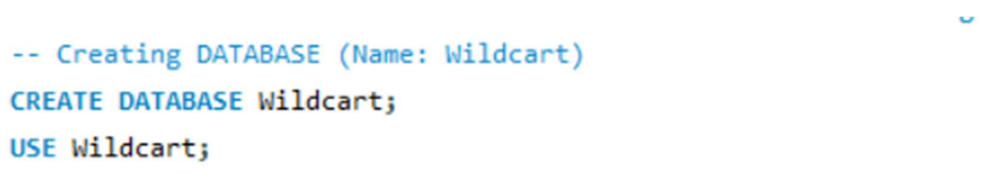
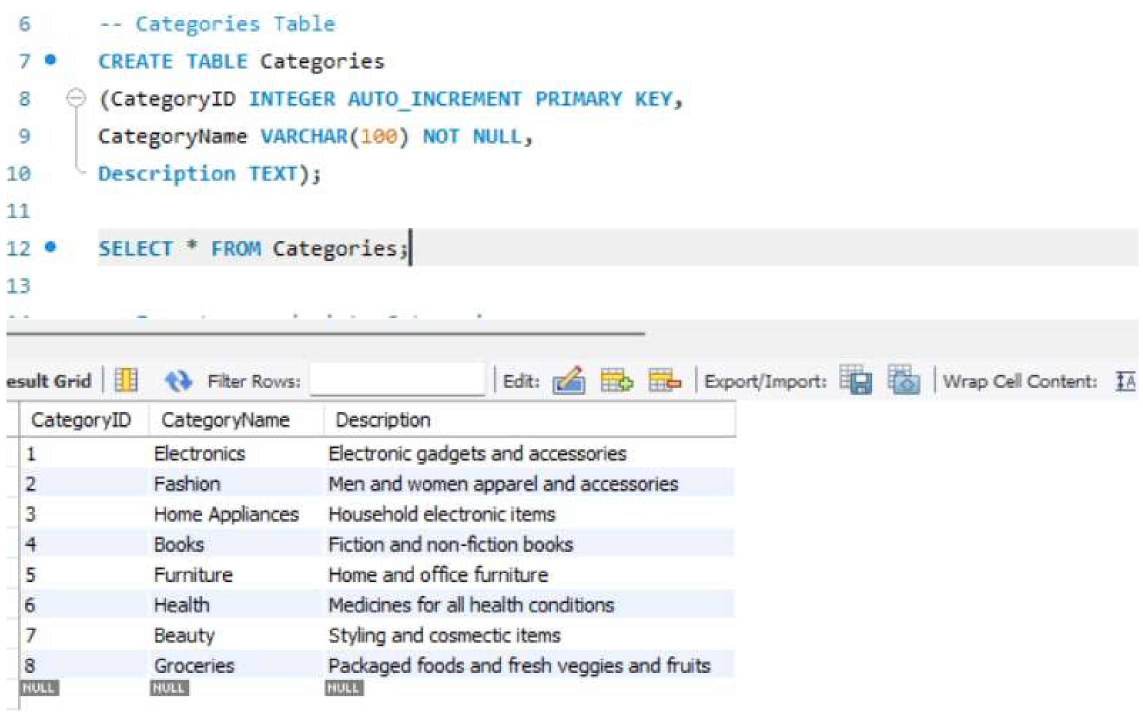
E-Commerce Database Management

Creating Database:



```sql
-- Creating DATABASE (Name: Wildcart)
CREATE DATABASE Wildcart;
USE Wildcart;
```

Categories Table:

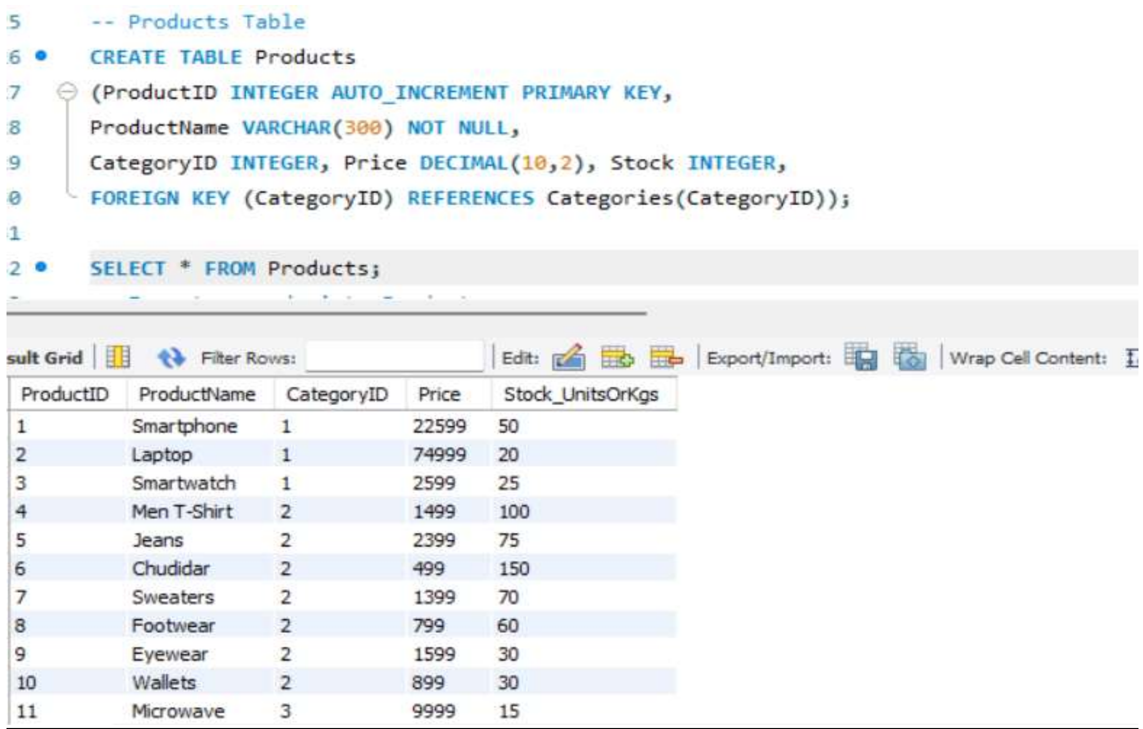


```sql
-- Categories Table
CREATE TABLE Categories
(CategoryID INTEGER AUTO_INCREMENT PRIMARY KEY,
CategoryName VARCHAR(100) NOT NULL,
Description TEXT);

SELECT * FROM Categories;
```

CategoryID	CategoryName	Description
1	Electronics	Electronic gadgets and accessories
2	Fashion	Men and women apparel and accessories
3	Home Appliances	Household electronic items
4	Books	Fiction and non-fiction books
5	Furniture	Home and office furniture
6	Health	Medicines for all health conditions
7	Beauty	Styling and cosmectic items
8	Groceries	Packaged foods and fresh veggies and fruits
NULL	NULL	NULL

Products Table:

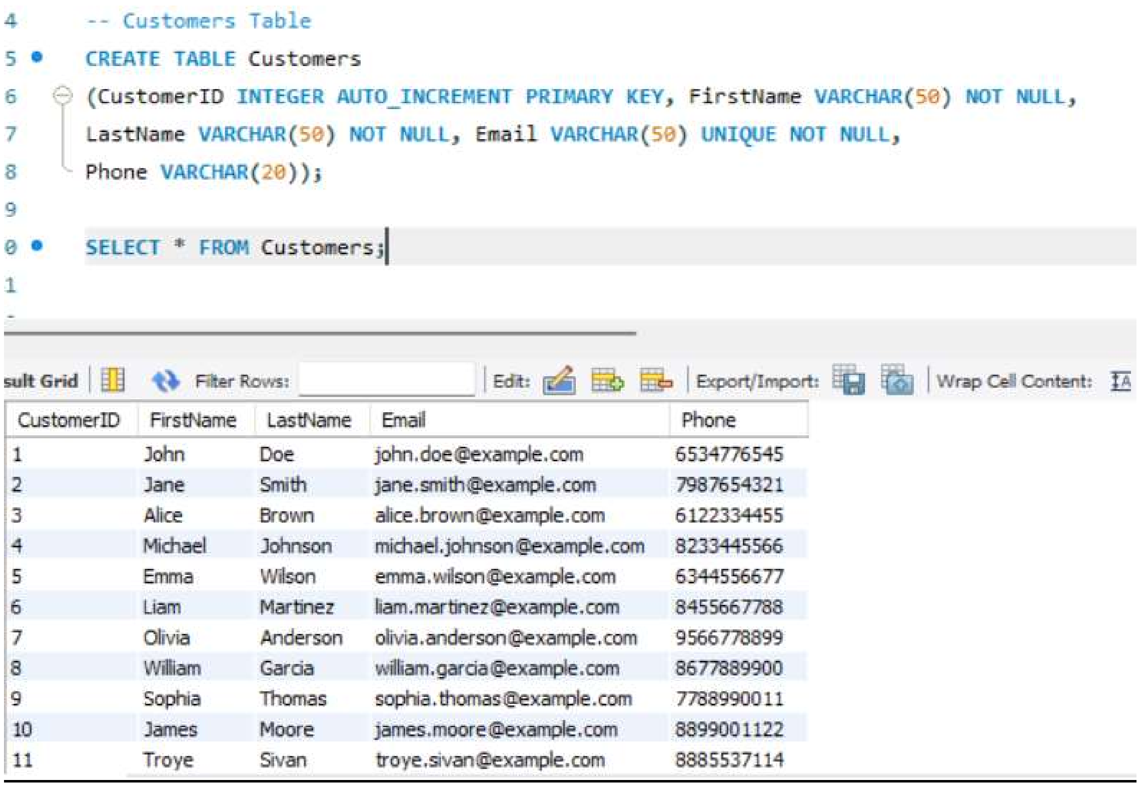


```sql
-- Products Table
CREATE TABLE Products
(ProductID INTEGER AUTO_INCREMENT PRIMARY KEY,
ProductName VARCHAR(300) NOT NULL,
CategoryID INTEGER, Price DECIMAL(10,2), Stock INTEGER,
FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID));

SELECT * FROM Products;
```

ProductID	ProductName	CategoryID	Price	Stock_UnitsOrKgs
1	Smartphone	1	22599	50
2	Laptop	1	74999	20
3	Smartwatch	1	2599	25
4	Men T-Shirt	2	1499	100
5	Jeans	2	2399	75
6	Chudidar	2	499	150
7	Sweaters	2	1399	70
8	Footwear	2	799	60
9	Eyewear	2	1599	30
10	Wallets	2	899	30
11	Microwave	3	9999	15

Customers Table:

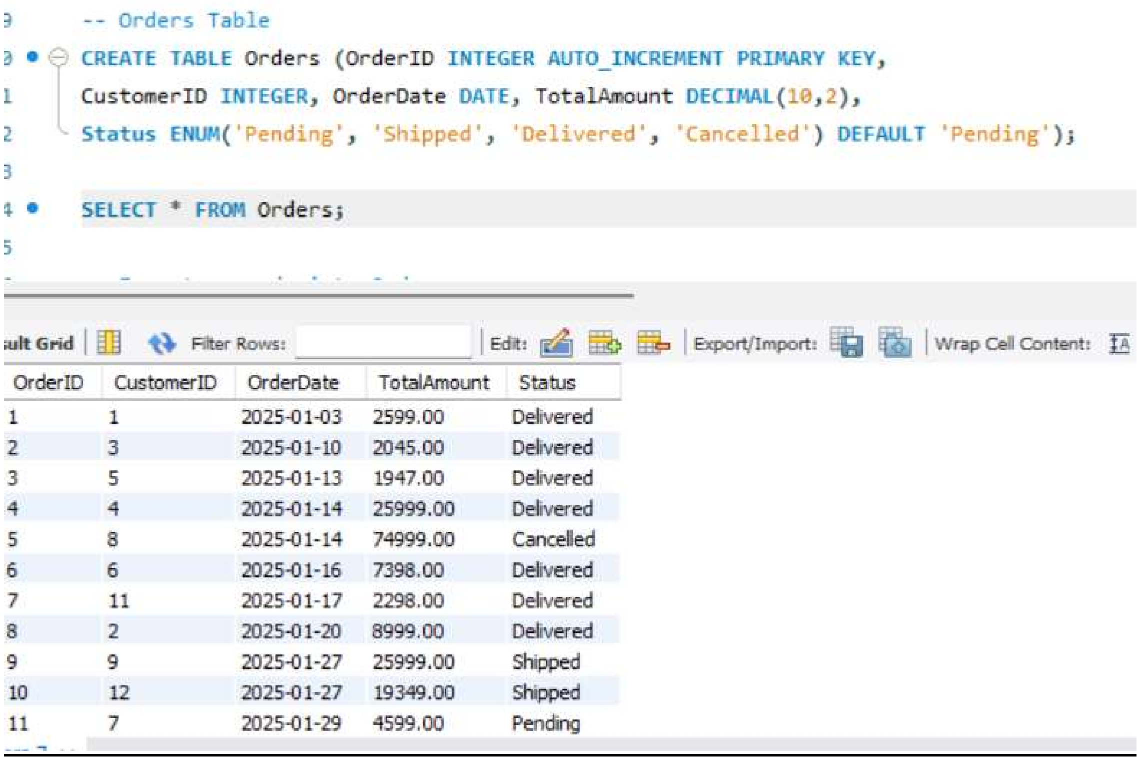


```sql
-- Customers Table
CREATE TABLE Customers
(CustomerID INTEGER AUTO_INCREMENT PRIMARY KEY, FirstName VARCHAR(50) NOT NULL,
LastName VARCHAR(50) NOT NULL, Email VARCHAR(50) UNIQUE NOT NULL,
Phone VARCHAR(20));

SELECT * FROM Customers;
```

CustomerID	FirstName	LastName	Email	Phone
1	John	Doe	john.doe@example.com	6534776545
2	Jane	Smith	jane.smith@example.com	7987654321
3	Alice	Brown	alice.brown@example.com	6122334455
4	Michael	Johnson	michael.johnson@example.com	8233445566
5	Emma	Wilson	emma.wilson@example.com	6344556677
6	Liam	Martinez	liam.martinez@example.com	8455667788
7	Olivia	Anderson	olivia.anderson@example.com	9566778899
8	William	Garcia	william.garcia@example.com	8677889900
9	Sophia	Thomas	sophia.thomas@example.com	7788990011
10	James	Moore	james.moore@example.com	8899001122
11	Troye	Sivan	troye.sivan@example.com	8885537114

Orders Table:

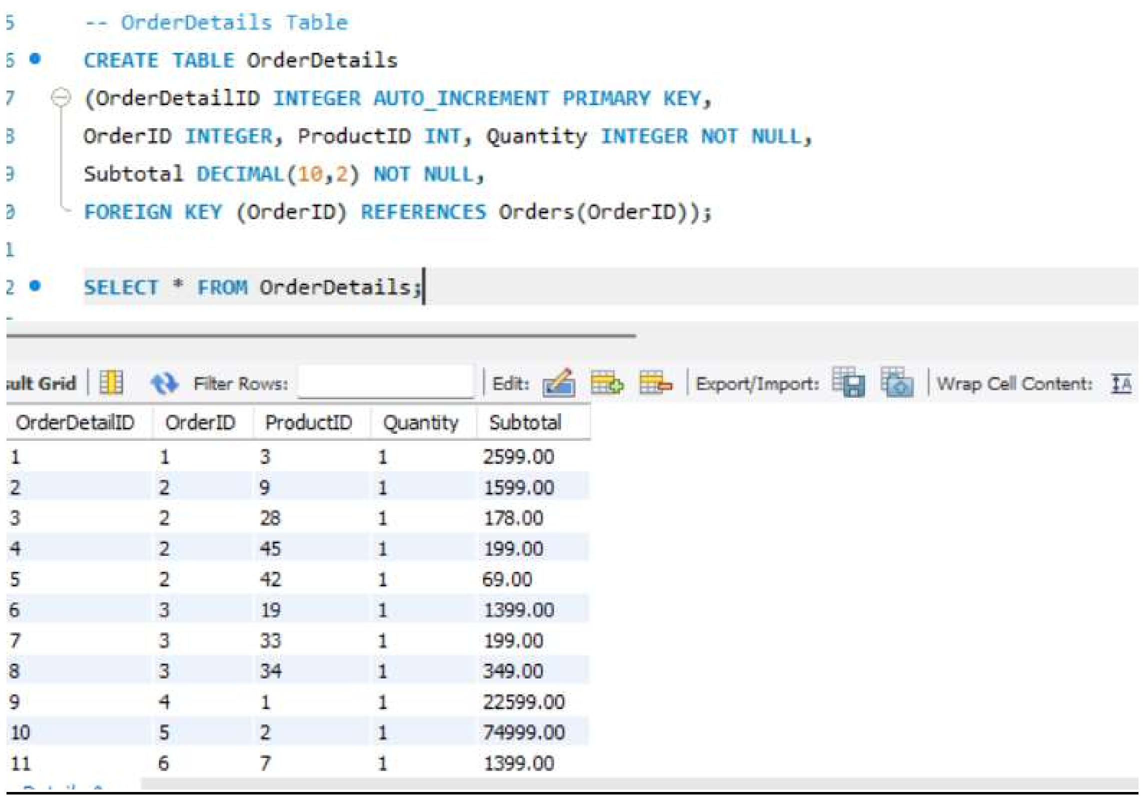


```sql
-- Orders Table
CREATE TABLE Orders (OrderID INTEGER AUTO_INCREMENT PRIMARY KEY,
CustomerID INTEGER, OrderDate DATE, TotalAmount DECIMAL(10,2),
Status ENUM('Pending', 'Shipped', 'Delivered', 'Cancelled') DEFAULT 'Pending');

SELECT * FROM Orders;
```

OrderID	CustomerID	OrderDate	TotalAmount	Status
1	1	2025-01-03	2599.00	Delivered
2	3	2025-01-10	2045.00	Delivered
3	5	2025-01-13	1947.00	Delivered
4	4	2025-01-14	25999.00	Delivered
5	8	2025-01-14	74999.00	Cancelled
6	6	2025-01-16	7398.00	Delivered
7	11	2025-01-17	2298.00	Delivered
8	2	2025-01-20	8999.00	Delivered
9	9	2025-01-27	25999.00	Shipped
10	12	2025-01-27	19349.00	Shipped
11	7	2025-01-29	4599.00	Pending

<u>Order Details Table:</u>

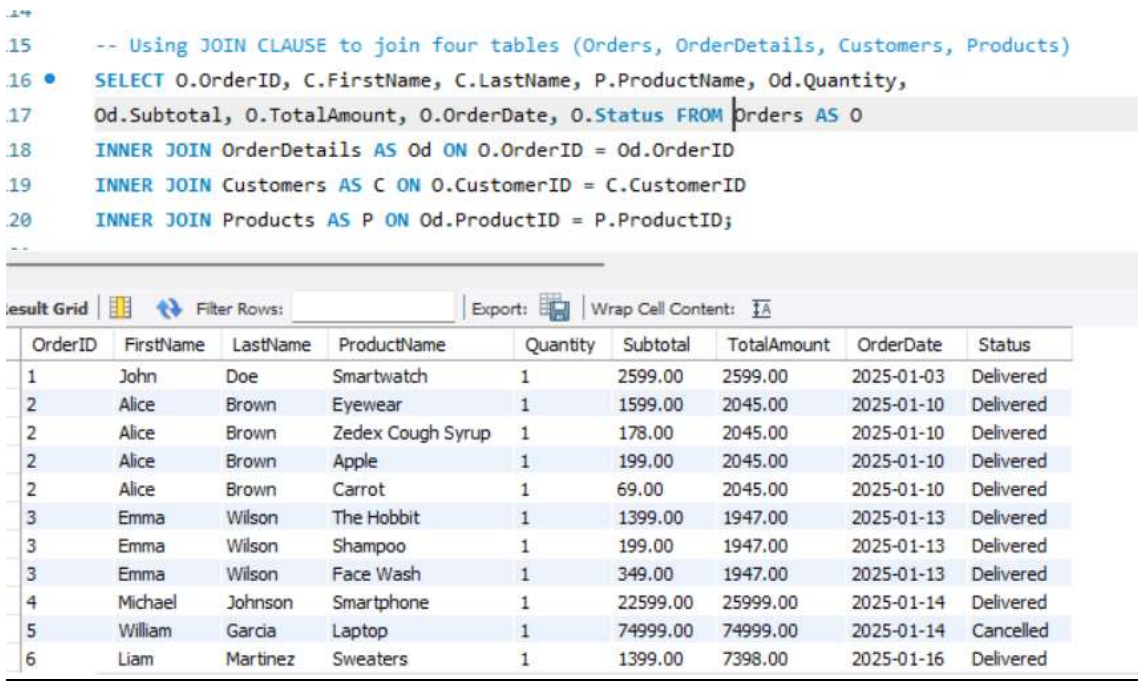


```sql
-- OrderDetails Table
CREATE TABLE OrderDetails
(OrderDetailID INTEGER AUTO_INCREMENT PRIMARY KEY,
OrderID INTEGER, ProductID INT, Quantity INTEGER NOT NULL,
Subtotal DECIMAL(10,2) NOT NULL,
FOREIGN KEY (OrderID) REFERENCES Orders(OrderID));

SELECT * FROM OrderDetails;
```

OrderDetailID	OrderID	ProductID	Quantity	Subtotal
1	1	3	1	2599.00
2	2	9	1	1599.00
3	2	28	1	178.00
4	2	45	1	199.00
5	2	42	1	69.00
6	3	19	1	1399.00
7	3	33	1	199.00
8	3	34	1	349.00
9	4	1	1	22599.00
10	5	2	1	74999.00
11	6	7	1	1399.00

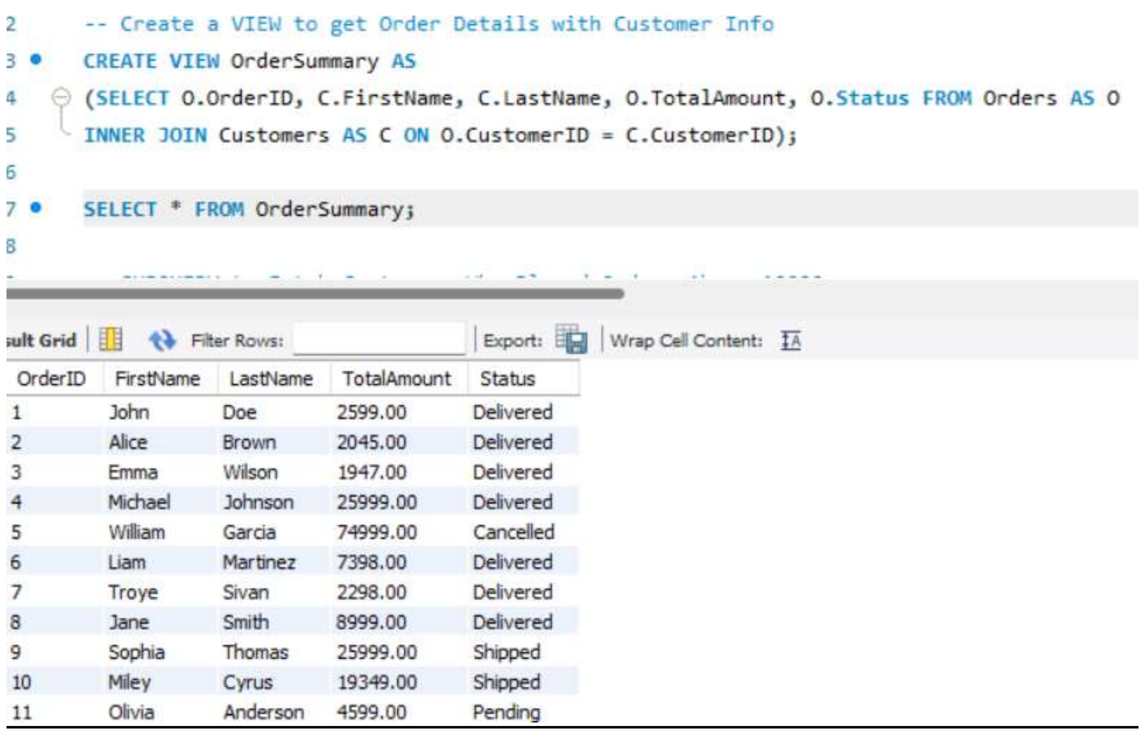
<u>Using JOIN CLAUSE:</u>



```sql
-- Using JOIN CLAUSE to join four tables (Orders, OrderDetails, Customers, Products)
SELECT O.OrderID, C.FirstName, C.LastName, P.ProductName, Od.Quantity,
Od.Subtotal, O.TotalAmount, O.OrderDate, O.Status FROM Orders AS O
INNER JOIN OrderDetails AS Od ON O.OrderID = Od.OrderID
INNER JOIN Customers AS C ON O.CustomerID = C.CustomerID
INNER JOIN Products AS P ON Od.ProductID = P.ProductID;
```

OrderID	FirstName	LastName	ProductName	Quantity	Subtotal	TotalAmount	OrderDate	Status
1	John	Doe	Smartwatch	1	2599.00	2599.00	2025-01-03	Delivered
2	Alice	Brown	Eyewear	1	1599.00	2045.00	2025-01-10	Delivered
2	Alice	Brown	Zedex Cough Syrup	1	178.00	2045.00	2025-01-10	Delivered
2	Alice	Brown	Apple	1	199.00	2045.00	2025-01-10	Delivered
2	Alice	Brown	Carrot	1	69.00	2045.00	2025-01-10	Delivered
3	Emma	Wilson	The Hobbit	1	1399.00	1947.00	2025-01-13	Delivered
3	Emma	Wilson	Shampoo	1	199.00	1947.00	2025-01-13	Delivered
3	Emma	Wilson	Face Wash	1	349.00	1947.00	2025-01-13	Delivered
4	Michael	Johnson	Smartphone	1	22599.00	25999.00	2025-01-14	Delivered
5	William	Garcia	Laptop	1	74999.00	74999.00	2025-01-14	Cancelled
6	Liam	Martinez	Sweaters	1	1399.00	7398.00	2025-01-16	Delivered

Creating a VIEW:

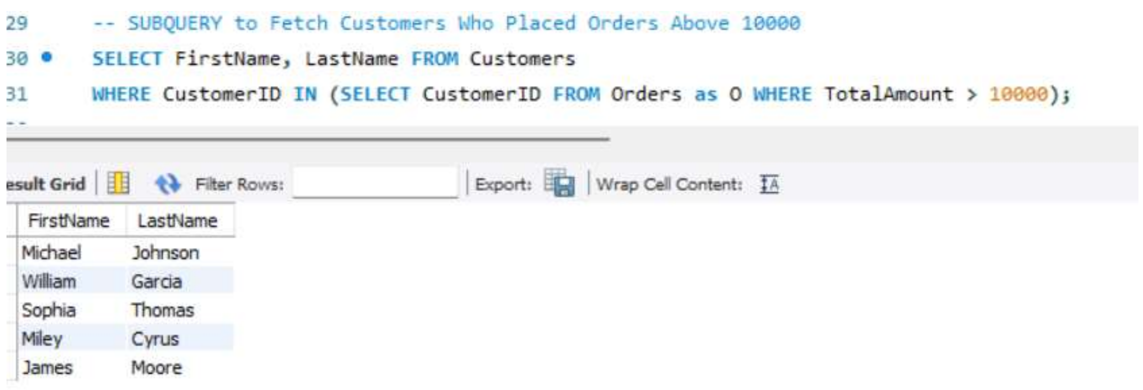


```sql
-- Create a VIEW to get Order Details with Customer Info
CREATE VIEW OrderSummary AS
(SELECT O.OrderID, C.FirstName, C.LastName, O.TotalAmount, O.Status FROM Orders AS O
INNER JOIN Customers AS C ON O.CustomerID = C.CustomerID);

SELECT * FROM OrderSummary;
```

OrderID	FirstName	LastName	TotalAmount	Status
1	John	Doe	2599.00	Delivered
2	Alice	Brown	2045.00	Delivered
3	Emma	Wilson	1947.00	Delivered
4	Michael	Johnson	25999.00	Delivered
5	William	Garcia	74999.00	Cancelled
6	Liam	Martinez	7398.00	Delivered
7	Troye	Sivan	2298.00	Delivered
8	Jane	Smith	8999.00	Delivered
9	Sophia	Thomas	25999.00	Shipped
10	Miley	Cyrus	19349.00	Shipped
11	Olivia	Anderson	4599.00	Pending

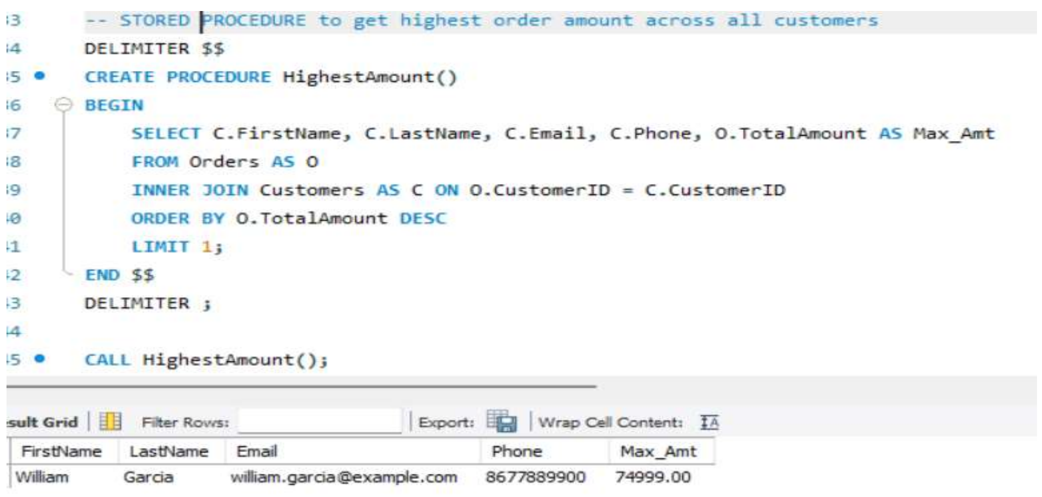
SUBQUERY to Fetch Customers Who Placed Orders Above Rs.10,000:



```sql
-- SUBQUERY to Fetch Customers Who Placed Orders Above 10000
SELECT FirstName, LastName FROM Customers
WHERE CustomerID IN (SELECT CustomerID FROM Orders as O WHERE TotalAmount > 10000);
```

FirstName	LastName
Michael	Johnson
William	Garcia
Sophia	Thomas
Miley	Cyrus
James	Moore

STORED PROCEDURE to get first highest order amount across all customers:

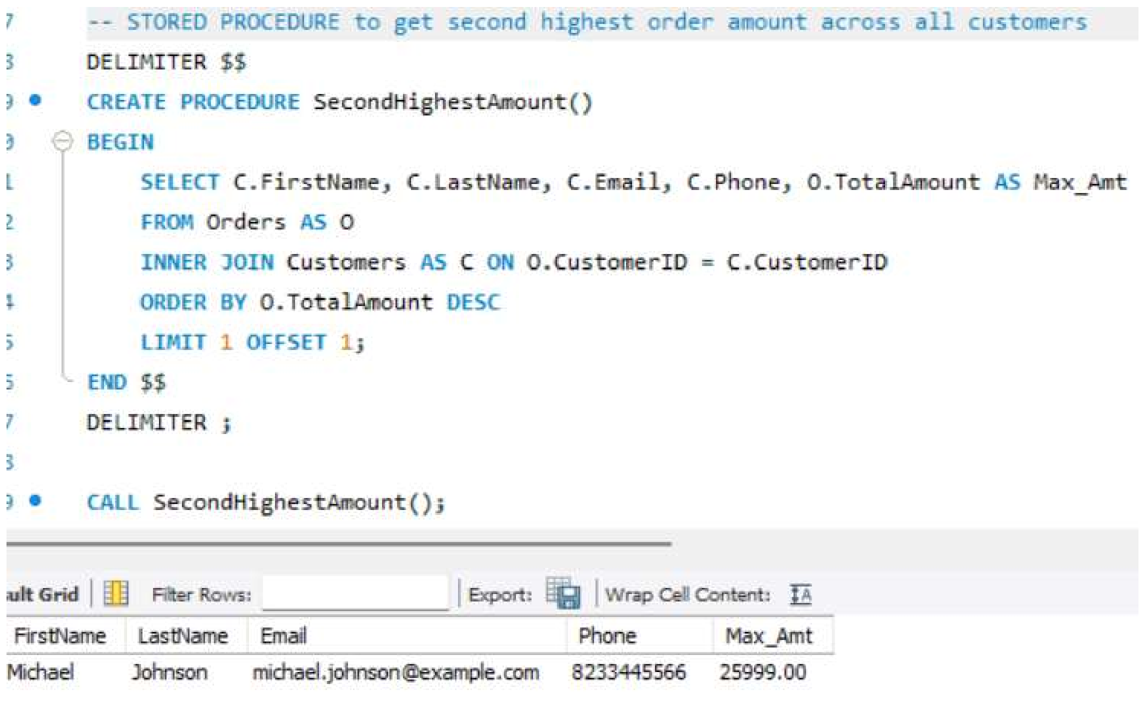


```sql
-- STORED PROCEDURE to get highest order amount across all customers
DELIMITER $$
CREATE PROCEDURE HighestAmount()
BEGIN
    SELECT C.FirstName, C.LastName, C.Email, C.Phone, O.TotalAmount AS Max_Amt
    FROM Orders AS O
    INNER JOIN Customers AS C ON O.CustomerID = C.CustomerID
    ORDER BY O.TotalAmount DESC
    LIMIT 1;
END $$
DELIMITER ;

CALL HighestAmount();
```

FirstName	LastName	Email	Phone	Max_Amt
William	Garcia	william.garcia@example.com	8677889900	74999.00

STORED PROCEDURE to get second highest order amount across all customers:



```
-- STORED PROCEDURE to get second highest order amount across all customers
DELIMITER $$
CREATE PROCEDURE SecondHighestAmount()
BEGIN
    SELECT C.FirstName, C.LastName, C.Email, C.Phone, O.TotalAmount AS Max_Amt
    FROM Orders AS O
    INNER JOIN Customers AS C ON O.CustomerID = C.CustomerID
    ORDER BY O.TotalAmount DESC
    LIMIT 1 OFFSET 1;
END $$
DELIMITER ;

CALL SecondHighestAmount();
```

FirstName	LastName	Email	Phone	Max_Amt
Michael	Johnson	michael.johnson@example.com	8233445566	25999.00