Assignment 1------------

create database TechShop;

use TechShop;

create table Customer(CustomerID int primary key,

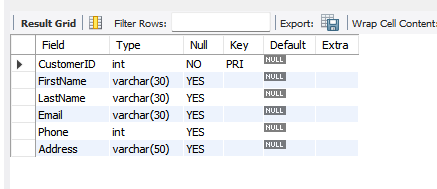
FirstName varchar(30),

LastName varchar(30),

Email varchar(30),

Phone int,

Address varchar(50));

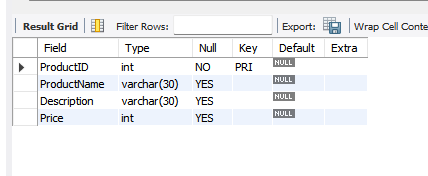


create table Products(ProductID int primary key,

ProductName varchar(30),

Description varchar(30),

Price int);

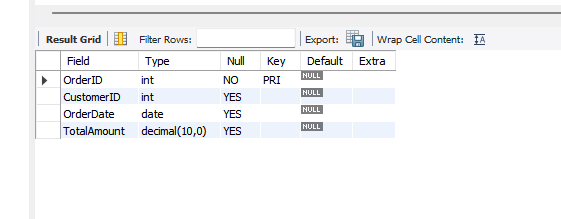


create table Orders(OrderID int primary key,

CustomerID int references Customer.CustomerID,

OrderDate date,

TotalAmount decimal);

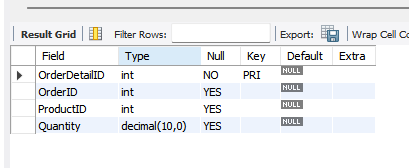


create table OrderDetails(OrderDetailID int primary key,

OrderID int references Orders.OrderID,

ProductID int references Products.ProductID,

Quantity decimal);

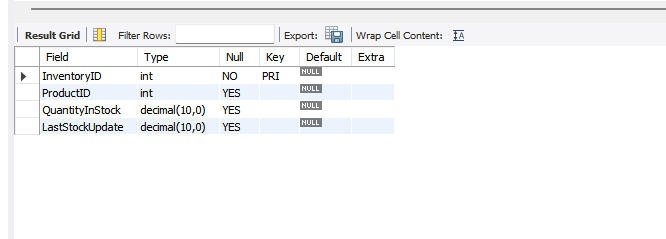


create table Inventory(InventoryID int primary key,

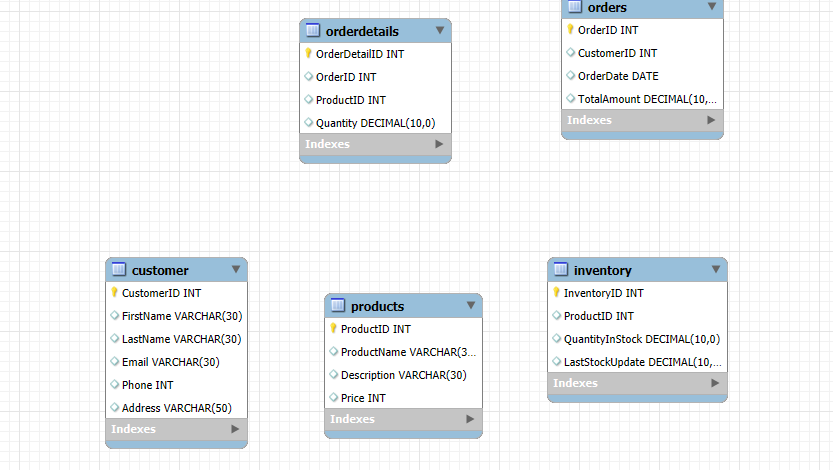
ProductID int references Products.ProductID,

QuantityInStock decimal,

LastStockUpdate decimal);



ER-diagram

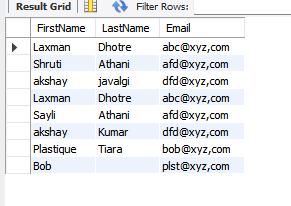


Task 2:

1. SELECT FirstName, LastName, Email

FROM Customer;

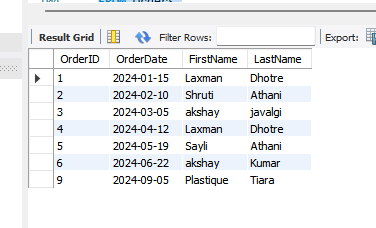
Output:



2. SELECT Orders.OrderID, Orders.OrderDate, Customer.FirstName, Customer.LastName

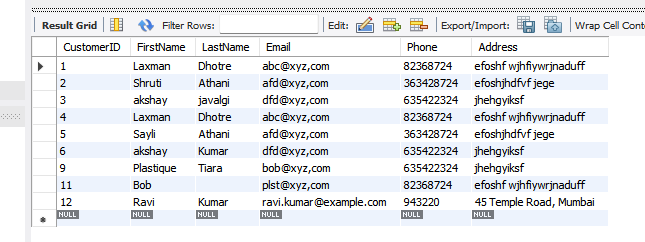
FROM Orders

JOIN Customer ON Orders.CustomerID = Customer.CustomerID;



3. INSERT INTO Customer (CustomerID, FirstName, LastName, Email, Phone, Address)

VALUES (12, 'Ravi', 'Kumar', 'ravi.kumar@example.com', 943220, '45 Temple Road, Mumbai');

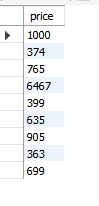
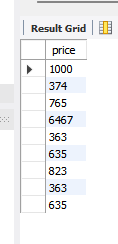


4. update Products

Set Price = Price \* 1.10

Where description LIKE '%electronic%';

Price before updating price after updating



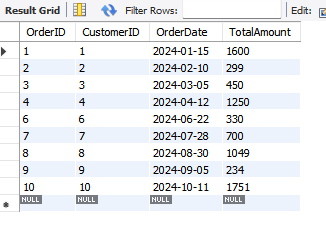
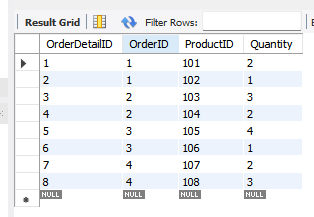
5. DELETE FROM OrderDetails

WHERE OrderID = 5;

DELETE FROM Orders

WHERE OrderID = 5;

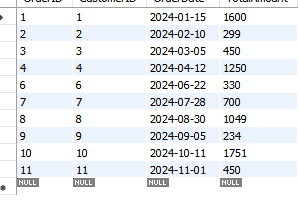
After Deletion:



6. INSERT INTO Orders (OrderID, CustomerID, OrderDate, TotalAmount)

VALUES (11, 11, '2024-11-01', 450);

Output after adding 11th order

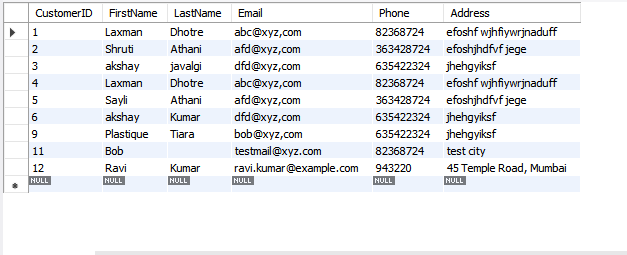


7. UPDATE Customer

SET Email = 'new.email@example.com', Address = 'New Address, City'

WHERE CustomerID = 11;

After updating 11th element:



8. UPDATE Orders

SET TotalAmount = (

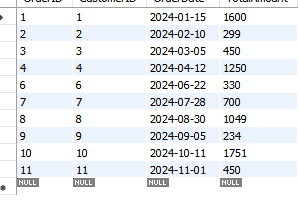
SELECT SUM(OD.Quantity \* P.Price)

FROM OrderDetails OD

JOIN Products P ON OD.ProductID = P.ProductID

WHERE OD.OrderID = Orders.OrderID

);



9. DELETE FROM OrderDetails

WHERE OrderID IN (

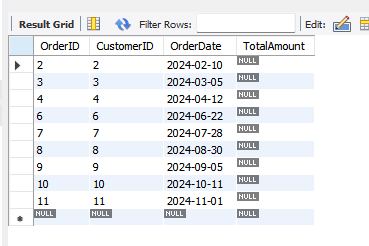
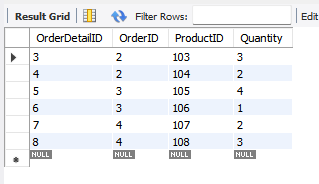
SELECT OrderID FROM Orders WHERE CustomerID = 1

);

DELETE FROM Orders

WHERE CustomerID = 1;

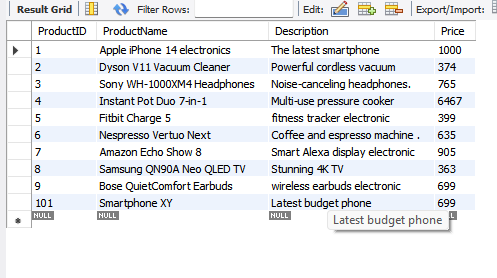
Tables after deleting order of customer 1



10. INSERT INTO Products (ProductID, ProductName, Description, Price)

VALUES (101, 'Smartphone XY', 'Latest budget phone', 699);

After adding product:

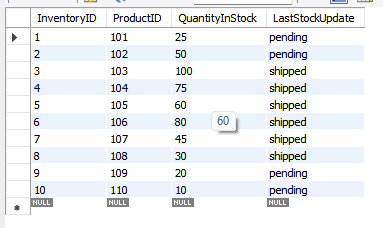


11. UPDATE Orderdetails

INNER JOIN inventory

ON orderdetails.productid = inventory.productid

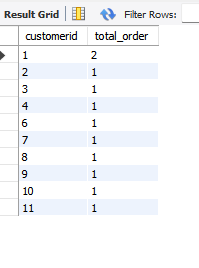
SET inventory.LastStockUpdate="shipped";



12. select customerid, count(\*) as total\_order

from orders

group by customerid;



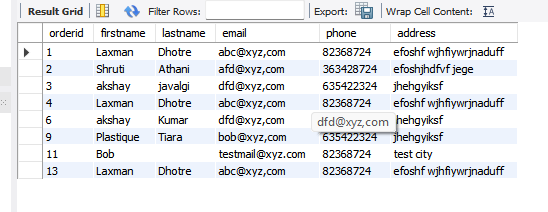
Task 3

1. select orderid, firstname, lastname, email, phone, address

from Orders

join customer on orders.customerid=customer.customerid

group by orderid;

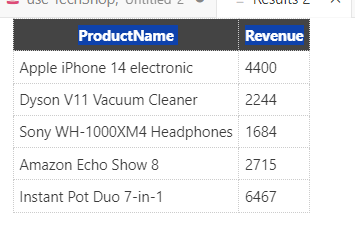


2. SELECT Products.ProductName, SUM(Products.Price \* OrderDetails.Quantity) AS Revenue

FROM Products

JOIN OrderDetails ON OrderDetails.ProductID = Products.ProductID

GROUP BY Products.ProductName;

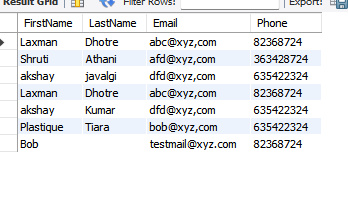


3. SELECT c.FirstName, c.LastName, c.Email, c.Phone

FROM Customer c

JOIN Orders o ON c.CustomerID = o.CustomerID

GROUP BY c.CustomerID, c.FirstName, c.LastName, c.Email, c.Phone

HAVING COUNT(o.OrderID) >= 1;

4.

select productname, quantity

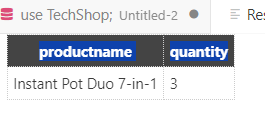
from products

join orderdetails on products.productid=orderdetails.OrderID

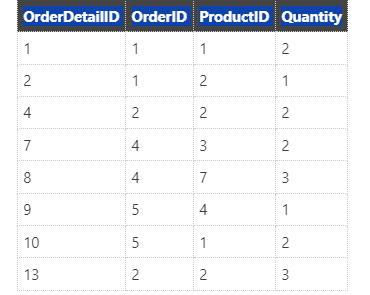
order by orderdetails.quantity desc

LIMIT 1;

Output :



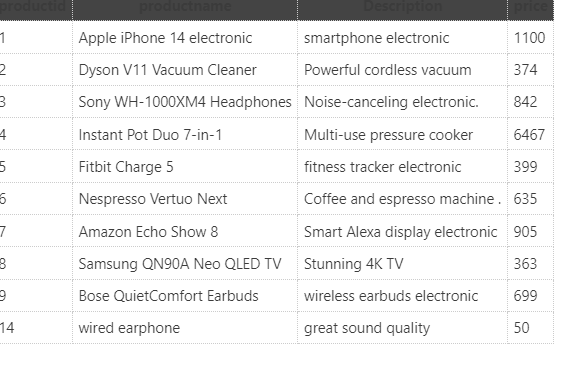
From table:



5. task 1 didnt have any category row in any table. So, i cant make any categories

select productid,productname, Description, price

from products;

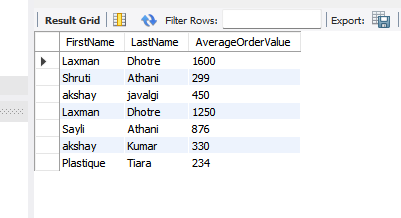


6. SELECT C.FirstName, C.LastName, sum(O.TotalAmount) AS AverageOrderValue

FROM Customer C

JOIN Orders O ON C.CustomerID = O.CustomerID

GROUP BY C.CustomerID, C.FirstName, C.LastName;



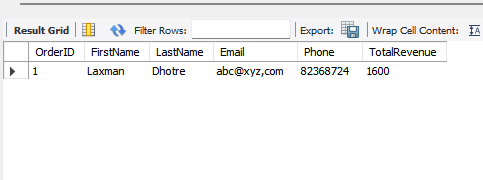
7. SELECT O.OrderID, C.FirstName, C.LastName, C.Email, C.Phone, O.TotalAmount AS TotalRevenue

FROM Orders O

JOIN Customer C ON O.CustomerID = C.CustomerID

ORDER BY O.TotalAmount DESC

LIMIT 1;



8. update orders TotalAmount=

select (sum(quantity\*price))

from orderdetails join products

on orderdetails.productid=products.productid

where orderdetails.orderid=orders.orderid;

Table after updation:



9. select firstname, lastname

from customer

join orders

on customer.customerid=orders.customerid

join orderdetails

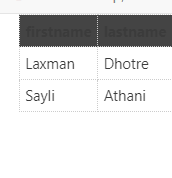
on orders.orderid=orderdetails.orderid

join products

on orderdetails.productid=products.productid

where products.productid=1;

For product id 1: output:



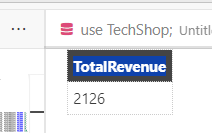
10.

SELECT SUM(TotalAmount) AS TotalRevenue

FROM Orders

WHERE OrderDate BETWEEN '2024-03-05' AND '2024-06-05';

Output:



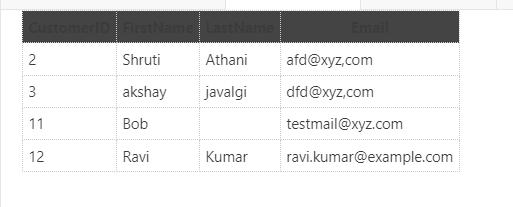
Task 4

1. SELECT Customer.CustomerID, Customer.FirstName, Customer.LastName, Customer.Email

FROM Customer

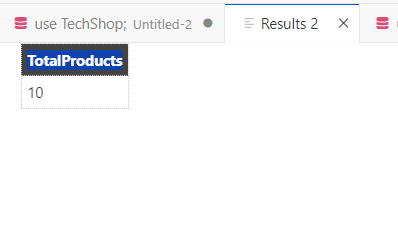
LEFT JOIN Orders ON Customer.CustomerID = Orders.CustomerID

WHERE Orders.OrderID IS NULL;



2. SELECT COUNT(\*) AS TotalProducts

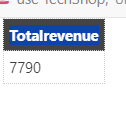
FROM Products;



3. SELECT sum(TotalAmount) AS Totalrevenue

FROM orders;

Output

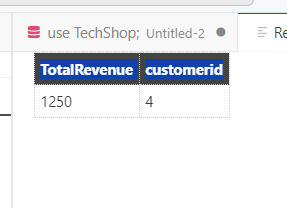


4.

5. SELECT SUM(TotalAmount) AS TotalRevenue

FROM Orders

WHERE CustomerID = 4;



6.(table modified to have customer with more thaan one order, inserted laxman with one more order)

SELECT Customer.FirstName, Customer.LastName, COUNT(Orders.OrderID) AS OrderCount

FROM Customer

JOIN Orders ON Customer.CustomerID = Orders.CustomerID

GROUP BY Customer.CustomerID, Customer.FirstName, Customer.LastName

ORDER BY OrderCount DESC

limit 1;



7.