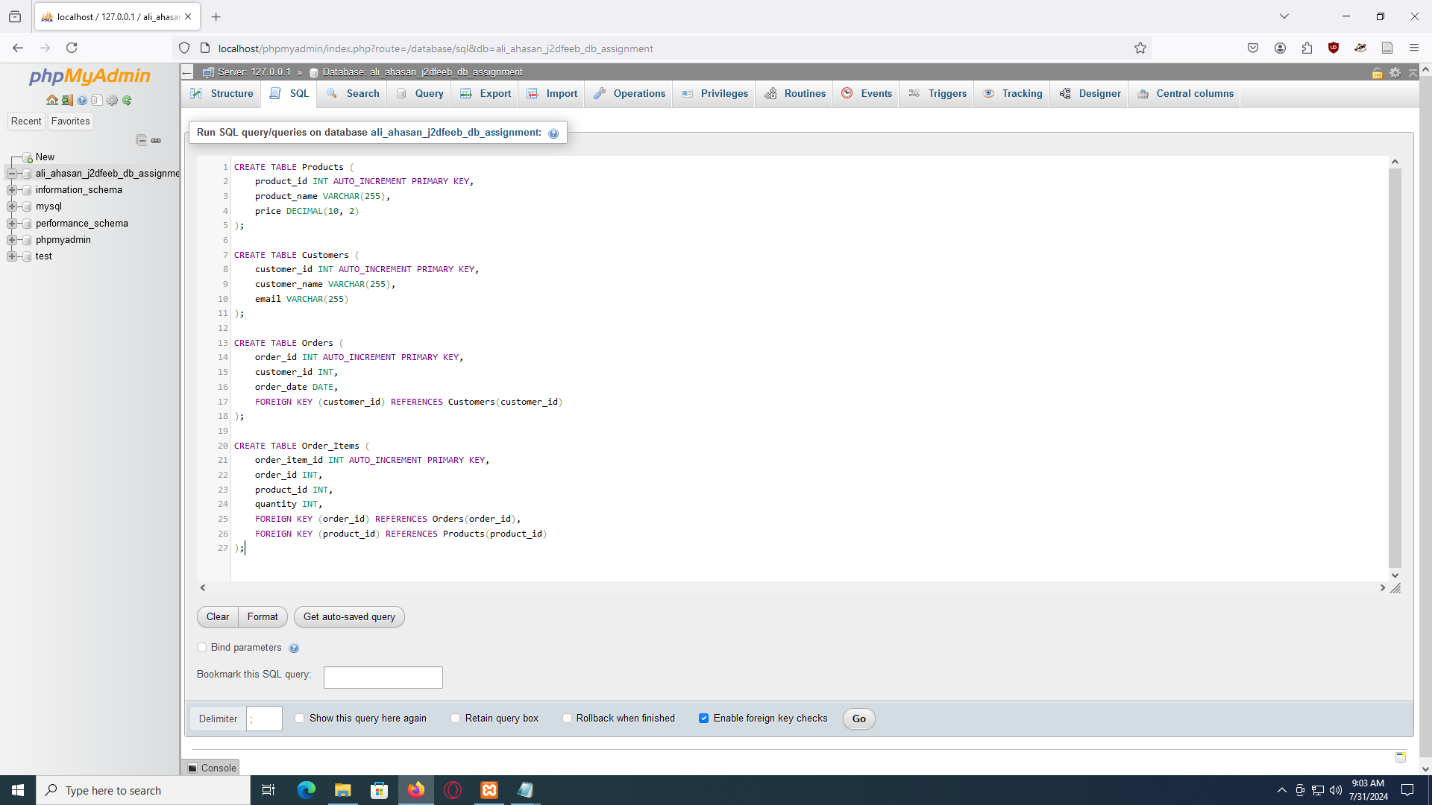
FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),

FOREIGN KEY (product\_id) REFERENCES Products(product\_id)

);

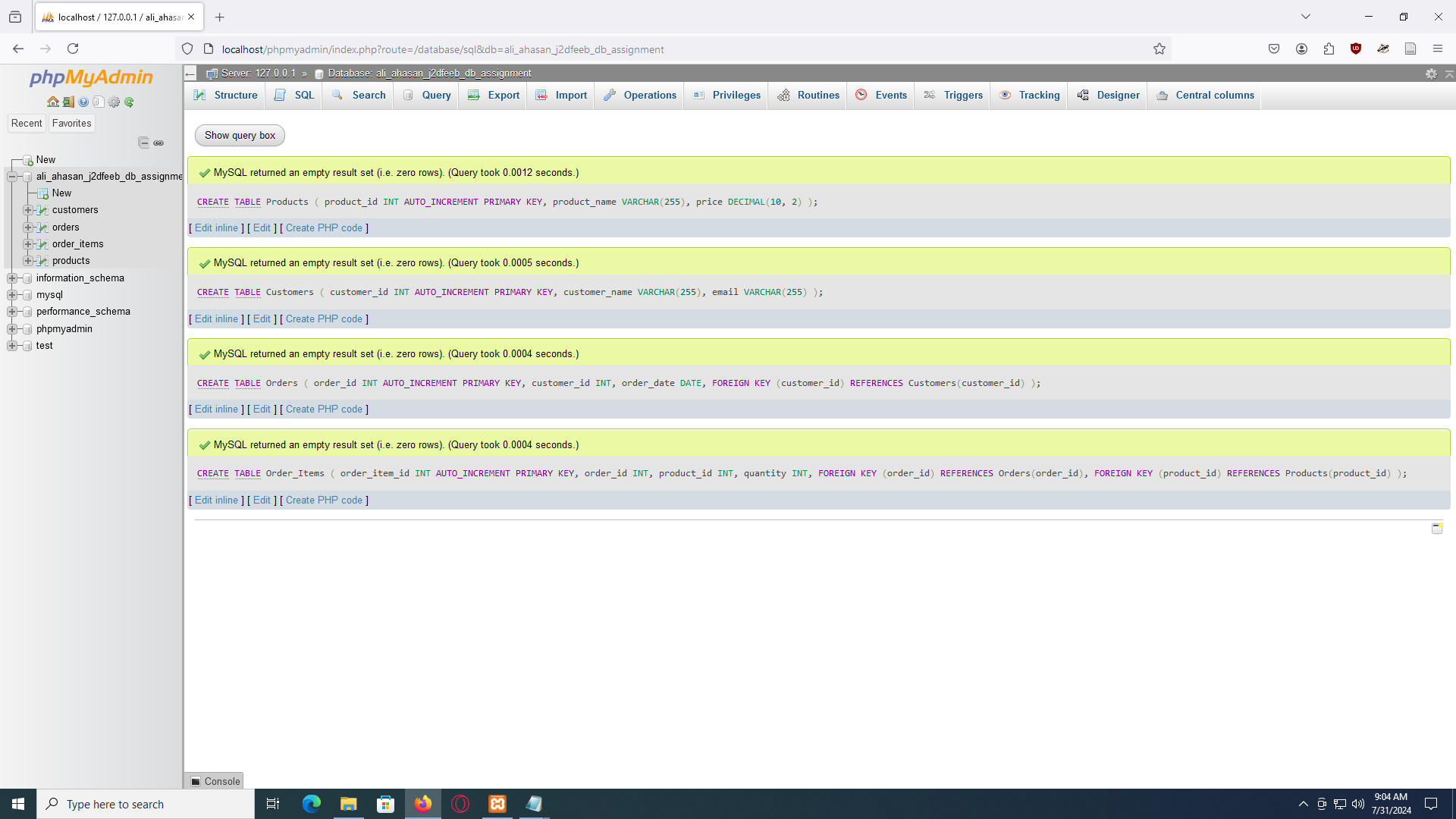
## Input:

1

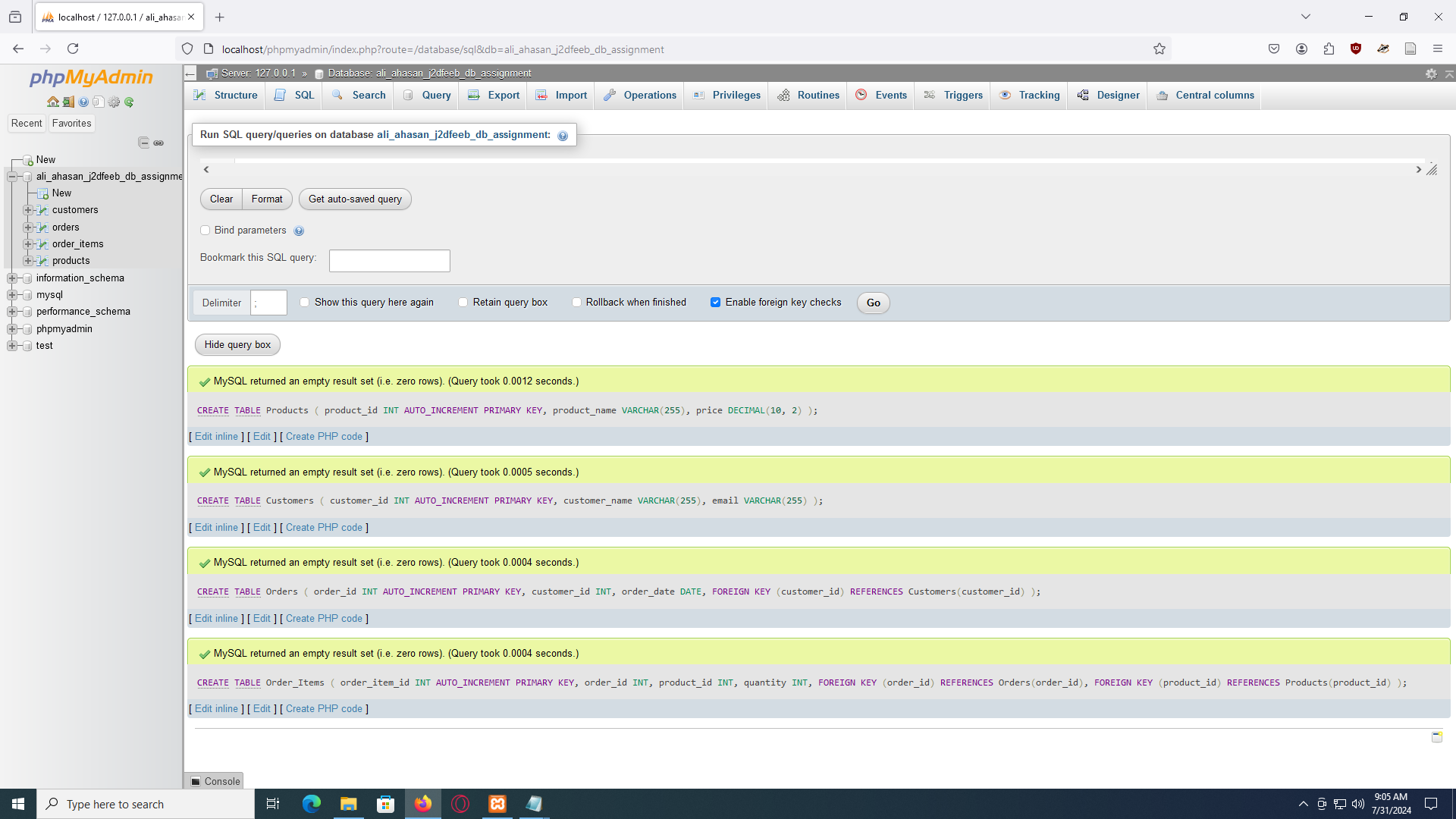


## Output:

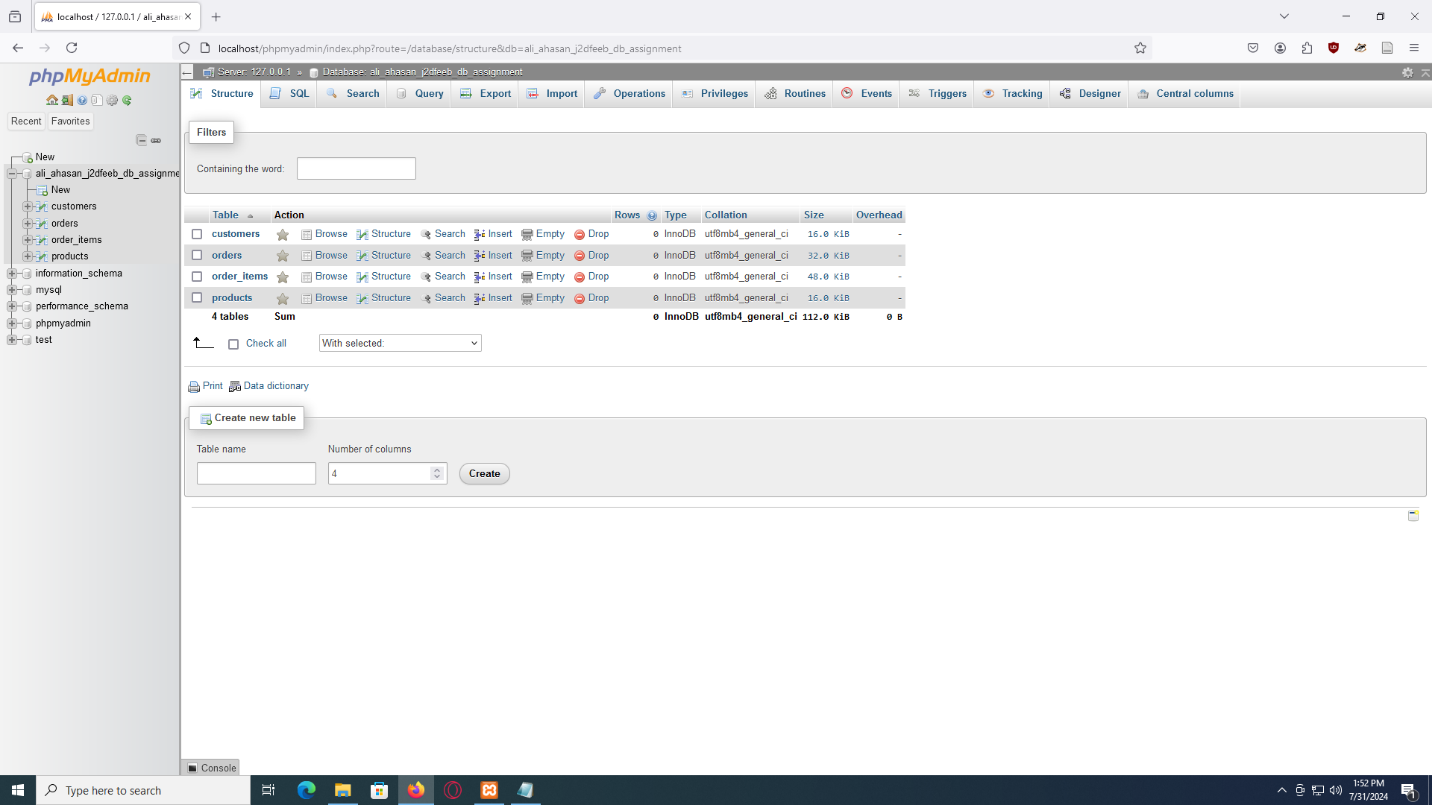
1



2



3



# Table Data Insertion:

Using DML (Data Manipulation Language) to insert the data.

## Code:

-- Insert or add example data into Products table

INSERT INTO Products (product\_name, price) VALUES

('Desktop', 72000.00),

('Headphones', 8000.00),

('Monitor', 15000.00),

('Tablet', 45000.00),

('Camera', 7000.00),

('Watch', 2500.00),

('Backpack', 700.00),

('Sunglasses', 850.00),

('Book', 200.00),

('Desk Lamp', 450.00),

('Mouse', 250.00),

('External Hard Drive', 7200.00),

('USB Flash Drive', 200.00),

('Smart TV', 15000.00),

('Portable Speaker', 1500.00),

('Charger', 250.00),

('Laptop Bag', 500.00),

('Gaming Console', 30000.00),

('Video Game', 600.00),

('Smartphone Case', 150.00),

('Earbuds', 1300.00);

-- Insert or add example data into Customers table

INSERT INTO Customers (customer\_name, email) VALUES

('Tarzan Smith', 'tarzan@example.com'),

('Anna Lockford', 'anna@example.com'),

('Alice Johnson', 'alice@example.com'),

('Bob Brown', 'bob@example.com'),

('Carol White', 'carol@example.com'),

('David Green', 'david@example.com'),

('Eva Blue', 'eva@example.com'),

('Frank Wright', 'frank@example.com'),

('Grace Hall', 'grace@example.com'),

('Henry Ford', 'henry@example.com'),

('Laura Craft', 'laura@example.com'),

('Omar Little', 'omar@example.com'),

('Terry Crews', 'terry@example.com'),

('Nina Williams', 'nina@example.com'),

('Larry Page', 'larry@example.com'),

('Sergey Brin', 'sergey@example.com'),

('Tim Cook', 'tim@example.com'),

('Sundar Pichai', 'sundar@example.com'),

('Elon Musk', 'elon@example.com'),

('Jeff Bezos', 'jeff@example.com');

-- Insert or add example data into Orders table

INSERT INTO Orders (customer\_id, order\_date) VALUES

(1, '2024-06-11'),

(2, '2024-06-22'),

(3, '2023-07-03'),

(4, '2023-07-04'),

(5, '2023-07-05'),

(6, '2023-07-06'),

(7, '2023-07-07'),

(8, '2023-07-08'),

(9, '2023-07-09'),

(10, '2023-07-10'),

(11, '2023-08-01'),

(12, '2023-08-02'),

(13, '2023-08-03'),

(14, '2023-08-04'),

(15, '2023-08-05'),

(16, '2023-08-06'),

(17, '2023-08-07'),

(18, '2023-08-08'),

(19, '2023-08-09'),

(20, '2023-08-10');

-- Insert or add example data into Order\_Items table

INSERT INTO Order\_Items (order\_id, product\_id, quantity) VALUES

(1, 1, 1),

(1, 3, 2),

(2, 2, 1),

(3, 4, 1),

(4, 5, 2),

(5, 6, 3),

(6, 7, 1),

(7, 8, 2),

(8, 9, 1),

(9, 1, 1),

(10, 2, 2),

(10, 3, 3),

(10, 14, 1),

(10, 15, 2),

(11, 11, 2),

(12, 12, 3),

(13, 13, 1),

(14, 14, 1),

(15, 15, 4),

(16, 16, 1),

(17, 17, 1),

(18, 18, 2),

(19, 19, 3),

(20, 20, 1),

(11, 12, 1),

(12, 11, 2),

(13, 13, 3),

(14, 11, 1),

(15, 18, 2),

(16, 16, 3),

(17, 18, 2),

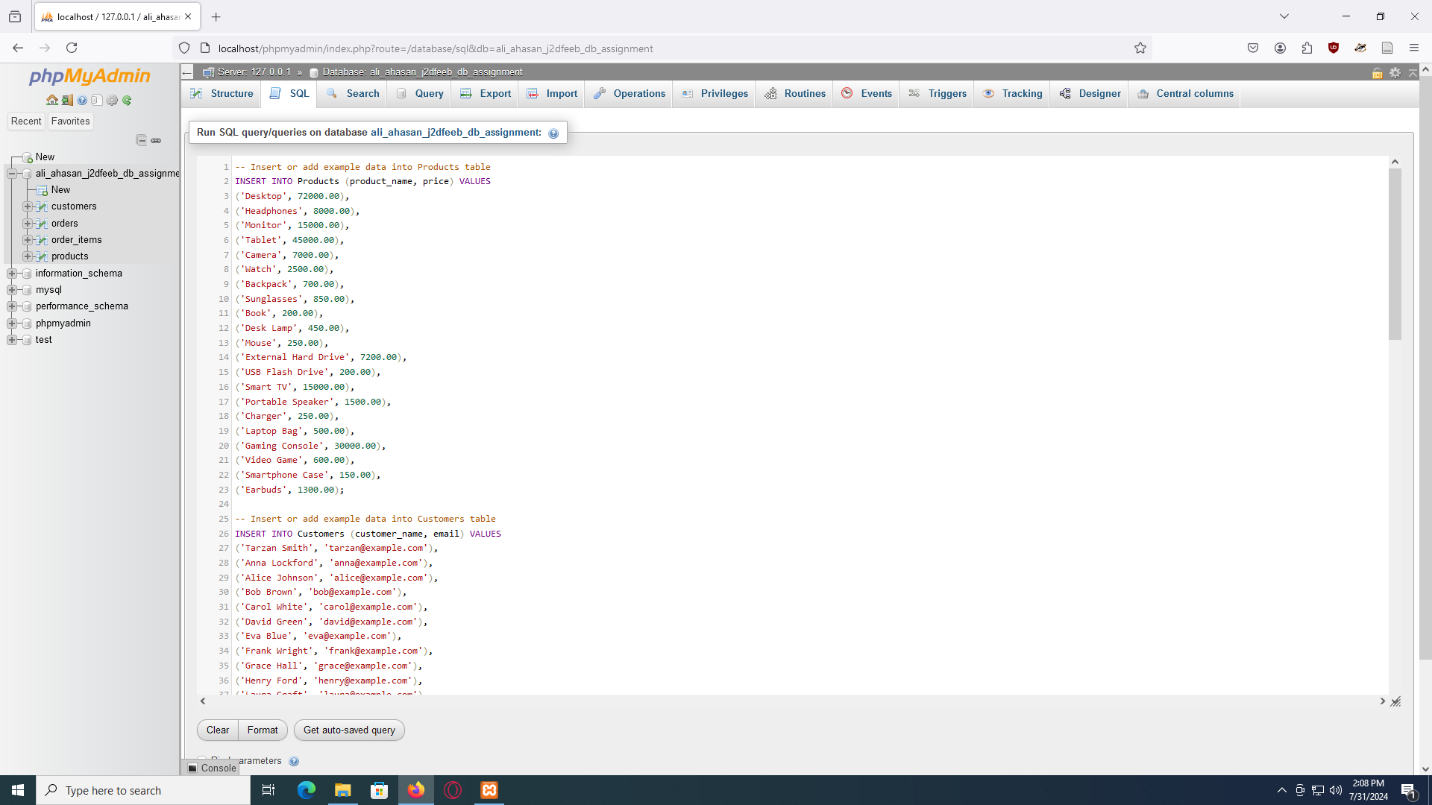
(18, 18, 1),

(19, 9, 1),

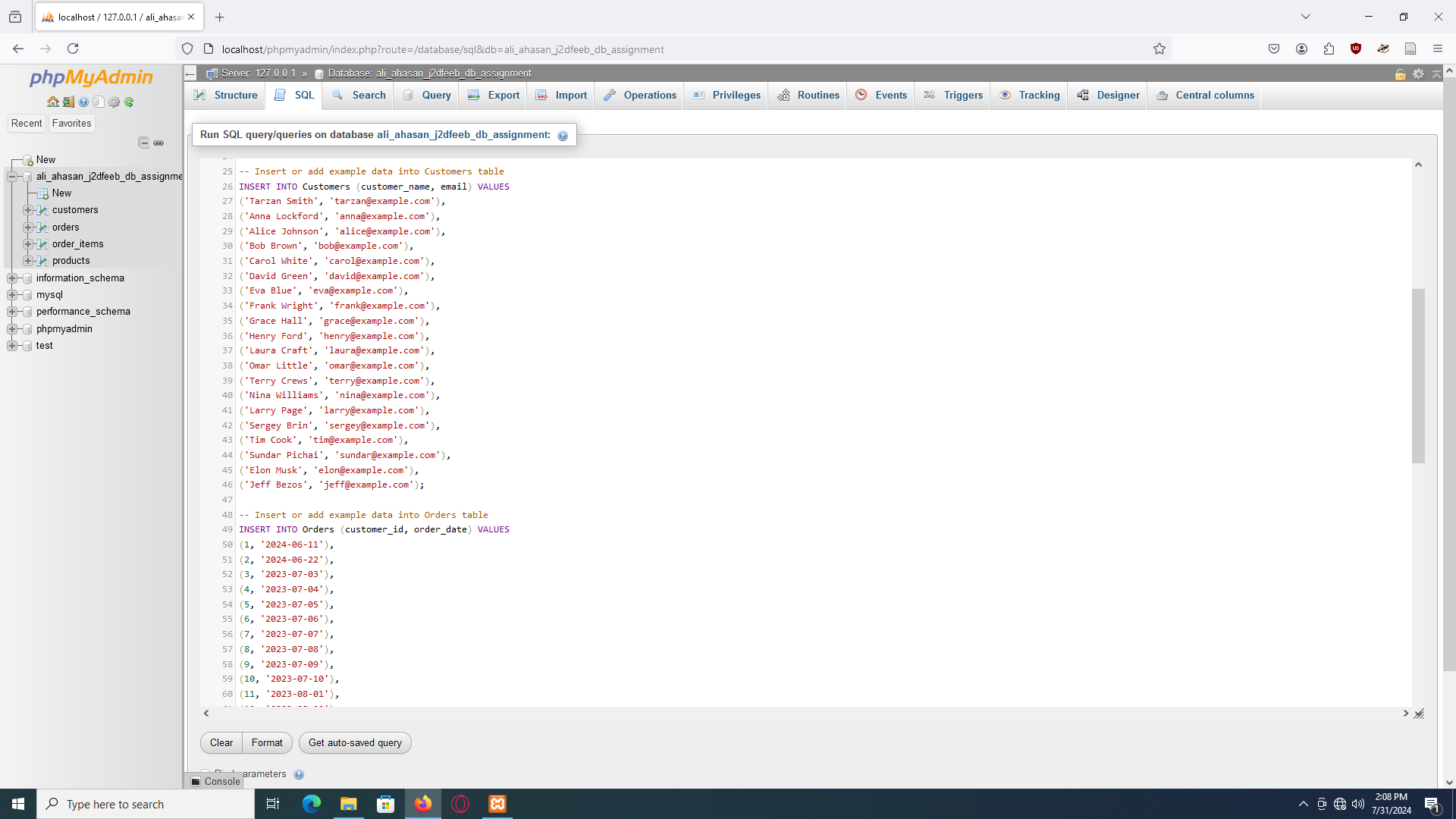
(20, 18, 2);

## Input:

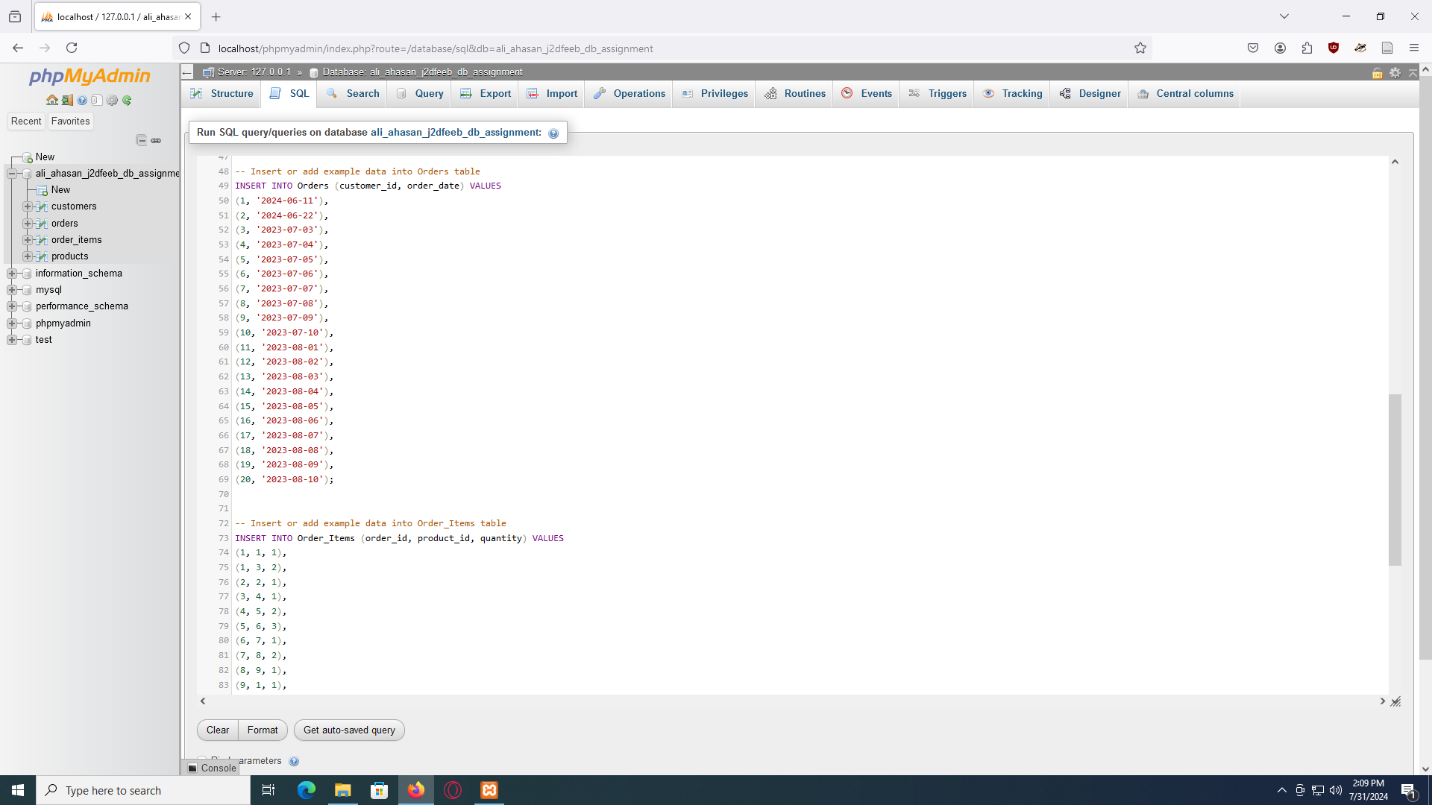
1



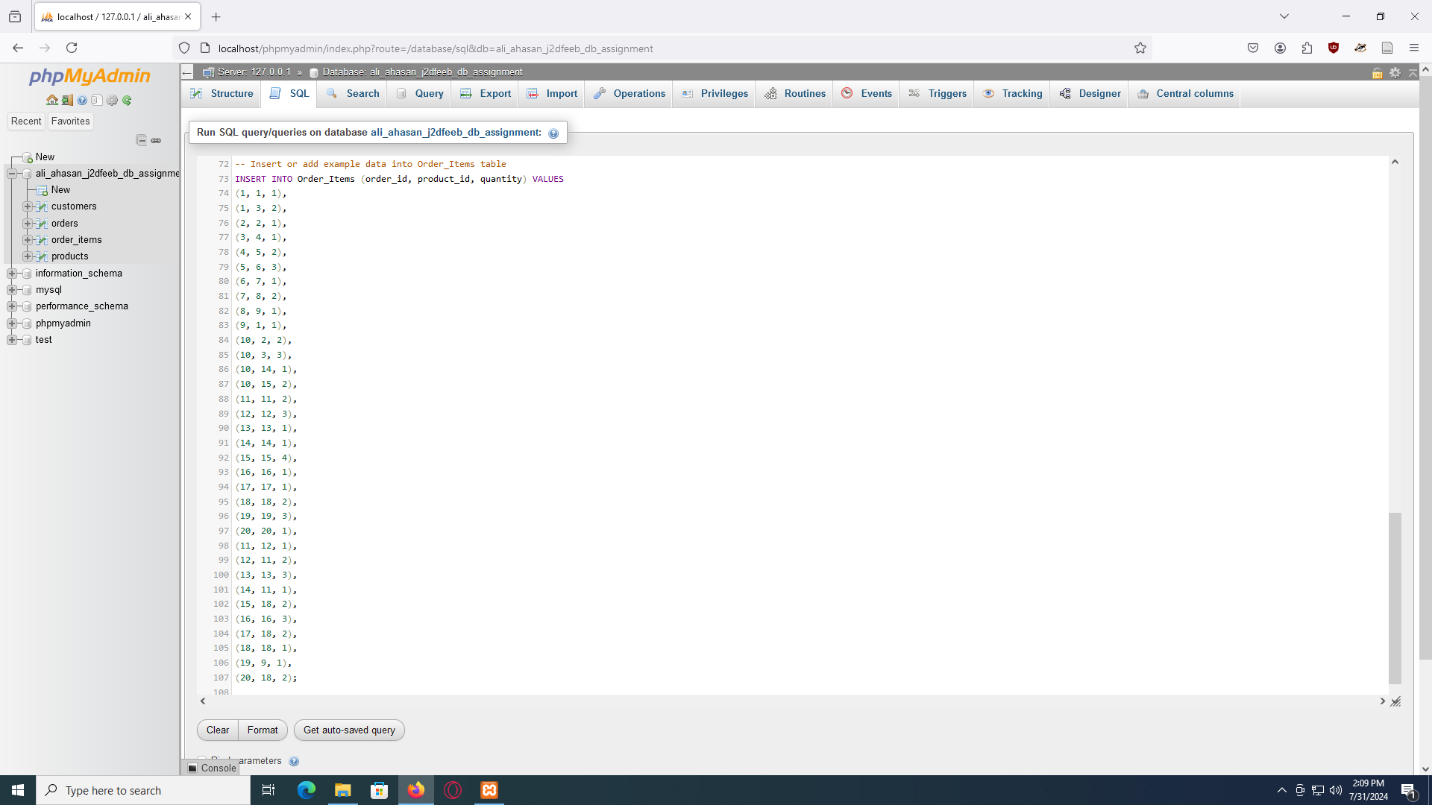
2



3

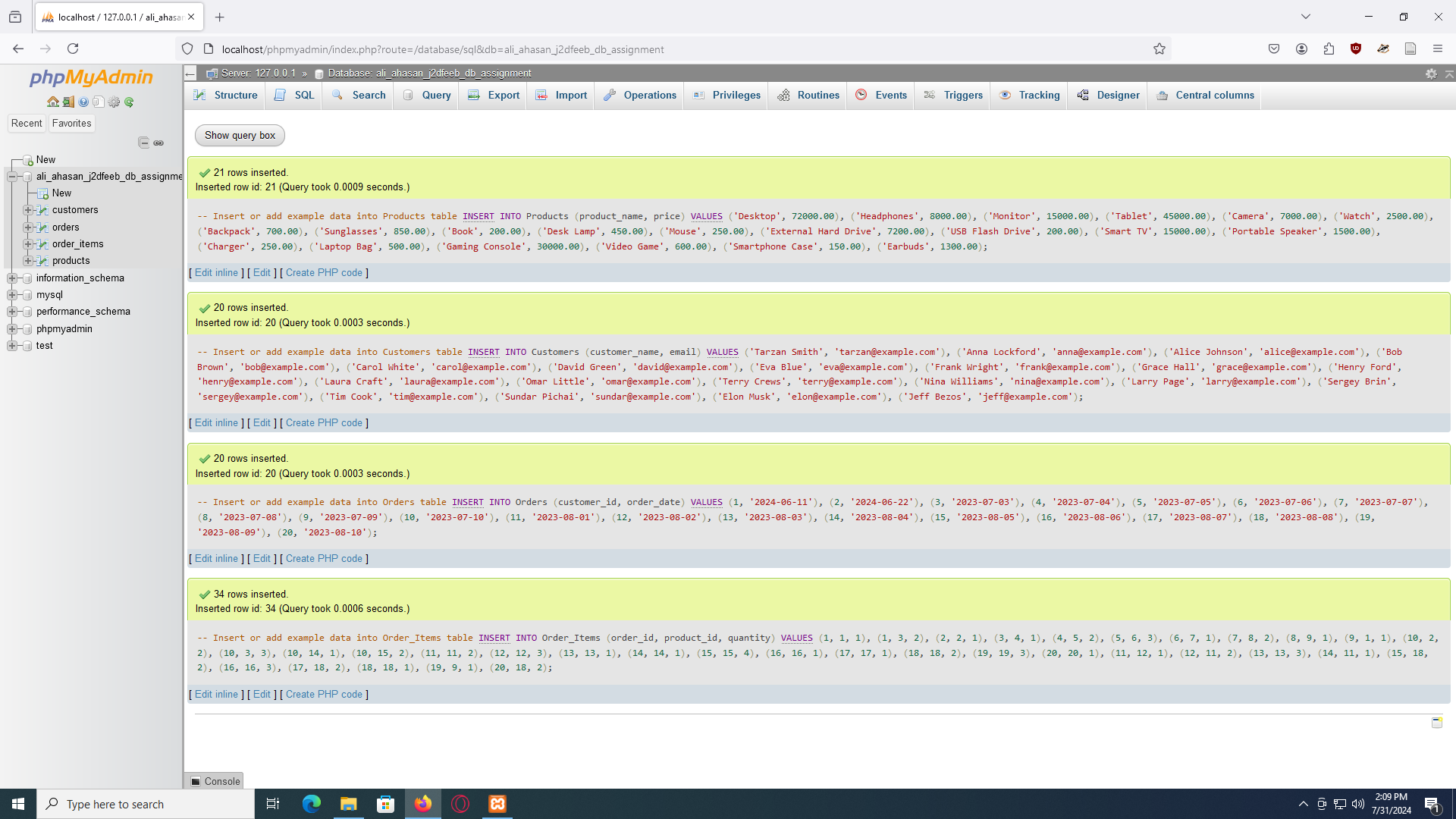


4

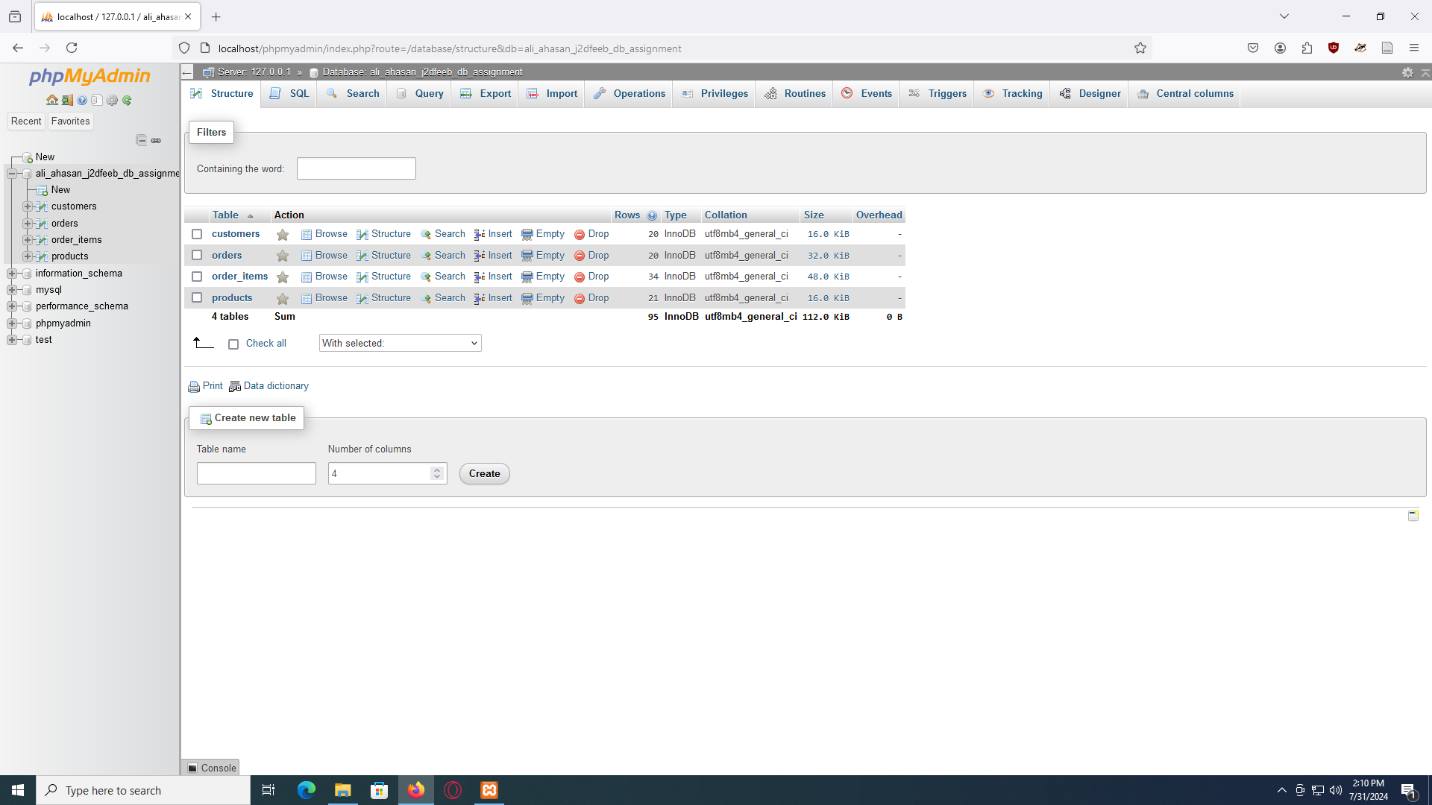


## Output:

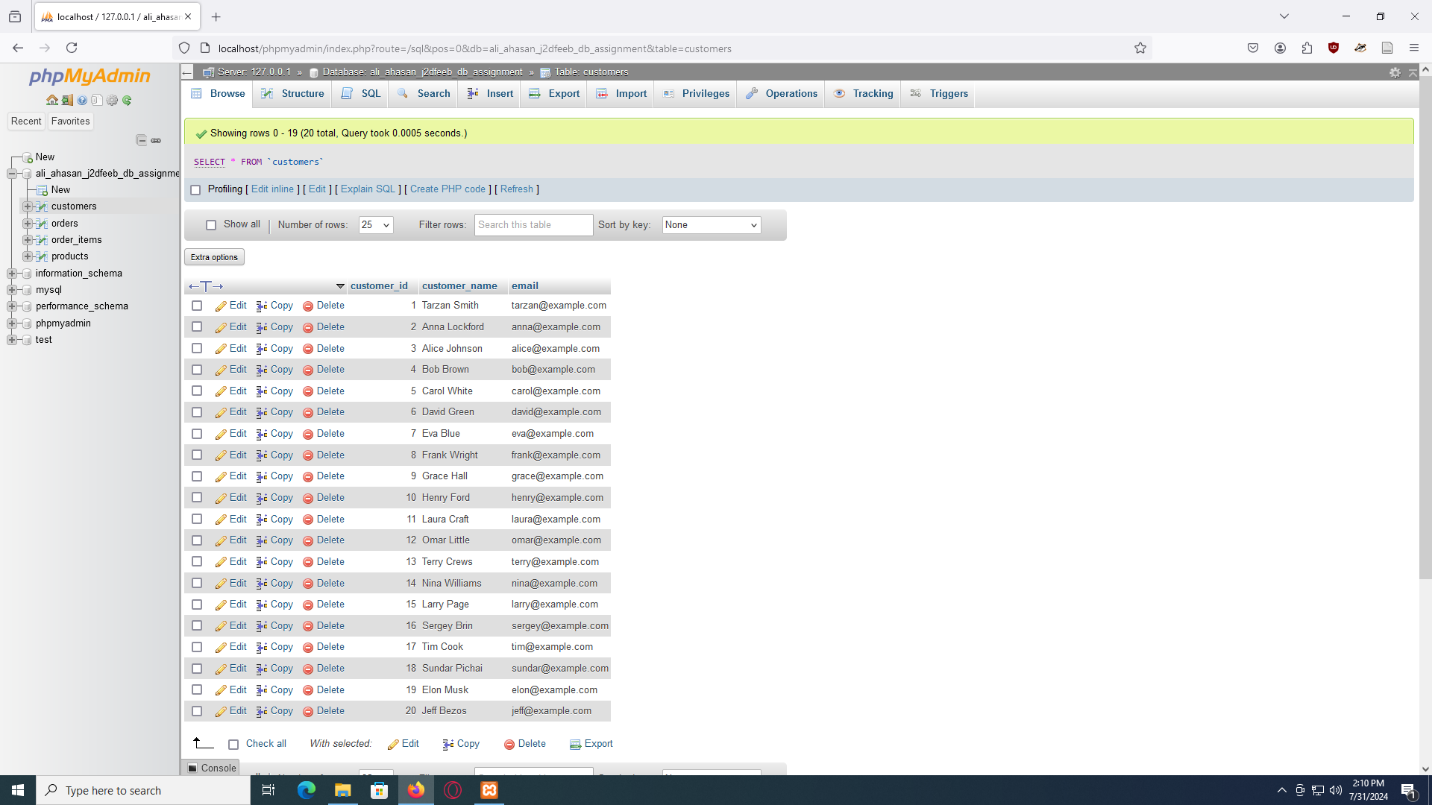
1



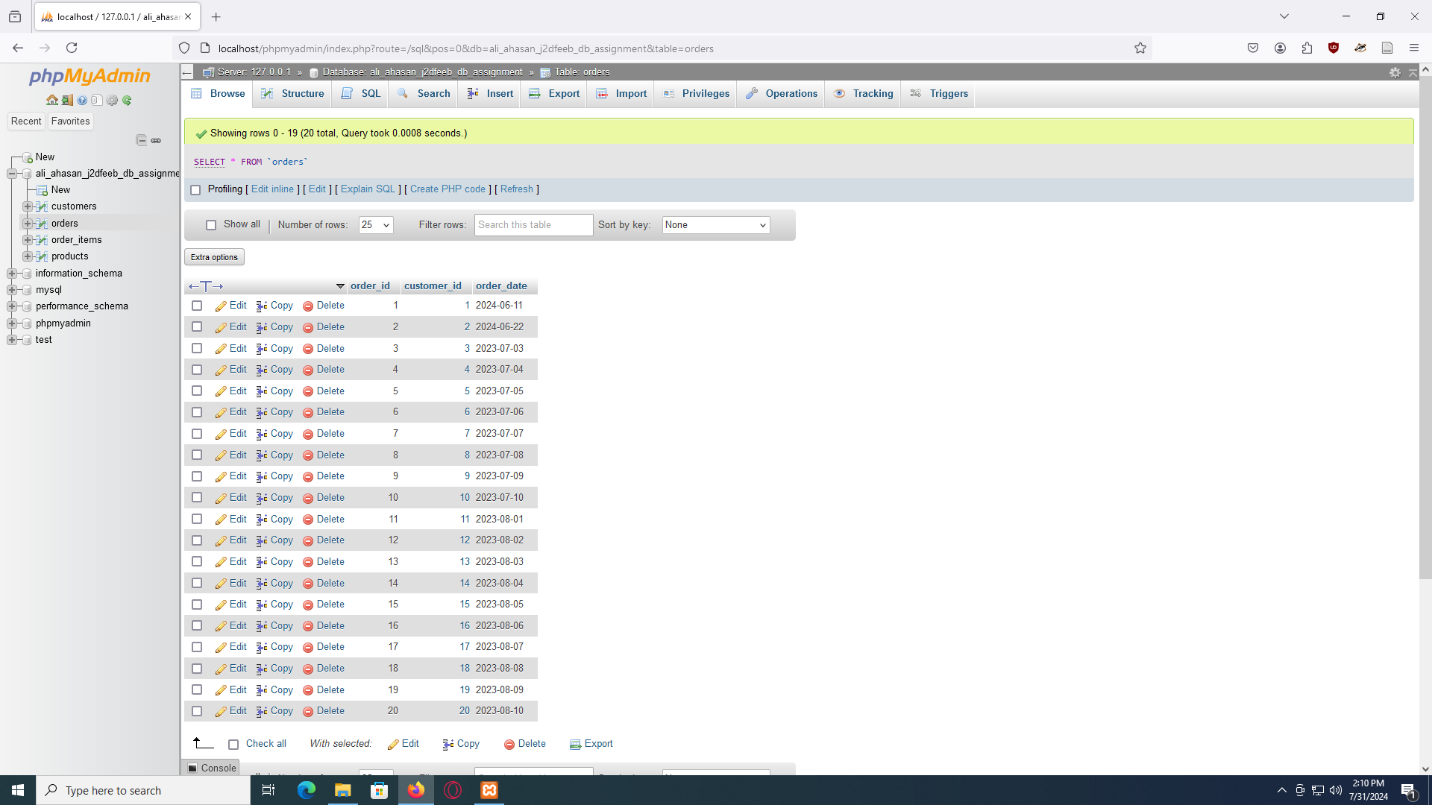
2



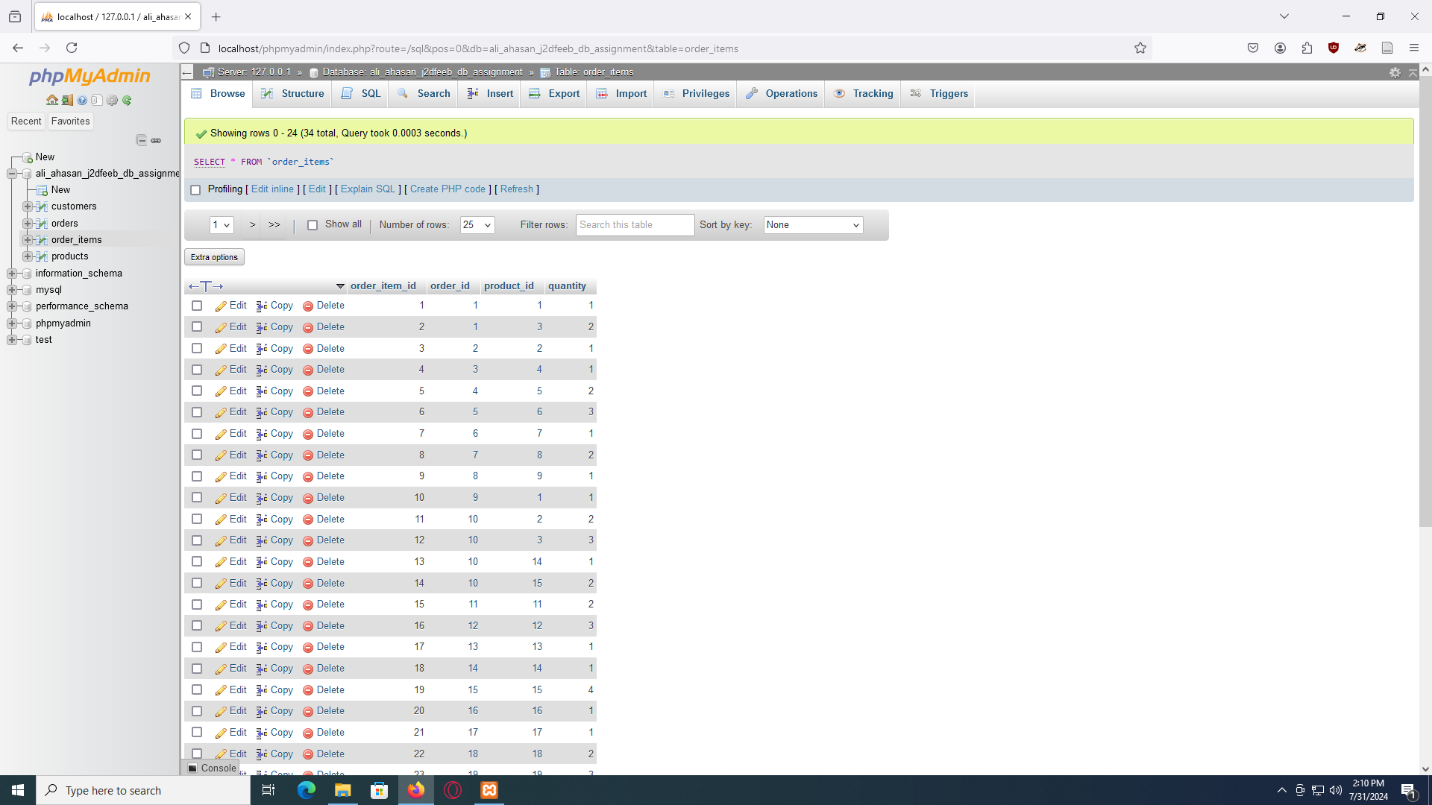
3



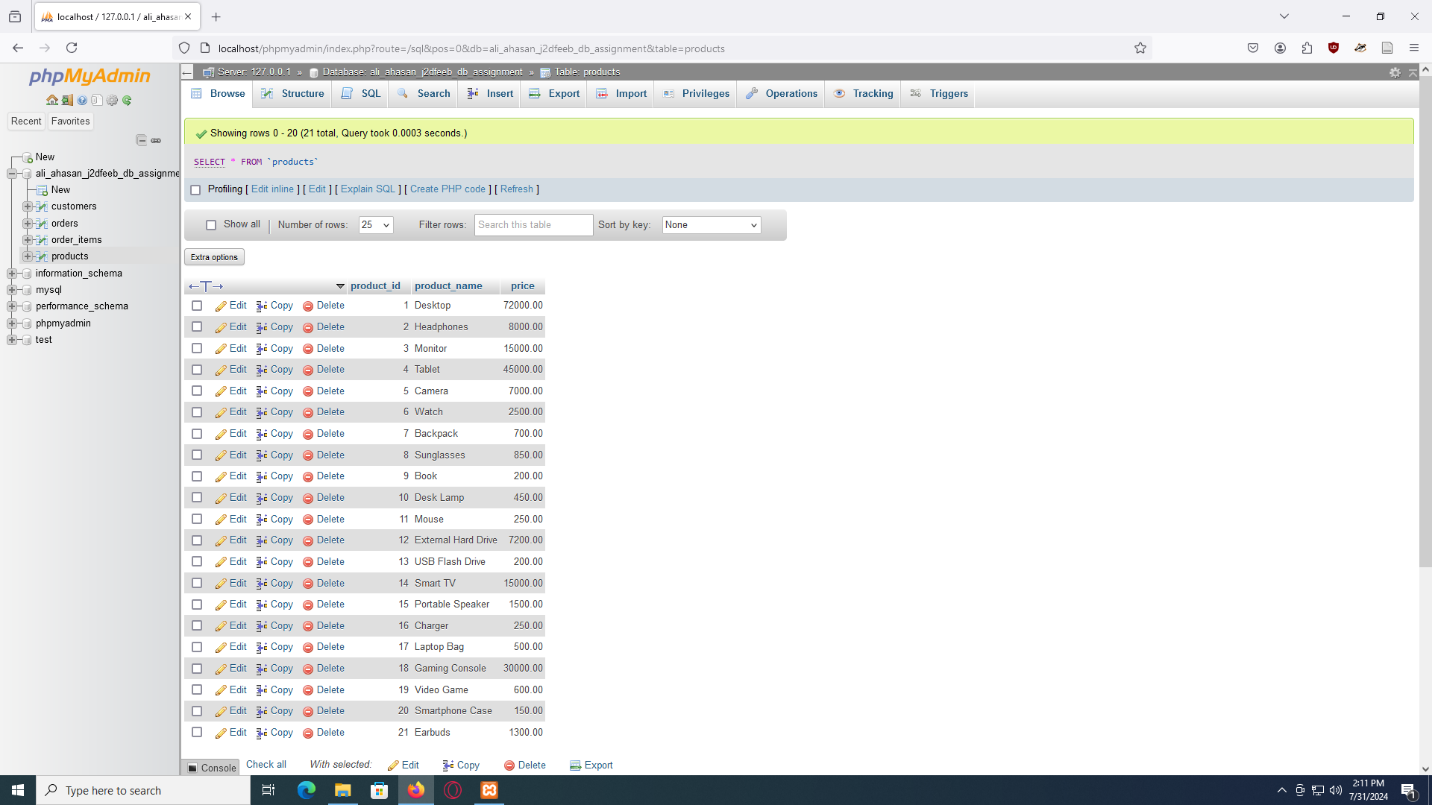
4



5



6



# Table Data Query:

With the help of query commands, Data query can be done.

## Query code:

-- Query to retrieve customer information based on a specific order

-- (SQL Query) --

SELECT \*

FROM Customers

WHERE customer\_id IN

(SELECT customer\_id

FROM Orders

WHERE order\_id = 2);

-- Query to obtain the total number of orders placed by each of customer

-- (SQL Query)--

SELECT order\_id AS customer\_id, SUM(quantity) AS total\_items\_ordered

FROM Order\_Items

GROUP BY order\_id;

-- Query to aquire the list of top 10 (Ten) customers who have ordered the most of the times

-- (SQL Query)--

SELECT order\_id AS customer\_id, COUNT(order\_id) AS orders\_count

FROM Order\_Items

GROUP BY customer\_id

ORDER BY orders\_count DESC LIMIT 10;

-- Query to aquire the list of products which have been ordered the most number of times

-- (SQL Query)--

SELECT p.product\_id, p.product\_name, COUNT(oi.order\_item\_id) AS times\_ordered

FROM Products p

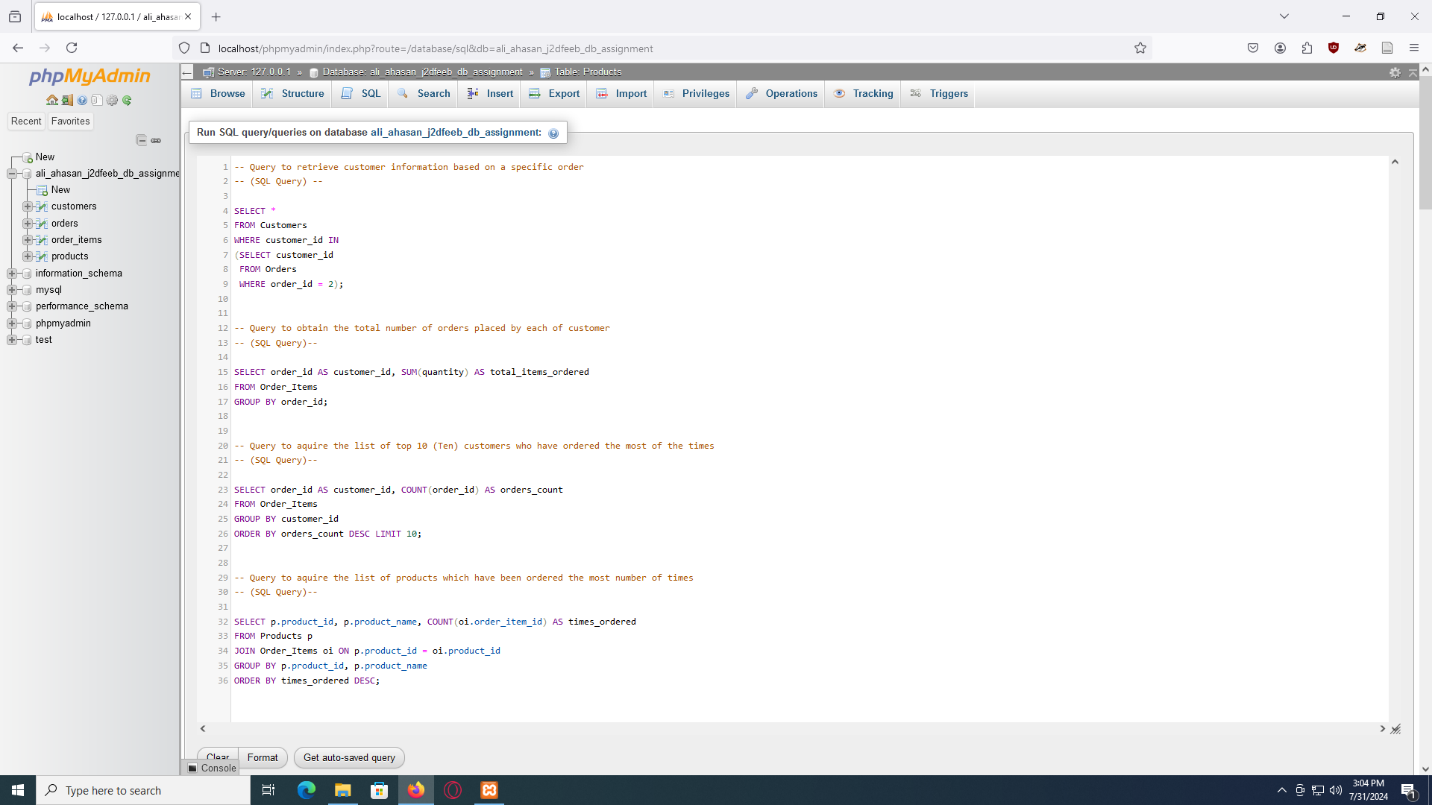
JOIN Order\_Items oi ON p.product\_id = oi.product\_id

GROUP BY p.product\_id, p.product\_name

ORDER BY times\_ordered DESC;

## Input:

1



## Output:

1

-- Query to retrieve customer information based on a specific order

-- (SQL Query) --

SELECT \*

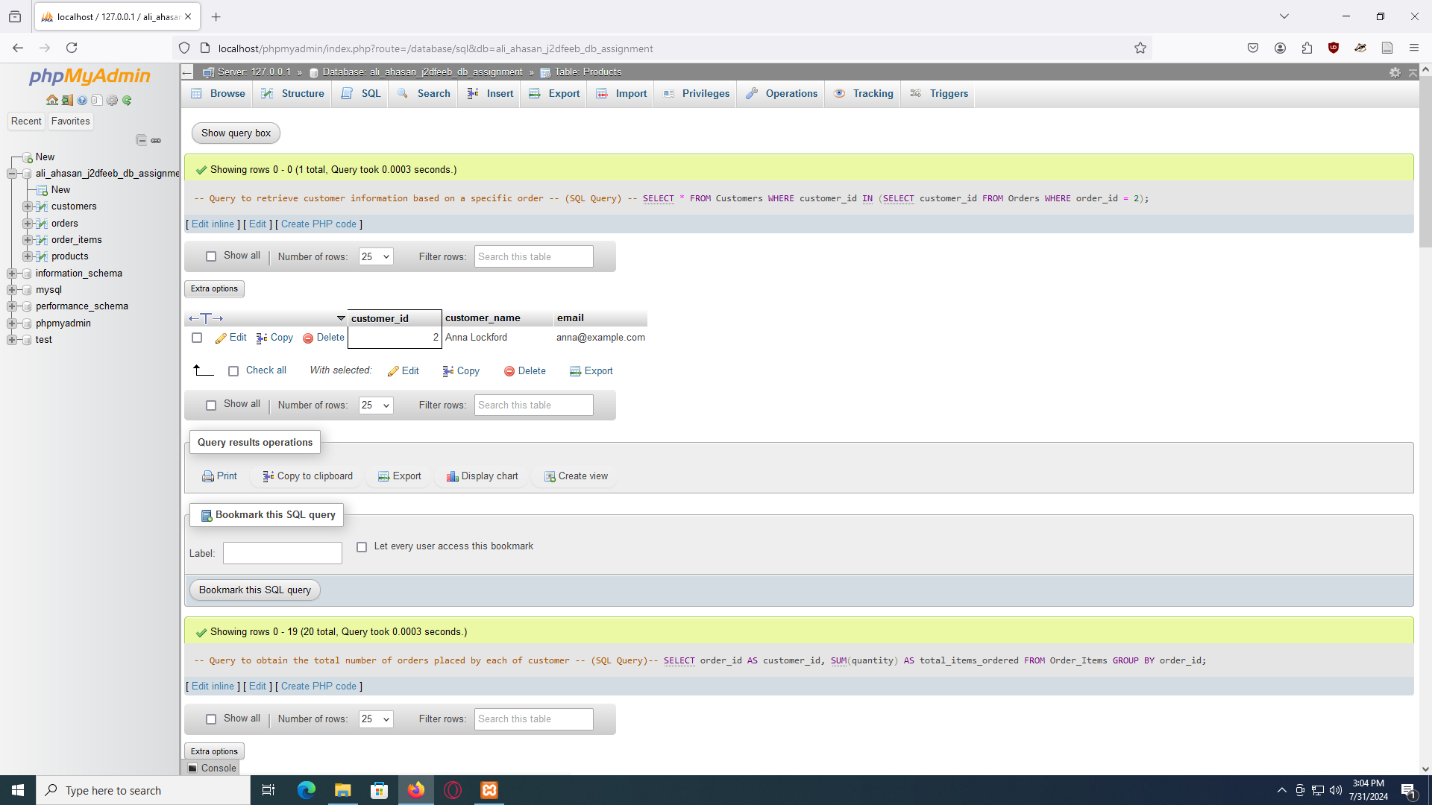
FROM Customers

WHERE customer\_id IN

(SELECT customer\_id

FROM Orders

WHERE order\_id = 2);



2

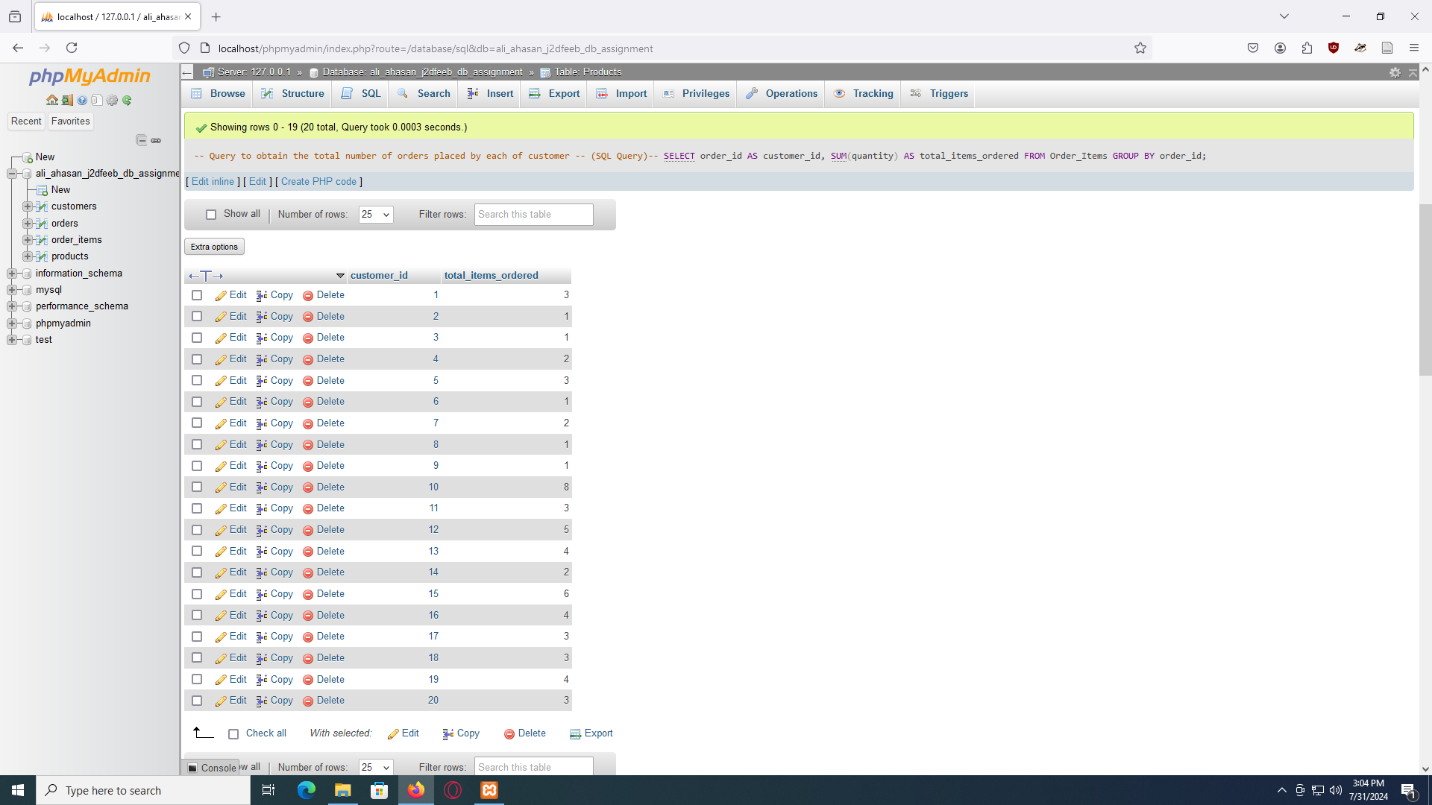
-- Query to obtain the total number of orders placed by each of customer

-- (SQL Query)--

SELECT order\_id AS customer\_id, SUM(quantity) AS total\_items\_ordered

FROM Order\_Items

GROUP BY order\_id;



3

-- Query to aquire the list of top 10 (Ten) customers who have ordered the most of the times

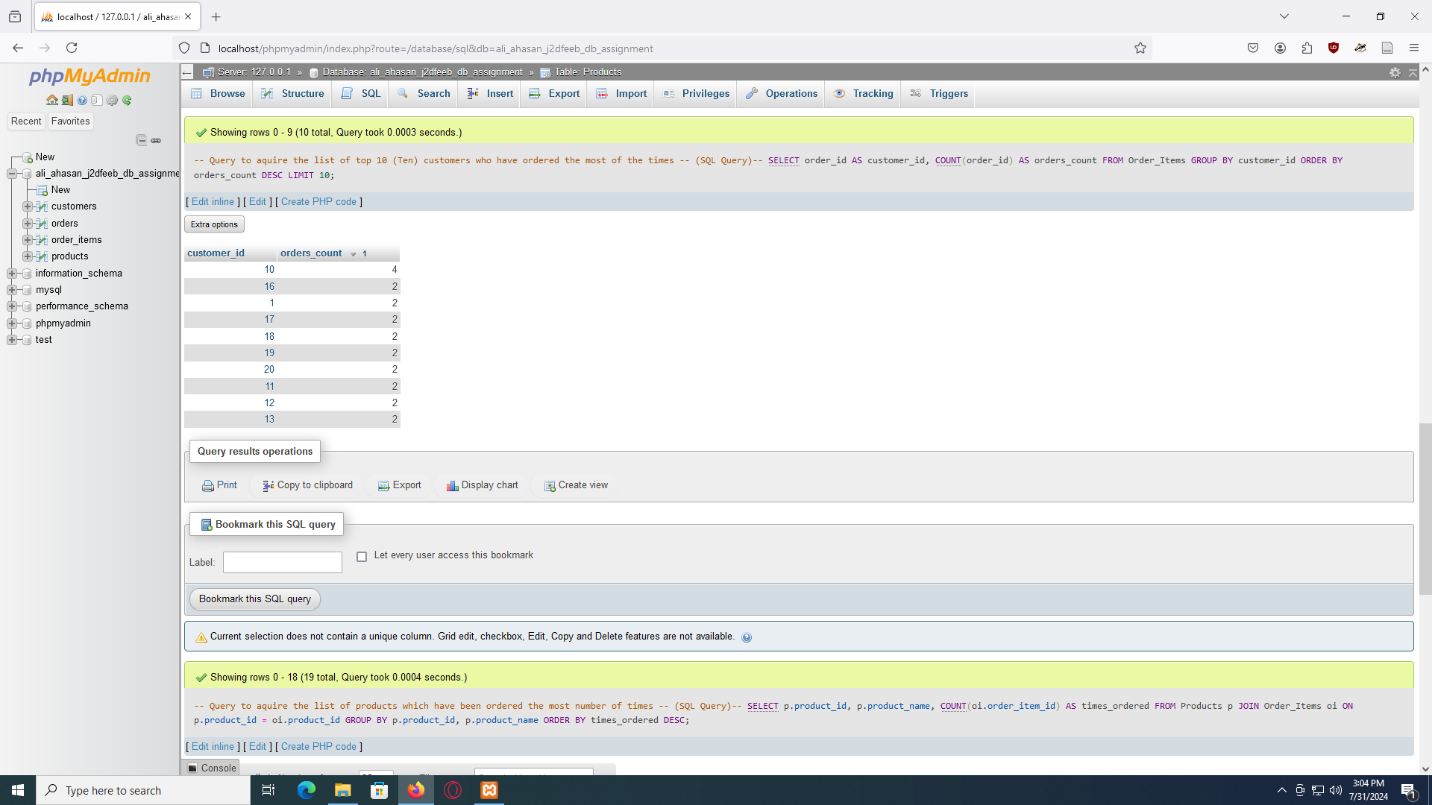
-- (SQL Query)--

SELECT order\_id AS customer\_id, COUNT(order\_id) AS orders\_count

FROM Order\_Items

GROUP BY customer\_id

ORDER BY orders\_count DESC LIMIT 10;



4

-- Query to aquire the list of products which have been ordered the most number of times

-- (SQL Query)--

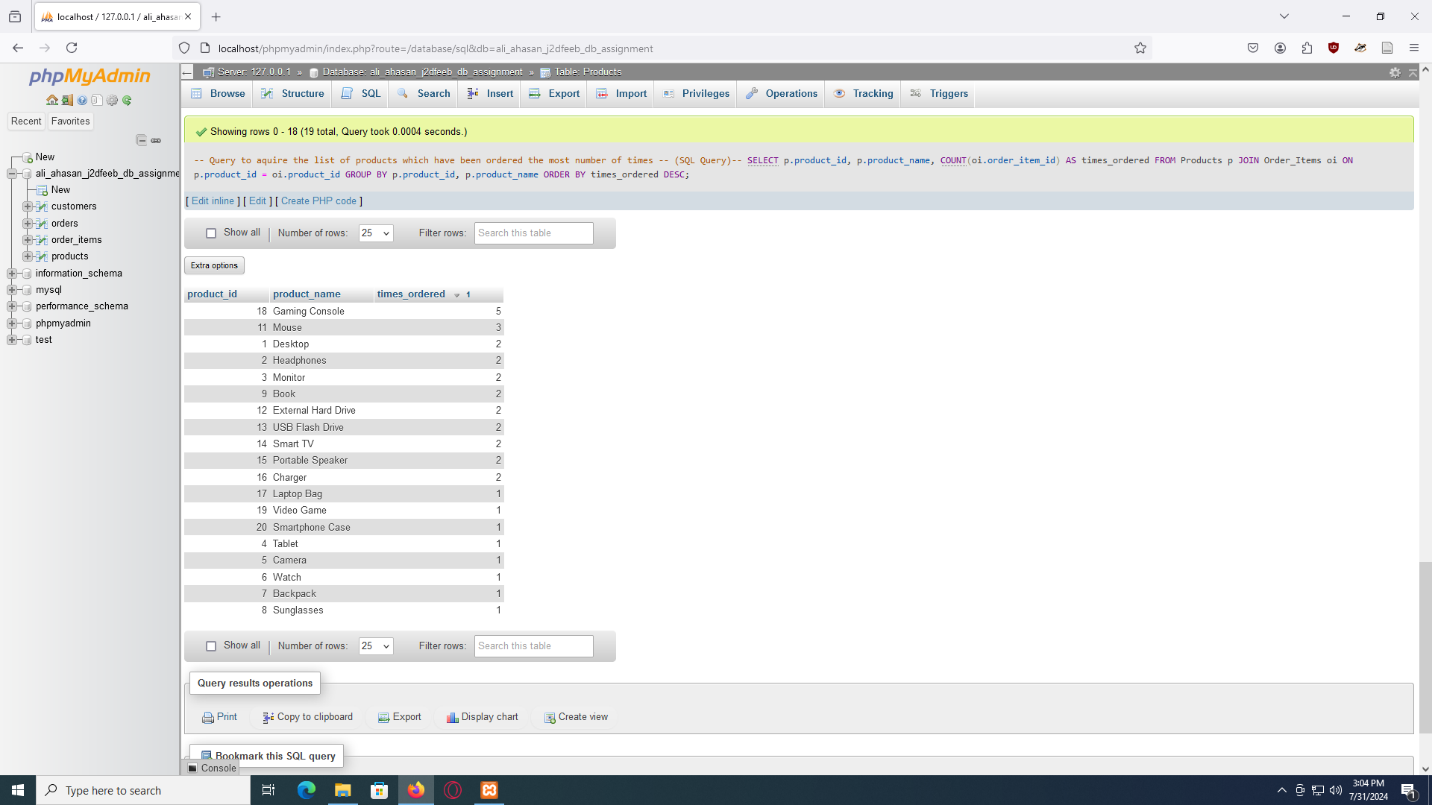
SELECT p.product\_id, p.product\_name, COUNT(oi.order\_item\_id) AS times\_ordered

FROM Products p

JOIN Order\_Items oi ON p.product\_id = oi.product\_id

GROUP BY p.product\_id, p.product\_name

ORDER BY times\_ordered DESC;



----