**Name : Sarthak Shandilya**

**Coding Challenge**

**Paper Solved :- Coding Challenge 6**

SQL Tables:

1. customers table:

• customer\_id (Primary Key)

• name

• email

• password

1. products table:

• product\_id (Primary Key)

• name

• price

• description

• stockQuantity

1. cart table:

• cart\_id (Primary Key)

• customer\_id (Foreign Key)

• product\_id (Foreign Key)

• quantity

1. orders table:

• order\_id (Primary Key)

• customer\_id (Foreign Key)

• order\_date

• total\_price

• shipping\_address

1. order\_items table (to store order details):

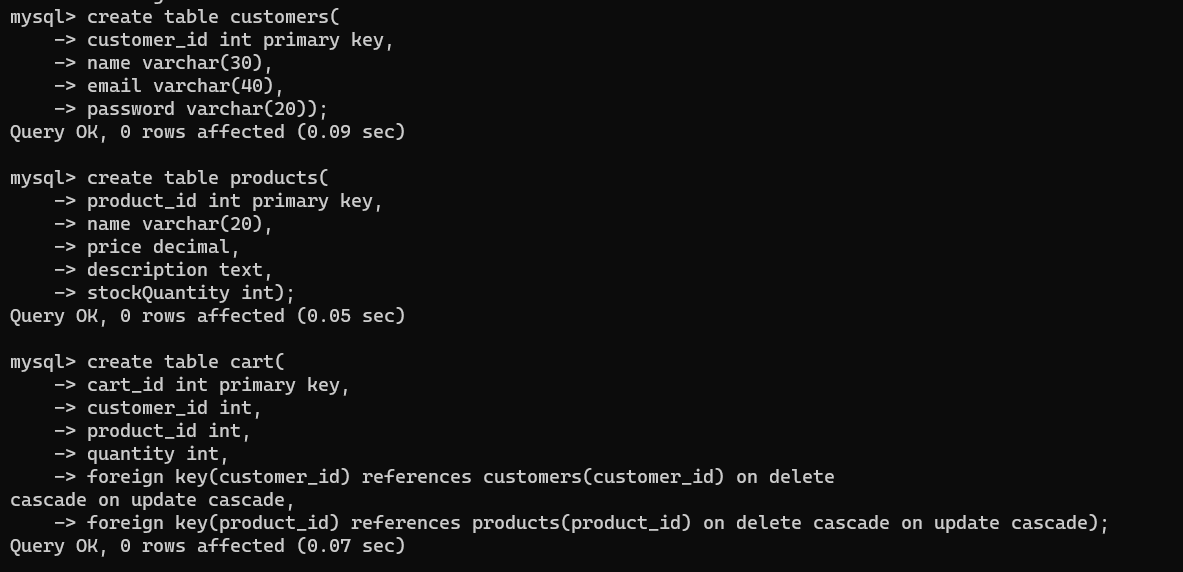
• order\_item\_id (Primary Key)

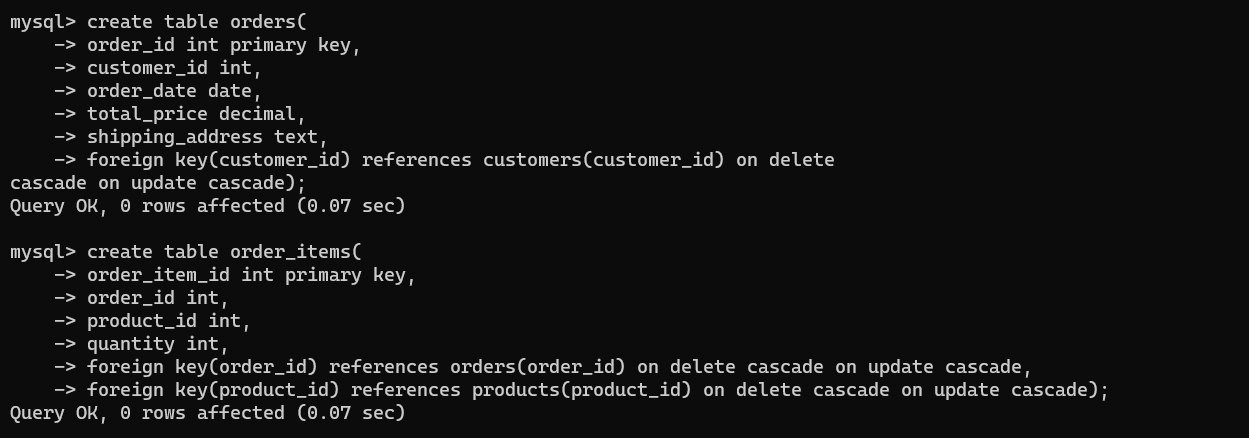
• order\_id (Foreign Key)

• product\_id (Foreign Key)

• quantity

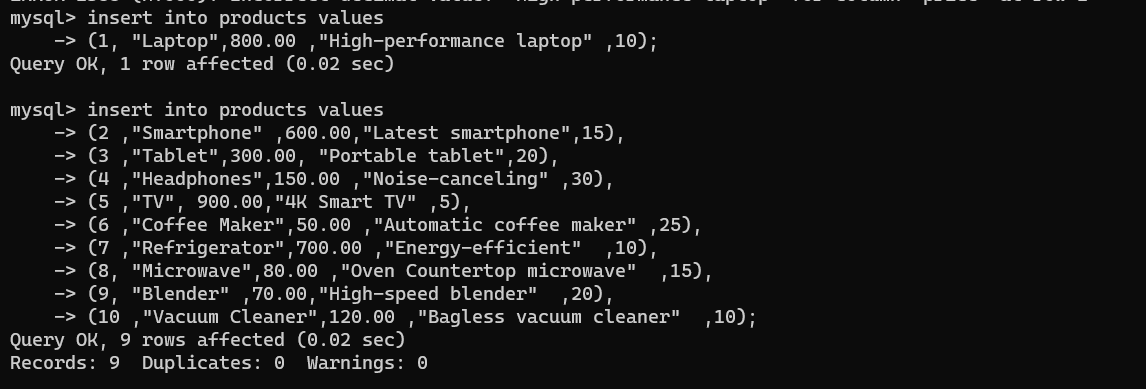
Created Tables:



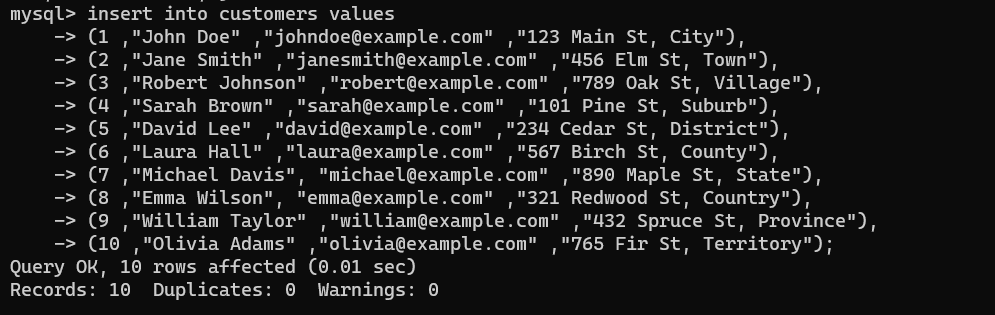


Records Inserted:

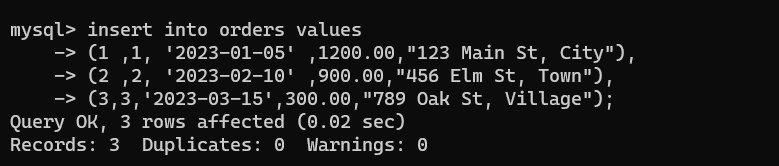
Products :-

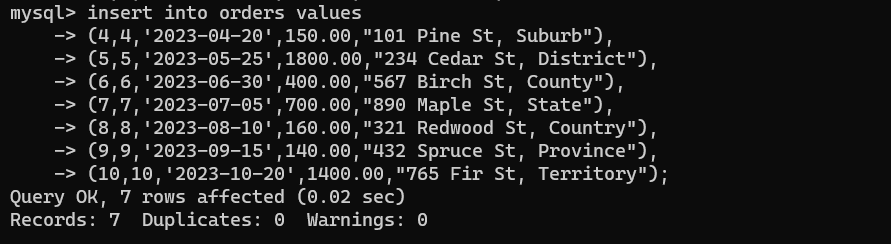


Customers:-

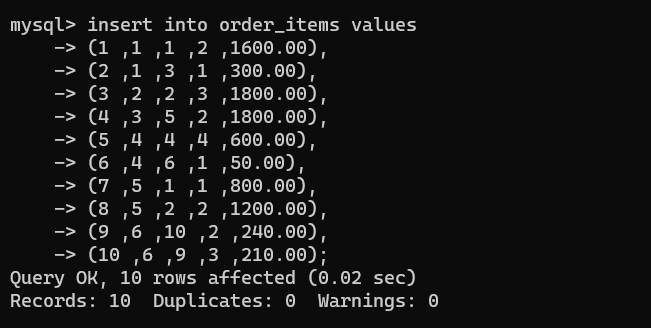


Orders:-

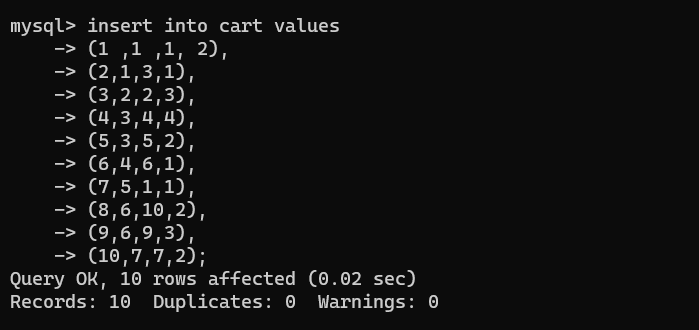




Order\_items:-

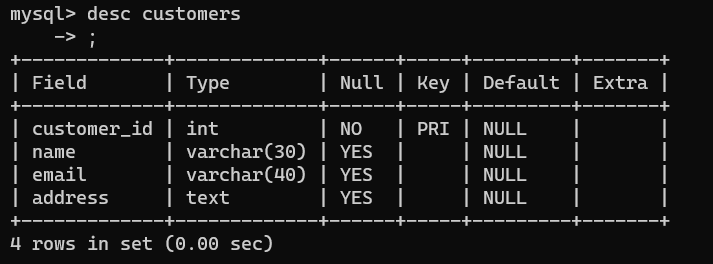


Cart:-



**Descriptions of my tables:-**

Customers:

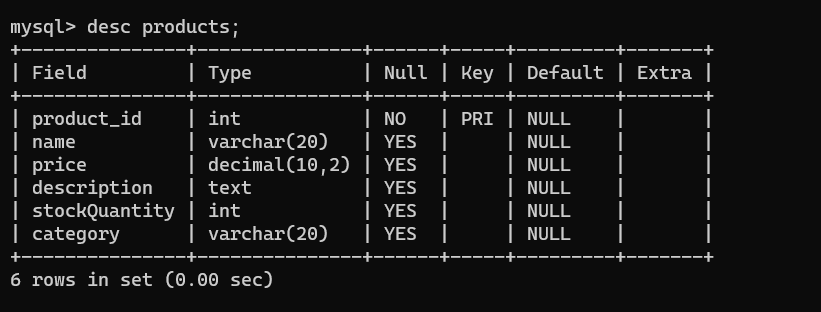
****

Products:

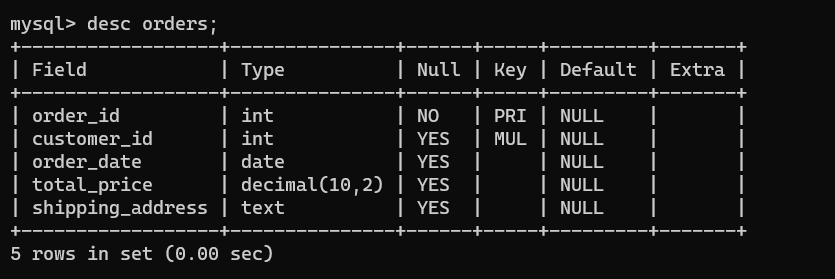
NOTE: I’ve later on added a category section in product table as it was required in a question.

I used the query as below:

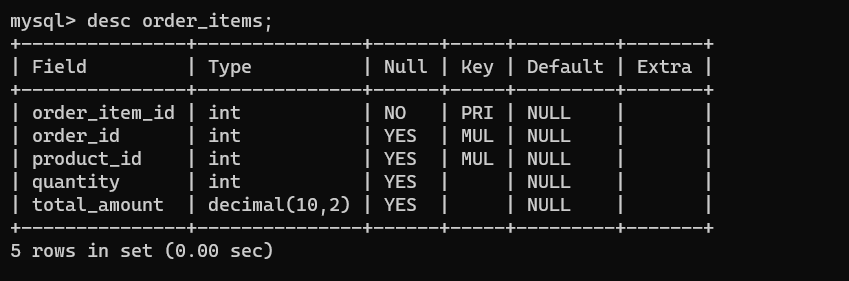
ALTER TABLE products ADD COLUMN category varchar(20);



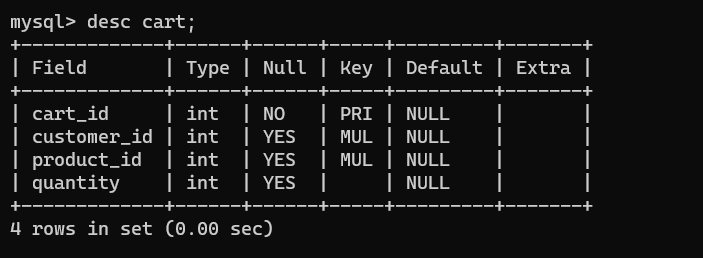
Orders:



Order\_items:

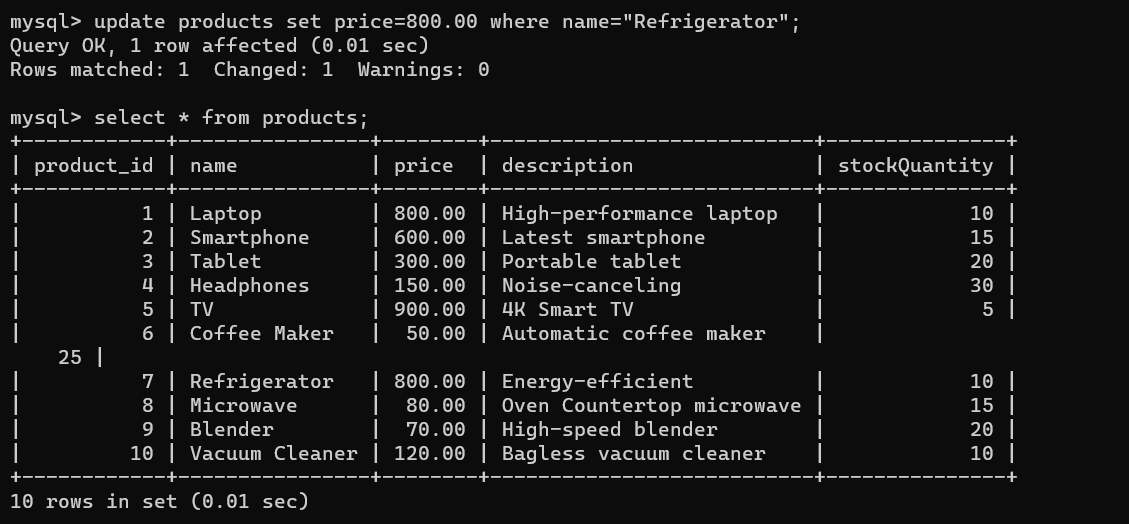


Cart:

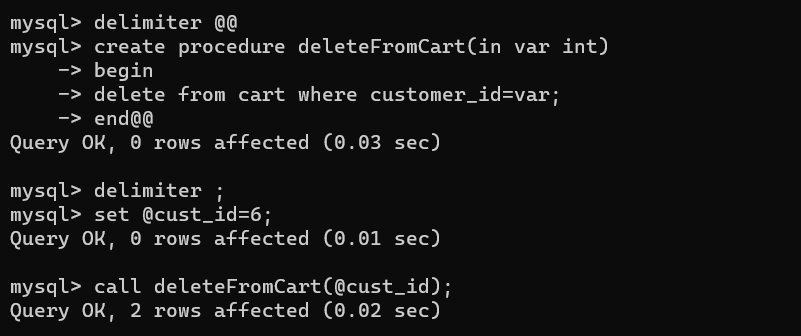


**Questions and Solutions:-**

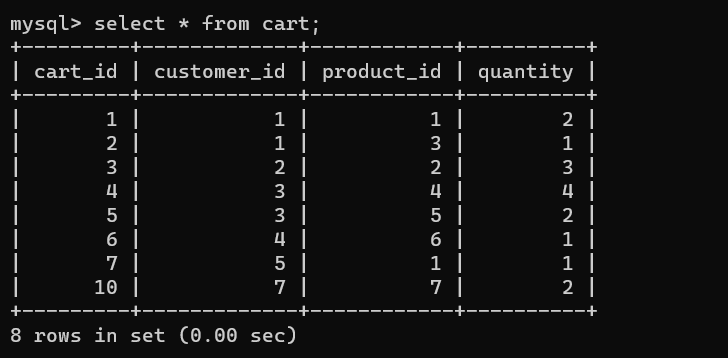
1. Update refrigerator product price to 800.



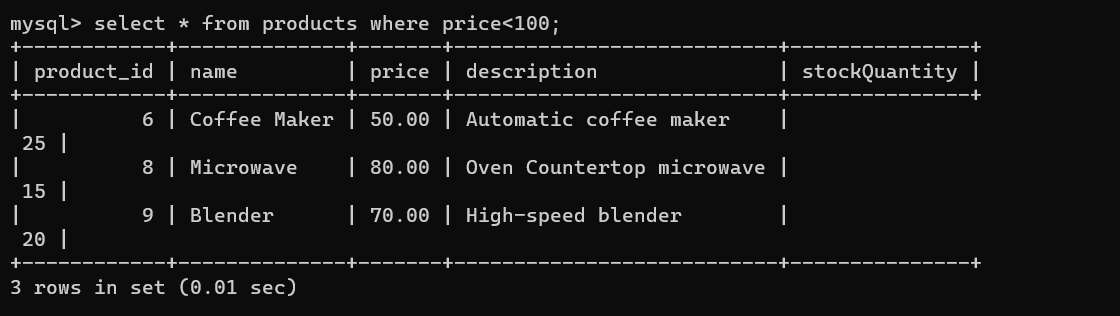
1. Remove all cart items for a specific customer.



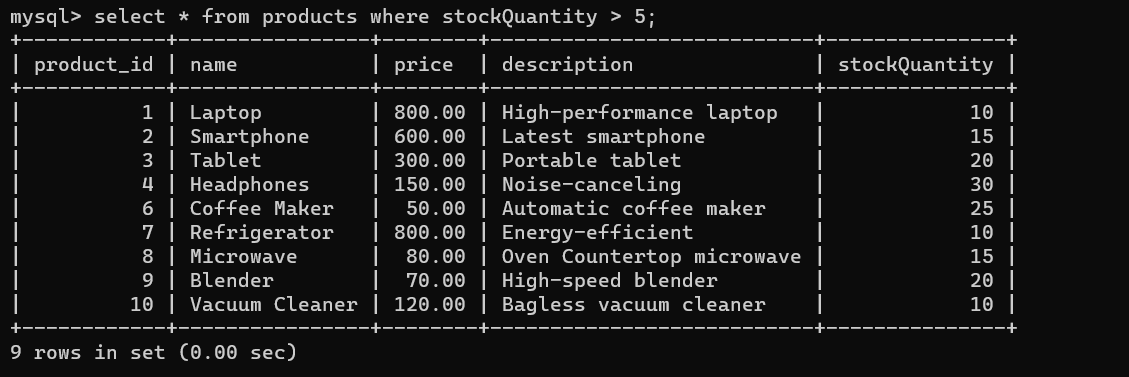
Cart item for customer with customer\_id = 6 deleted



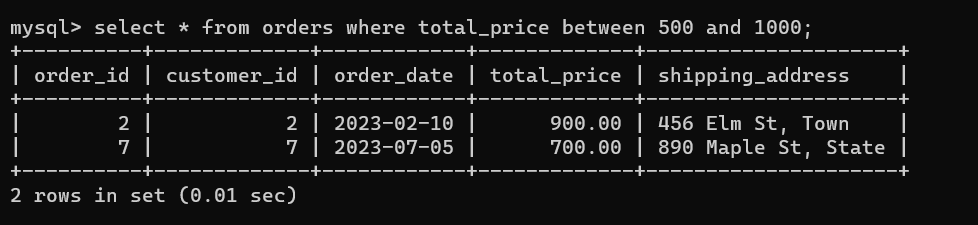
1. Retrieve Products Priced Below $100.



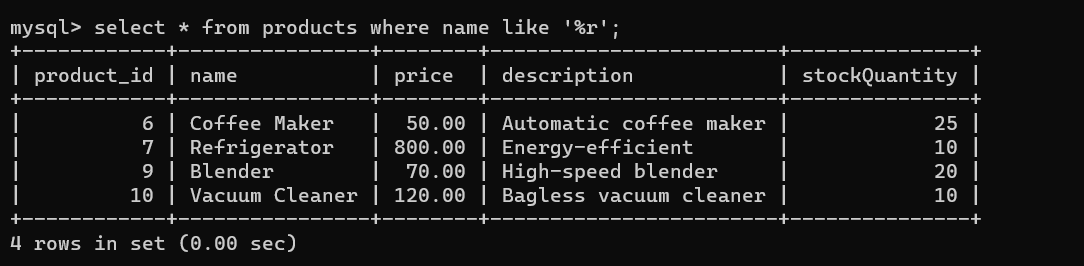
1. Find Products with Stock Quantity Greater Than 5.



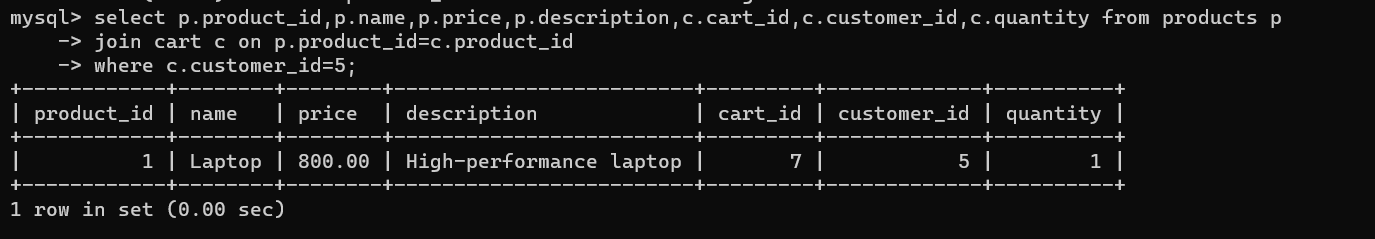
1. Retrieve Orders with Total Amount Between $500 and $1000.



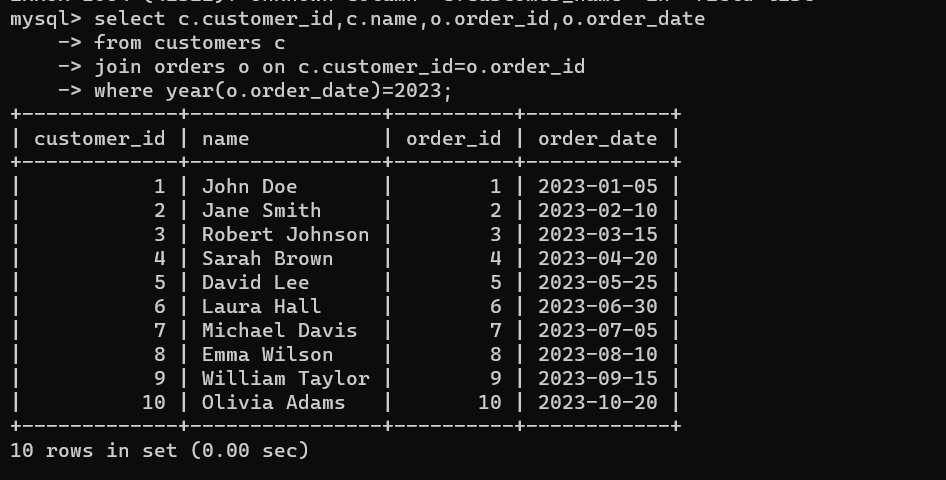
1. Find Products which name end with letter ‘r’.



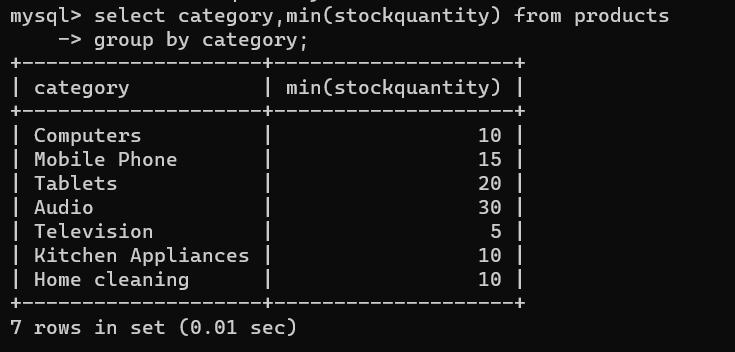
1. Retrieve Cart Items for Customer 5.



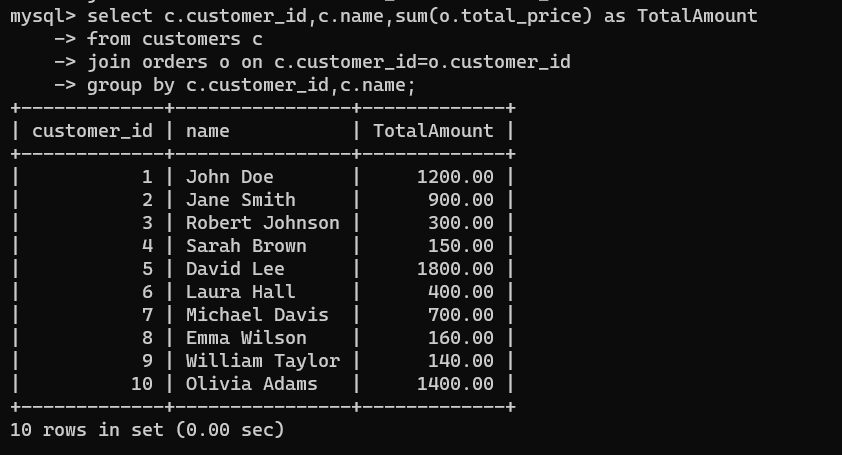
1. Find Customers Who Placed Orders in 2023.



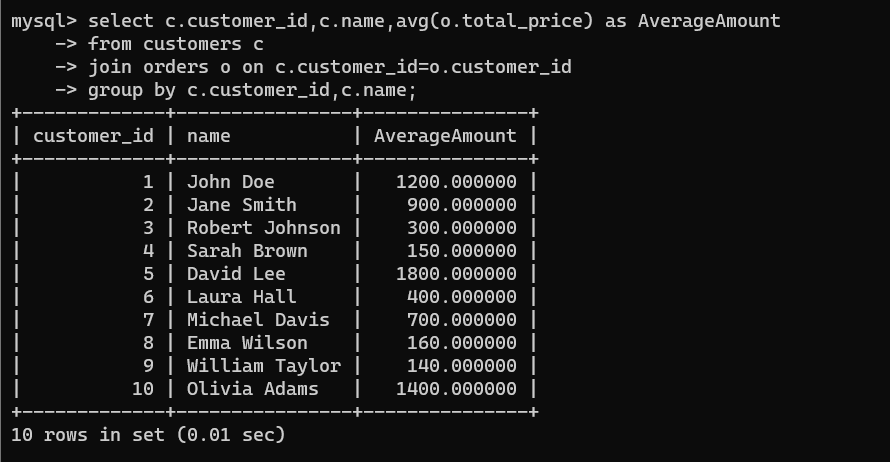
1. Determine the Minimum Stock Quantity for Each Product Category.



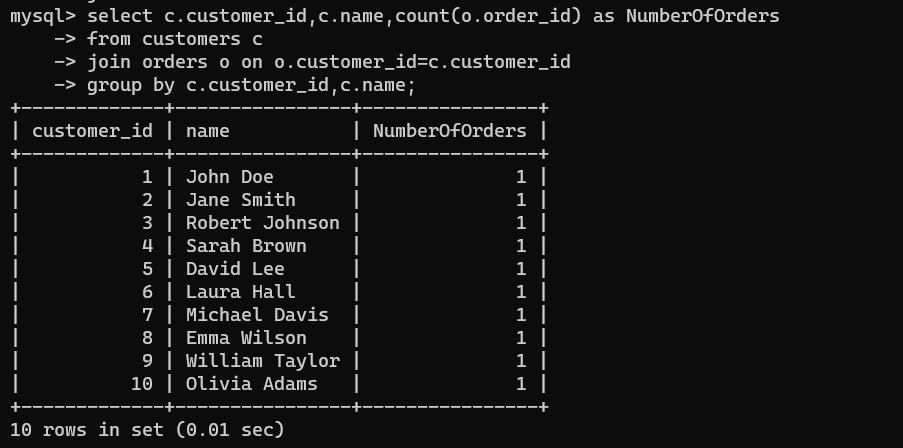
1. Calculate the Total Amount Spent by Each Customer.



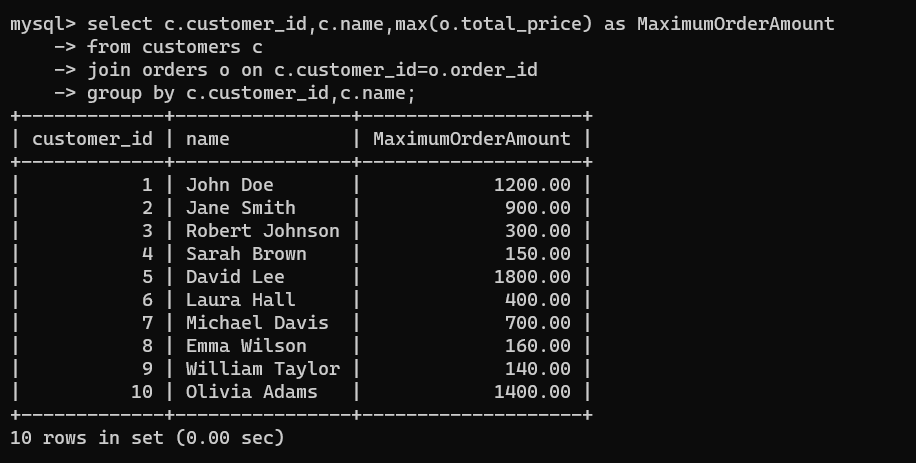
1. Find the Average Order Amount for Each Customer.



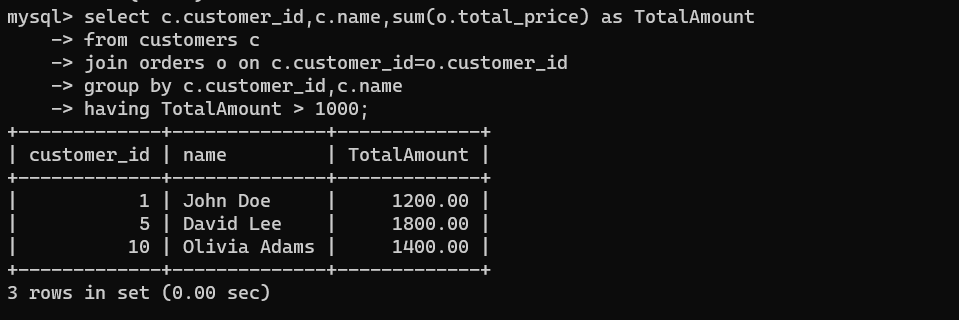
1. Count the Number of Orders Placed by Each Customer.



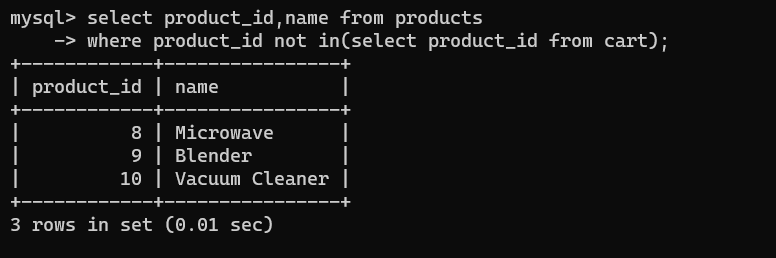
1. Find the Maximum Order Amount for Each Customer.



1. Get Customers Who Placed Orders Totaling Over $1000.



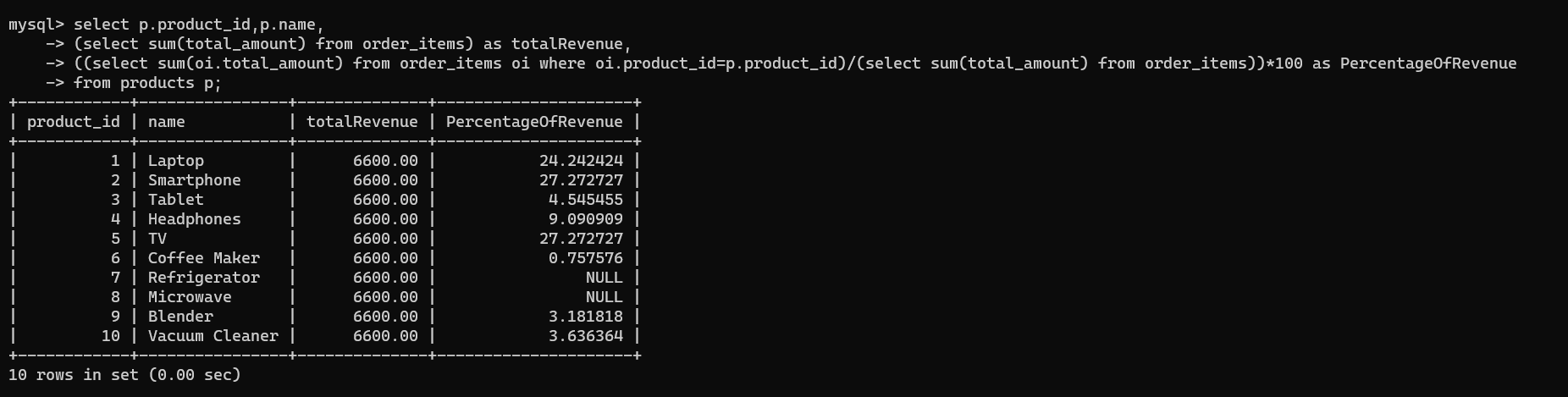
1. Subquery to Find Products Not in the Cart.



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

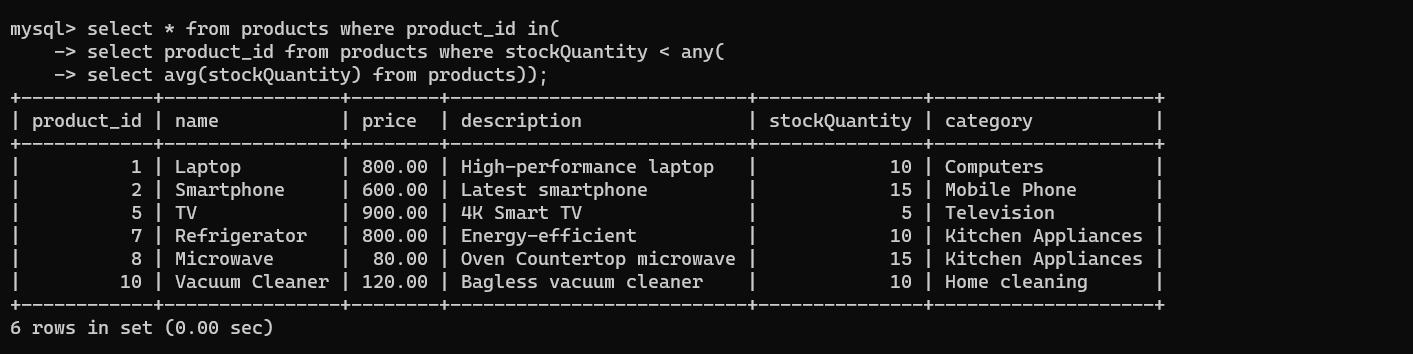


1. Subquery to Calculate the Percentage of Total Revenue for a Product.



1. Subquery to Find Products with Low Stock.

**Note:- I’ve considered a product’s stock to be low if it’s available stock quantity is less than total average stock quanity**



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

