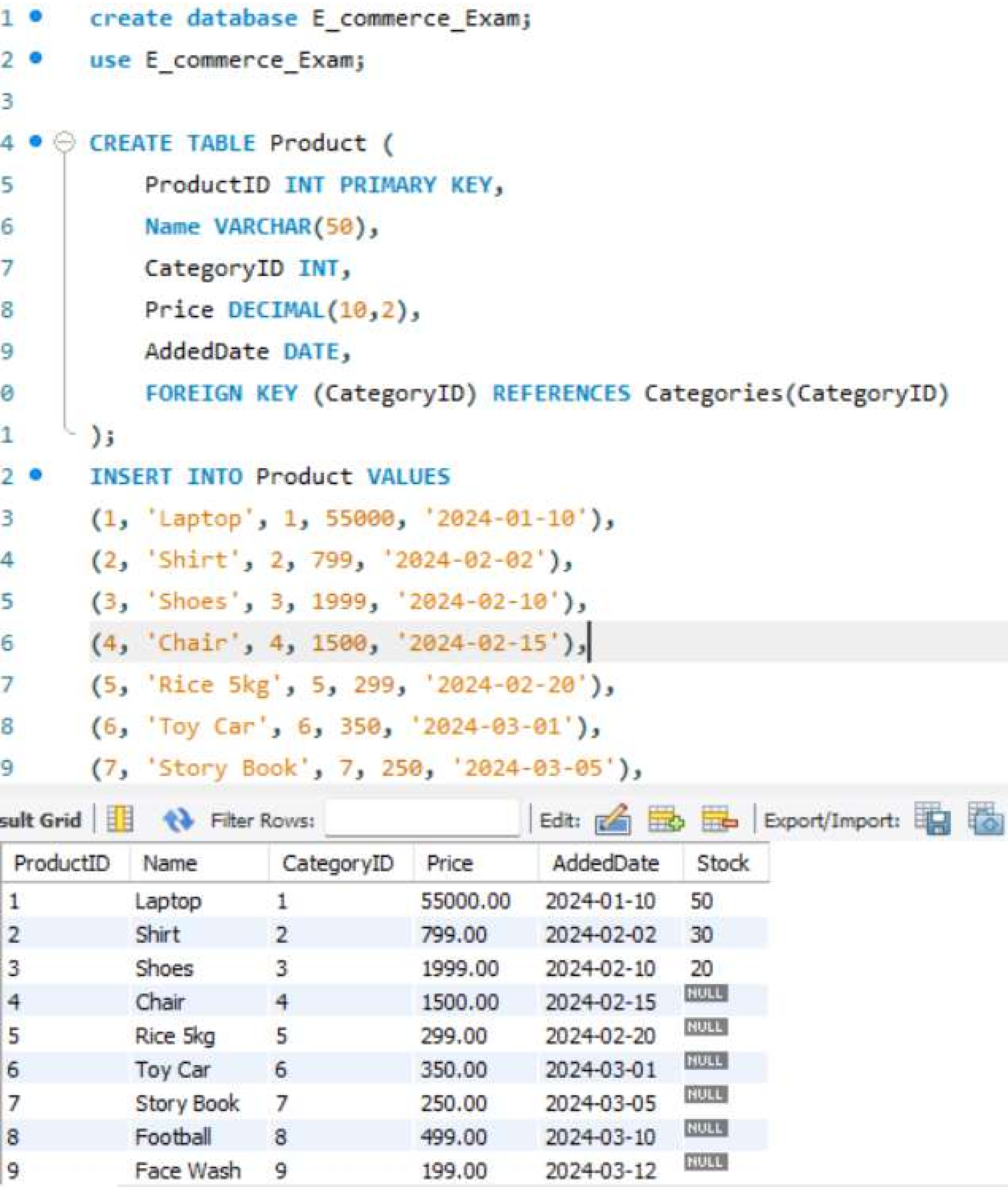
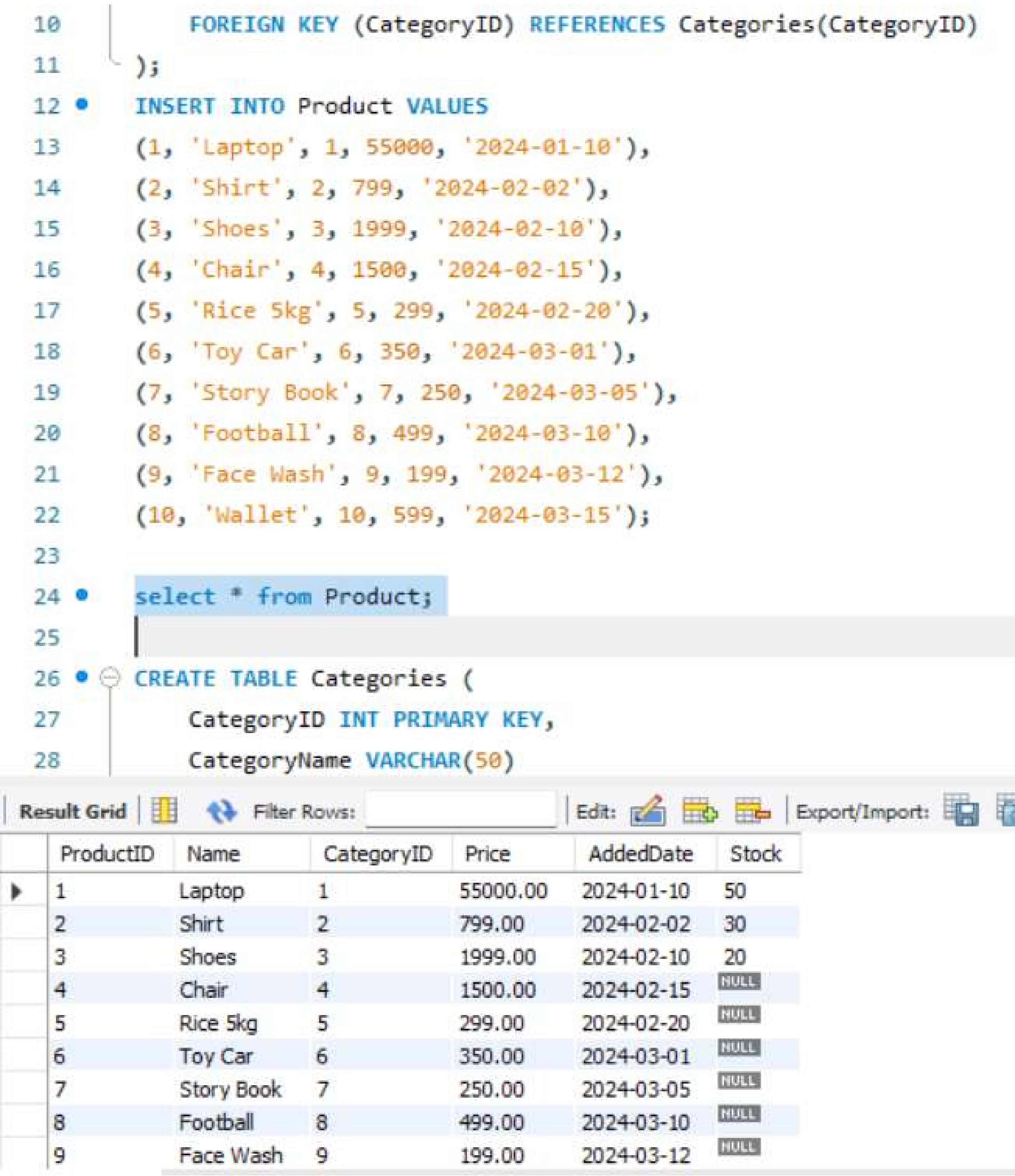


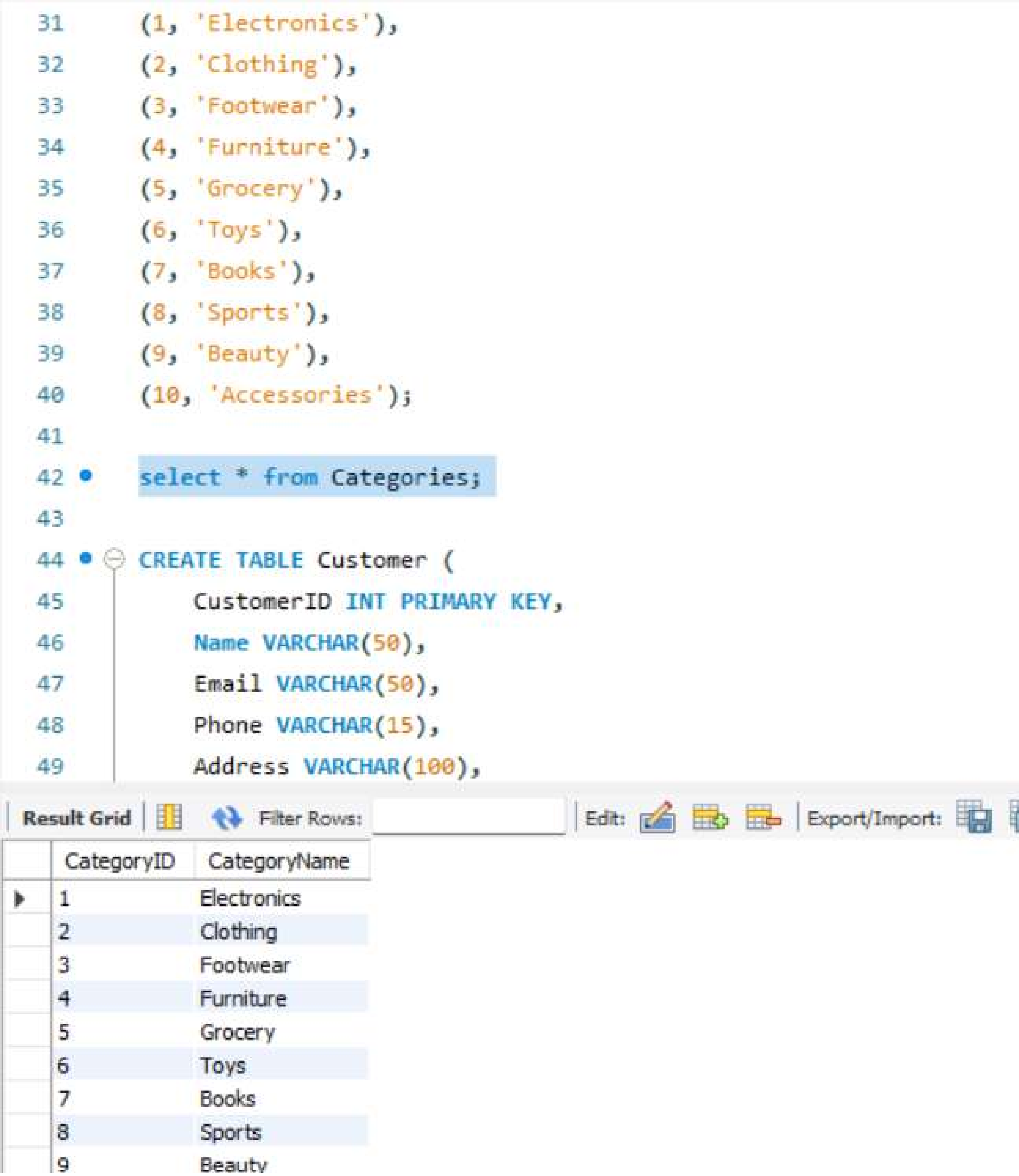
```
create database E_commerce_Exam;
use E_commerce_Exam;

CREATE TABLE Product (
    ProductID INT PRIMARY KEY,
    Name VARCHAR(50),
    CategoryID INT,
    Price DECIMAL(10,2),
    AddedDate DATE,
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
);
INSERT INTO Product VALUES
(1, 'Laptop', 1, 55000, '2024-01-10'),
(2, 'Shirt', 2, 799, '2024-02-02'),
(3, 'Shoes', 3, 1999, '2024-02-10'),
(4, 'Chair', 4, 1500, '2024-02-15'),
(5, 'Rice 5kg', 5, 299, '2024-02-20'),
(6, 'Toy Car', 6, 350, '2024-03-01'),
(7, 'Story Book', 7, 250, '2024-03-05'),
```

ProductID	Name	CategoryID	Price	AddedDate	Stock
1	Laptop	1	55000.00	2024-01-10	50
2	Shirt	2	799.00	2024-02-02	30
3	Shoes	3	1999.00	2024-02-10	20
4	Chair	4	1500.00	2024-02-15	NULL
5	Rice 5kg	5	299.00	2024-02-20	NULL
6	Toy Car	6	350.00	2024-03-01	NULL
7	Story Book	7	250.00	2024-03-05	NULL
8	Football	8	499.00	2024-03-10	NULL
9	Face Wash	9	199.00	2024-03-12	NULL



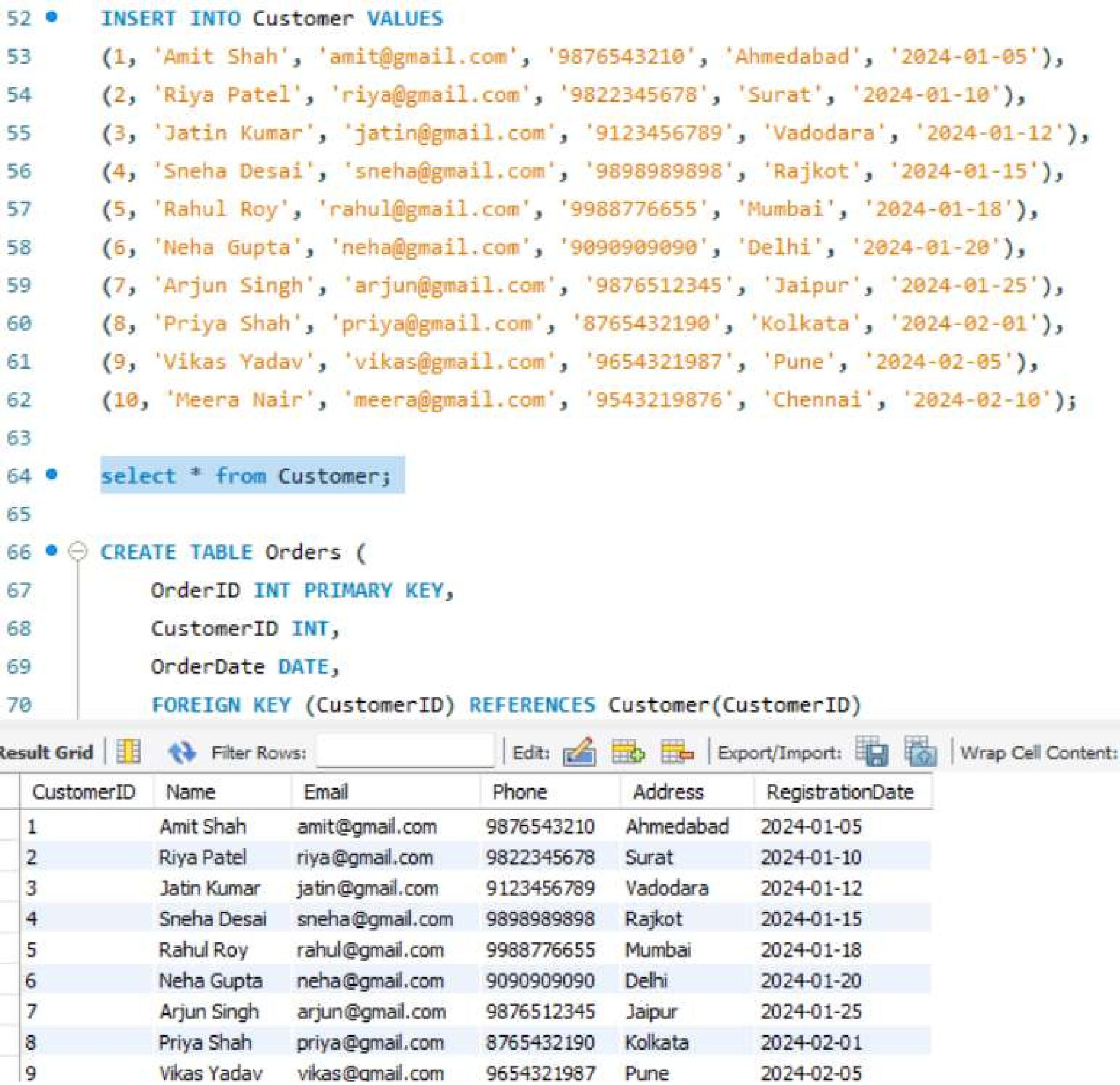
```
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
);
INSERT INTO Product VALUES
(1, 'Laptop', 1, 55000, '2024-01-10'),
(2, 'Shirt', 2, 799, '2024-02-02'),
(3, 'Shoes', 3, 1999, '2024-02-10'),
(4, 'Chair', 4, 1500, '2024-02-15'),
(5, 'Rice 5kg', 5, 299, '2024-02-20'),
(6, 'Toy Car', 6, 350, '2024-03-01'),
(7, 'Story Book', 7, 250, '2024-03-05'),
(8, 'Football', 8, 499, '2024-03-10'),
(9, 'Face Wash', 9, 199, '2024-03-12'),
(10, 'Wallet', 10, 599, '2024-03-15');

select * from Product;

CREATE TABLE Categories (
    CategoryID INT PRIMARY KEY,
    CategoryName VARCHAR(50)
```

ProductID	Name	CategoryID	Price	AddedDate	Stock
1	Laptop	1	55000.00	2024-01-10	50
2	Shirt	2	799.00	2024-02-02	30
3	Shoes	3	1999.00	2024-02-10	20
4	Chair	4	1500.00	2024-02-15	NULL
5	Rice 5kg	5	299.00	2024-02-20	NULL
6	Toy Car	6	350.00	2024-03-01	NULL
7	Story Book	7	250.00	2024-03-05	NULL
8	Football	8	499.00	2024-03-10	NULL
9	Face Wash	9	199.00	2024-03-12	NULL



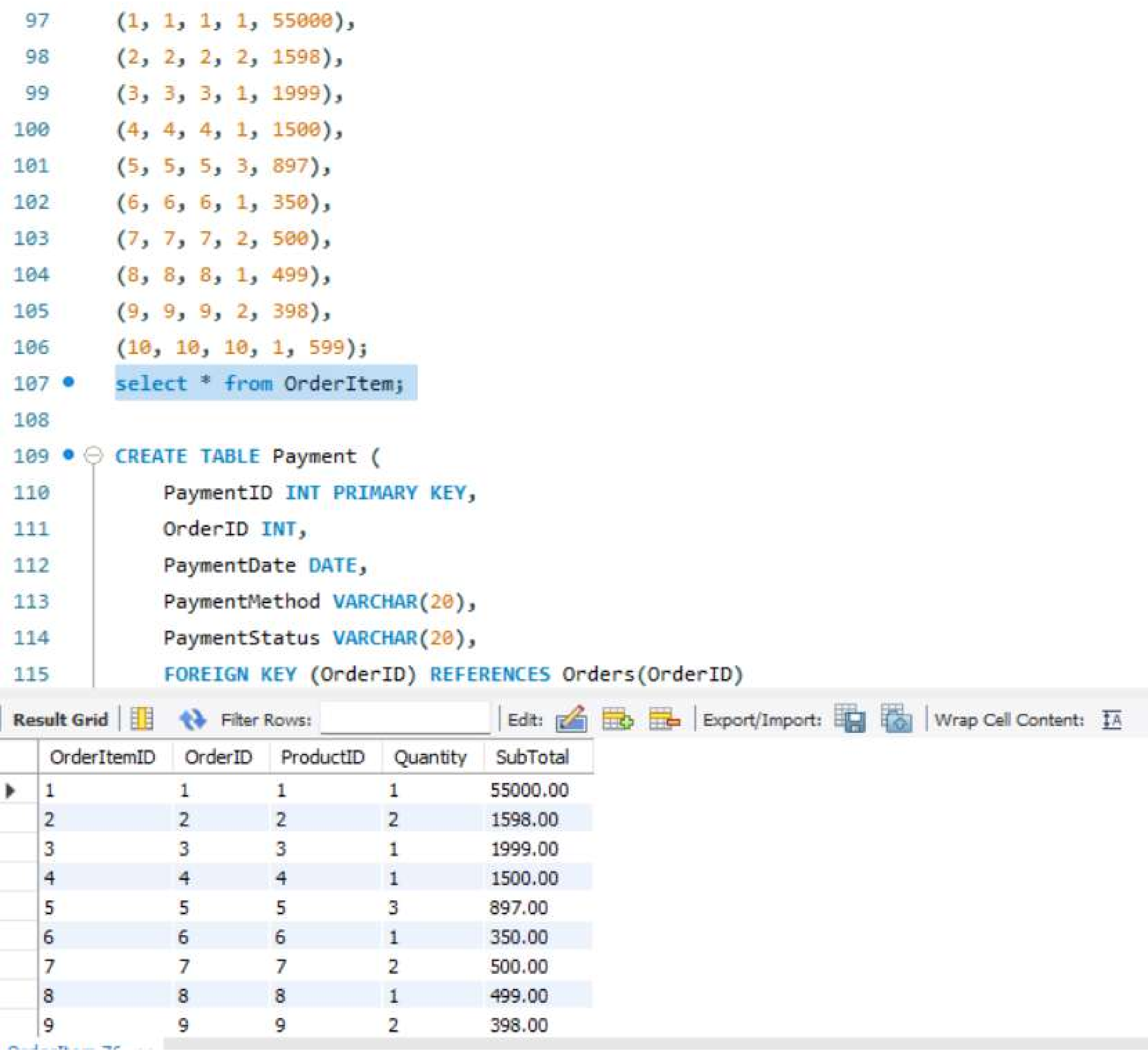
```
(1, 'Electronics'),
(2, 'Clothing'),
(3, 'Footwear'),
(4, 'Furniture'),
(5, 'Grocery'),
(6, 'Toys'),
(7, 'Books'),
(8, 'Sports'),
(9, 'Beauty'),
(10, 'Accessories');

select * from Categories;

CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(50),
    Email VARCHAR(50),
    Phone VARCHAR(15),
    Address VARCHAR(100),
```

CategoryID	CategoryName
1	Electronics
2	Clothing
3	Footwear
4	Furniture
5	Grocery
6	Toys
7	Books
8	Sports
9	Beauty



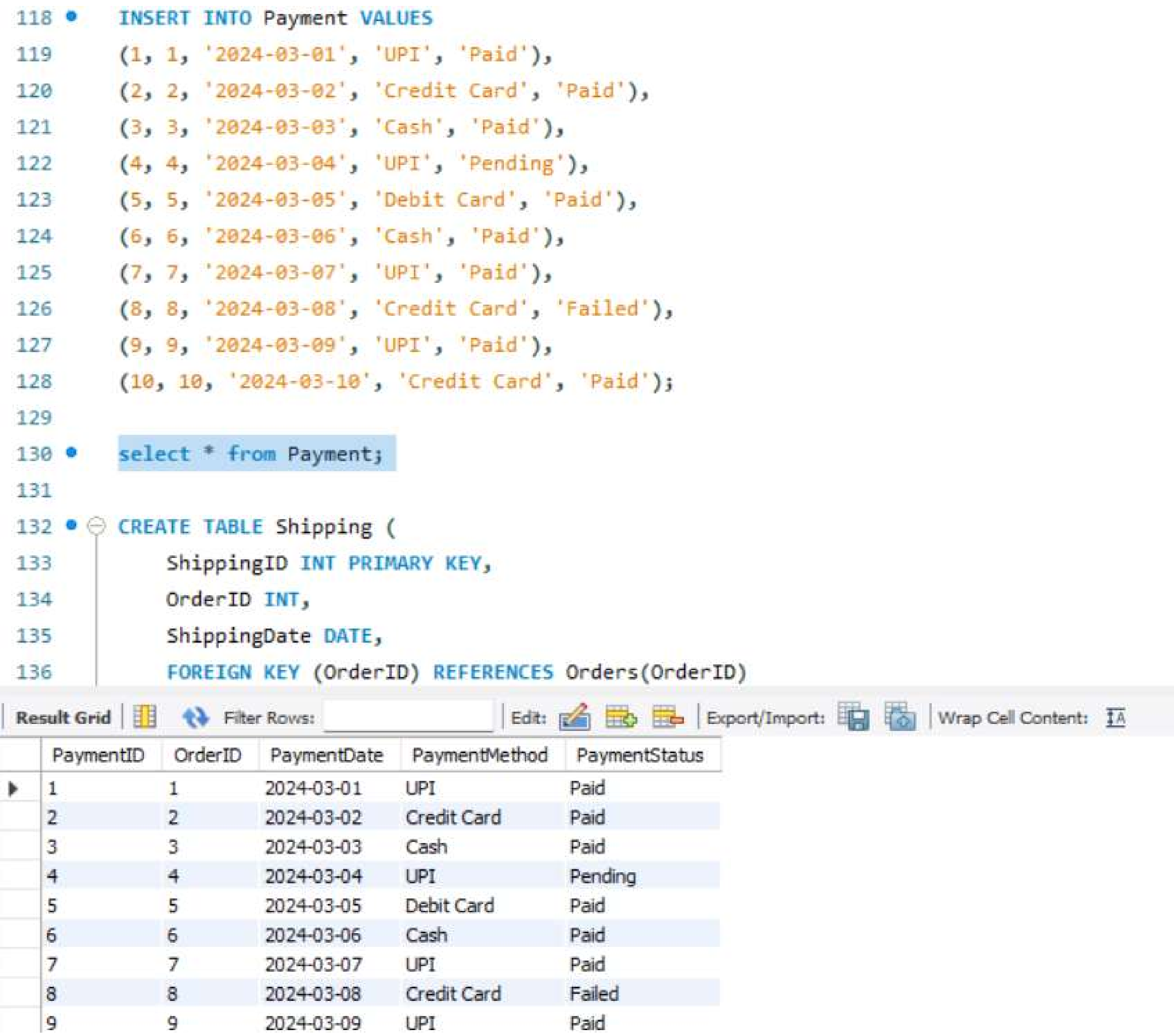
```sql
INSERT INTO Customer VALUES
(1, 'Amit Shah', 'amit@gmail.com', '9876543210', 'Ahmedabad', '2024-01-05'),
(2, 'Riya Patel', 'riya@gmail.com', '9822345678', 'Surat', '2024-01-10'),
(3, 'Jatin Kumar', 'jatin@gmail.com', '9123456789', 'Vadodara', '2024-01-12'),
(4, 'Sneha Desai', 'sneha@gmail.com', '9898989898', 'Rajkot', '2024-01-15'),
(5, 'Rahul Roy', 'rahul@gmail.com', '9988776655', 'Mumbai', '2024-01-18'),
(6, 'Neha Gupta', 'neha@gmail.com', '9090909090', 'Delhi', '2024-01-20'),
(7, 'Arjun Singh', 'arjun@gmail.com', '9876512345', 'Jaipur', '2024-01-25'),
(8, 'Priya Shah', 'priya@gmail.com', '8765432190', 'Kolkata', '2024-02-01'),
(9, 'Vikas Yadav', 'vikas@gmail.com', '9654321987', 'Pune', '2024-02-05'),
(10, 'Meera Nair', 'meera@gmail.com', '9543219876', 'Chennai', '2024-02-10');

select * from Customer;

CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
```

CustomerID	Name	Email	Phone	Address	RegistrationDate
1	Amit Shah	amit@gmail.com	9876543210	Ahmedabad	2024-01-05
2	Riya Patel	riya@gmail.com	9822345678	Surat	2024-01-10
3	Jatin Kumar	jatin@gmail.com	9123456789	Vadodara	2024-01-12
4	Sneha Desai	sneha@gmail.com	9898989898	Rajkot	2024-01-15
5	Rahul Roy	rahul@gmail.com	9988776655	Mumbai	2024-01-18
6	Neha Gupta	neha@gmail.com	9090909090	Delhi	2024-01-20
7	Arjun Singh	arjun@gmail.com	9876512345	Jaipur	2024-01-25
8	Priya Shah	priya@gmail.com	8765432190	Kolkata	2024-02-01
9	Vikas Yadav	vikas@gmail.com	9654321987	Pune	2024-02-05



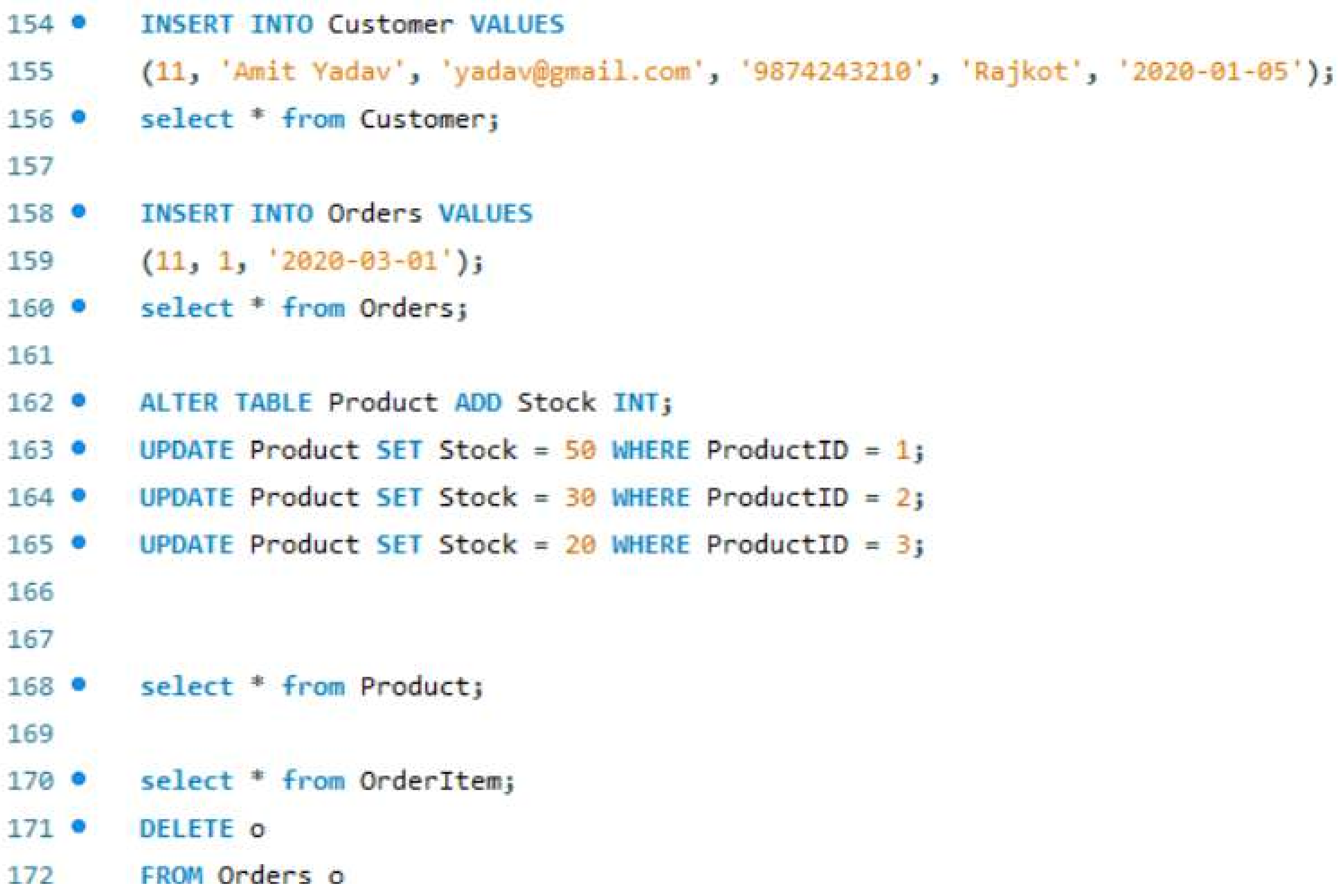
```
    (1, 1, 1, 1, 55000),
    (2, 2, 2, 2, 1598),
    (3, 3, 3, 1, 1999),
    (4, 4, 4, 1, 1500),
    (5, 5, 5, 3, 897),
    (6, 6, 6, 1, 350),
    (7, 7, 7, 2, 500),
    (8, 8, 8, 1, 499),
    (9, 9, 9, 2, 398),
    (10, 10, 10, 1, 599);
    select * from OrderItem;

    CREATE TABLE Payment (
        PaymentID INT PRIMARY KEY,
        OrderID INT,
        PaymentDate DATE,
        PaymentMethod VARCHAR(20),
        PaymentStatus VARCHAR(20),
        FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
```

OrderItemID	OrderID	ProductID	Quantity	SubTotal
1	1	1	1	55000.00
2	2	2	2	1598.00
3	3	3	1	1999.00
4	4	4	1	1500.00
5	5	5	3	897.00
6	6	6	1	350.00
7	7	7	2	500.00
8	8	8	1	499.00
9	9	9	2	398.00

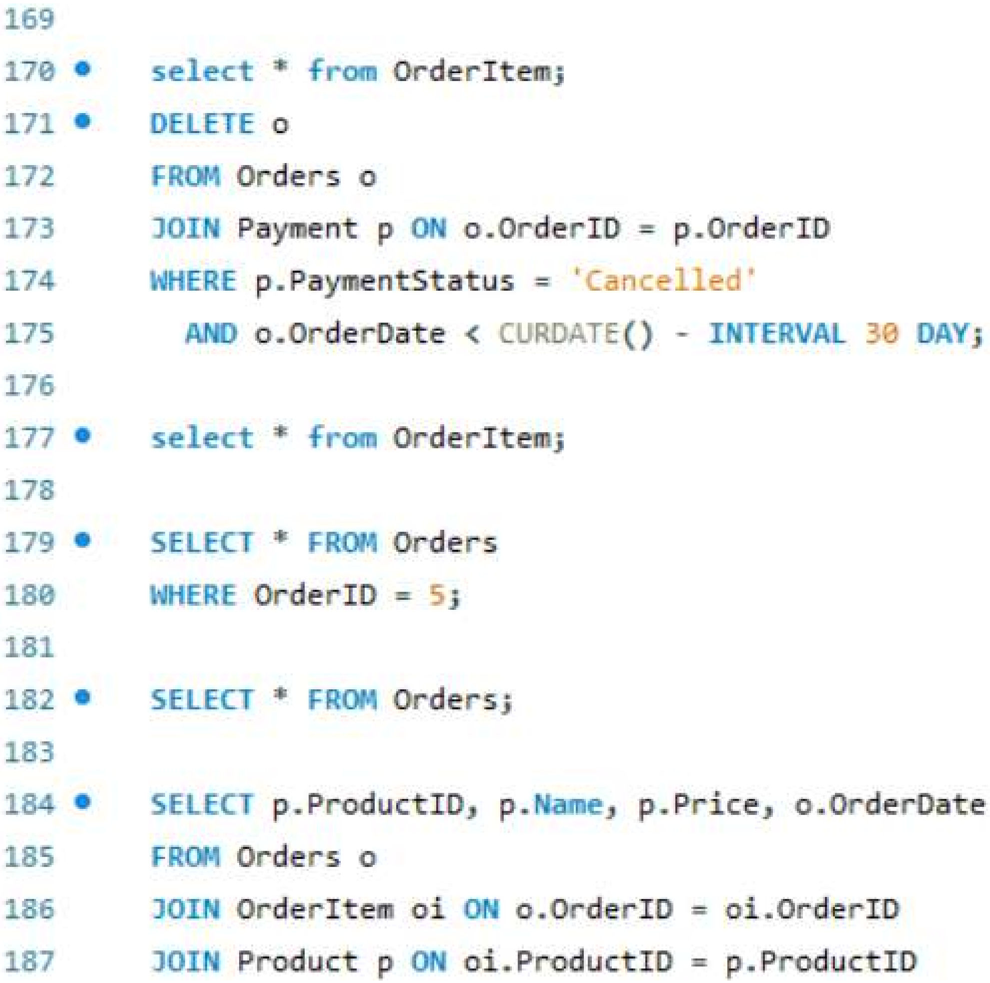


```sql
INSERT INTO Payment VALUES
(1, 1, '2024-03-01', 'UPI', 'Paid'),
(2, 2, '2024-03-02', 'Credit Card', 'Paid'),
(3, 3, '2024-03-03', 'Cash', 'Paid'),
(4, 4, '2024-03-04', 'UPI', 'Pending'),
(5, 5, '2024-03-05', 'Debit Card', 'Paid'),
(6, 6, '2024-03-06', 'Cash', 'Paid'),
(7, 7, '2024-03-07', 'UPI', 'Paid'),
(8, 8, '2024-03-08', 'Credit Card', 'Failed'),
(9, 9, '2024-03-09', 'UPI', 'Paid'),
(10, 10, '2024-03-10', 'Credit Card', 'Paid');

select * from Payment;

CREATE TABLE Shipping (
    ShippingID INT PRIMARY KEY,
    OrderID INT,
    ShippingDate DATE,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
```

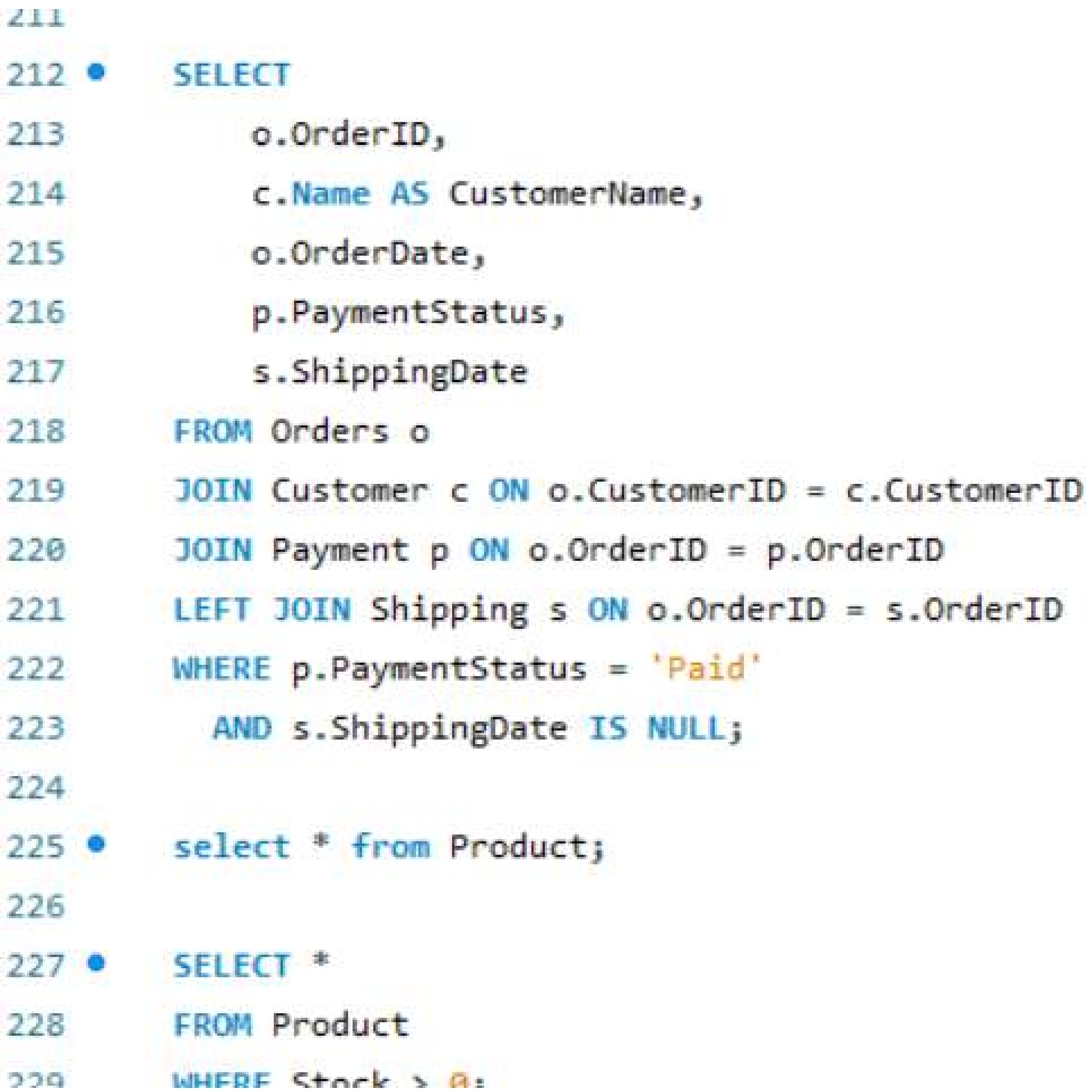
PaymentID	OrderID	PaymentDate	PaymentMethod	PaymentStatus
1	1	2024-03-01	UPI	Paid
2	2	2024-03-02	Credit Card	Paid
3	3	2024-03-03	Cash	Paid
4	4	2024-03-04	UPI	Pending
5	5	2024-03-05	Debit Card	Paid
6	6	2024-03-06	Cash	Paid
7	7	2024-03-07	UPI	Paid
8	8	2024-03-08	Credit Card	Failed
9	9	2024-03-09	UPI	Paid



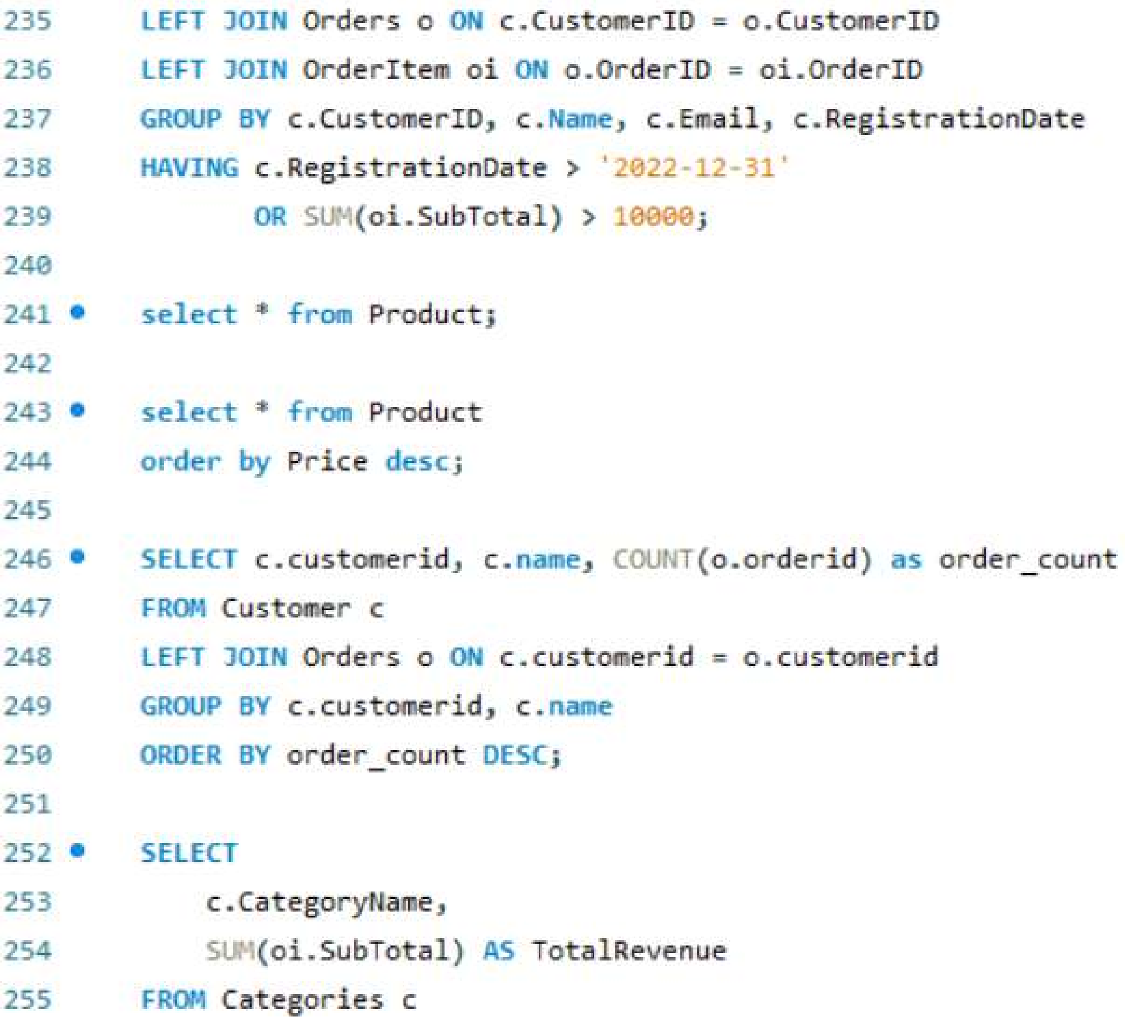
```
INSERT INTO Customer VALUES
(11, 'Amit Yadav', 'yadav@gmail.com', '9874243210', 'Rajkot', '2020-01-05');
select * from Customer;

INSERT INTO Orders VALUES
(11, 1, '2020-03-01');
select * from Orders;

ALTER TABLE Product ADD Stock INT;
UPDATE Product SET Stock = 50 WHERE ProductID = 1;
UPDATE Product SET Stock = 30 WHERE ProductID = 2;
UPDATE Product SET Stock = 20 WHERE ProductID = 3;


select * from Product;

select * from OrderItem;
DELETE o
FROM Orders o
```



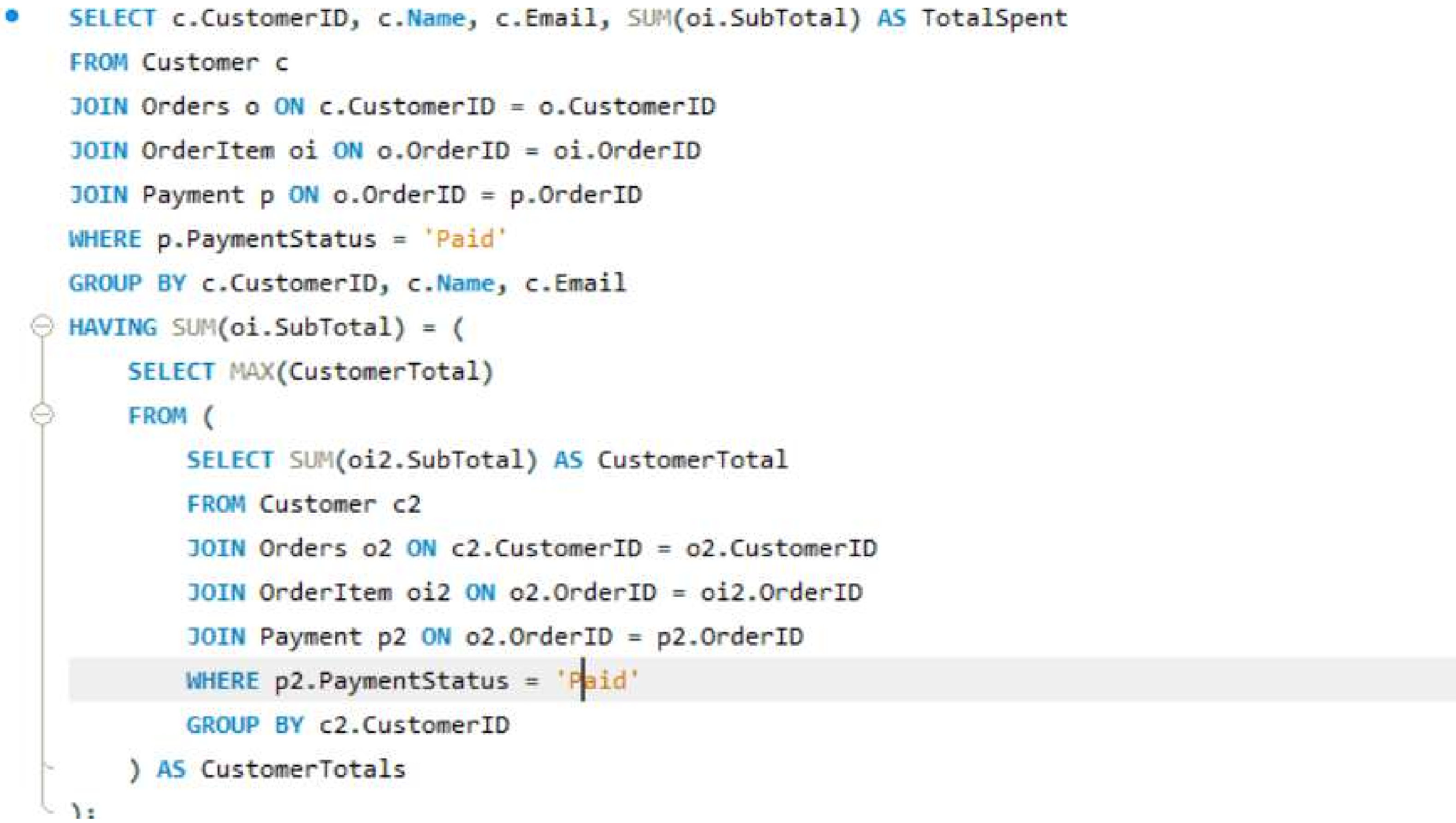
```sql

select * from OrderItem;
DELETE o
FROM Orders o
JOIN Payment p ON o.OrderID = p.OrderID
WHERE p.PaymentStatus = 'Cancelled'
  AND o.OrderDate < CURDATE() - INTERVAL 30 DAY;

select * from OrderItem;

SELECT * FROM Orders
WHERE OrderID = 5;

SELECT * FROM Orders;

SELECT p.ProductID, p.Name, p.Price, o.OrderDate
FROM Orders o
JOIN OrderItem oi ON o.OrderID = oi.OrderID
JOIN Product p ON oi.ProductID = p.ProductID
```



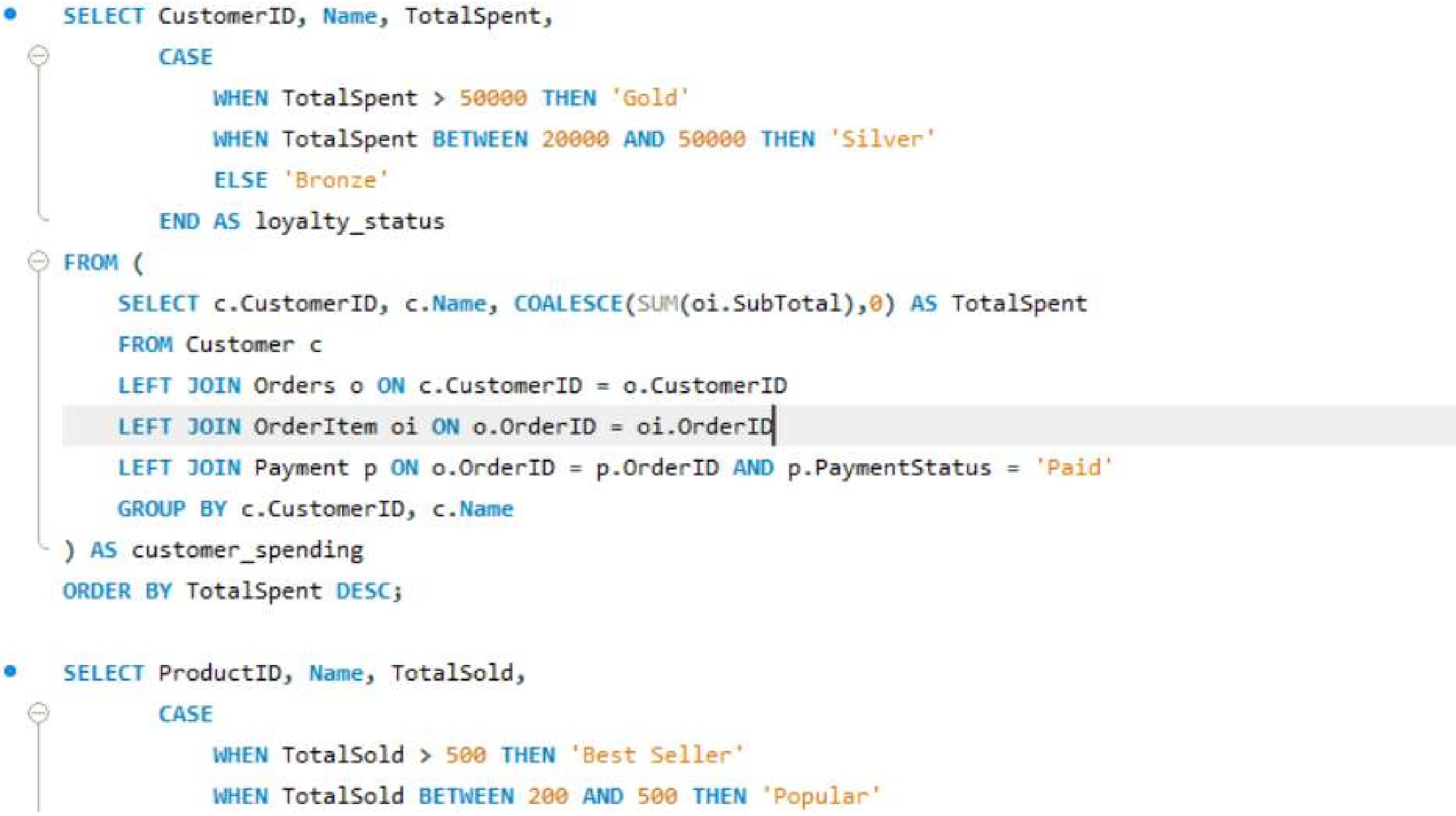
```sql
SELECT
    o.OrderID,
    c.Name AS CustomerName,
    o.OrderDate,
    p.PaymentStatus,
    s.ShippingDate
FROM Orders o
JOIN Customer c ON o.CustomerID = c.CustomerID
JOIN Payment p ON o.OrderID = p.OrderID
LEFT JOIN Shipping s ON o.OrderID = s.OrderID
WHERE p.PaymentStatus = 'Paid'
  AND s.ShippingDate IS NULL;

select * from Product;

SELECT *
FROM Product
WHERE Stock > 0;
```



```sql
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
LEFT JOIN OrderItem oi ON o.OrderID = oi.OrderID
GROUP BY c.CustomerID, c.Name, c.Email, c.RegistrationDate
HAVING c.RegistrationDate > '2022-12-31'
       OR SUM(oi.SubTotal) > 10000;

select * from Product;

select * from Product
order by Price desc;

SELECT c.customerid, c.name, COUNT(o.orderid) as order_count
FROM Customer c
LEFT JOIN Orders o ON c.customerid = o.customerid
GROUP BY c.customerid, c.name
ORDER BY order_count DESC;

SELECT
    c.CategoryName,
    SUM(oi.SubTotal) AS TotalRevenue
FROM Categories c
```



```sql
SELECT c.CustomerID, c.Name, c.Email, SUM(oi.SubTotal) AS TotalSpent
FROM Customer c
JOIN Orders o ON c.CustomerID = o.CustomerID
JOIN OrderItem oi ON o.OrderID = oi.OrderID
JOIN Payment p ON o.OrderID = p.OrderID
WHERE p.PaymentStatus = 'Paid'
GROUP BY c.CustomerID, c.Name, c.Email
HAVING SUM(oi.SubTotal) = (
    SELECT MAX(CustomerTotal)
    FROM (
        SELECT SUM(oi2.SubTotal) AS CustomerTotal
        FROM Customer c2
        JOIN Orders o2 ON c2.CustomerID = o2.CustomerID
        JOIN OrderItem oi2 ON o2.OrderID = oi2.OrderID
        JOIN Payment p2 ON o2.OrderID = p2.OrderID
        WHERE p2.PaymentStatus = 'Paid'
        GROUP BY c2.CustomerID
    ) AS CustomerTotals
);
```



```sql
SELECT CustomerID, Name, TotalSpent,
        CASE
            WHEN TotalSpent > 50000 THEN 'Gold'
            WHEN TotalSpent BETWEEN 20000 AND 50000 THEN 'Silver'
            ELSE 'Bronze'
        END AS loyalty_status
FROM (
    SELECT c.CustomerID, c.Name, COALESCE(SUM(oi.SubTotal),0) AS TotalSpent
    FROM Customer c
    LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
    LEFT JOIN OrderItem oi ON o.OrderID = oi.OrderID
    LEFT JOIN Payment p ON o.OrderID = p.OrderID AND p.PaymentStatus = 'Paid'
    GROUP BY c.CustomerID, c.Name
) AS customer_spending
ORDER BY TotalSpent DESC;

SELECT ProductID, Name, TotalSold,
        CASE
            WHEN TotalSold > 500 THEN 'Best Seller'
            WHEN TotalSold BETWEEN 200 AND 500 THEN 'Popular'
```