Instructions:

- This assignment is designed to test your knowledge of MySQL database concepts in the context of a stock management system for an e-commerce platform.
- Write SQL queries for each task and provide explanations where necessary.
- Submit your assignment as a single SQL script file.

Task 1: Create a Database Create a new MySQL database named `ecommerce_stock_db`.

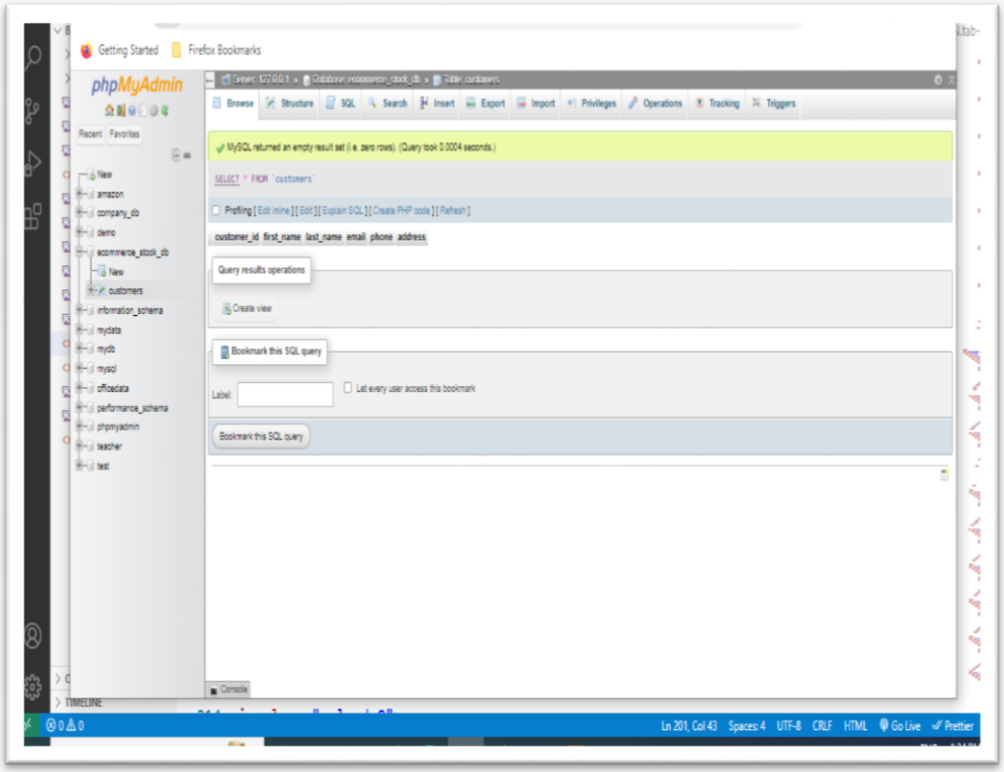
QUERY- CREATE DATABASE `ecommerce_stock_db ;`

Task 2: Create Tables Inside the `ecommerce_stock_db`, create the following tables:

1. **`customers` table with the following columns:**
   - **`customer_id` (Primary Key, Auto Increment)**
   - **`first_name`**
   - **`last_name`**
   - **`email`**
   - **`Phone`**
   - **`address`**

QUERY-2

```
CREATE TABLE customers(
customer_id int NOT NULL AUTO_INCREMENT,
first_name varchar(50) null,
last_name varchar(50) null,
email  varchar(50) null,
phone int null,
address text null,
PRIMARY KEY(customer_id))
```

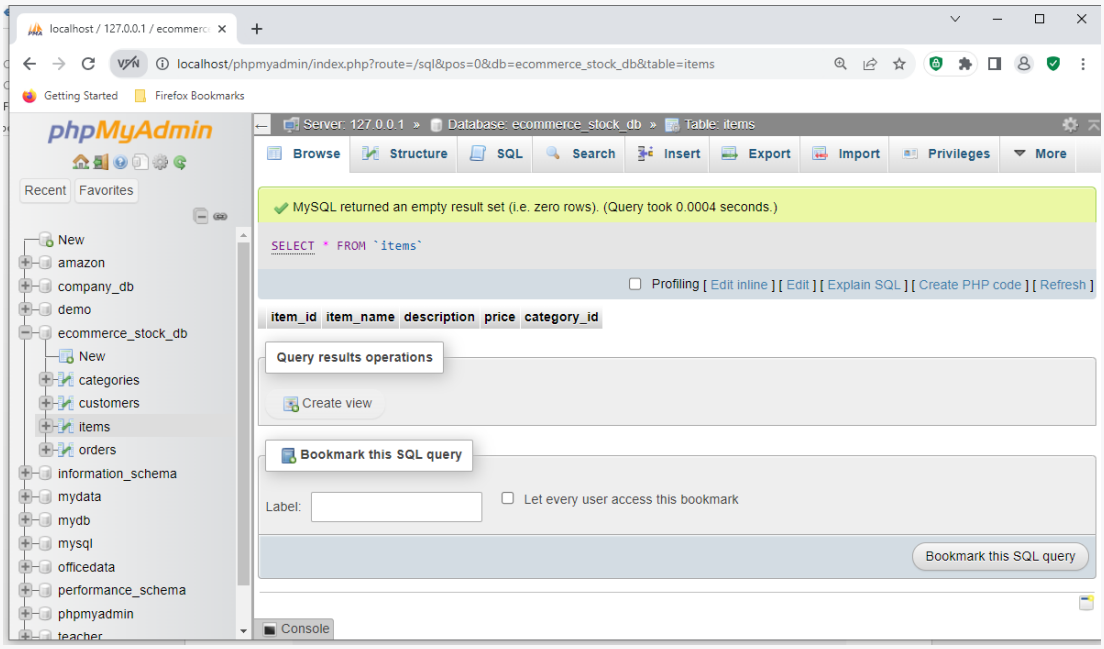


2- `items` table with the following columns:

- `item_id` (Primary Key, Auto Increment)
- `item_name`
- `description`
- `price`
- `stock_quantity`
- `category_id` (Foreign Key referencing `category_id` in the `categories` table)

QUERY-

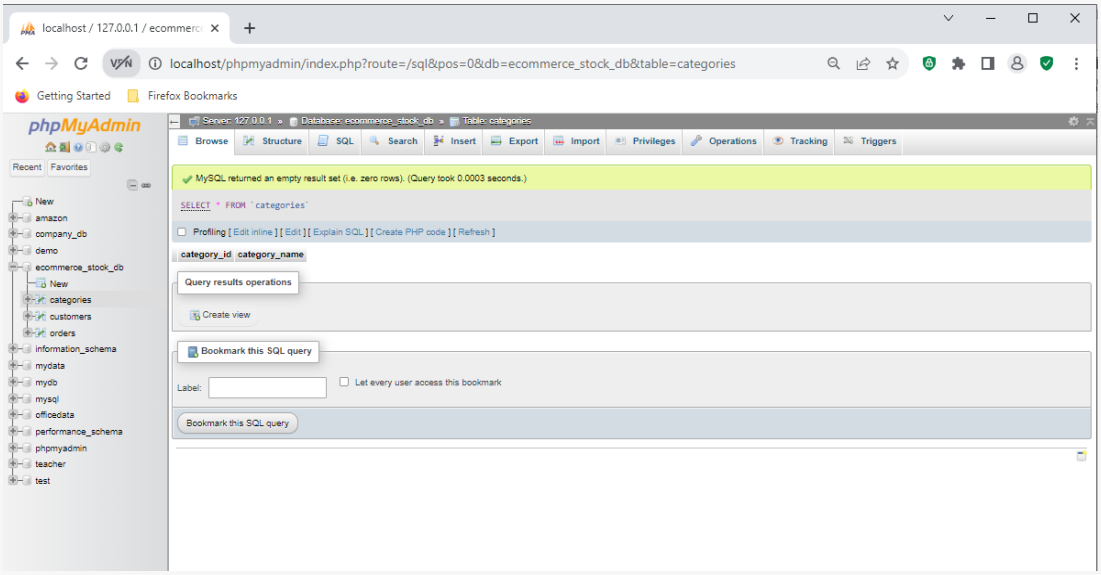
```
CREATE TABLE items(
item_id int not null AUTO_INCREMENT,
item_name varchar(200) not null,
description  text,
price float not null,
category_id int not null,
PRIMARY KEY (item_id),
FOREIGN KEY(category_id) REFERENCES categories(category_id))
```



3- `categories` table with the following columns:

- **`category_id` (Primary Key, Auto Increment)**
- **`category_name`**

QUERY- CREATE TABLE categories(
category_id int not null AUTO_INCREMENT,
category_name varchar(200) not null,
PRIMARY KEY(category_id))

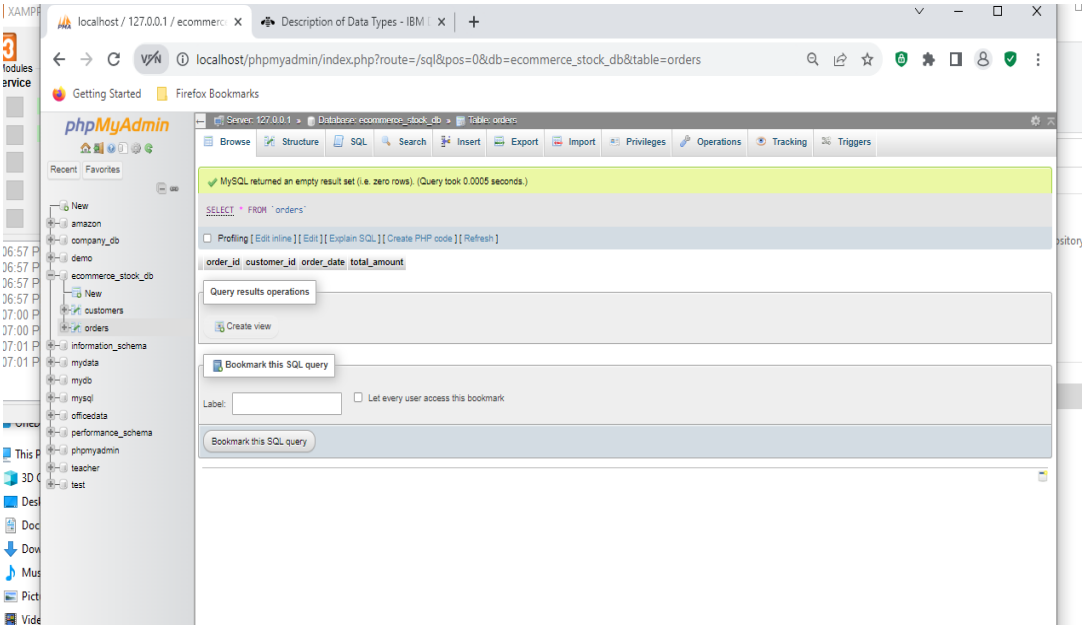


4-`orders` table with the following columns:

- **`order_id` (Primary Key, Auto Increment)**
- **`customer_id` (Foreign Key referencing `customer_id` in the `customers` table)**
- **`order_date`**
- **`total_amount`**

```
QUERY- CREATE TABLE orders(

    order_id int not null AUTO_INCREMENT,
```

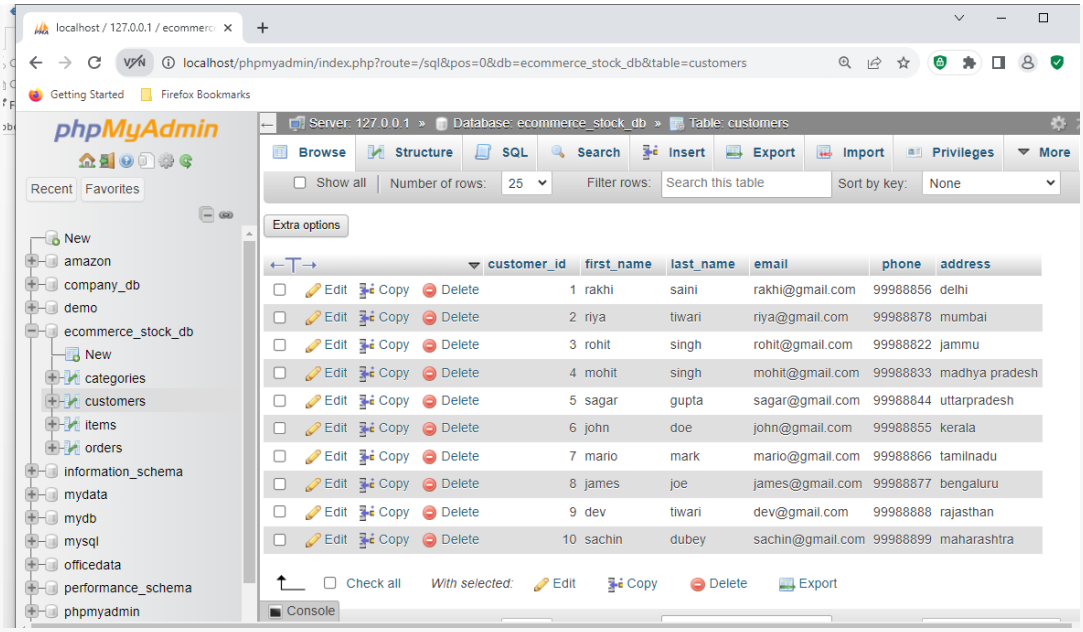
```
customer_id int not null,

order_date DATE not null,

total_amount float  not null,

PRIMARY KEY(order_id),

FOREIGN KEY(customer_id)REFERENCES customers(customer_id));
```



Task 3: Insert Data Insert at least 10 records into the `customers` table, 20 records into the `items` table, 5 records into the `categories` table, and 30 records into the `orders` table. Ensure that items are assigned to categories and orders are associated with customers and items.

QUERY- **Inserted 10 records into the customer table**

```
INSERT INTO `customers`( `first_name`, `last_name`, `email`, `phone`, `address`)

VALUES

('rakhi','saini','rakhi@gmail.com','099988856','delhi'),

('riya','tiwari','riya@gmail.com','099988878','mumbai'),

('rohit','singh','rohit@gmail.com','099988822','jammu'),

('mohit','singh','mohit@gmail.com','099988833','madhya pradesh'),

('sagar','gupta','sagar@gmail.com','099988844','uttarpradesh'),

('john','doe','john@gmail.com','099988855','kerala'),

('mario','mark','mario@gmail.com','099988866','tamilnadu'),

('james','joe','james@gmail.com','099988877','bengaluru'),

('dev','tiwari','dev@gmail.com','099988888','rajasthan'),

('sachin','dubey','sachin@gmail.com','099988899','maharashtra');
```



2. **Inserted 5 records into the categories table**

Query- INSERT INTO `categories`(`category_name`)

VALUES

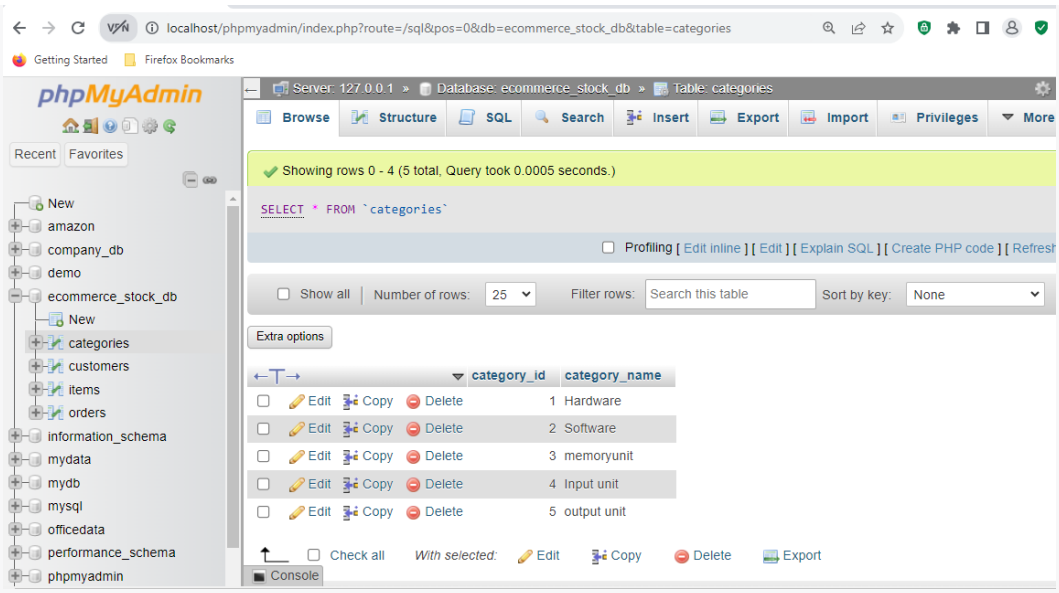
('Hardware'),

('Software'),

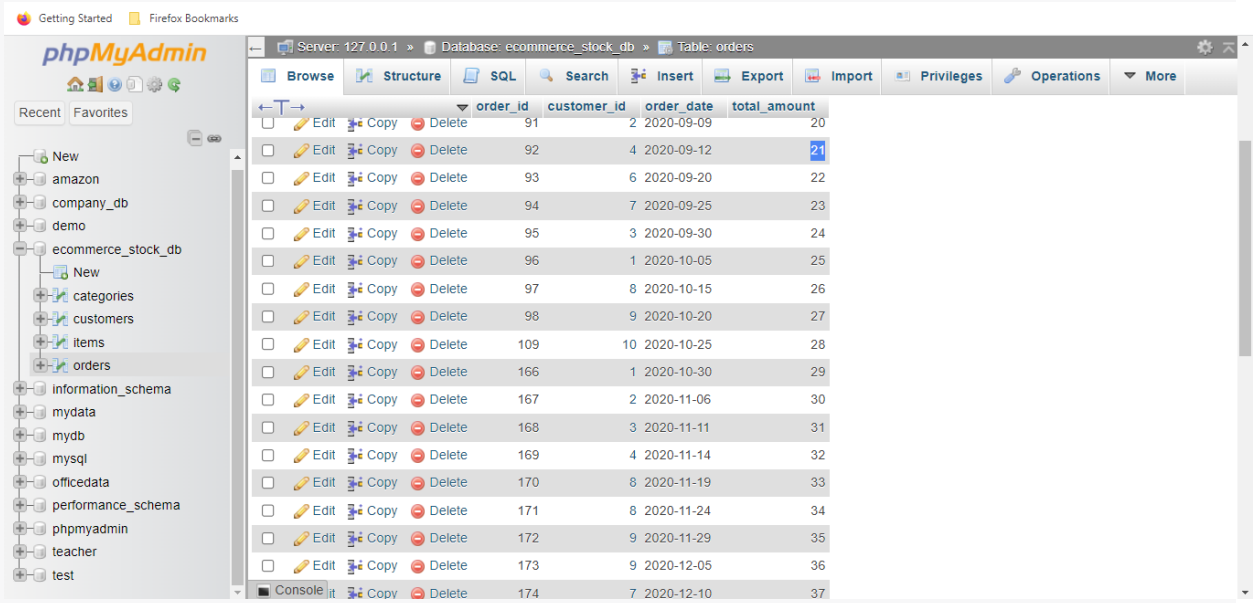
('memoryunit'),

('Input unit'),

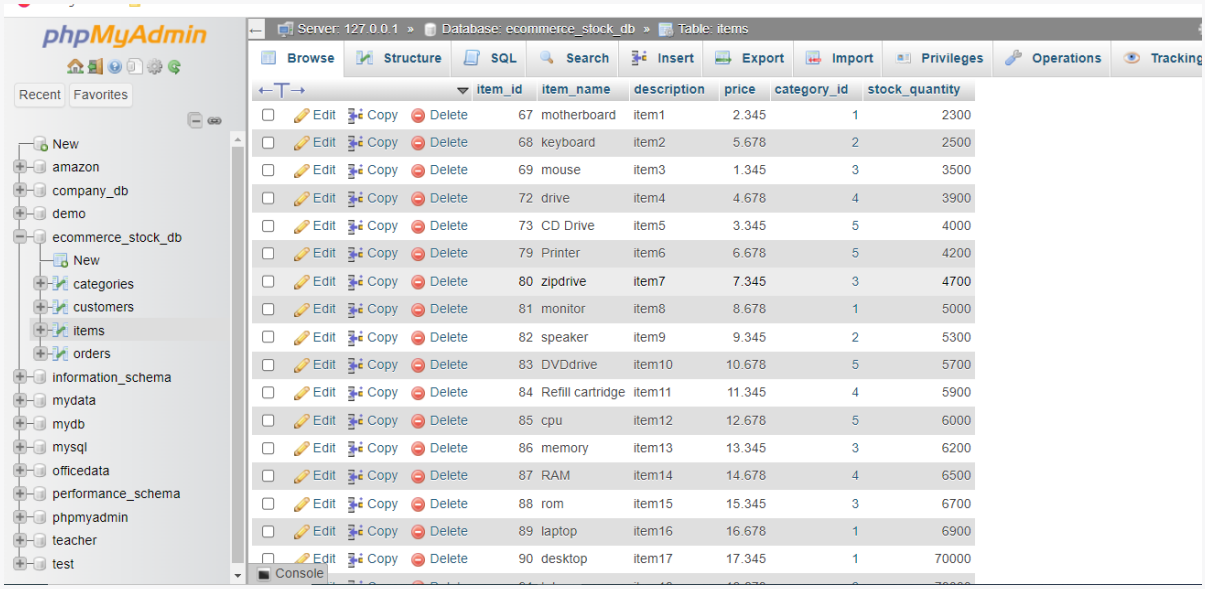
('output unit')



Inserted 30 records into orders table-



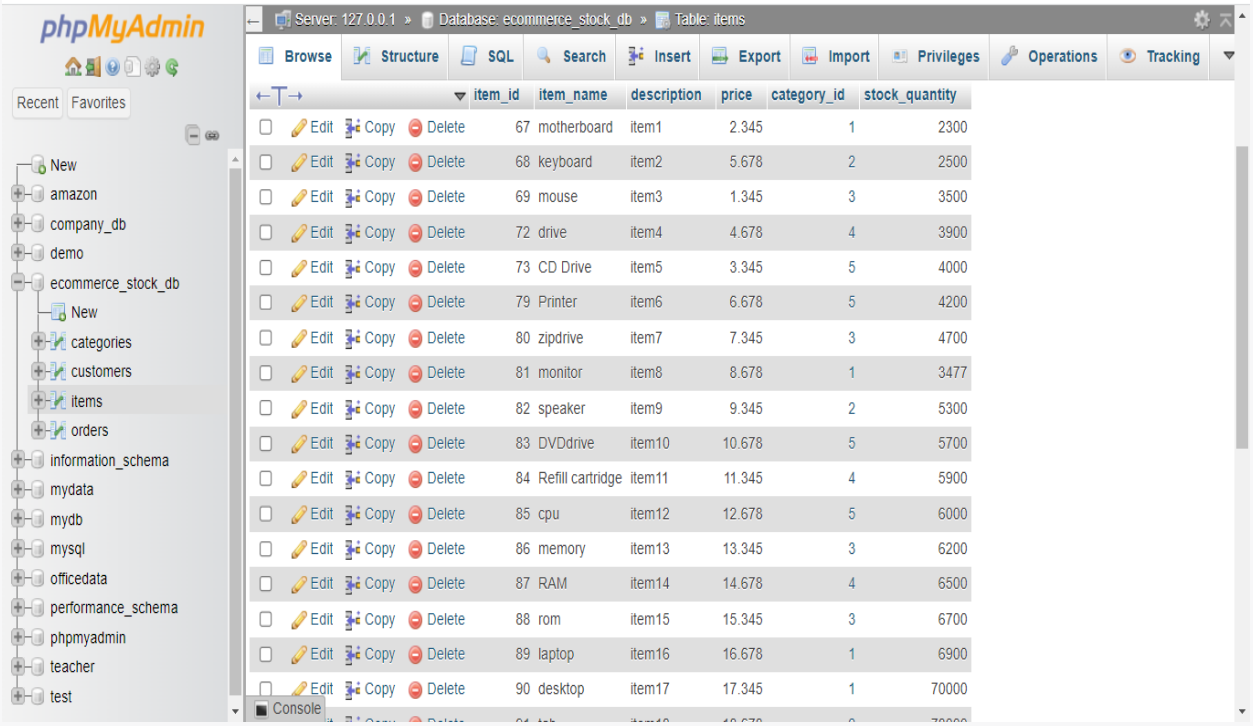
Inserted 20 records into item table-



	item_id	item_name	description	price	category_id	stock_quantity
Edit Copy Delete	67	motherboard	item1	2.345	1	2300
Edit Copy Delete	68	keyboard	item2	5.678	2	2500
Edit Copy Delete	69	mouse	item3	1.345	3	3500
Edit Copy Delete	72	drive	item4	4.678	4	3900
Edit Copy Delete	73	CD Drive	item5	3.345	5	4000
Edit Copy Delete	79	Printer	item6	6.678	5	4200
Edit Copy Delete	80	zipdrive	item7	7.345	3	4700
Edit Copy Delete	81	monitor	item8	8.678	1	5000
Edit Copy Delete	82	speaker	item9	9.345	2	5300
Edit Copy Delete	83	DVDdrive	item10	10.678	5	5700
Edit Copy Delete	84	Refill cartridge	item11	11.345	4	5900
Edit Copy Delete	85	cpu	item12	12.678	5	6000
Edit Copy Delete	86	memory	item13	13.345	3	6200
Edit Copy Delete	87	RAM	item14	14.678	4	6500
Edit Copy Delete	88	rom	item15	15.345	3	6700
Edit Copy Delete	89	laptop	item16	16.678	1	6900
Edit Copy Delete	90	desktop	item17	17.345	1	70000

Task 4: Update Data Write a SQL query to update the stock quantity of a specific item in the `items` table.

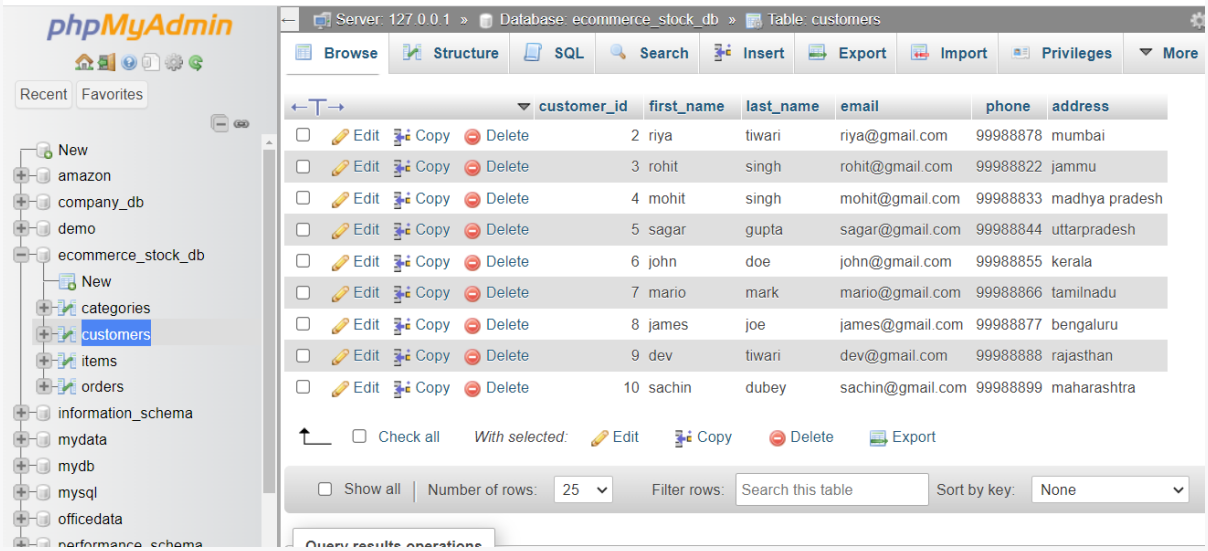
QUERY- UPDATE items set stock_quantity ='3477' WHERE item_id=81;

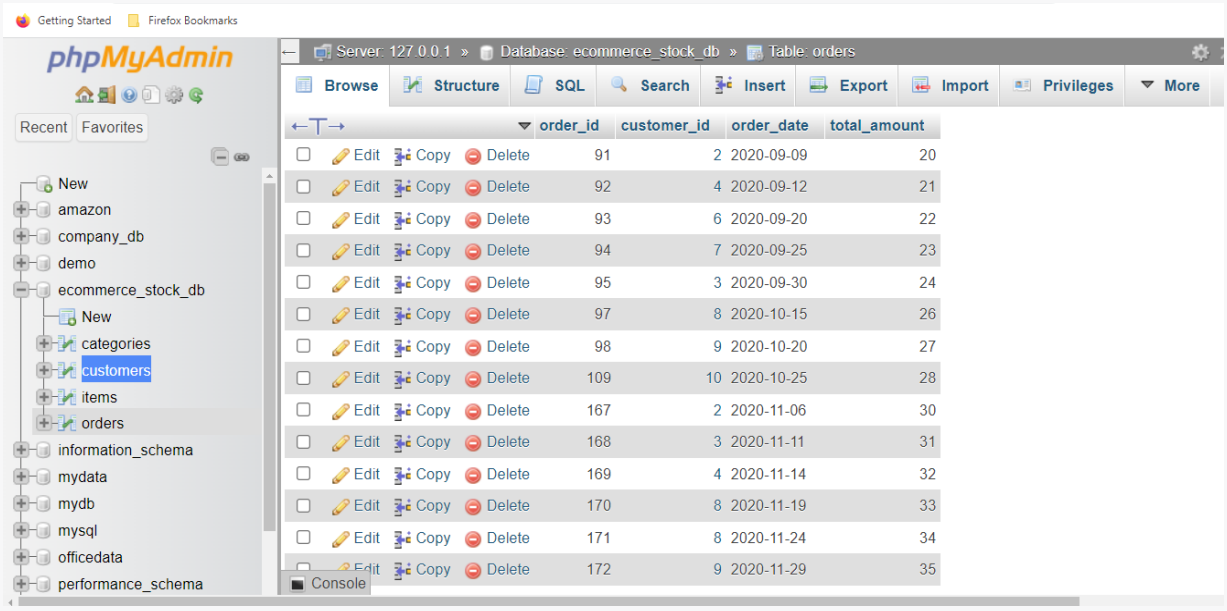


Task 5: Delete Data Write a SQL query to delete a customer from the `customers` table and all related orders (if any) in the database.

QUERY- DELETE FROM `orders` WHERE customer_id=1;

DELETE FROM `customers` WHERE customer_id=1;





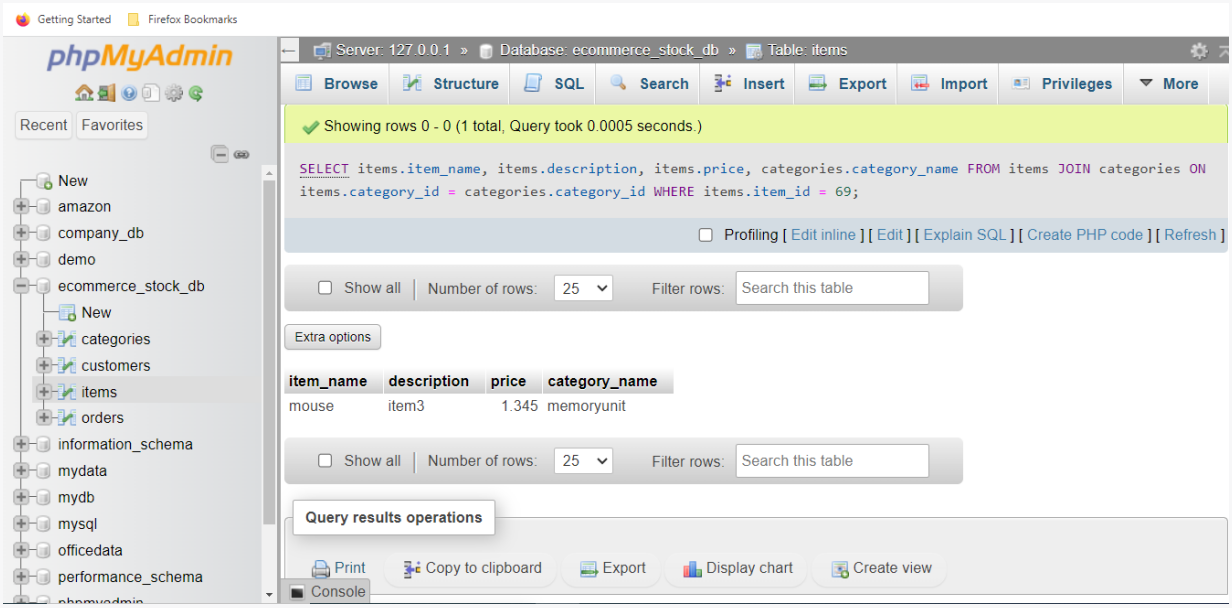
Task 6: Retrieve Item Information Write an SQL query to retrieve information about an item, including its name, description, price, and category name, for a specific item.

SELECT items.item_name, items.description, items.price, categories.category_name

FROM items

JOIN categories ON items.category_id = categories.category_id

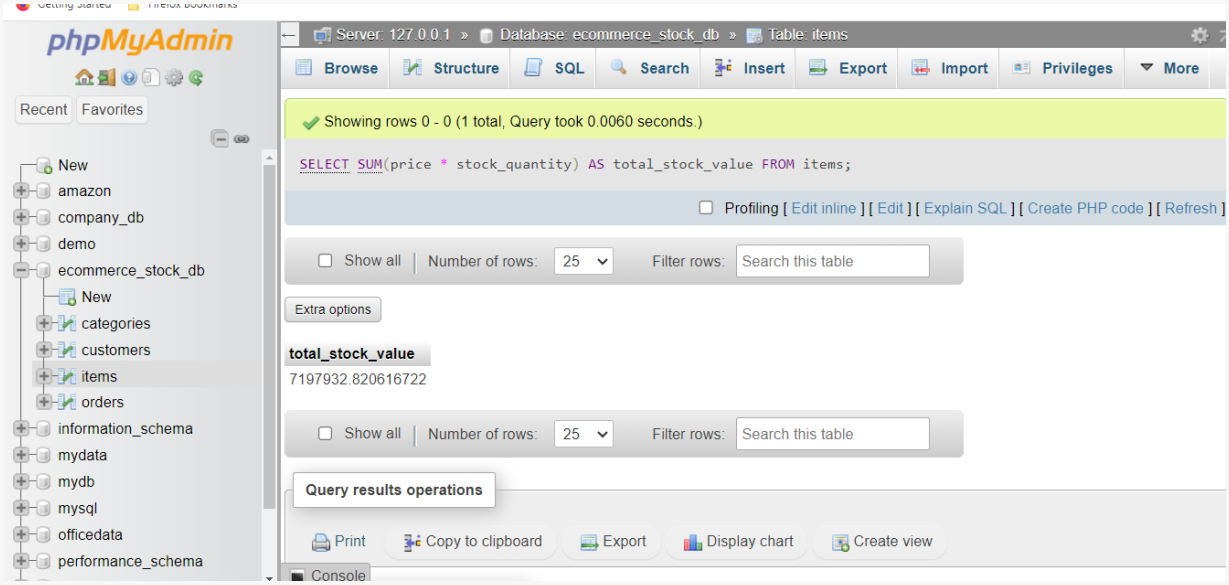
WHERE items.item_id = 69;



Task 7: Calculate Total Stock Value Write an SQL query to calculate the total value of all items in stock, considering the price and stock quantity.

QUERY- SELECT SUM(price * stock_quantity) AS total_stock_value

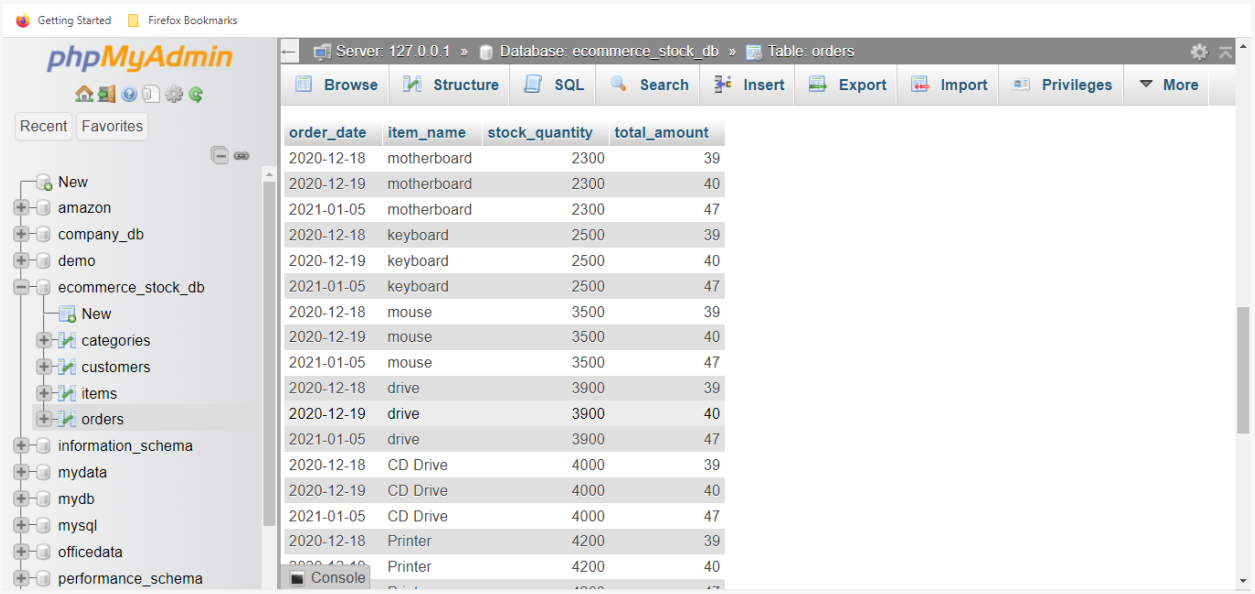
FROM items;



Task 8: Retrieve Customer Order History Write an SQL query to retrieve the order history for a specific customer, including order dates, item names, quantities, and total amounts spent.

QUERY-

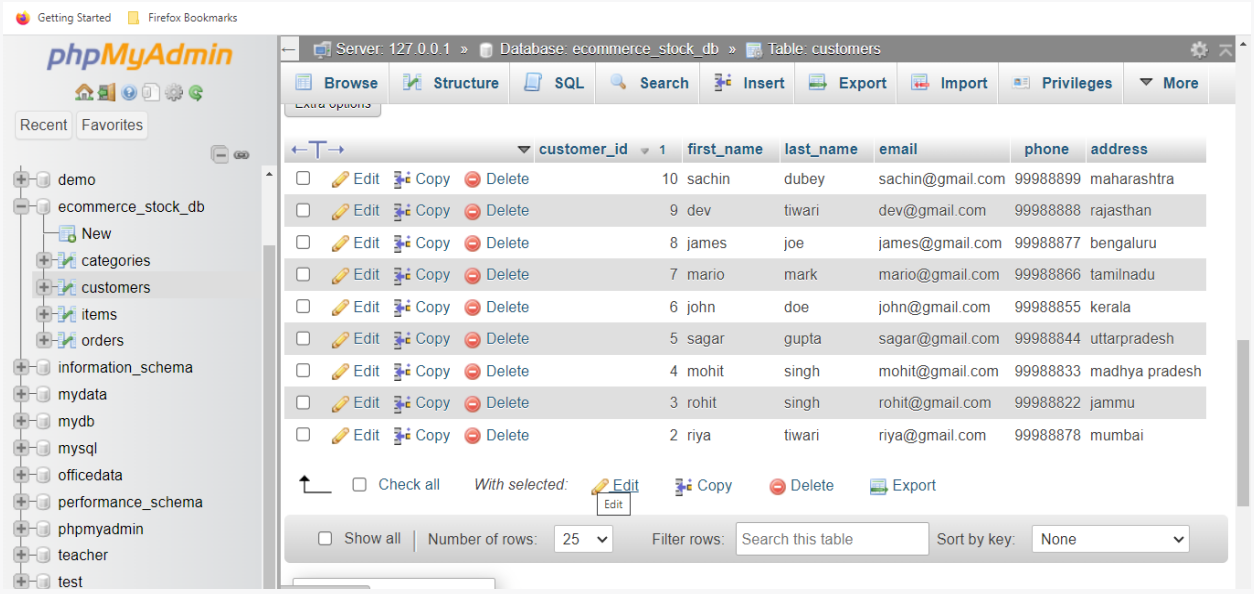
```
SELECT orders.order_date, items.item_name, items.stock_quantity, orders.total_amount

FROM orders

JOIN items ON items.item_id = items.item_id

WHERE orders.customer_id = 5;
```



Task 9: Implement ORDER BY DESC Write an SQL query to retrieve a list of all tables, ordering them in descending order by the `primary_key or id` column.

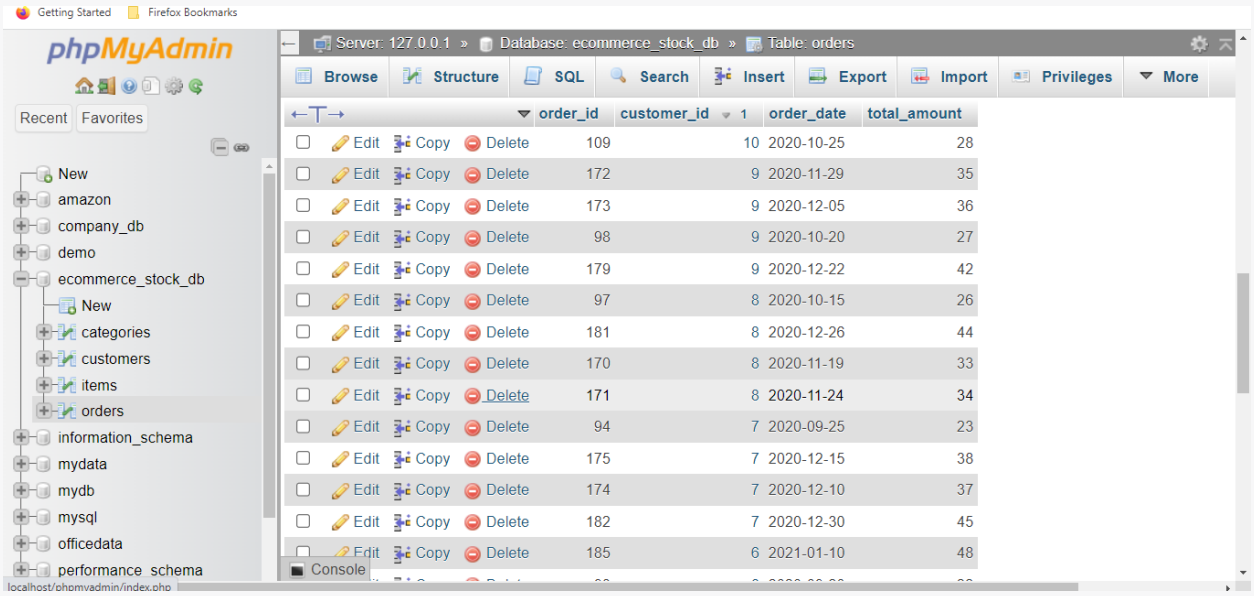
QUERY- SELECT * FROM Customers

ORDER BY customer_id DESC;



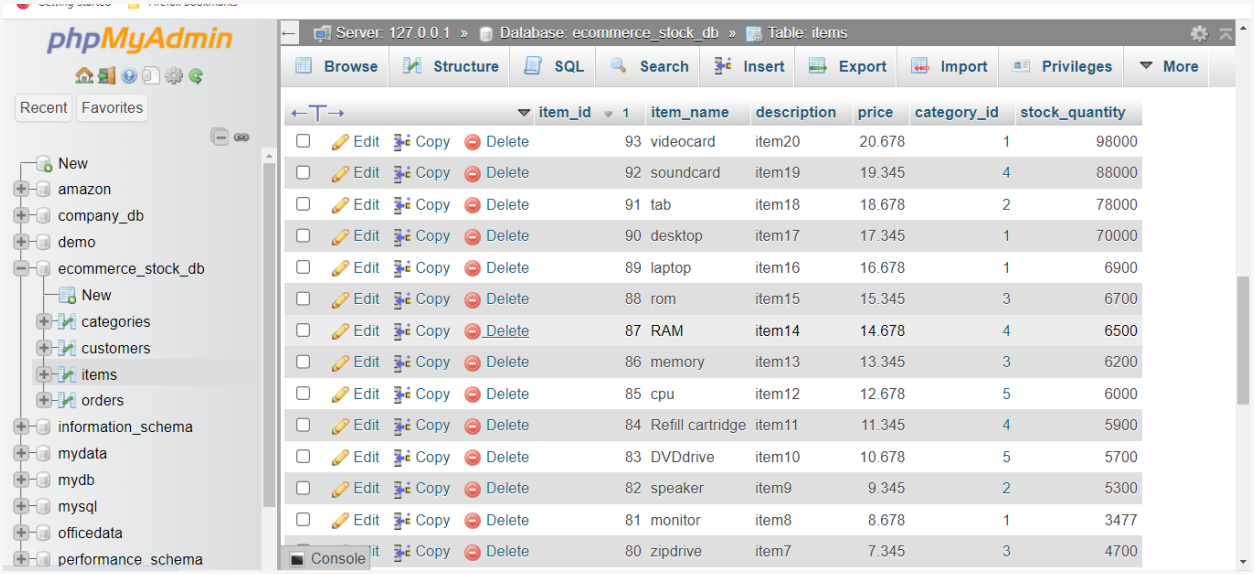
Query- SELECT * FROM orders

ORDER BY customer_id DESC;



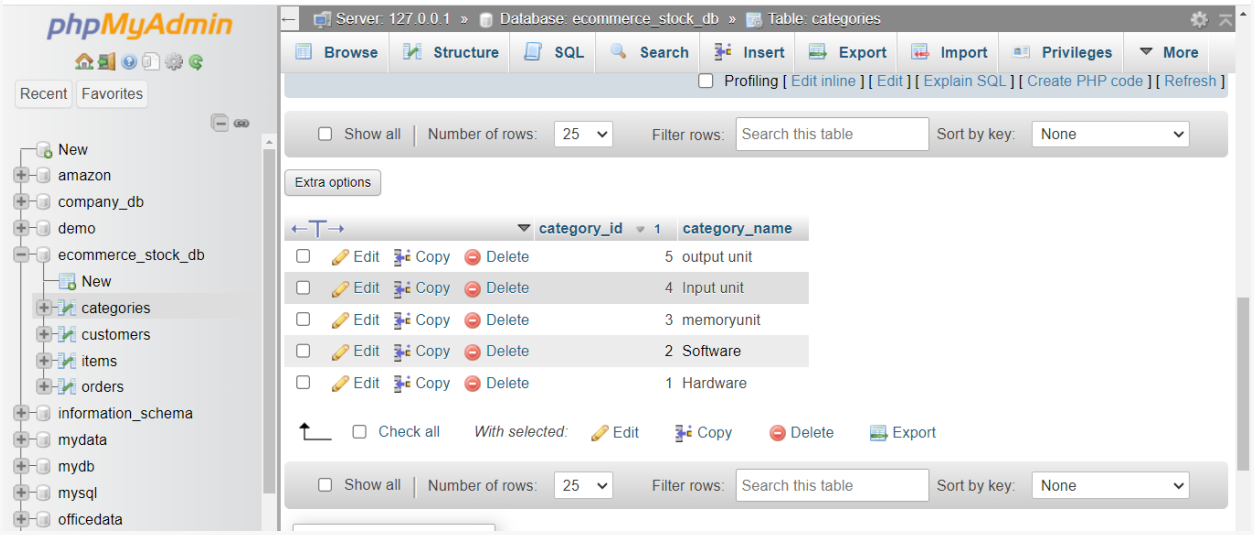
Query- SELECT * FROM items

ORDER BY item_id DESC;



Query- SELECT * FROM `categories`

ORDER BY category_id DESC;



Task 10: Backup and Restore Export a backup of the `ecommerce_stock_db` database to a SQL file. Then, write SQL statements to restore the database from the backup fileSubmission:

- Save your SQL script with a meaningful filename (e.g., `ecommerce_assignment.sql`).
- Submit the SQL script by the due date specified.