Name : Tanishka Prasad Shinde

Emp ID:2000147767

Batch 4

Coding Challenge : Ecommerce

Tables:

```
create database ecommerce ;

use ecommerce;

1.

create table customers(
        customer_id INT primary key,
        f_name varchar(20),
        l_name varchar(20),
        email varchar(30),
        c_add varchar(40));

insert into customers values (1 , "John" , " Doe " , "johndoe@example.com" , "123 Main St,City"),

(2 , "Jane", "Smith" , "janesmith@example.com" , "456 Eim St , Town"),

(3 , "Robert" , "Johnson" , "robert@example.com" , "789 Oak St,Village"),

(4 , "Sarah" , "Brown" , "sarah@example.com" , "101 Pine St,Suburb"),

(5 , "David", "Lee" , "david@example.com" , "234 Cedar St,District"),

(6 , "Laura", "Hall" , "laura@example.com" , "567 Birch St County"),

(7 , "Michael" , "Davis" , "michael@example.com" , "890 Maple St, State"),

(8 , "Emma" , "Wilson" , "emma@example.com" , "321 Redwood St,Country"),

(9 , "William" , "Taylor" , "william@example.com" , "432 Spouce St,Province"),

(10 , "Olivia" , "Adams" , "olovia@example.com" , "765 Fir St,Territory");
```

```
create table products(
                product_id int primary key,
        namee varchar(30),
        descriptionn varchar(30),
        price double,
        stockQuantity int);
insert into products ( product_id, namee, descriptionn, price, stockQuantity) values
        (1, "Laptop","High-Performance Laptop",800,10),
    (2,"Smartphone","latest smartphone",600,15),
    (3,"Tablet","Portable tablet", 300,20),
    (4,"Headphones","Noice-cancelling",150,30),
    (5, "TV", "4K Smart TV",900,5),
    (6,"Coffee Maker", "Automatic coffee maker", 50, 25),
    (7,"Refrigerator","Energy-efficient", 700,10),
    (8,"Microwave over","Countertop microwave",80,15),
    (9,"Blender","High-speed blender",70, 20),
    (10,"Vaccum cleaner","Bagless vaccum cleaner",120,10);

select *from products;


create table cart(
                cart_id int primary key,
        customer_id int,
        product_id int,
        quantity int);
insert into cart(cart_id, customer_id, product_id, quantity) values
        (1,1,1,2),
```

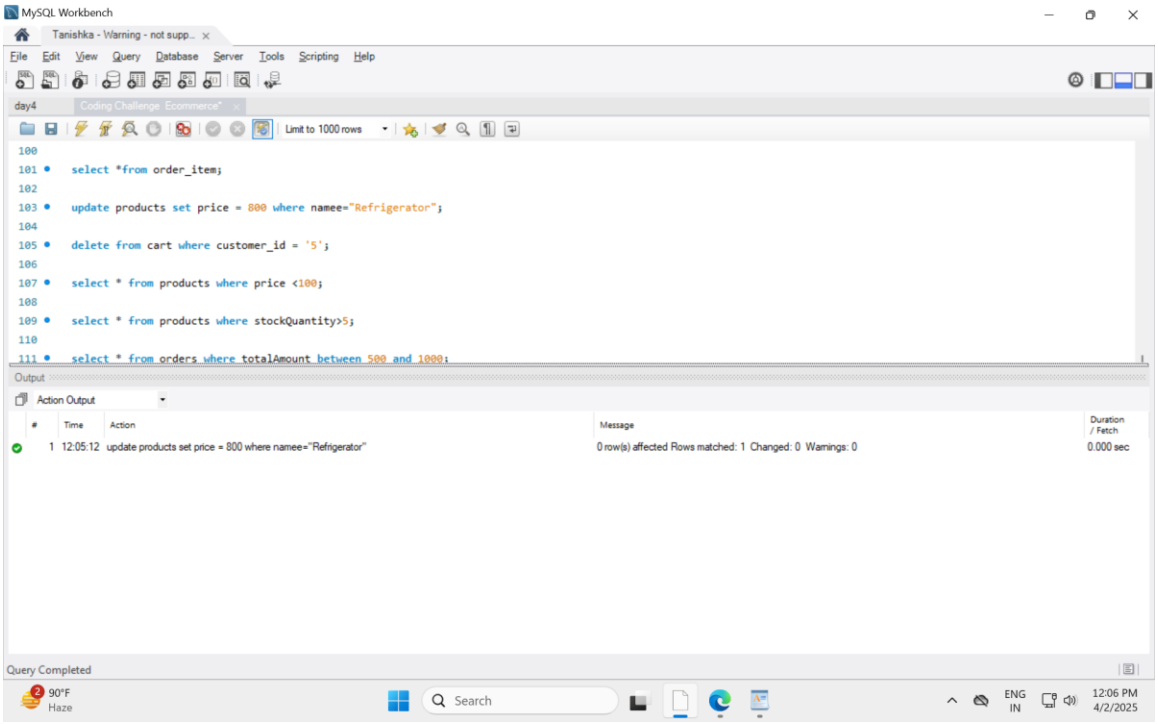
```
(2,1,3,1),
(3,2,2,3),
(4,3,4,4),
(5,3,5,2),
(6,4,6,1),
(7,5,1,1),
(8,6,10,2),
(9,6,9,3),
(10,7,7,2);
select *from cart;

create table orders(
        order_id int primary key,
    customer_id int,
    order_date date,
    totalAmount double);
insert into orders values
(1 , 1 , "2023-01-05" , 1200.00),
(2 , 2 , "2023-02-10" , 900.00),
(3 , 3 , "2023-03-15" , 300.00),
(4 , 4 , "2023-04-20" , 150.00),
(5 , 5 , "2023-05-25" , 1800.00),
(6 , 6 , "2023-06-30" , 400.00),
(7 , 7 , "2023-07-05" , 700.00),
(8 , 8 , "2023-08-10" , 160.00),
(9 , 9 , "2023-09-15" , 140.00),
(10 , 10 , "2023-10-20" , 1400.00);
```

```
select *from orders;
create table order_item(
                order_item_id int primary key,
        order_id int,
        product_id int,
        quantity int,
        itemAmount double);
Insert into order_item values
(1 , 1 , 1, 2 , 1600.00),
(2 , 1 , 3 , 1 ,300.00) ,
(3 , 2 , 2 , 3 , 1800.00) ,
(4 , 3 , 5 , 2 , 1800.00),
(5 , 4 , 4 , 4, 600.00) ,
(6 , 4 , 6 , 1 , 50.00),
(7 , 5 , 1 , 1 , 800.00),
(8 , 5 , 2 , 2 , 1200.00) ,
(9 , 6 , 10 , 2 , 240.00),
(10 , 6 , 9 , 3 , 210.00);
```

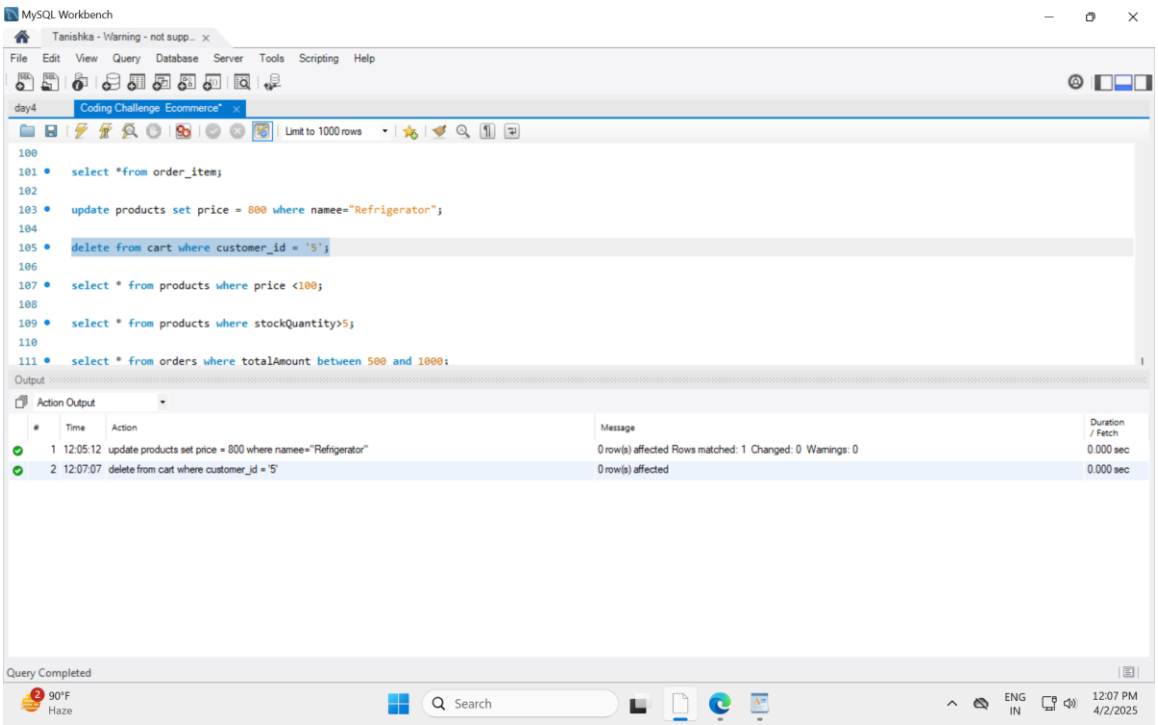
1. Update refrigerator product price to 800.

```
update products set price = 800 where namee="Refrigerator";
```



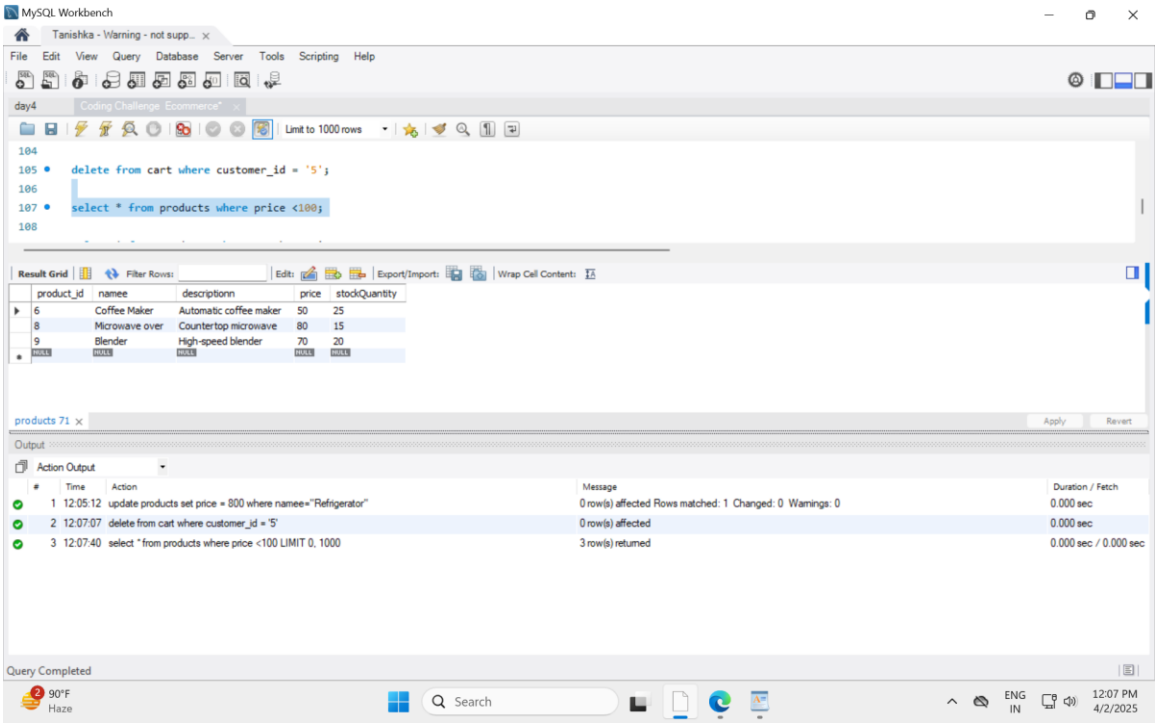
2. Remove all cart items for a specific customer.

delete from cart where customer_id = '5';



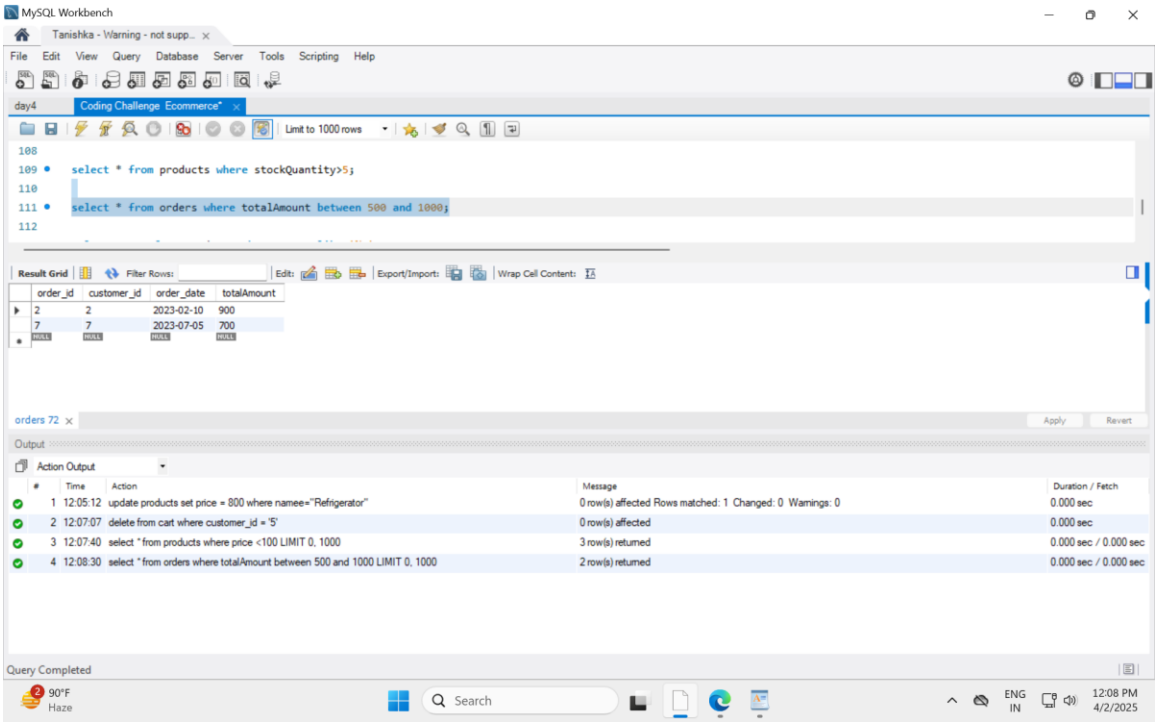
3. Retrieve Products Priced Below $100.

select * from products where price <100;



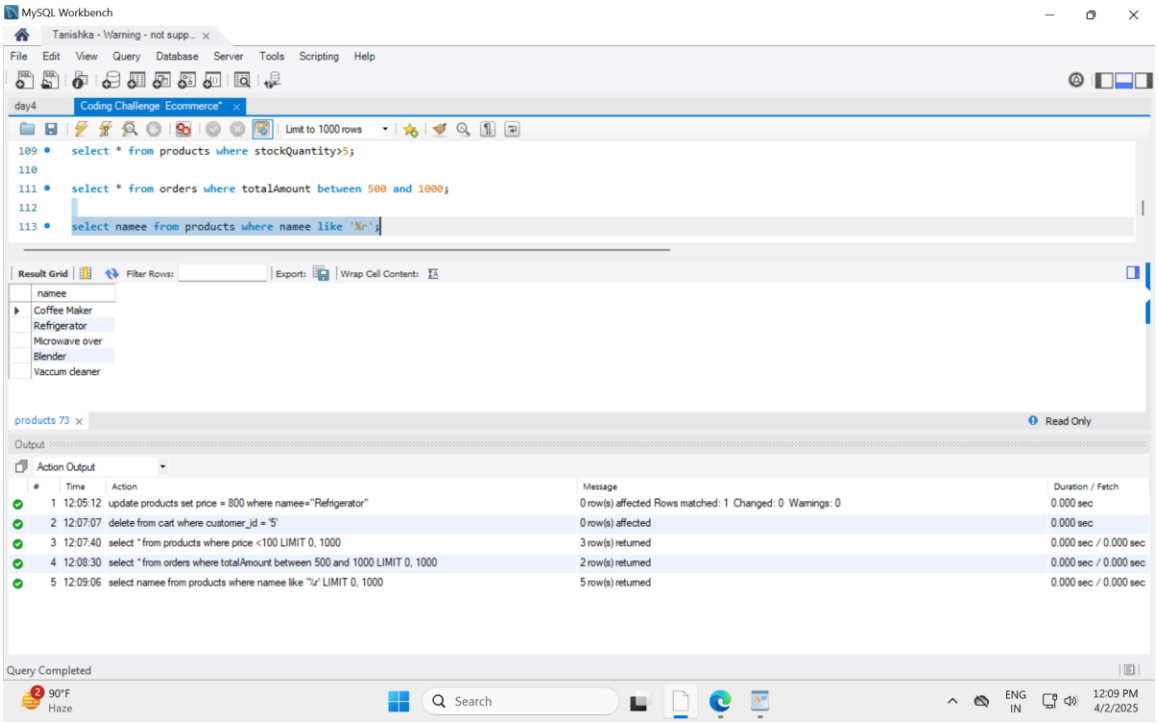
4. Find Products with Stock Quantity Greater Than 5

select * from products where stockQuantity>5;



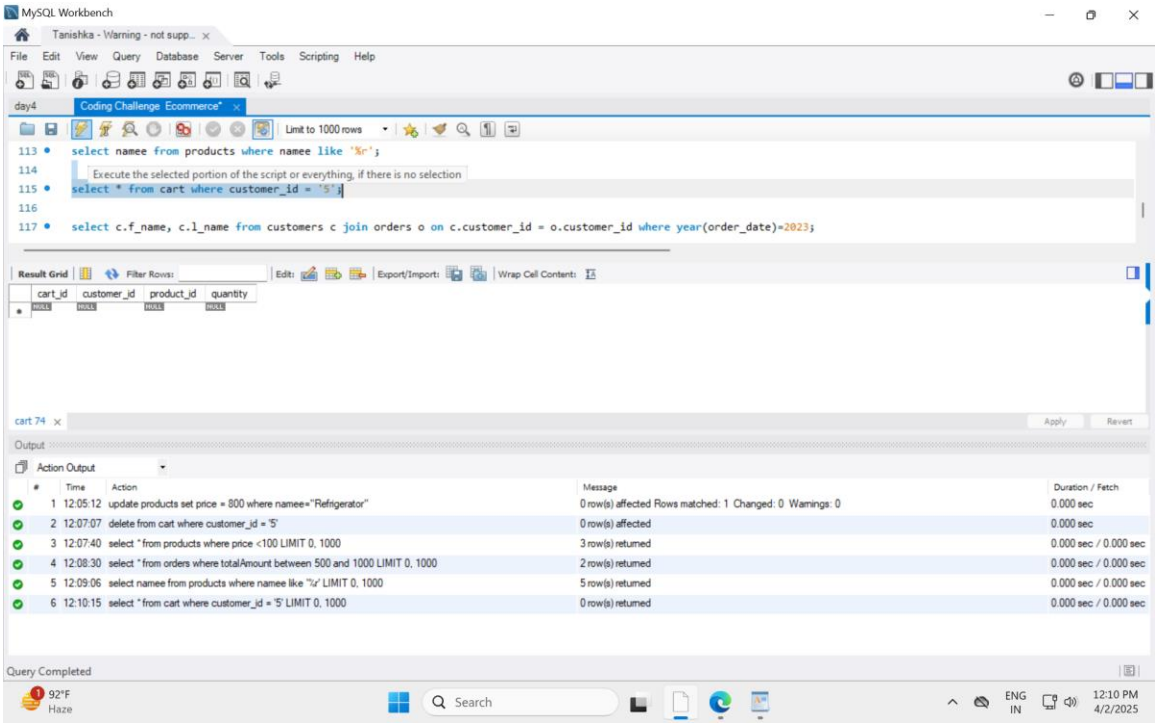
5. Retrieve Orders with Total Amount Between $500 and $1000.

select * from orders where totalAmount between 500 and 1000;



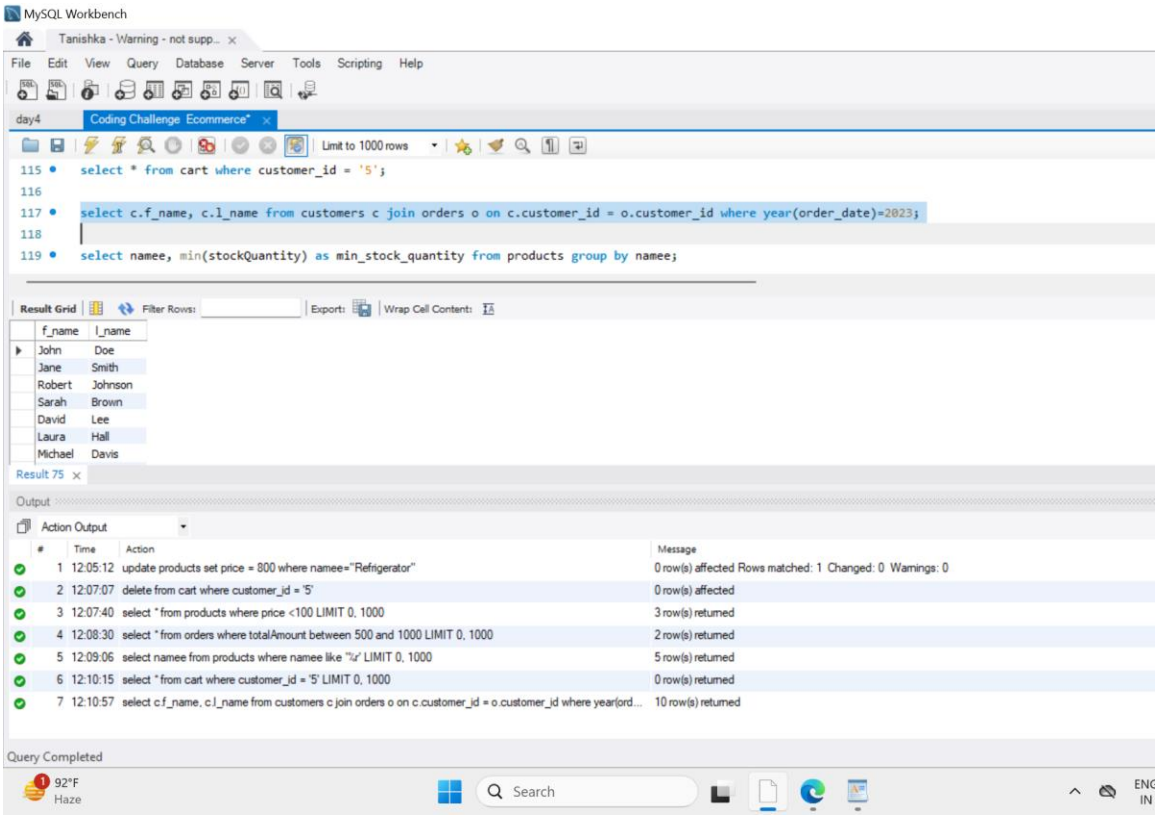
6. Find Products which name end with letter ‘r’.

select namee from products where namee like '%r';



7. Retrieve Cart Items for Customer 5.

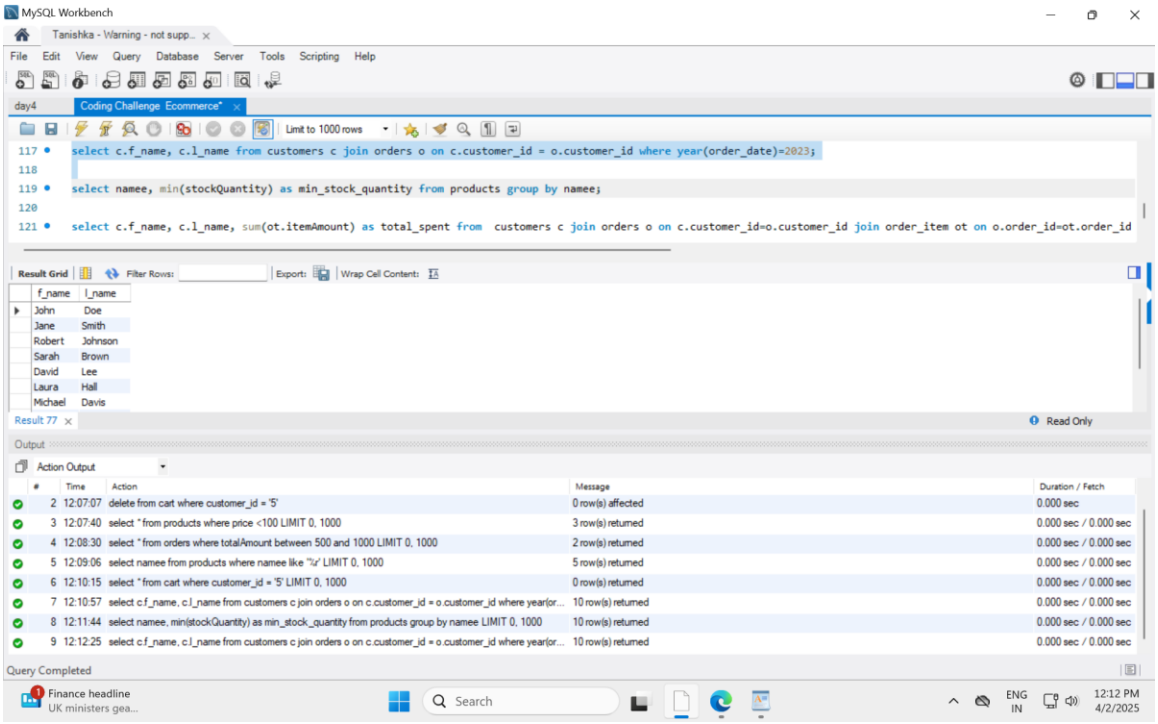
select * from cart where customer_id = '5';



8. Find Customers Who Placed Orders in 2023.

select c.f_name, c.l_name from customers c

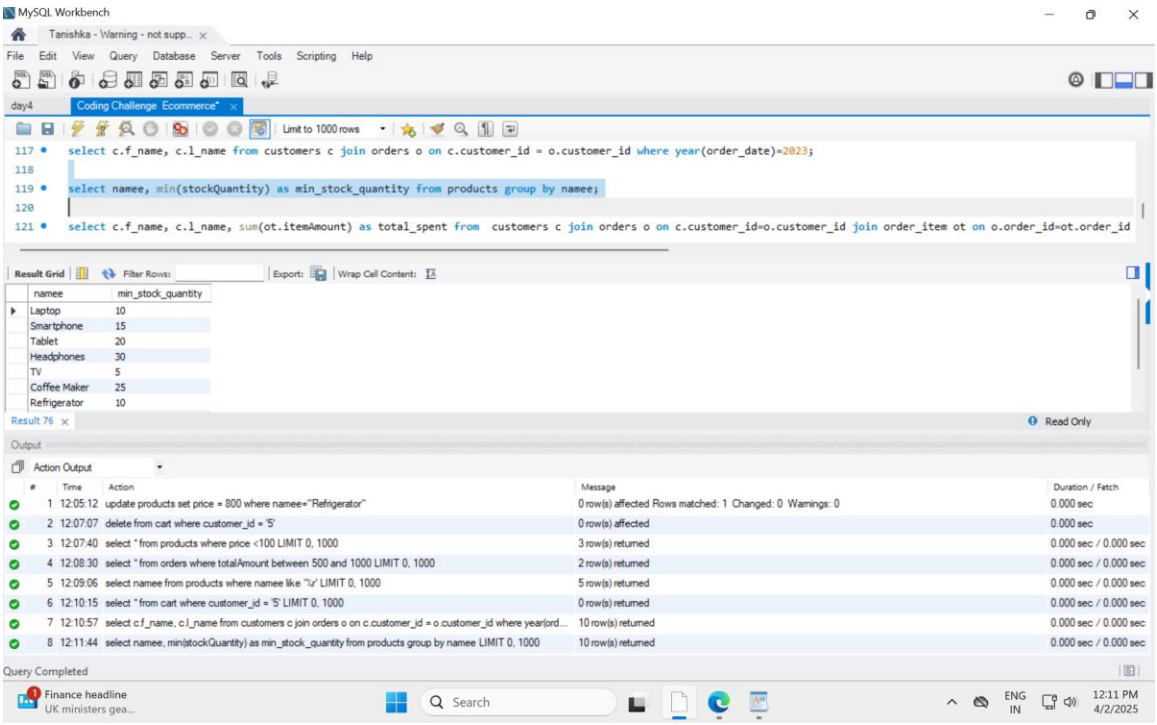
join orders o on c.customer_id = o.customer_id where year(order_date)=2023;



9. Determine the Minimum Stock Quantity for Each Product Category.

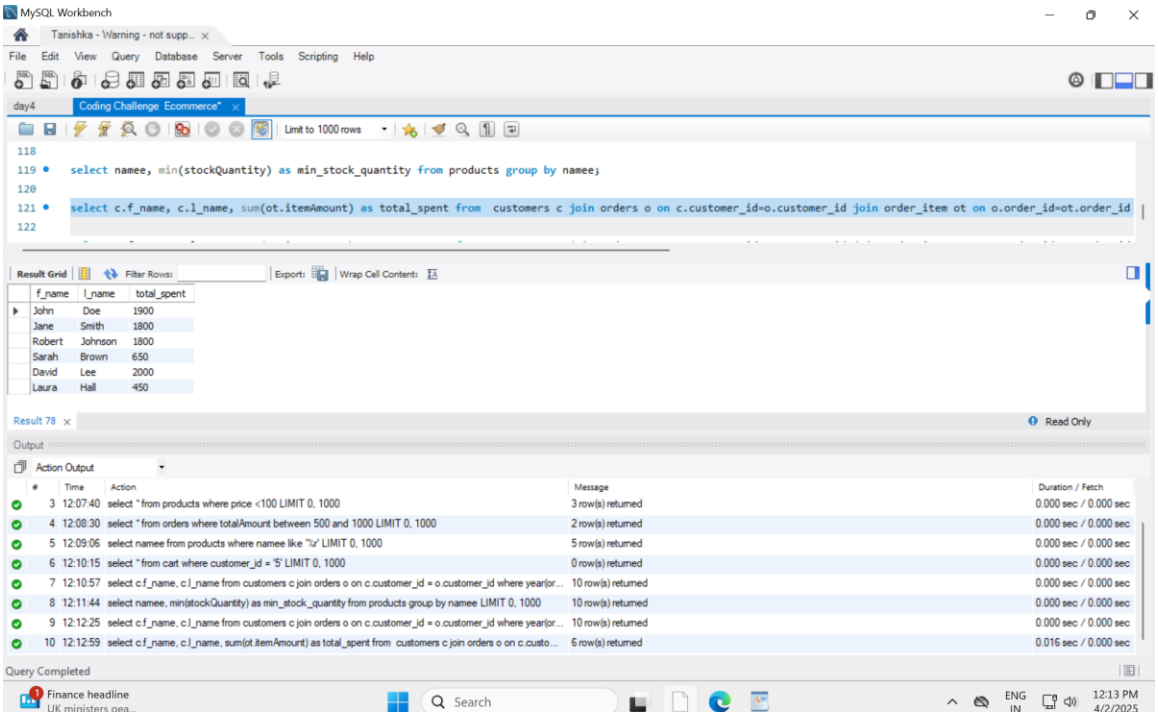
select namee, min(stockQuantity) as min_stock_quantity

from products group by namee;



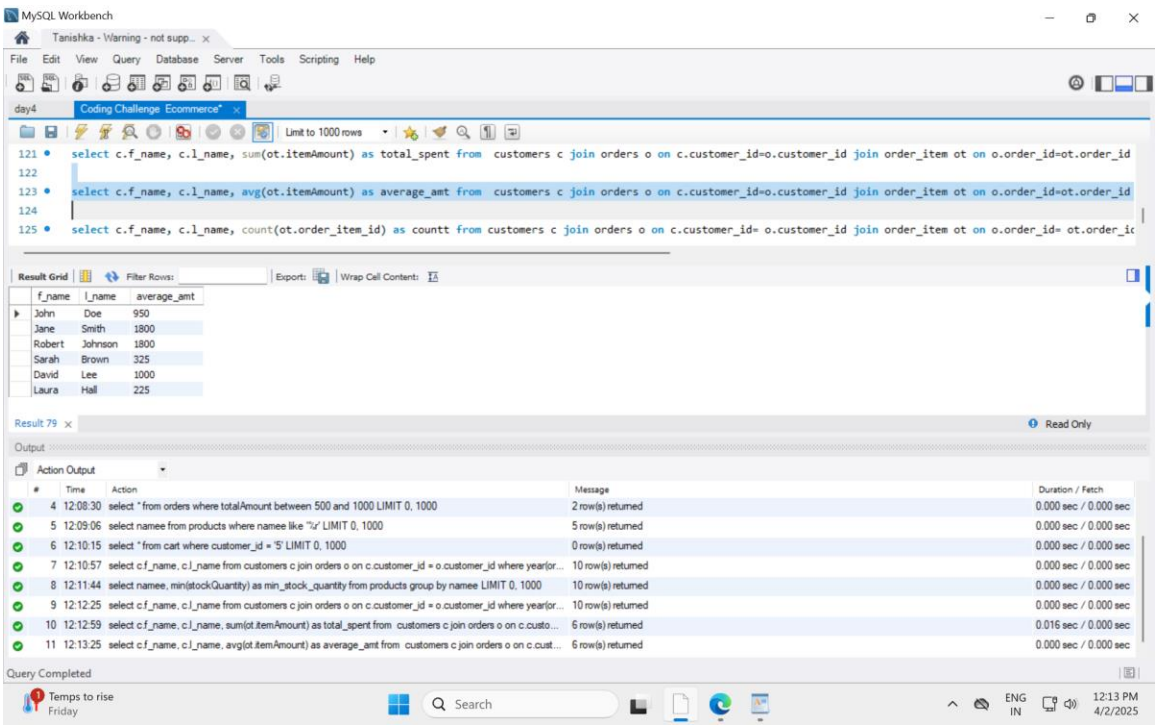
10. Calculate the Total Amount Spent by Each Customer.

select c.f_name, c.l_name, sum(ot.itemAmount) as total_spent from customers c join orders o on c.customer_id=o.customer_id join order_item ot on o.order_id=ot.order_id group by o.order_id;



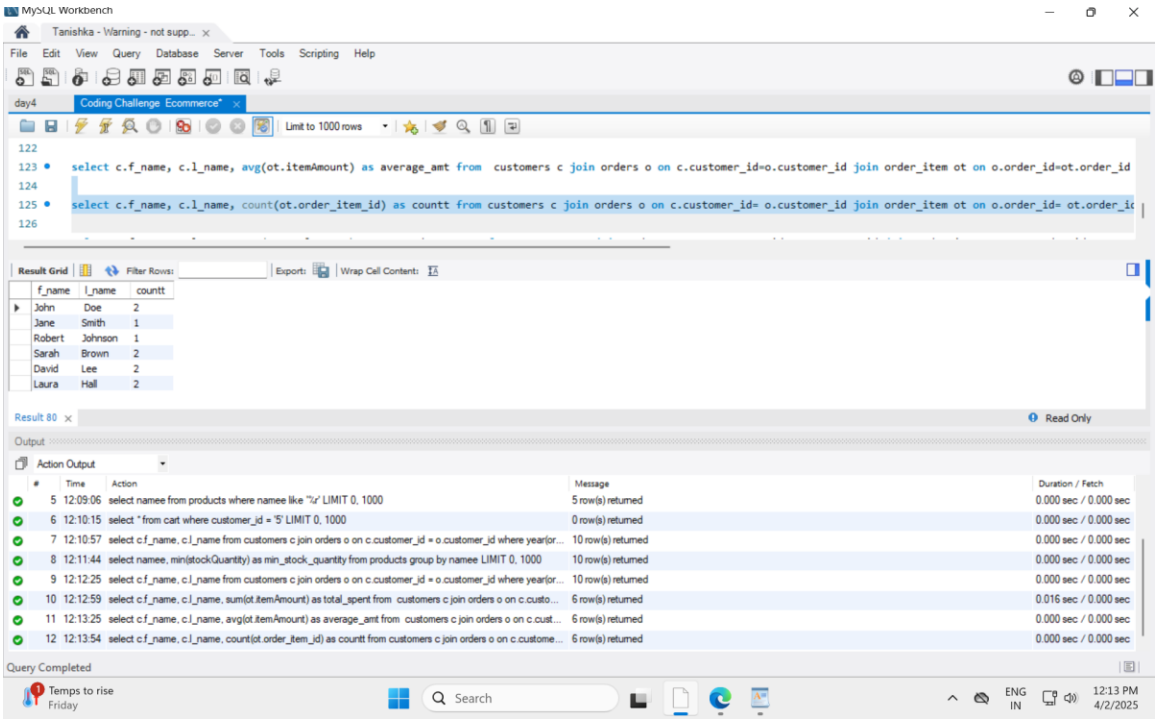
11. Find the Average Order Amount for Each Customer.

select c.f_name, c.l_name, avg(ot.itemAmount) as average_amt from customers c join orders o on c.customer_id=o.customer_id join order_item ot on o.order_id=ot.order_id group by o.order_id;



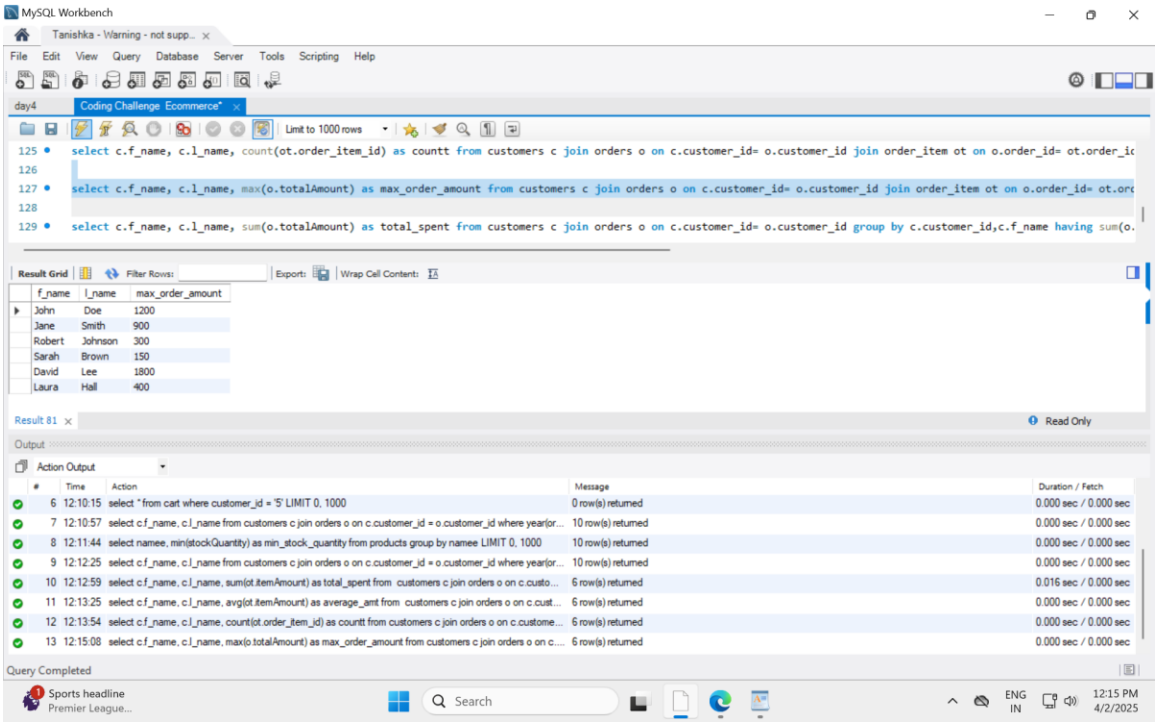
12. Count the Number of Orders Placed by Each Customer.

select c.f_name, c.l_name, count(ot.order_item_id) as countt from customers c join orders o on c.customer_id= o.customer_id join order_item ot on o.order_id= ot.order_id group by c.customer_id;



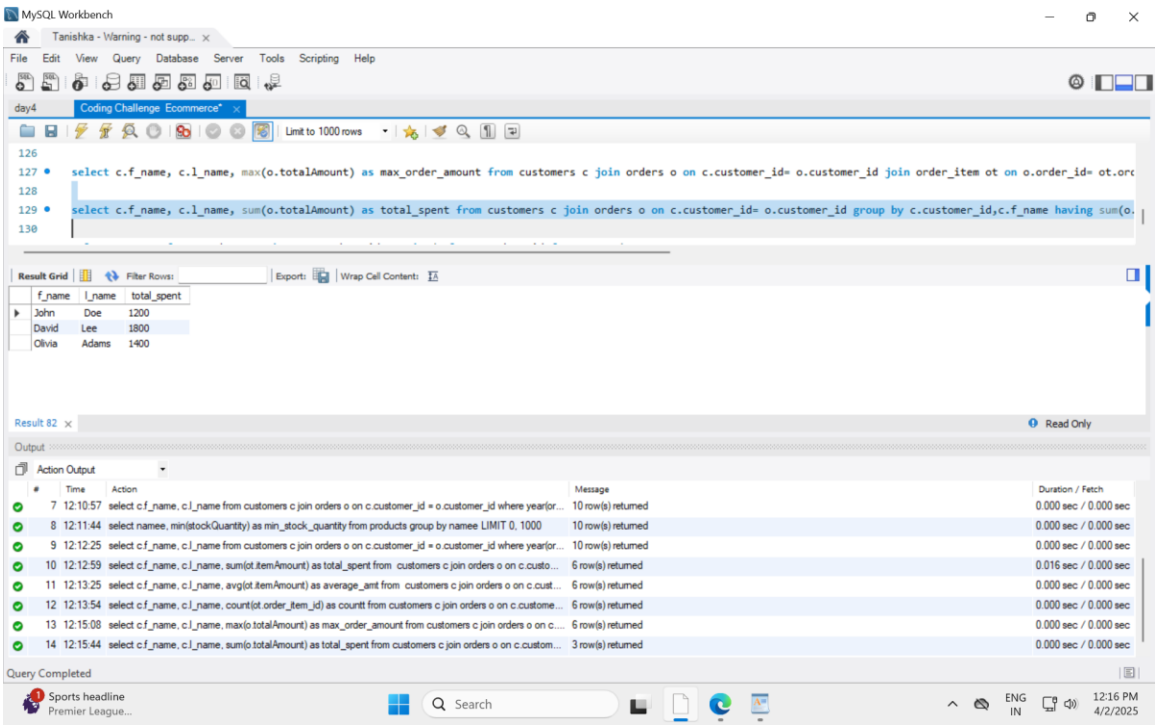
13. Find the Maximum Order Amount for Each Customer.

select c.f_name, c.l_name, max(o.totalAmount) as max_order_amount from customers c join orders o on c.customer_id= o.customer_id join order_item ot on o.order_id= ot.order_id group by c.customer_id;



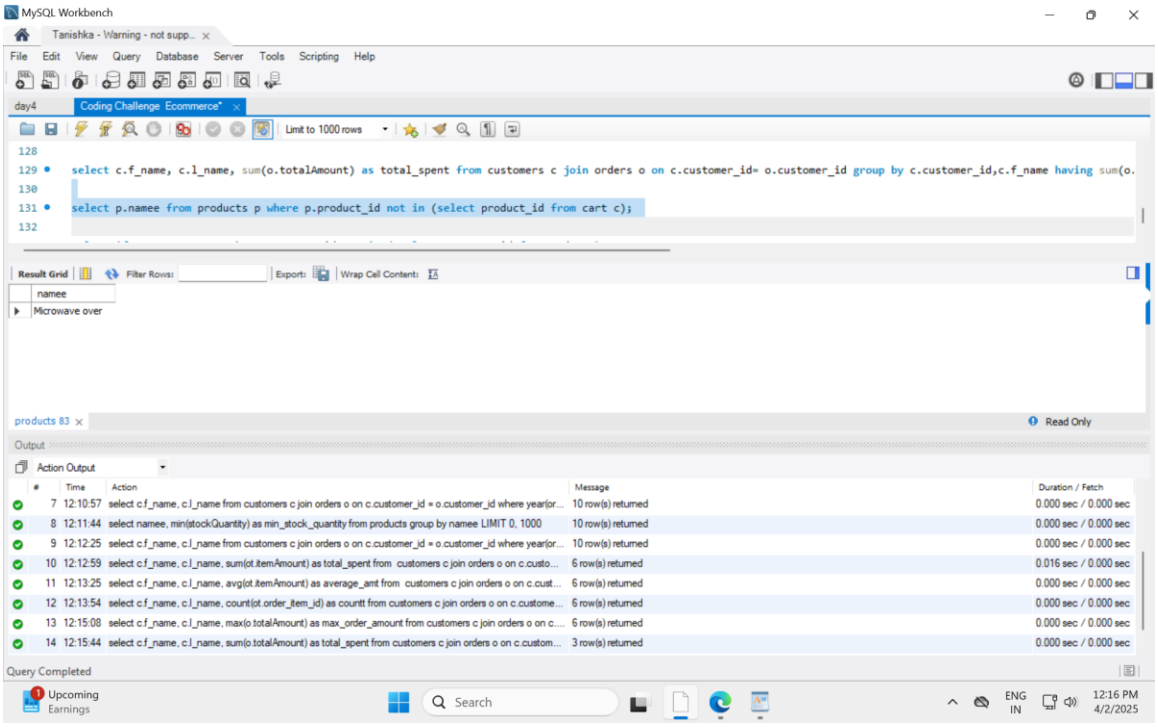
14. Get Customers Who Placed Orders Totaling Over $1000

select c.f_name, c.l_name, sum(o.totalAmount) as total_spent from customers c join orders o on c.customer_id= o.customer_id group by c.customer_id,c.f_name having sum(o.totalAmount)>1000;



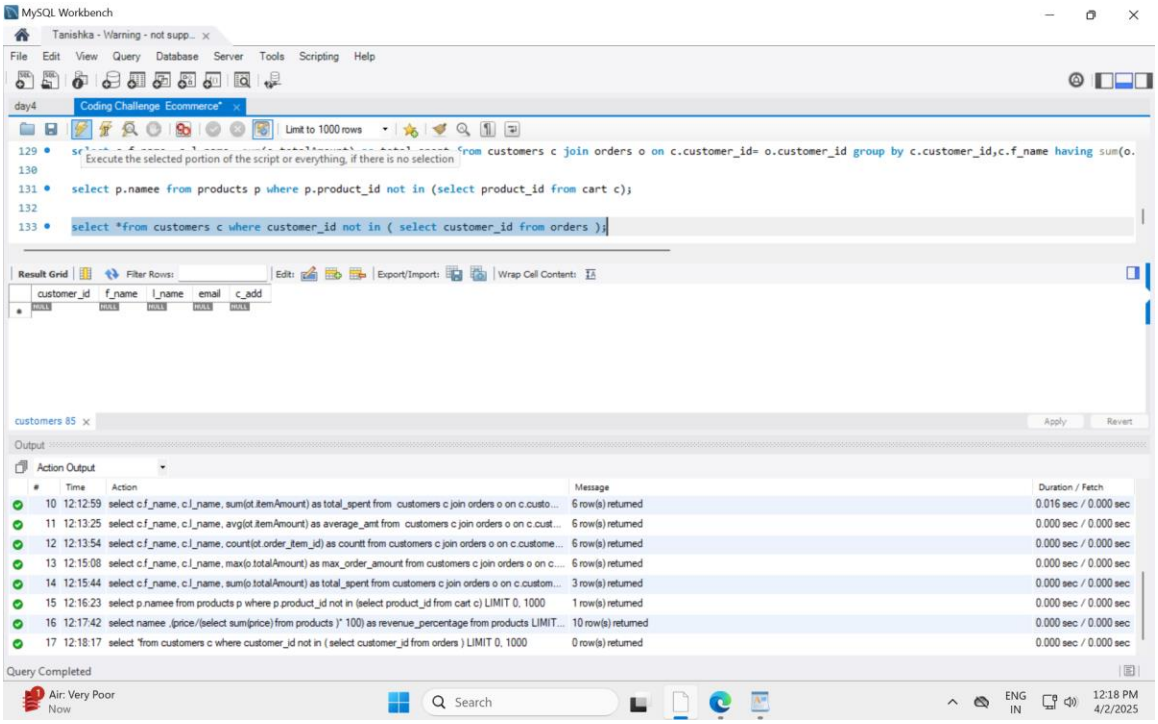
15. Subquery to Find Products Not in the Cart.

select p.namee from products p where p.product_id not in (select product_id from cart c);



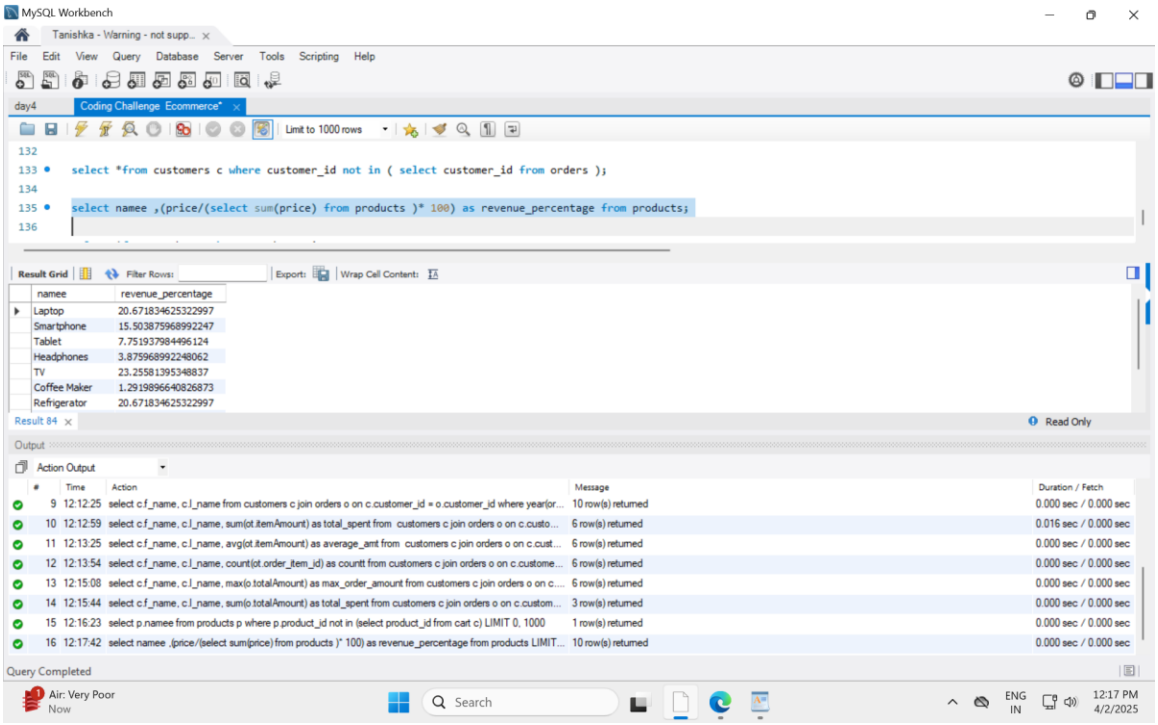
16. Subquery to Find Customers Who Haven't Placed Orders.

select *from customers c where customer_id not in (select customer_id from orders);



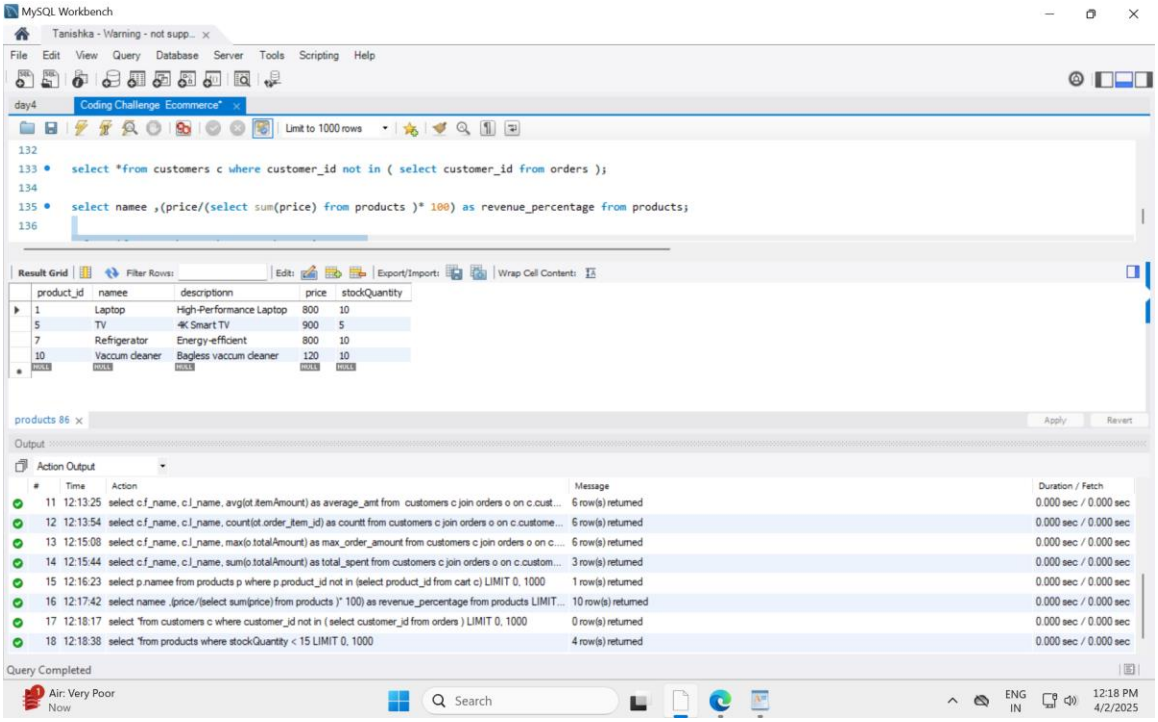
17. Subquery to Calculate the Percentage of Total Revenue for a Product.

select namee ,(price/(select sum(price) from products)* 100) as revenue_percentage from products;



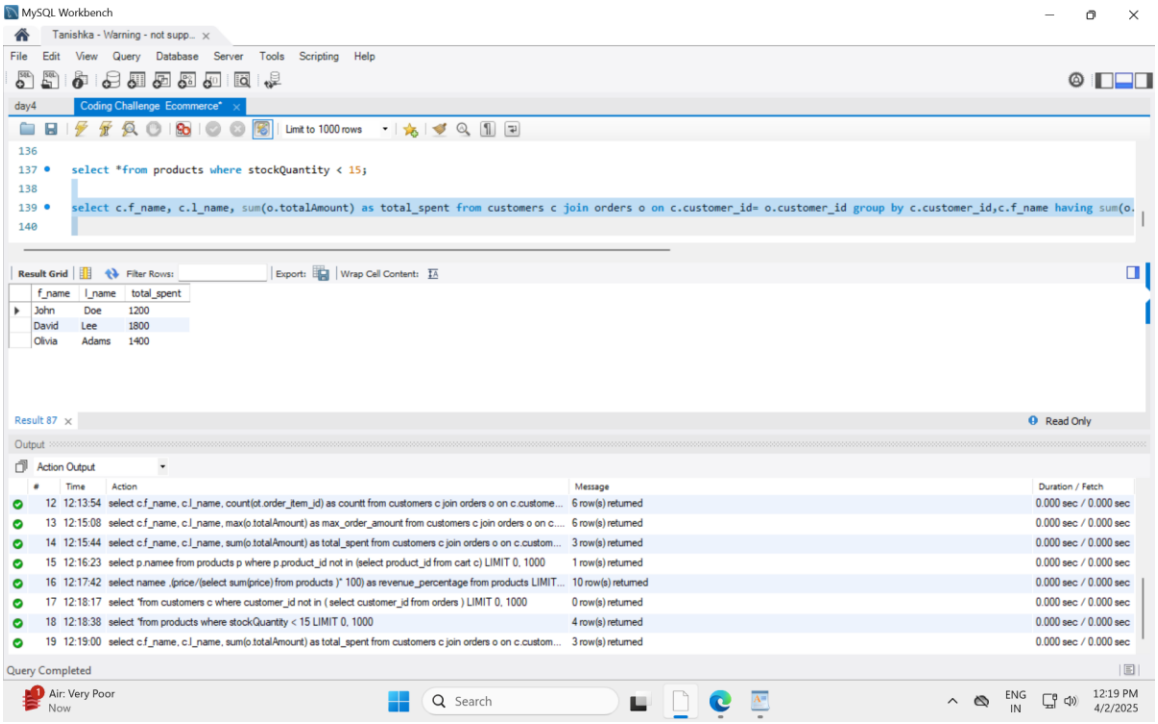
18. Subquery to Find Products with Low Stock.

select *from products where stockQuantity < 15;



19. Subquery to Find Customers Who Placed High-Value Orders

select c.f_name, c.l_name, sum(o.totalAmount) as total_spent from customers c join orders o on c.customer_id= o.customer_id group by c.customer_id,c.f_name having sum(o.totalAmount)>1000;


MySQL Workbench
Tanishka - Warning - not supp...
File Edit View Query Database Server Tools Scripting Help
day4
Coding Challenge_Ecommerce*
Limit to 1000 rows
136
137 select *from products where stockQuantity < 15;
138
139 select c.f_name, c.l_name, sum(o.totalAmount) as total_spent from customers c join orders o on c.customer_id= o.customer_id group by c.customer_id,c.f_name having sum(o.
140
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
f_name l_name total_spent
John Doe 1200
David Lee 1800
Olivia Adams 1400
Result 87
Read Only
Output
Action Output
Time Action Message Duration / Fetch
12 12:13:54 select c.f_name, c.l_name, count(ot.order_item_id) as countt from customers c join orders o on c.custome... 6 row(s) returned 0.000 sec / 0.000 sec
13 12:15:08 select c.f_name, c.l_name, max(o.totalAmount) as max_order_amount from customers c join orders o on c.... 6 row(s) returned 0.000 sec / 0.000 sec
14 12:15:44 select c.f_name, c.l_name, sum(o.totalAmount) as total_spent from customers c join orders o on c.custom... 3 row(s) returned 0.000 sec / 0.000 sec
15 12:16:23 select p.namee from products p where p.product_id not in (select product_id from cart c) LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec
16 12:17:42 select namee ,(price/(select sum(price) from products)* 100) as revenue_percentage from products LIMIT... 10 row(s) returned 0.000 sec / 0.000 sec
17 12:18:17 select *from customers c where customer_id not in (select customer_id from orders) LIMIT 0, 1000 0 row(s) returned 0.000 sec / 0.000 sec
18 12:18:38 select *from products where stockQuantity < 15 LIMIT 0, 1000 4 row(s) returned 0.000 sec / 0.000 sec
19 12:19:00 select c.f_name, c.l_name, sum(o.totalAmount) as total_spent from customers c join orders o on c.custom... 3 row(s) returned 0.000 sec / 0.000 sec
Query Completed
Air: Very Poor
Now
Search
ENG
IN
12:19 PM
4/2/2025