Ecommerce – SQL

create database SQLassignment;

use SQLassignment;

Customer's Table:

create table customers(customer_id int primary key,first_name varchar(60),last_name varchar(60),email varchar(60),address varchar(60),password varchar(60));

insert into customers values(1,"John","Doe","johndoe@example.com","123 mainstret","1234");

insert into customers values (2, "Jane", "Smith", "janesmith@example.com", "456 Elm St, Town", 2345);

insert into customers values (3, "Robert", "Johnson", "robert@example.com", "789 Oak St, Village", 3456);

insert into customers values (4, "Sarah", "Brown", "sarah@example.com", "101 Pine St, Suburb", 4567);

insert into customers values (5, "David", "Lee", "david@example.com", "234 Cedar St, District", 5678);

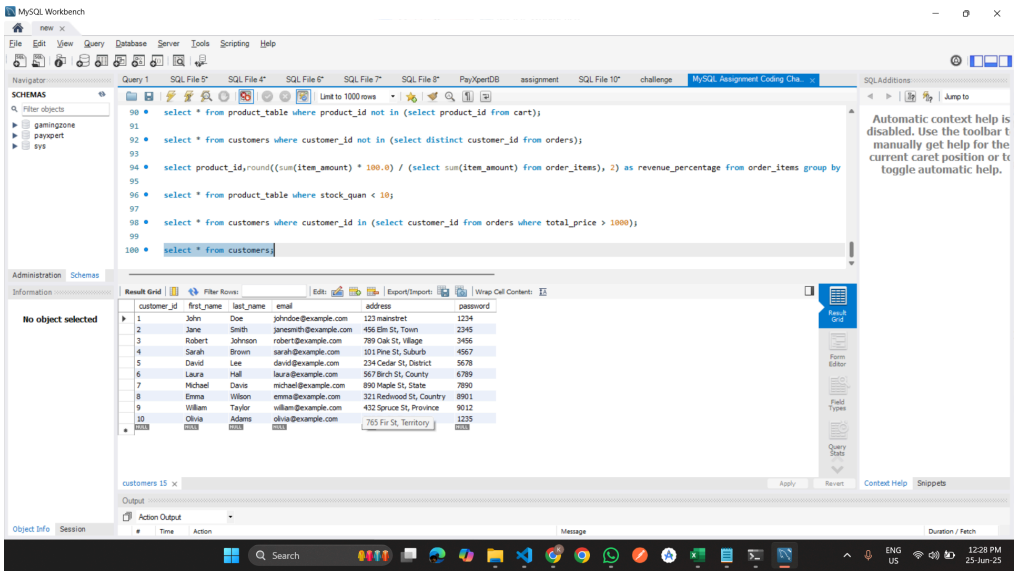
insert into customers values (6, "Laura", "Hall", "laura@example.com", "567 Birch St, County", 6789);

insert into customers values (7, "Michael", "Davis", "michael@example.com", "890 Maple St, State", 7890);

insert into customers values (8, "Emma", "Wilson", "emma@example.com", "321 Redwood St, Country", 8901);

insert into customers values (9, "William", "Taylor", "william@example.com", "432 Spruce St, Province", 9012);

insert into customers values (10, "Olivia", "Adams", "olivia@example.com", "765 Fir St, Territory", 1235);



Product Table:

create table product_table(product_id int primary key,prod_name varchar(60),price decimal(10,2),description varchar(60),stock_quan int);

insert into product_table values (1, "Laptop", 800.00, "High-performance laptop", 10);

insert into product_table values (2, "Smartphone", 600.00, "Latest smartphone", 15);

insert into product_table values (3, "Tablet", 300.00, "Portable tablet", 20);

insert into product_table values (4, "Headphones", 150.00, "Noise-canceling", 30);

insert into product_table values (5, "TV", 900.00, "4K Smart TV", 5);

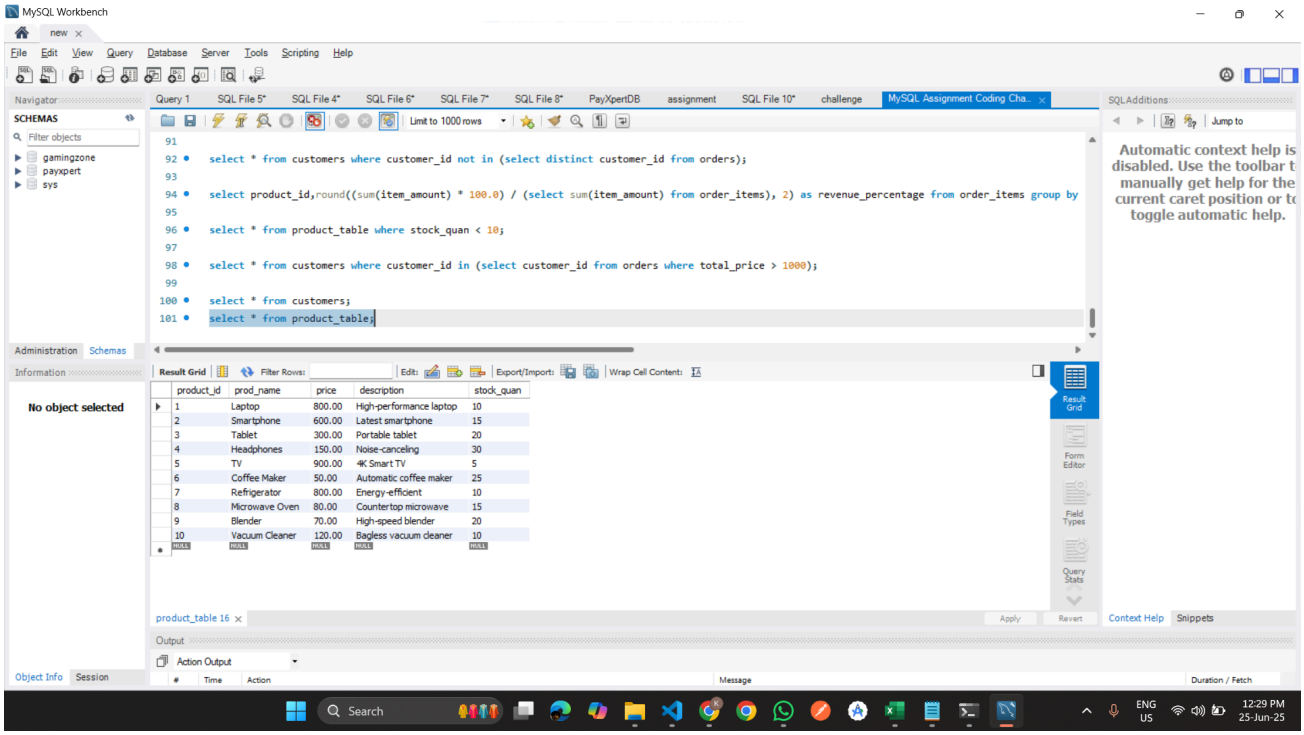
insert into product_table values (6, "Coffee Maker", 50.00, "Automatic coffee maker", 25);

insert into product_table values (7, "Refrigerator", 700.00, "Energy-efficient", 10);

insert into product_table values (8, "Microwave Oven", 80.00, "Countertop microwave", 15);

insert into product_table values (9, "Blender", 70.00, "High-speed blender", 20);

insert into product_table values (10, "Vacuum Cleaner", 120.00, "Bagless vacuum cleaner", 10);



Orders Table:

create table orders (order_id int primary key,customer_id int,order_date date,total_price decimal(10,2),foreign key (customer_id) references customers(customer_id));

insert into orders values (1, 1, '2023-01-05', 1200.00);

insert into orders values (2, 2, '2023-02-10', 900.00);

insert into orders values (3, 3, '2023-03-15', 300.00);

insert into orders values (4, 4, '2023-04-20', 150.00);

insert into orders values (5, 5, '2023-05-25', 1800.00);

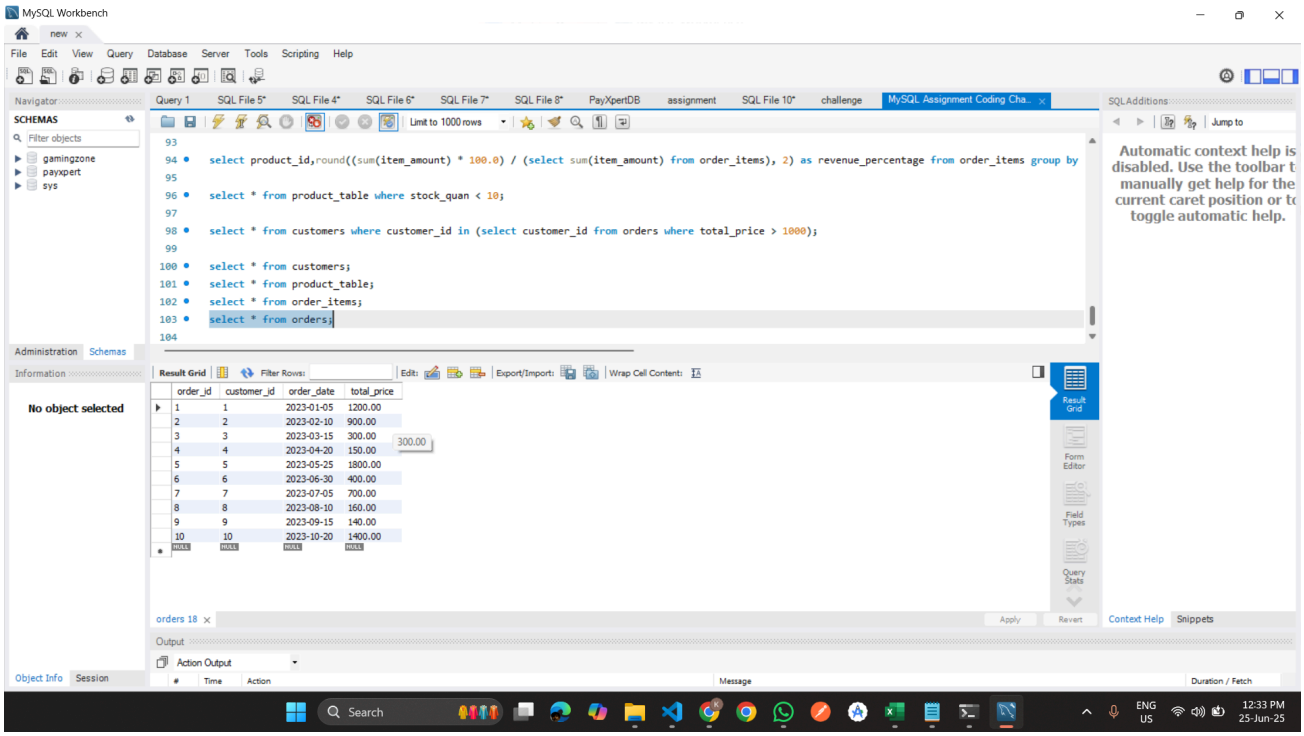
insert into orders values (6, 6, '2023-06-30', 400.00);

insert into orders values (7, 7, '2023-07-05', 700.00);

insert into orders values (8, 8, '2023-08-10', 160.00);

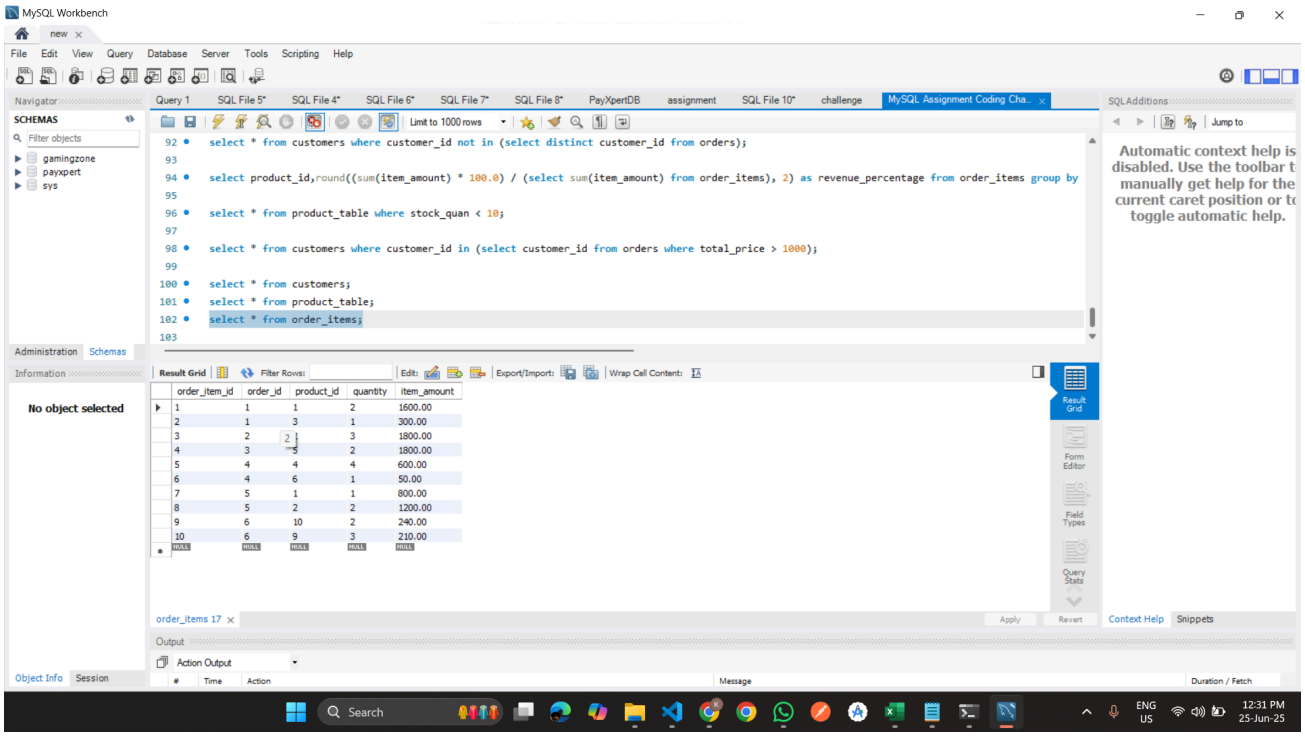
insert into orders values (9, 9, '2023-09-15', 140.00);

insert into orders values (10, 10, '2023-10-20', 1400.00);



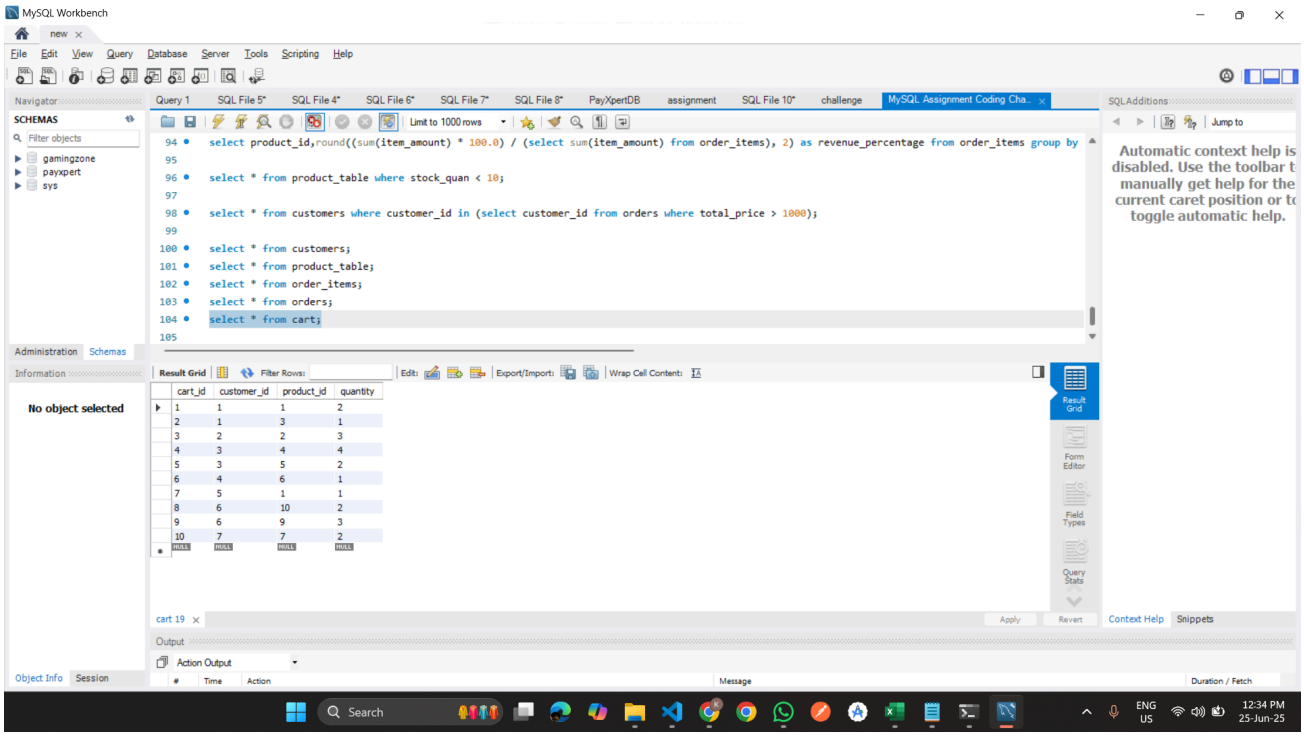
Order Items Table:

```
create table order_items (order_item_id int primary key,order_id int,product_id int,quantity int,item_amount decimal(10,2),foreign key (order_id) references orders(order_id),foreign key (product_id) references product_table(product_id));
insert into order_items values (1, 1, 1, 2, 1600.00);
insert into order_items values (2, 1, 3, 1, 300.00);
insert into order_items values (3, 2, 2, 3, 1800.00);
insert into order_items values (4, 3, 5, 2, 1800.00);
insert into order_items values (5, 4, 4, 4, 600.00);
insert into order_items values (6, 4, 6, 1, 50.00);
insert into order_items values (7, 5, 1, 1, 800.00);
insert into order_items values (8, 5, 2, 2, 1200.00);
insert into order_items values (9, 6, 10, 2, 240.00);
insert into order_items values (10, 6, 9, 3, 210.00);
```



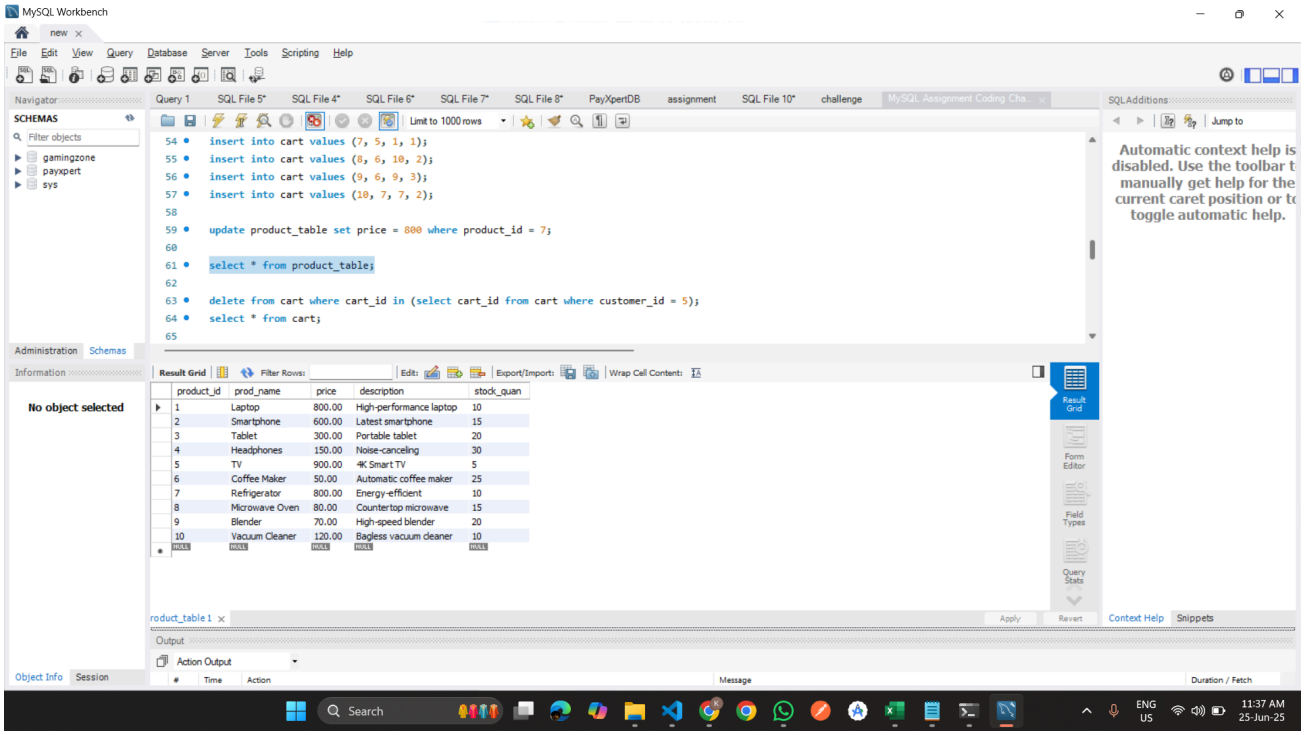
Cart Table:

```
create table cart (cart_id int primary key,customer_id int,product_id int,quantity int,foreign key (customer_id) references customers(customer_id),foreign key (product_id) references product_table(product_id));
insert into cart values (1, 1, 1, 2);
insert into cart values (2, 1, 3, 1);
insert into cart values (3, 2, 2, 3);
insert into cart values (4, 3, 4, 4);
insert into cart values (5, 3, 5, 2);
insert into cart values (6, 4, 6, 1);
insert into cart values (7, 5, 1, 1);
insert into cart values (8, 6, 10, 2);
insert into cart values (9, 6, 9, 3);
insert into cart values (10, 7, 7, 2);
```



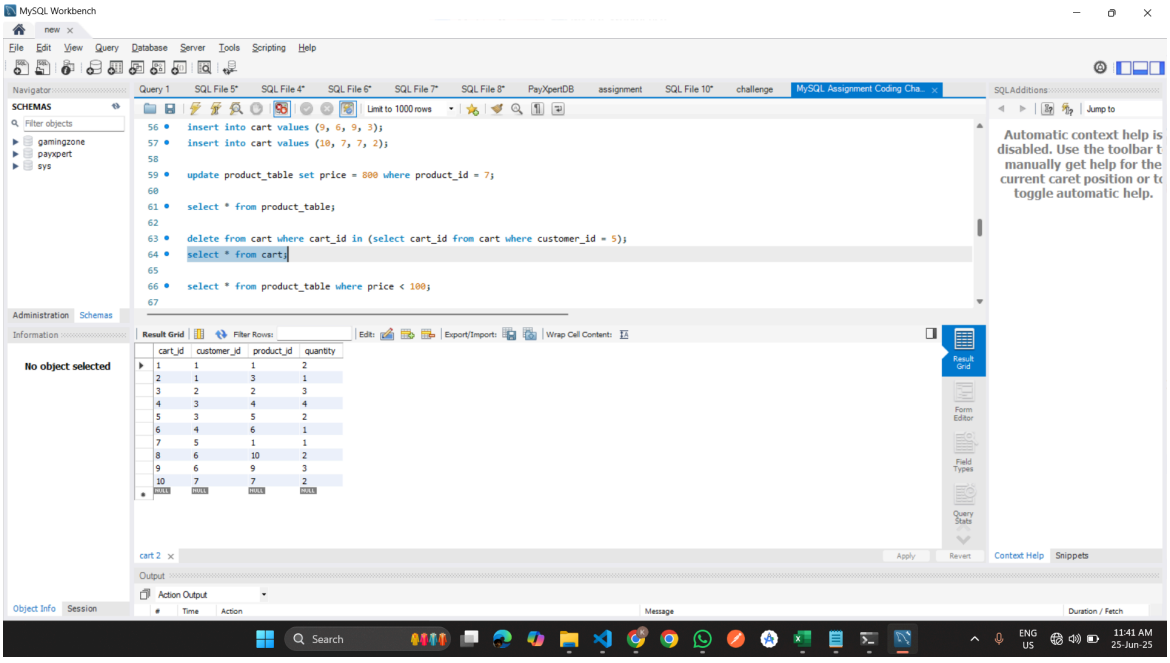
1.Update refrigerator product price to 800.

->update product_table set price = 800 where product_id = 7;



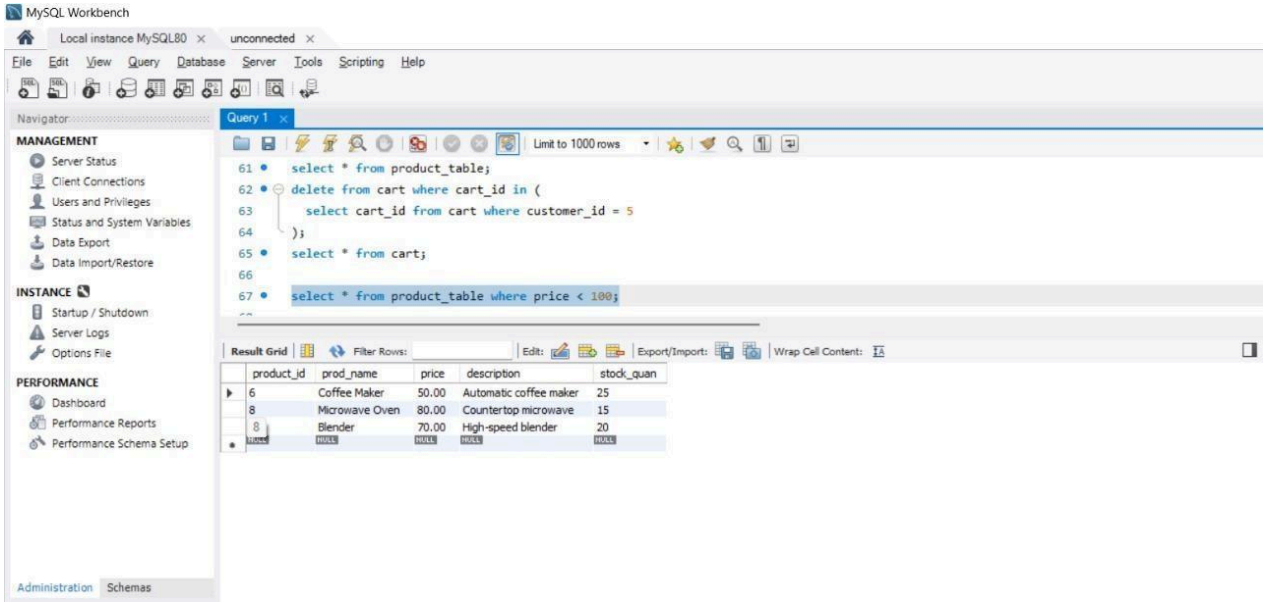
2.Remove all cart items for a specific customer

->delete from cart where cart_id in (select cart_id from cart where customer_id = 5);



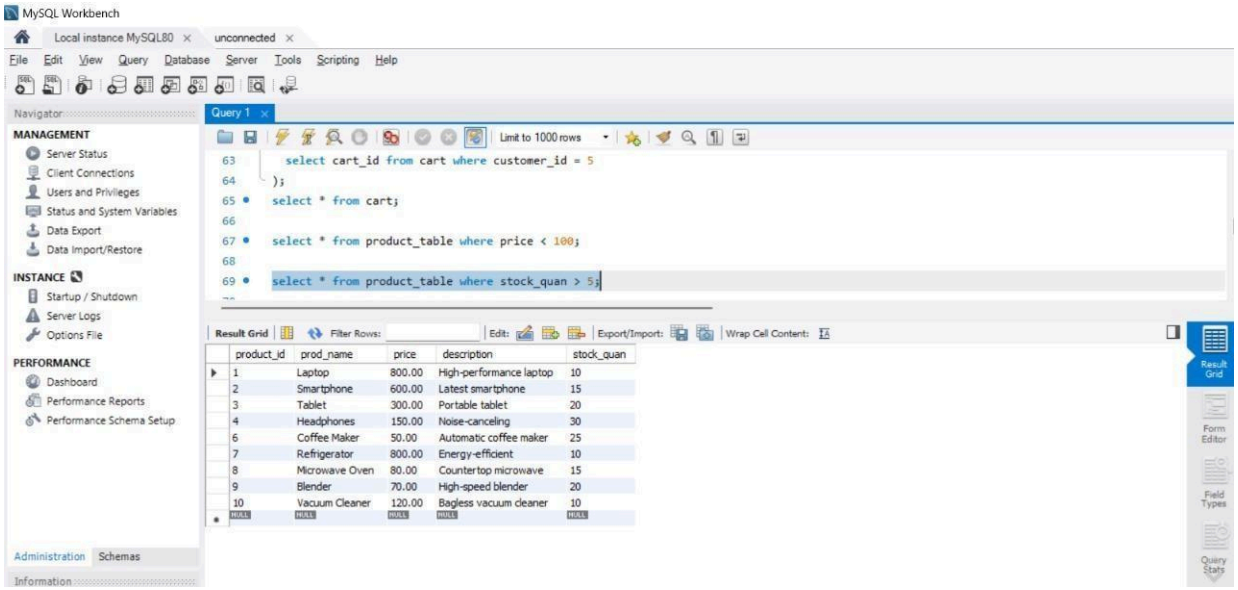
3.Retrieve Products Priced Below $100.

->select * from product_table where price < 100;



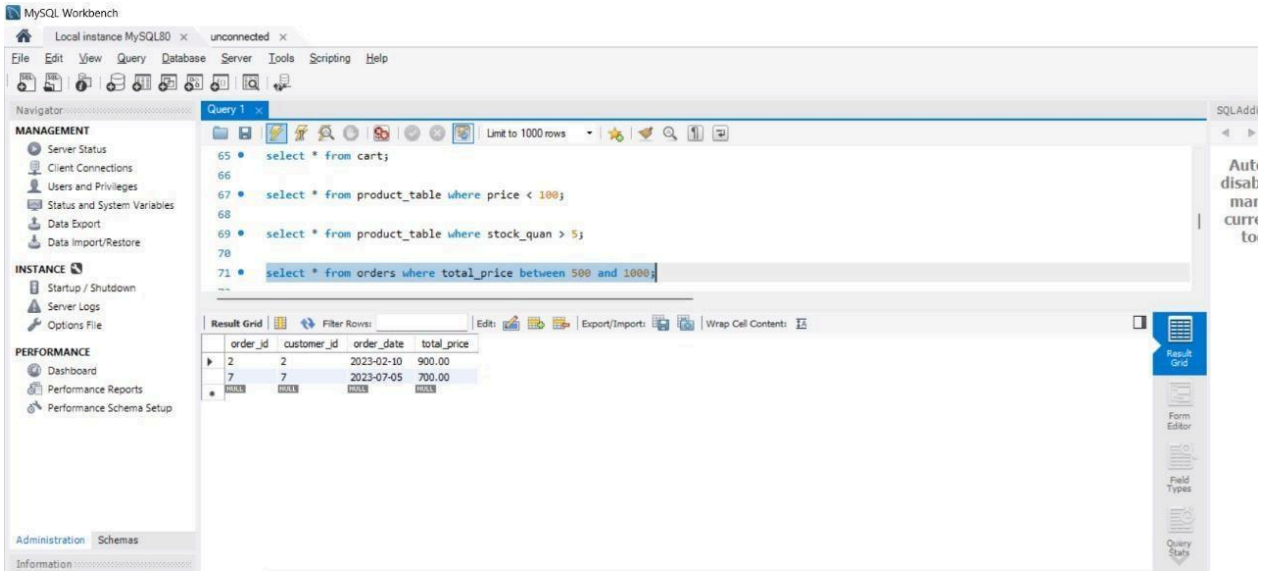
4.Find Products with Stock Quantity Greater Than 5.

->select * from product_table where stock_quan > 5;



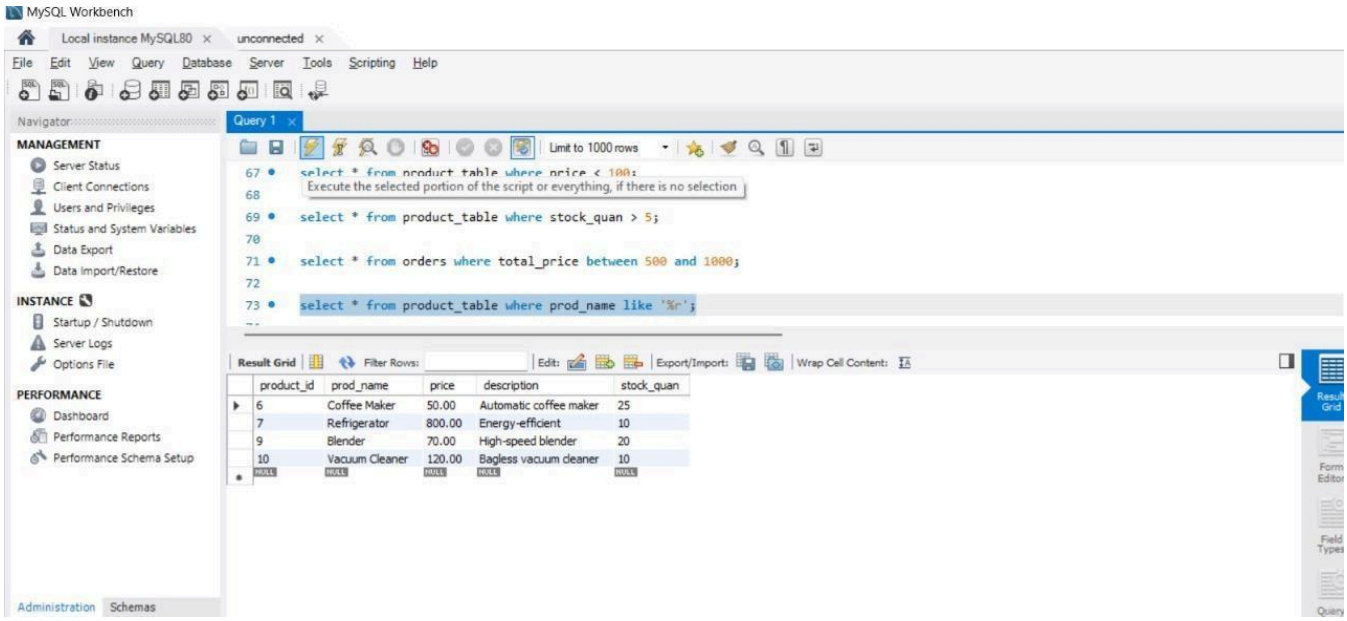
5)Retrieve Orders with Total Amount Between $500 and $1000.

->select * from orders where total_price between 500 and 1000;



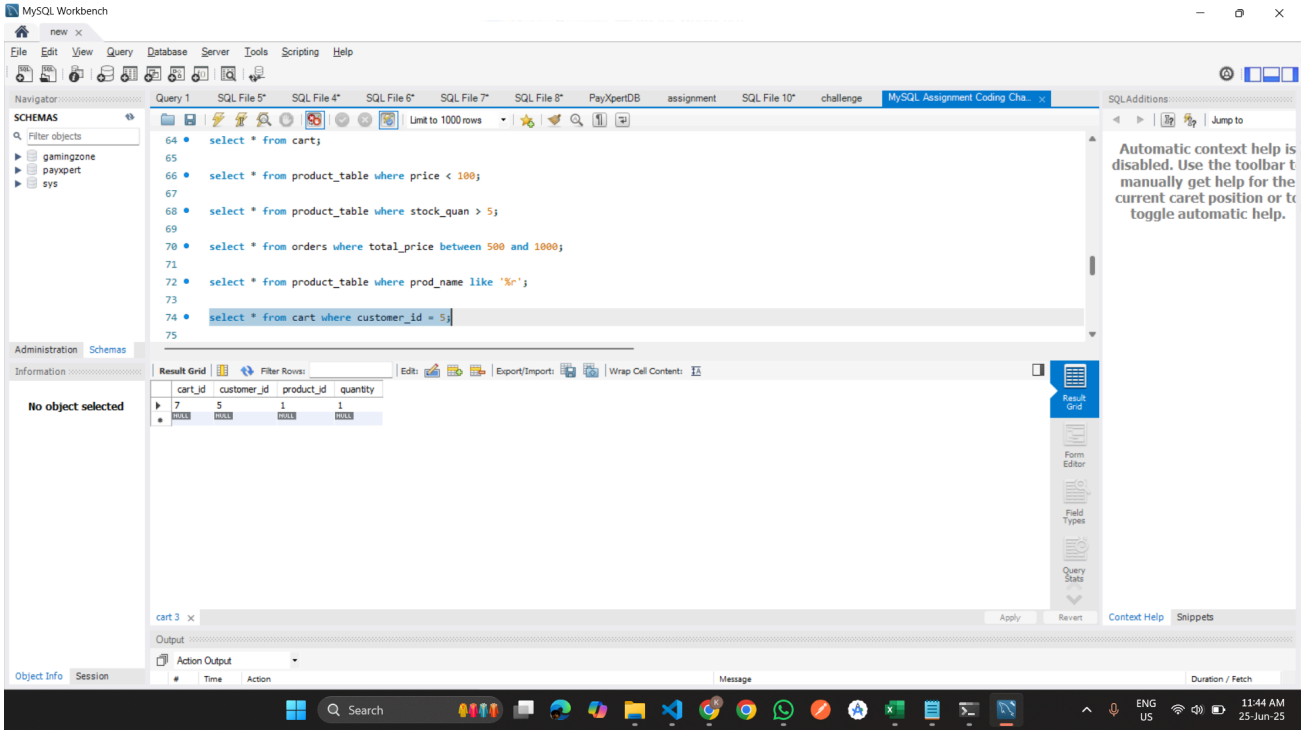
6)Find Products whose name ends with the letter ‘r’.

->select * from product_table where prod_name like '%r';



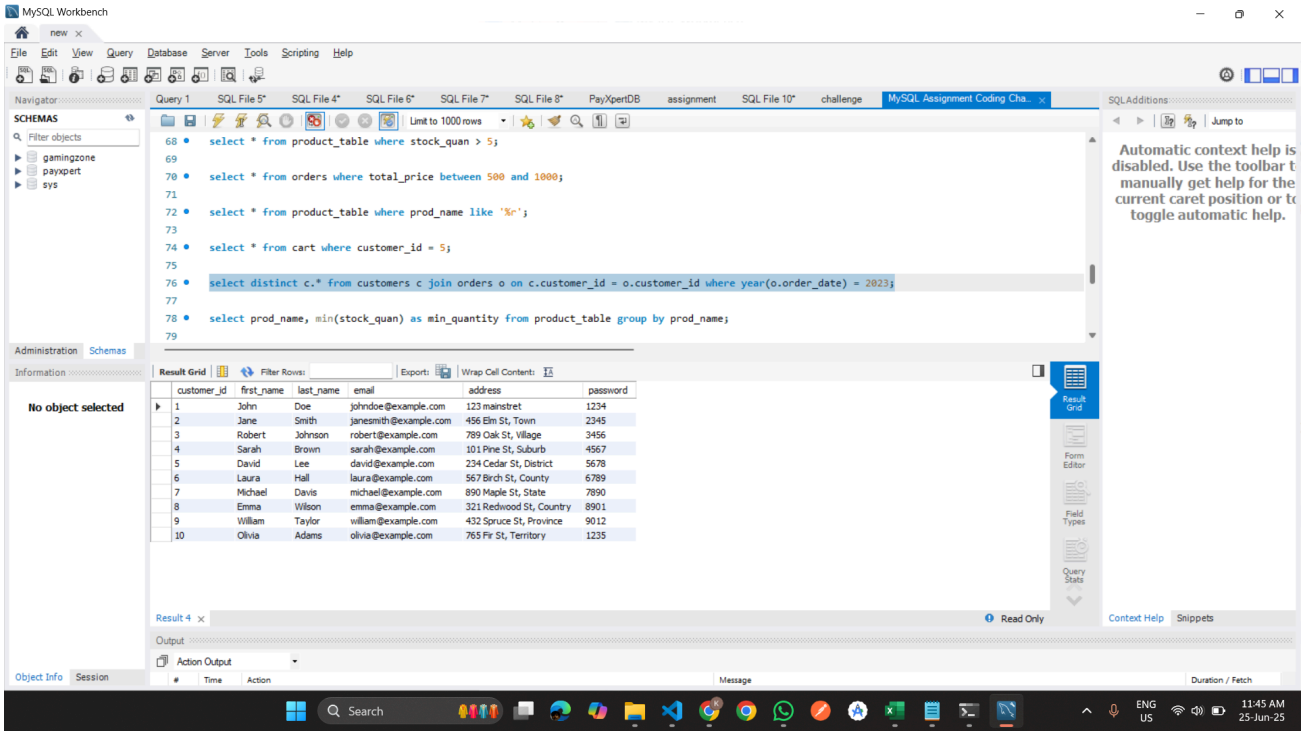
7)Retrieve Cart Items for Customer 5.

->select * from cart where customer_id = 5;



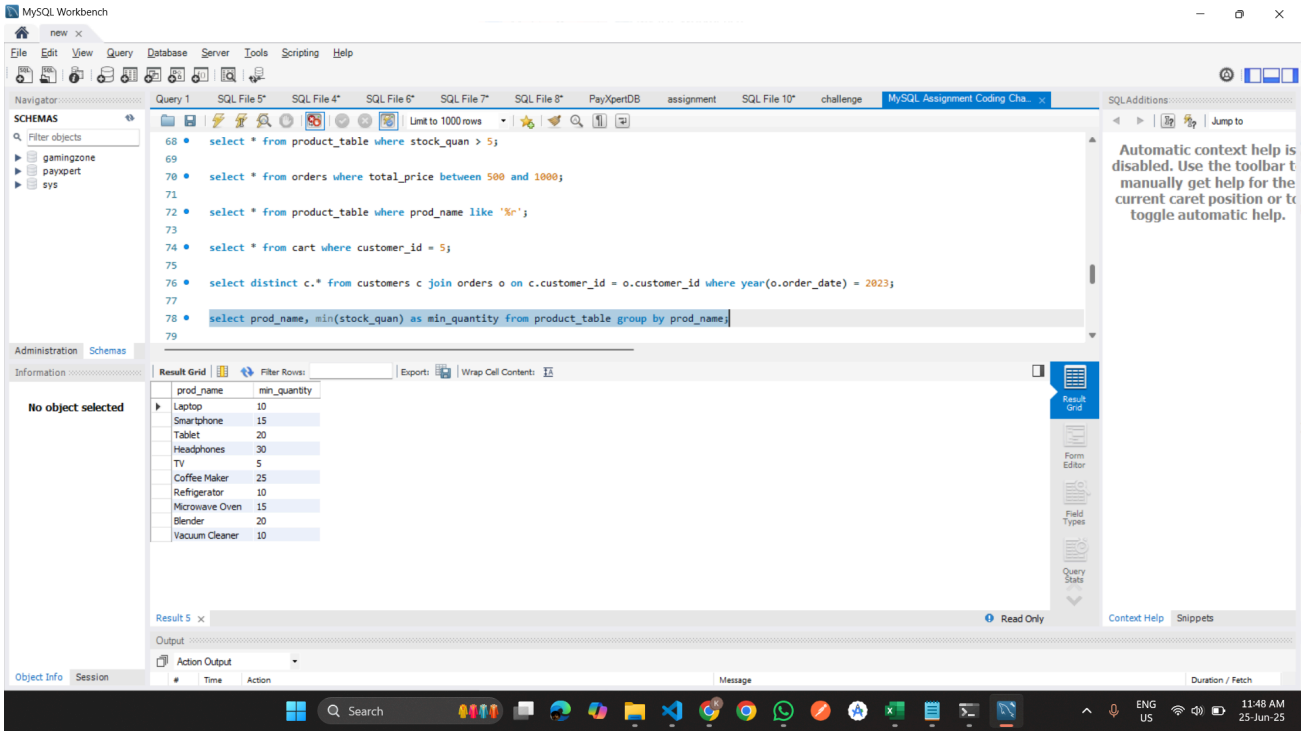
8)Find Customers Who Placed Orders in 2023.

->select distinct c.* from customers c join orders o on c.customer_id = o.customer_id where year(o.order_date) = 2023;



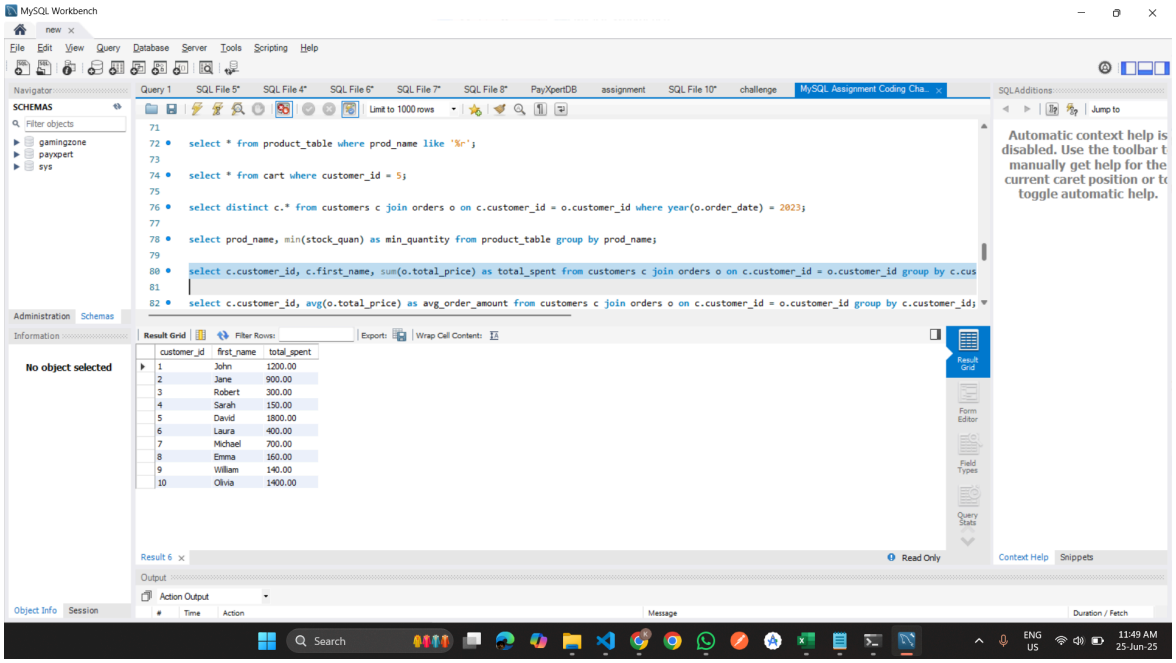
9)Determine the Minimum Stock Quantity for Each Product Category.

->select prod_name, min(stock_quan) as min_quantity from product_table group by prod_name;



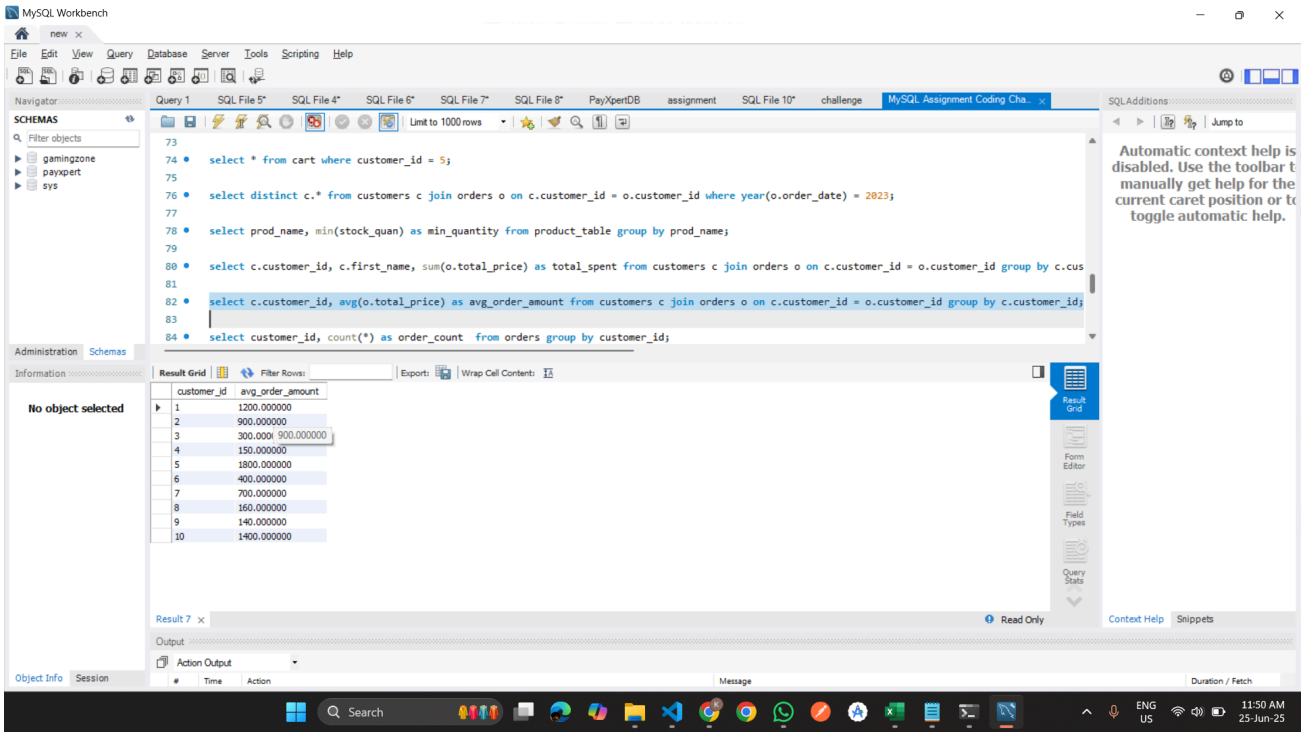
10)Calculate the Total Amount Spent by Each Customer.

->select c.customer_id, c.first_name, sum(o.total_price) as total_spenfrom customers c join orders o on c.customer_id = o.customer_id group by c.customer_id;



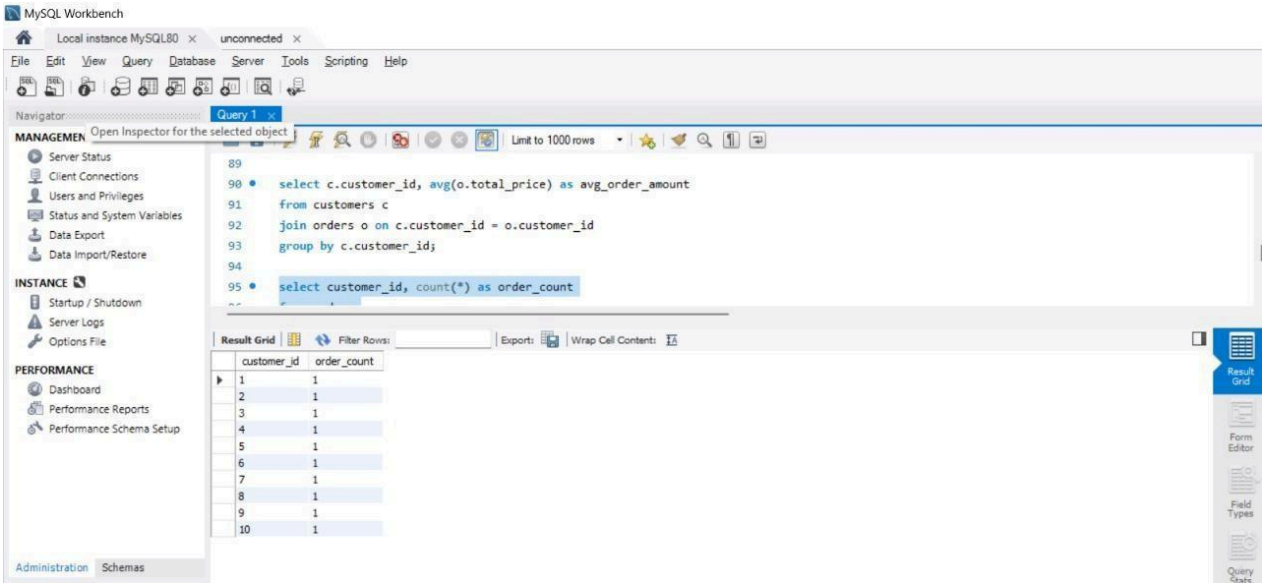
11)Find the Average Order Amount for Each Customer.

->select c.customer_id, avg(o.total_price) as avg_order_amount from customers c join orders o on c.customer_id = o.customer_id group by c.customer_id;



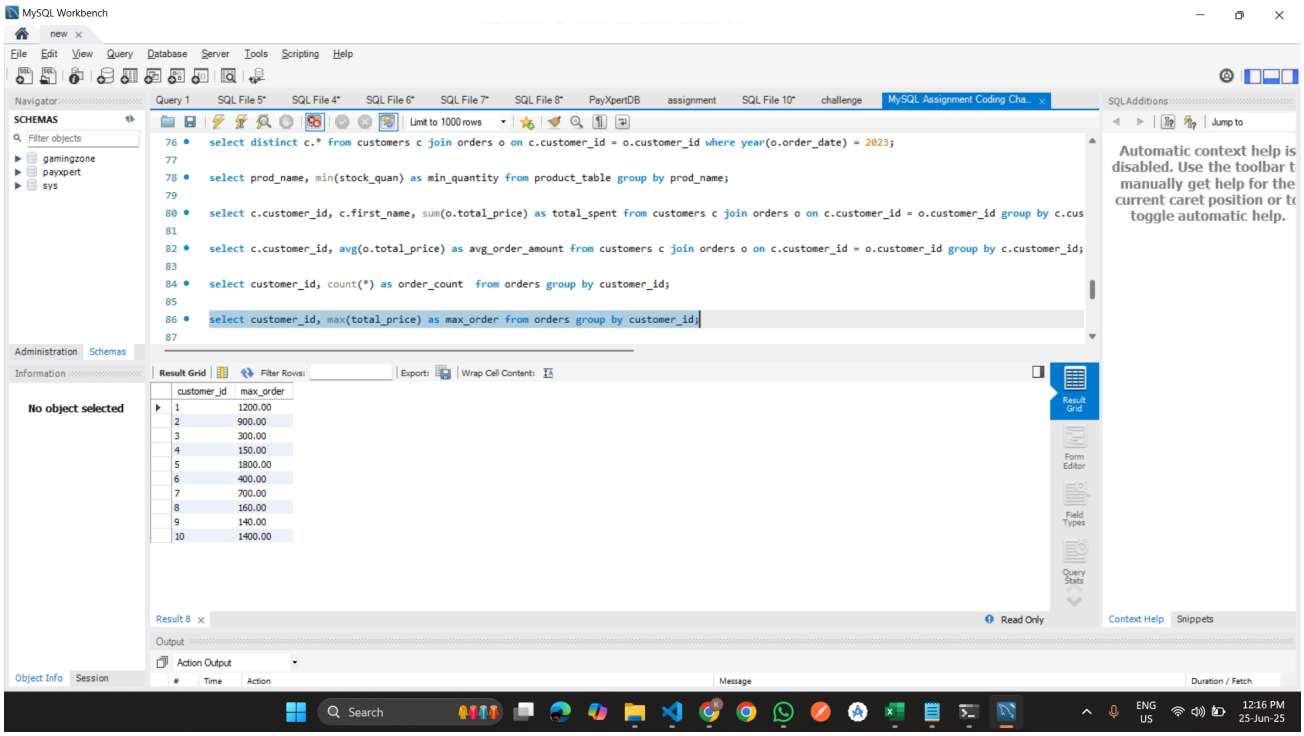
12)Count the Number of Orders Placed by Each Customer.

->select customer_id, count(*) as order_count from orders group by customer_id;



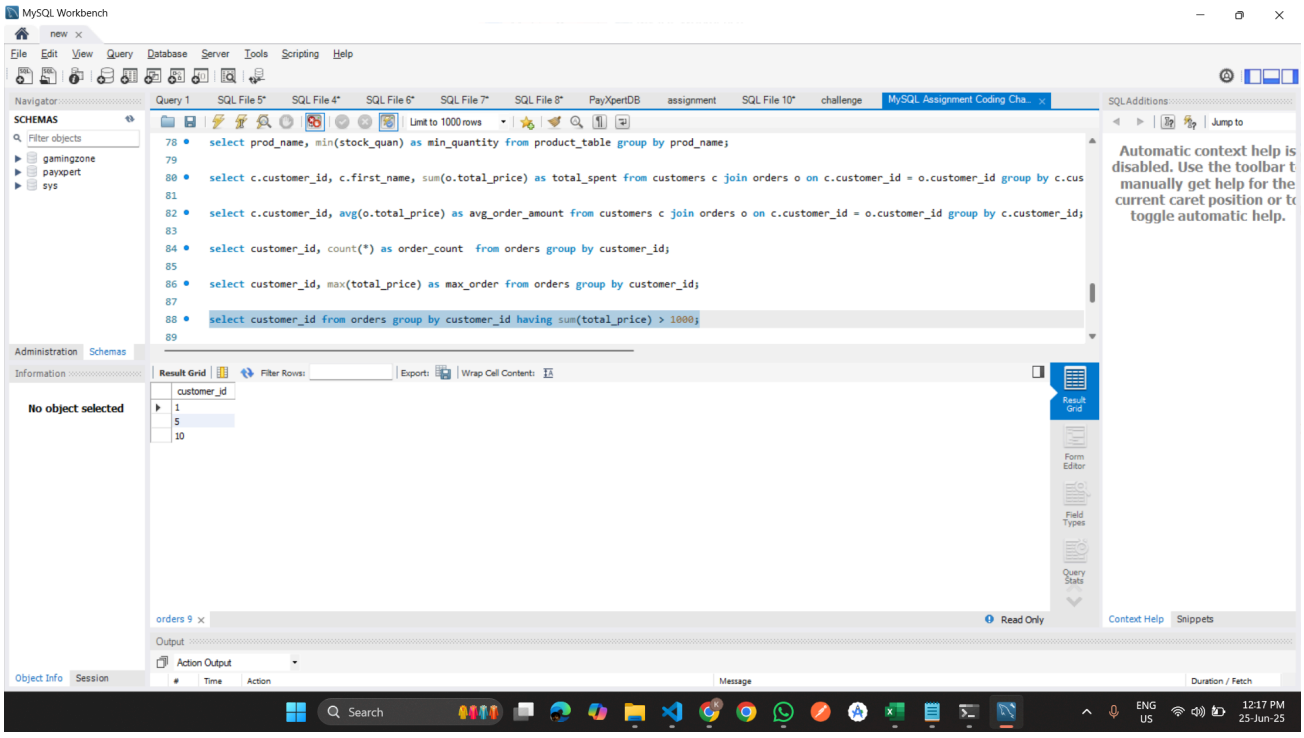
13)Find the Maximum Order Amount for Each Customer.

->select customer_id, max(total_price) as max_order from orders group by customer_id;



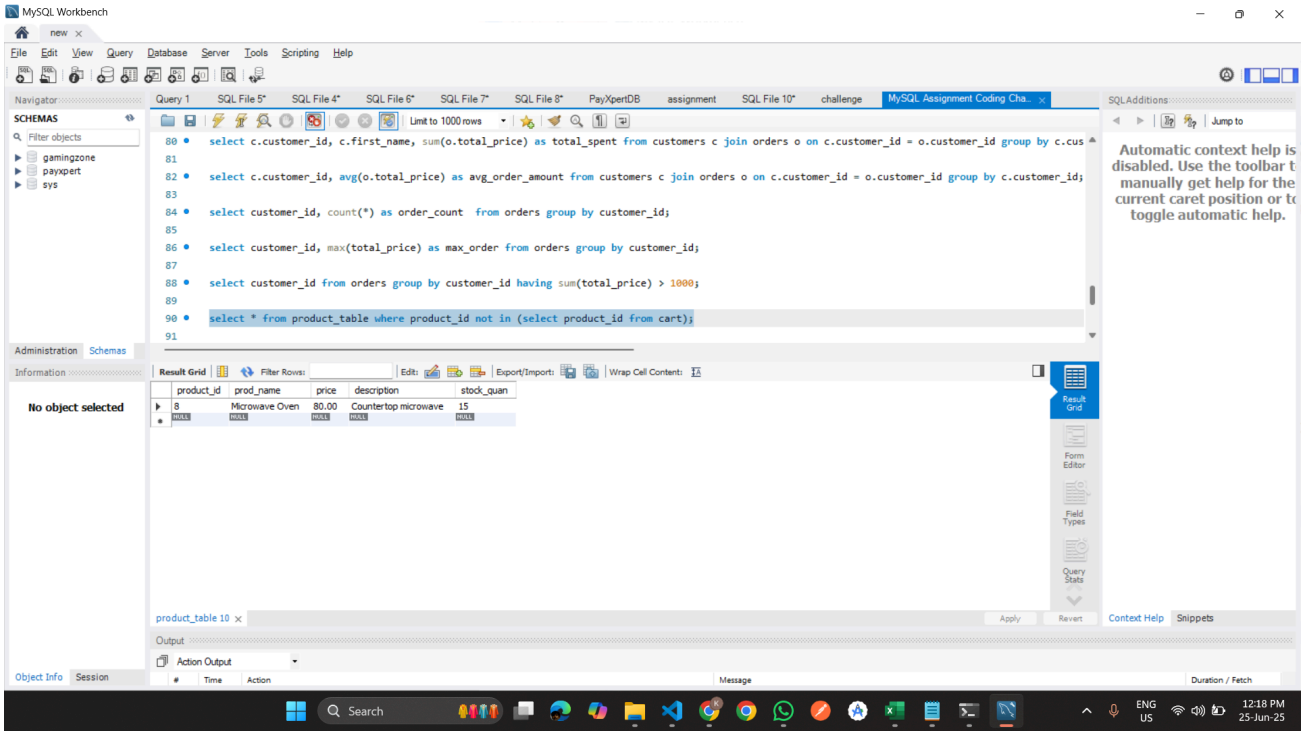
14)Get Customers Who Placed Orders Totaling Over $1000.

->select customer_id from orders group by customer_id having sum(total_price) > 1000;



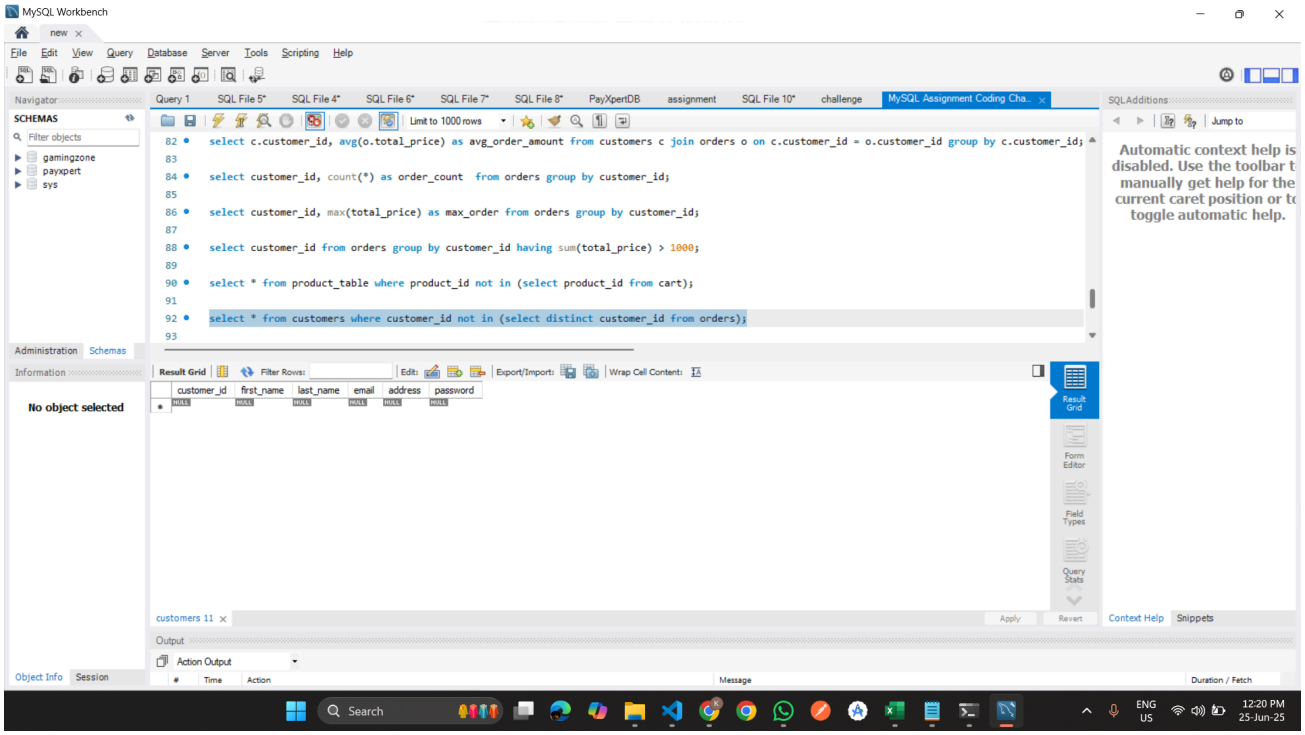
15)Subquery to Find Products Not in the Cart.

->select * from product_table where product_id not in (select product_id from cart);



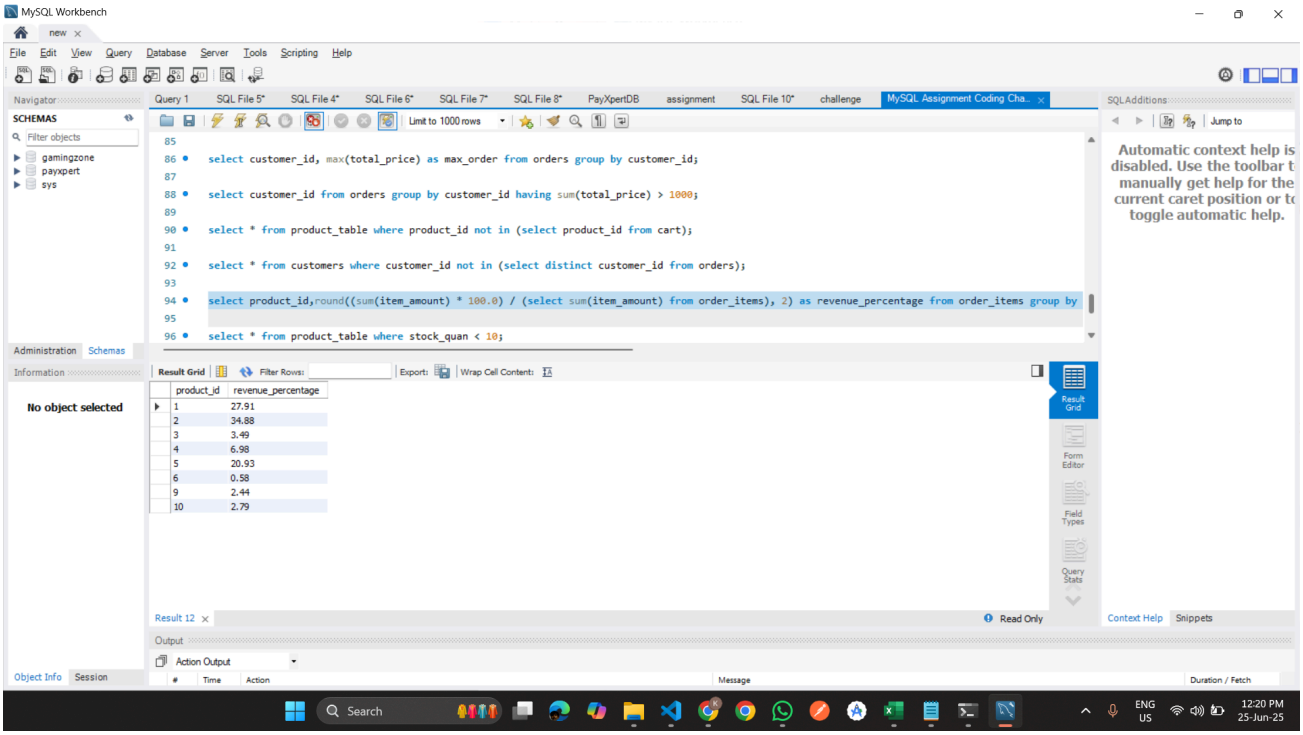
16)Subquery to Find Customers Who Haven't Placed Orders.

->select * from customers where customer_id not in (select distinct customer_id from orders);



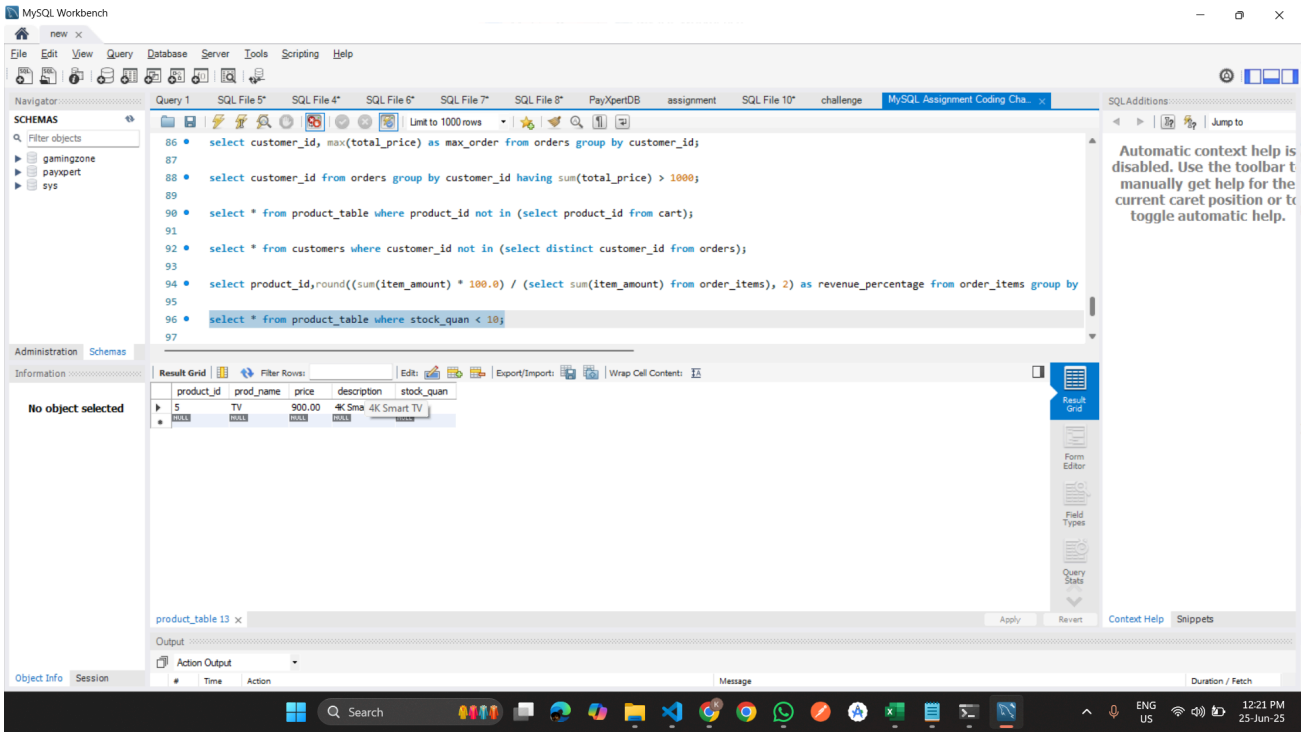
17)Subquery to Calculate the Percentage of Total Revenue for a Product.

->select product_id,round((sum(item_amount) * 100.0) / (select sum(item_amount) from order_items), 2) as revenue_percentage from order_items group by product_id;



18)Subquery to Find Products with Low Stock.

->select * from product_table where stock_quan < 10;



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

->select * from customers where customer_id in (select customer_id from orders where total_price > 1000);