create table customer (

id int AUTO\_INCREMENT,

name VARCHAR(250),

email varchar(255),

address varchar(250));

create table orders (

id int PRIMARY key AUTO\_INCREMENT,

customer\_id INT,

order\_date date,

total float,

FOREIGN key (customer\_id) REFERENCES customer(id));

create table products (

id int PRIMARY key AUTO\_INCREMENT,

name varchar(255),

price float,

descriptions text);

INSERT into customer(name,email,address)

values("karthikeyan.R","[karthikeyan@gmail.com](mailto:karthikeyan@gmail.com)","303,thirumagalam");

INSERT into orders(customer\_id,order\_date,total)

values(1,2025-03-01,560),(2,2025-02-21,560),(3,2025-03-31,560);

INSERT into products(name,price,descriptions)

values("trouser",799,”m-size trouser brand basics),("tshirt",599,"peter england tshirt"),(“shirt”,1999,"blackberry-shirt");

**Retrieve all customers who have placed an order in the last 30 days:**

– SELECT \* FROM `customer` INNER JOIN orders on orders.customer\_id = customer.id WHERE orders.order\_date > CURDATE() - INTERVAL '30' DAY;

**Get the total amount of all orders placed by each customer**:

SELECT total,name FROM `customer` INNER JOIN orders on customer.id=orders.customer\_id

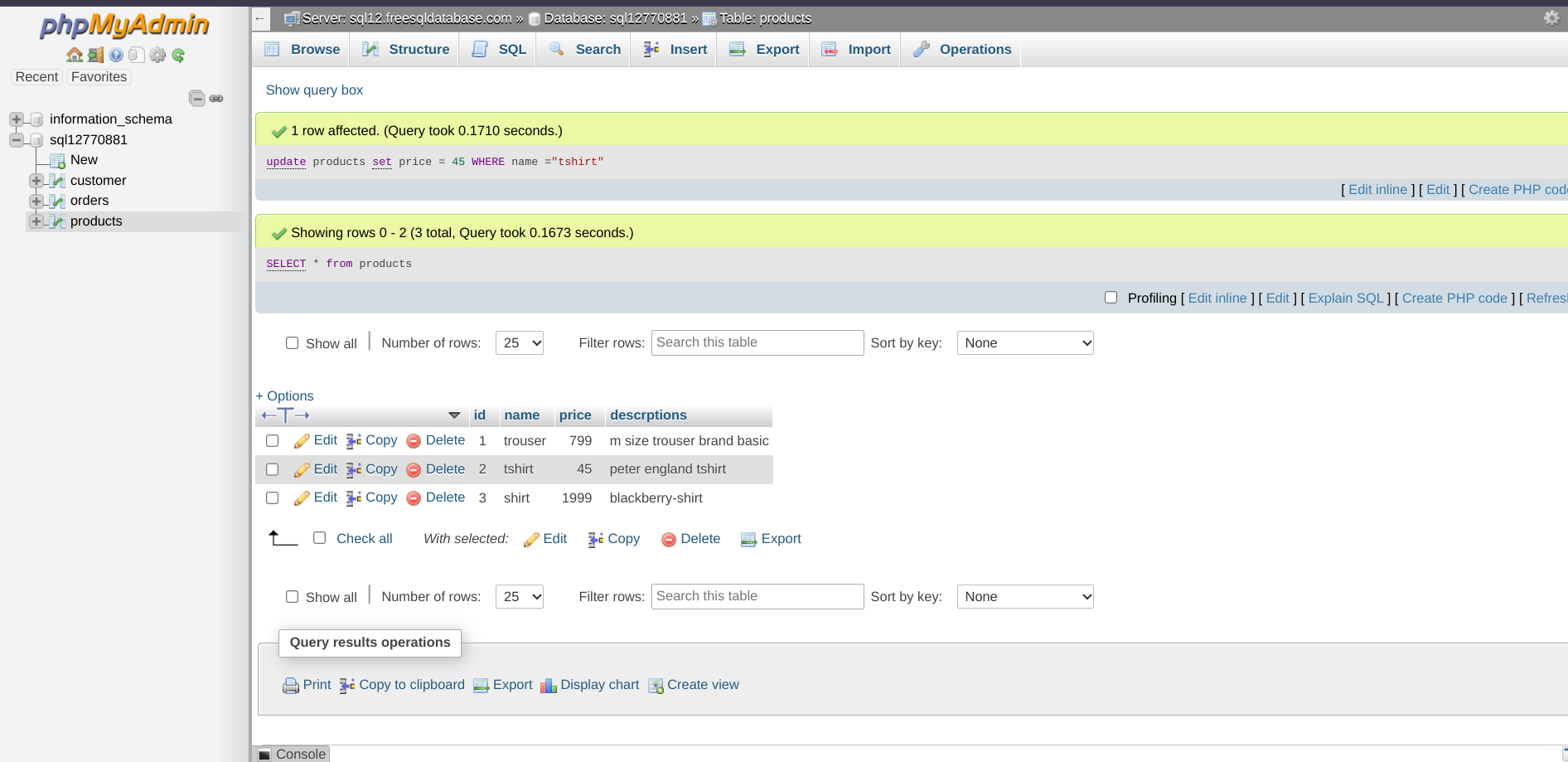
**Update the price of Product C to 45.00**

update products

set price = 45

WHERE name ="tshirt";

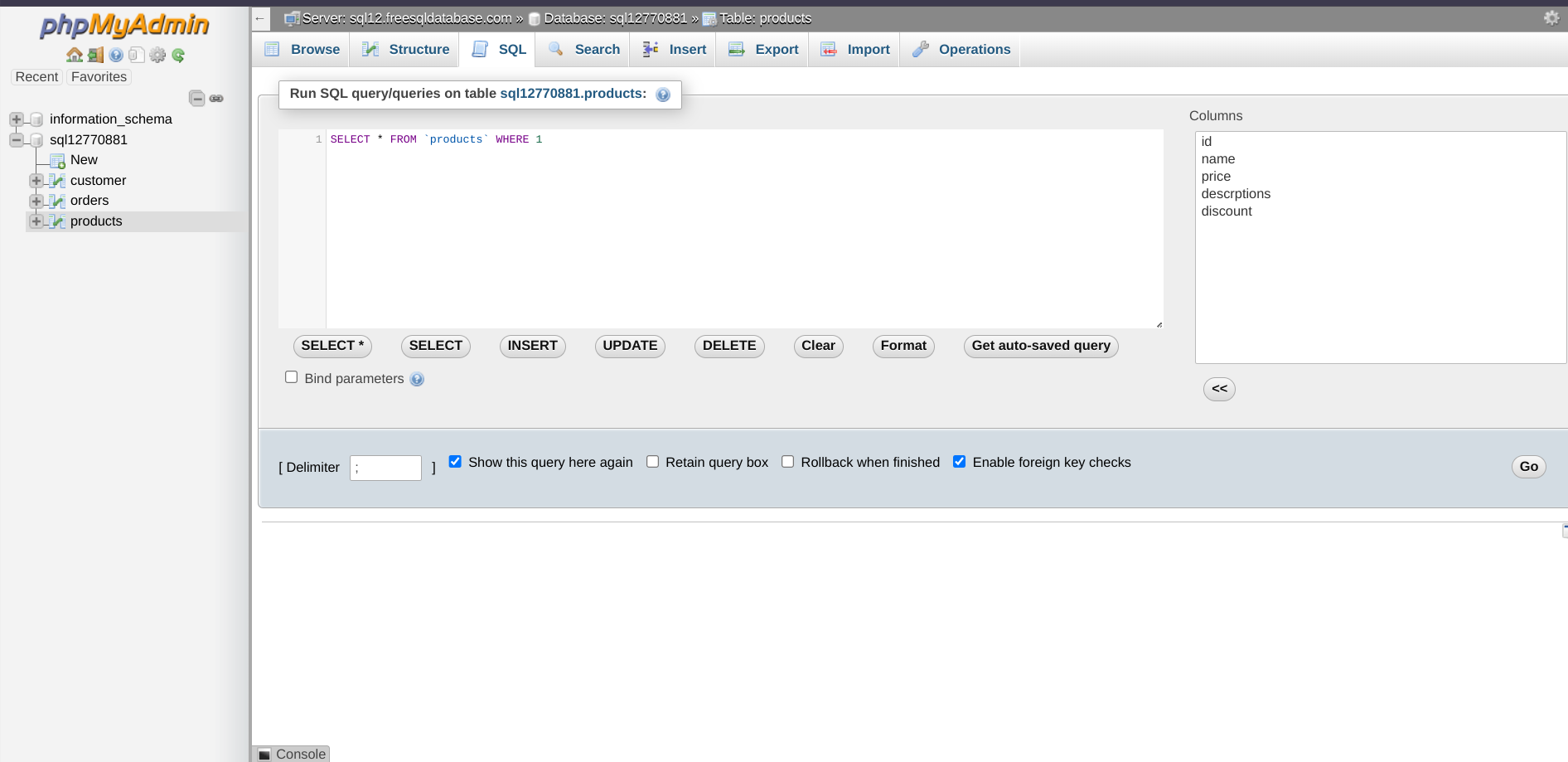
SELECT \* from products

****

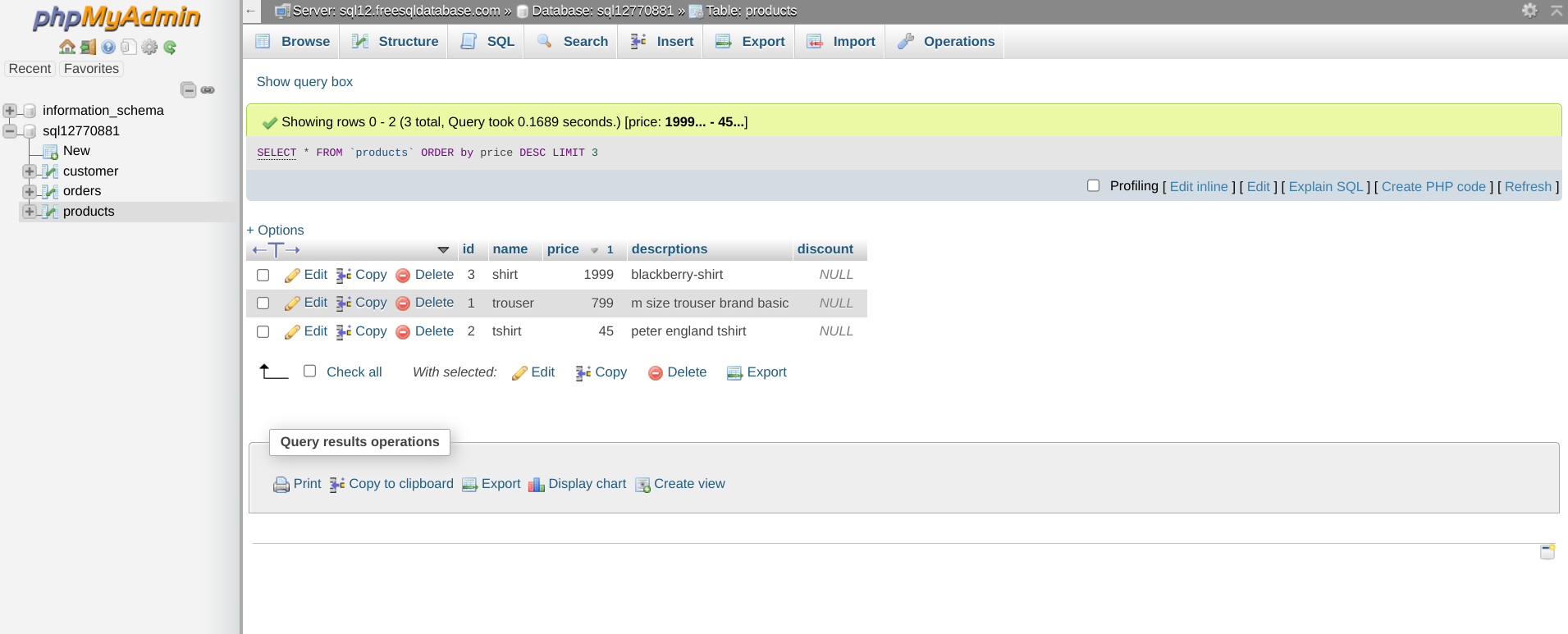
**Add a New column discount to the product table:**

ALTER table products

Add discount float ;

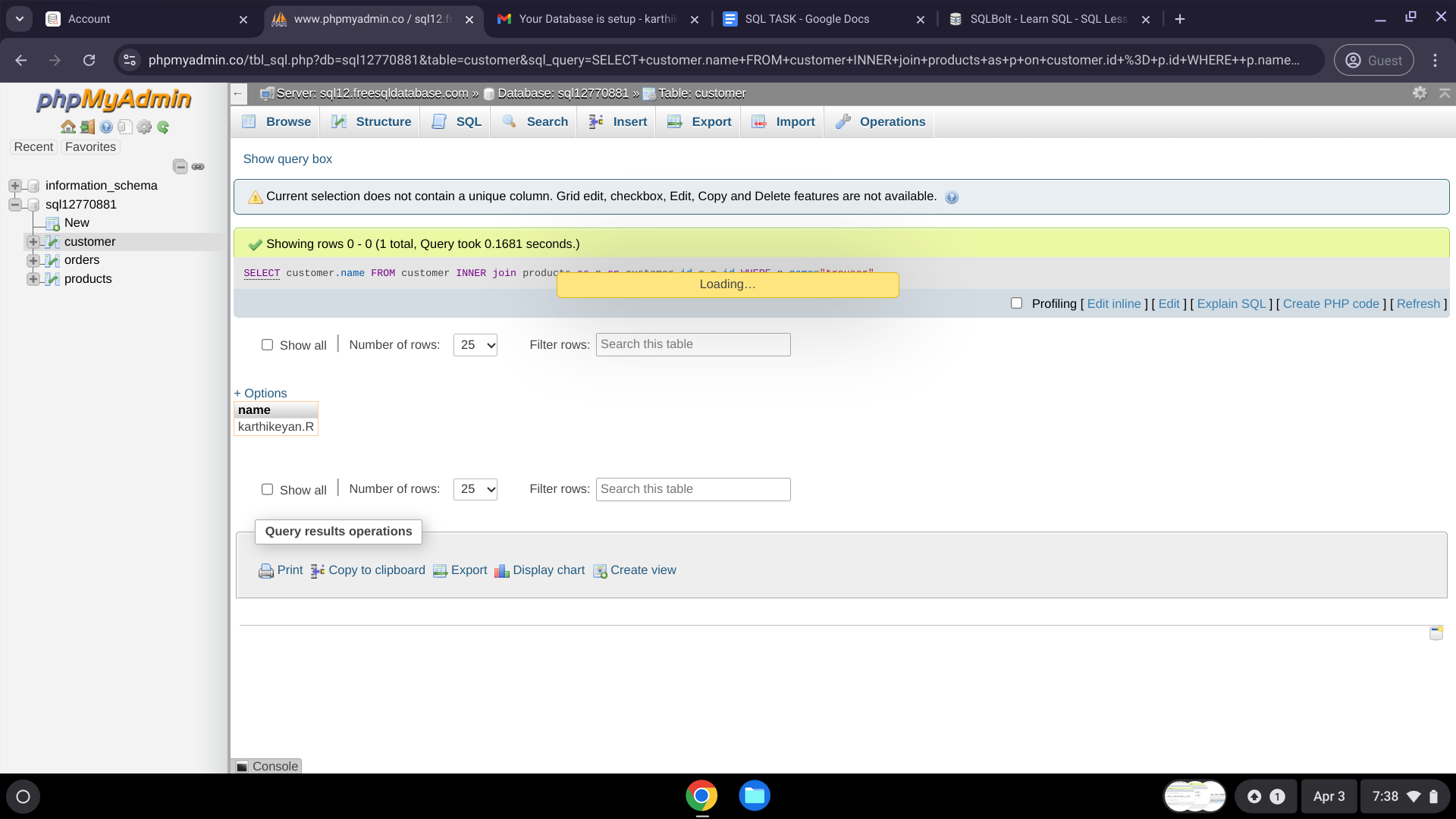


**Retrieve the top 3 products with highest price :**

SELECT \* FROM `products` ORDER by price DESC LIMIT 3; ****

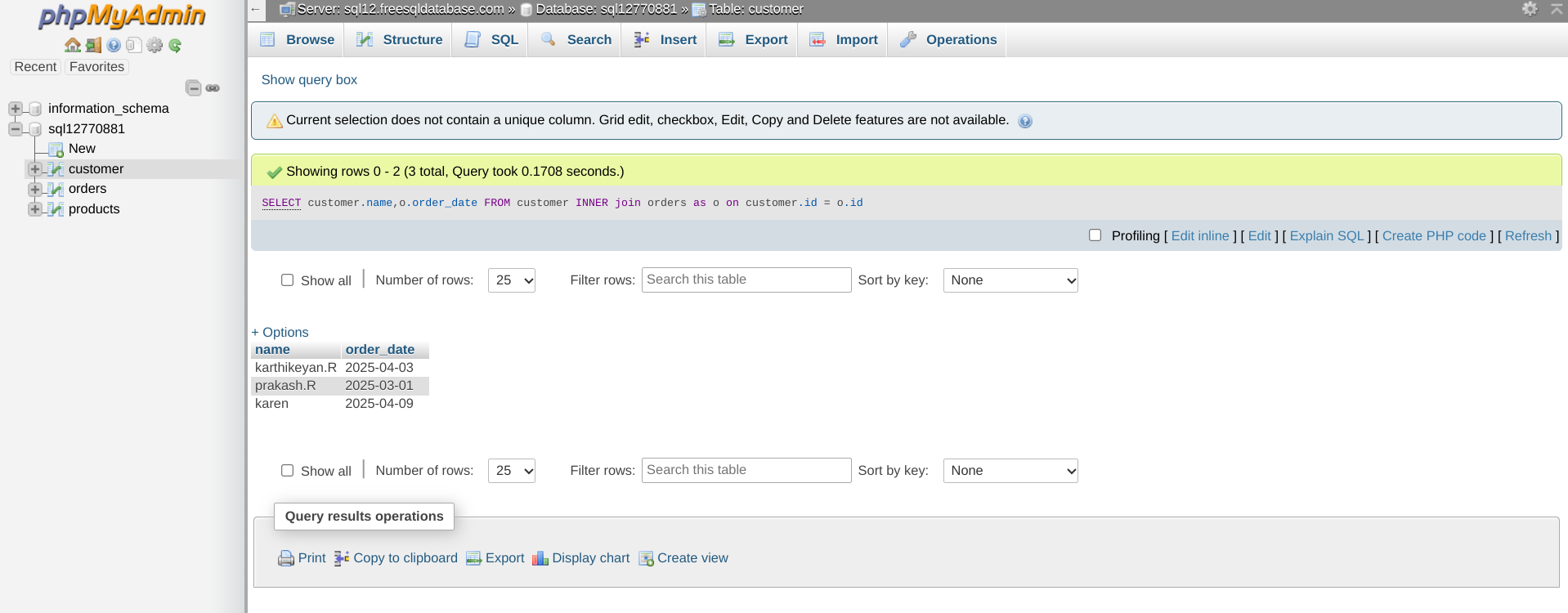
**Get the names of customer who have ordered product A :**

SELECT customer.name FROM customer INNER join products as p on customer.id = p.id WHERE p.name="trouser";

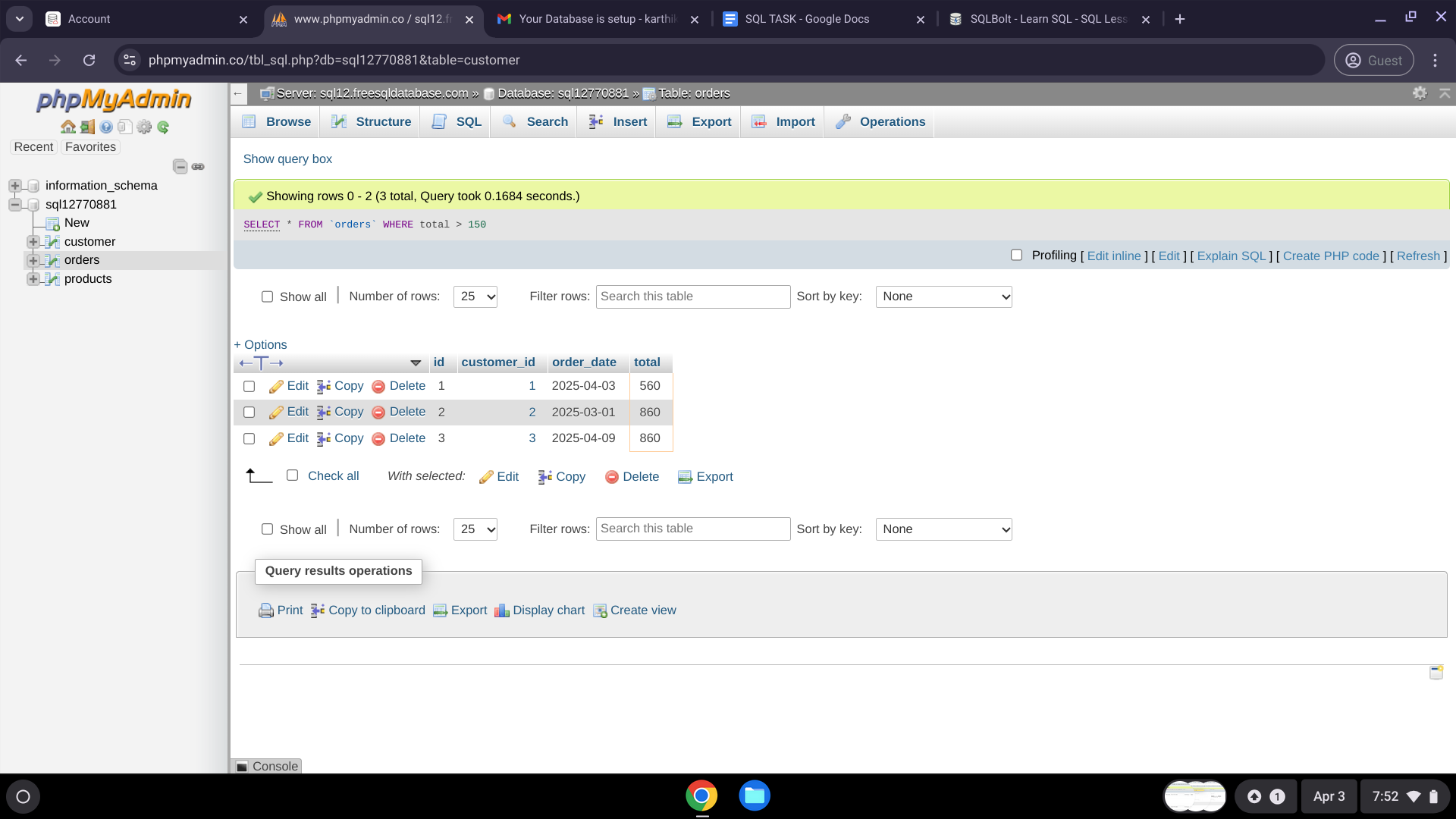
****

**Join the orders and customers tables to retrieve the customer name and order date for each order:**

**SELECT customer.name,o.order\_date FROM customer INNER join orders as o on customer.id = o.id ;**

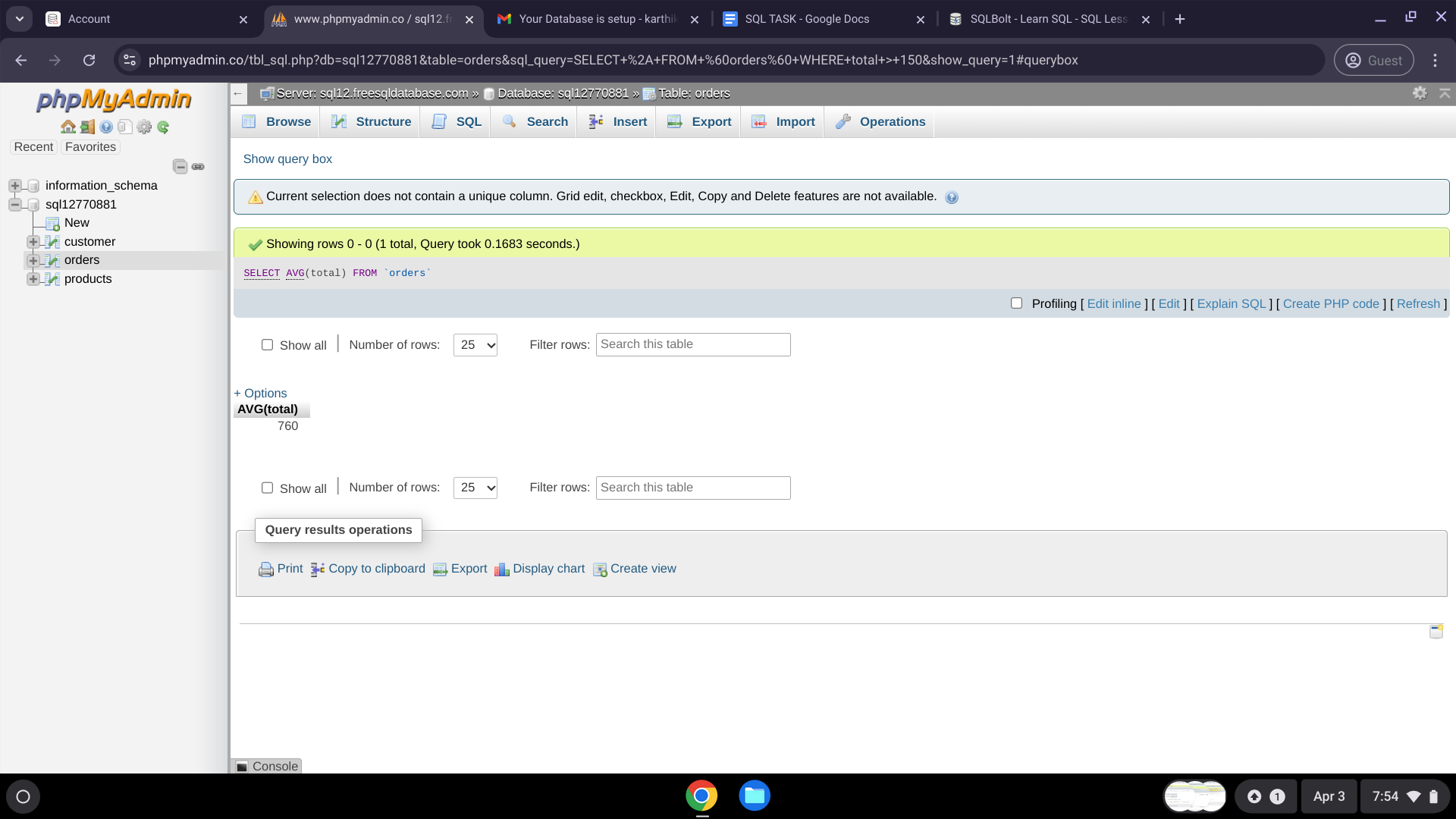
****

**Retrieve the order with a total amount greater than 150.00**SELECT \* FROM `orders` WHERE total > 150;

****

**Retrieve the average total of all orders :**

SELECT AVG(total) FROM `orders` ;



**Normalize the database by creating a separate table for order item and updating the order table to reference the order\_item table.**create table order\_item (

id int PRIMARY key AUTO\_INCREMENT,

customer\_id INT,

Order\_item text,

FOREIGN key (customer\_id) REFERENCES orders(id));

