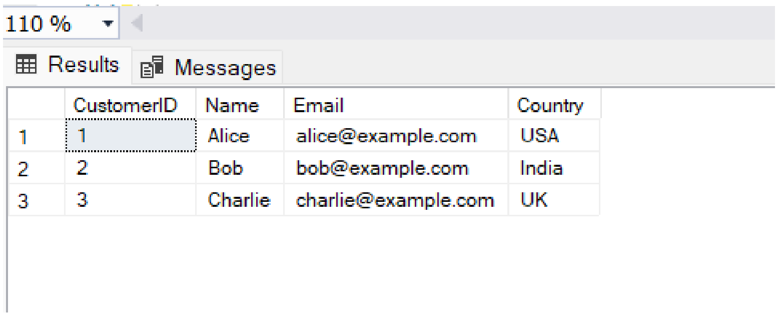
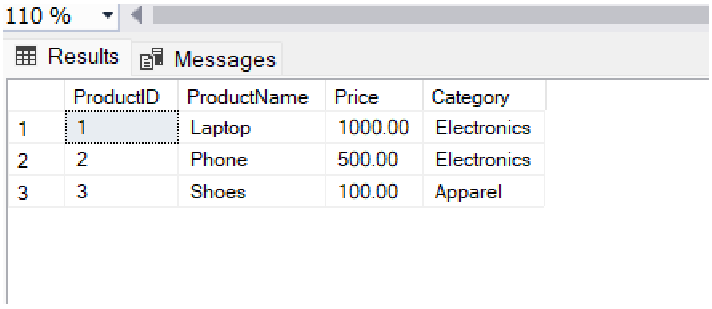
```
create database ecomm
use ecomm

-- Customers table
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Country VARCHAR(50)
);

-- Products table
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Price DECIMAL(10, 2),
    Category VARCHAR(50)
);

-- Orders table
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

-- Customers
INSERT INTO Customers VALUES
(1, 'Alice', 'alice@example.com', 'USA'),
(2, 'Bob', 'bob@example.com', 'India'),
(3, 'Charlie', 'charlie@example.com', 'UK');

-- Products
INSERT INTO Products VALUES
(1, 'Laptop', 1000, 'Electronics'),
(2, 'Phone', 500, 'Electronics'),
(3, 'Shoes', 100, 'Apparel');

-- Orders
INSERT INTO Orders VALUES
(1, 1, '2025-04-01'),
(2, 2, '2025-04-03'),
(3, 1, '2025-04-07');


-- OrderDetails table
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
select * from Customers
```



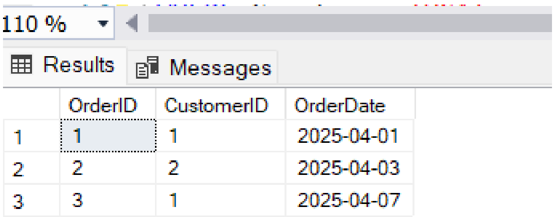
	CustomerID	Name	Email	Country
1	1	Alice	alice@example.com	USA
2	2	Bob	bob@example.com	India
3	3	Charlie	charlie@example.com	UK

```
select * from  Products
```



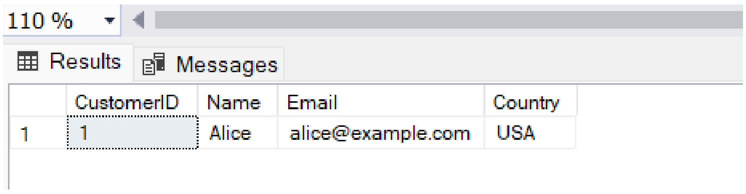
	ProductID	ProductName	Price	Category
1	1	Laptop	1000.00	Electronics
2	2	Phone	500.00	Electronics
3	3	Shoes	100.00	Apparel

```
select * from  Orders
```



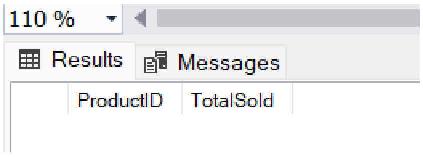
	OrderID	CustomerID	OrderDate
1	1	1	2025-04-01
2	2	2	2025-04-03
3	3	1	2025-04-07

```
-- Get customers from USA, ordered by name
SELECT * FROM Customers
WHERE Country = 'USA'
ORDER BY Name;
```



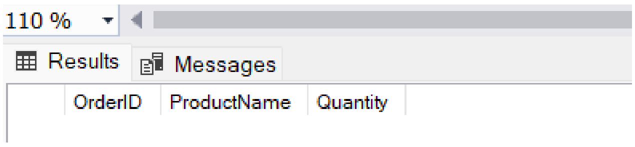
	CustomerID	Name	Email	Country
1	1	Alice	alice@example.com	USA

```
-- Total quantity of products ordered grouped by ProductID
SELECT ProductID, SUM(Quantity) AS TotalSold
FROM OrderDetails
GROUP BY ProductID;
```



	ProductID	TotalSold

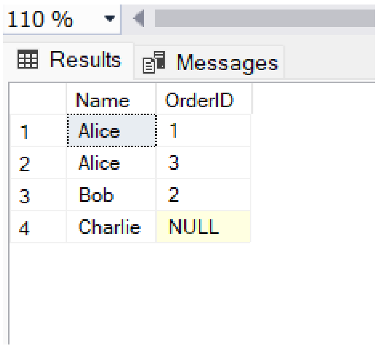
```
-- INNER JOIN to get order details with product names
SELECT o.OrderID, p.ProductName, od.Quantity
FROM Orders o
INNER JOIN OrderDetails od ON o.OrderID = od.OrderID
INNER JOIN Products p ON od.ProductID = p.ProductID;
```



	OrderID	ProductName	Quantity

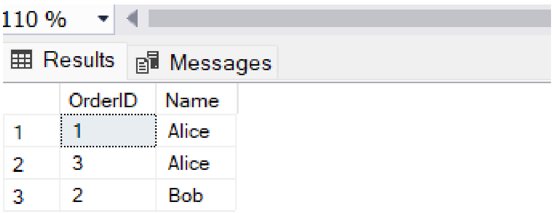
```
-- LEFT JOIN: Show all customers with their orders (if any)
SELECT c.Name, o.OrderID
```

```
FROM Customers c

LEFT JOIN Orders o ON c.CustomerID = o.CustomerID;
```



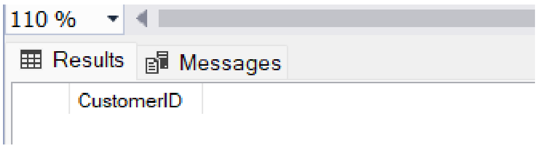
```
   Name     OrderID
1  Alice    1
2  Alice    3
3  Bob      2
4  Charlie  NULL
```

```
-- RIGHT JOIN: Show all orders and corresponding customers

SELECT o.OrderID, c.Name

FROM Customers c

RIGHT JOIN Orders o ON c.CustomerID = o.CustomerID;
```



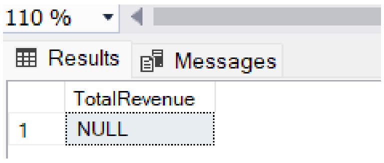
```
   OrderID  Name
1  1        Alice
2  3        Alice
3  2        Bob
```

```
-- Customers who ordered more than 1 product in a single order

SELECT DISTINCT CustomerID

FROM Orders

WHERE OrderID IN (

    SELECT OrderID

    FROM OrderDetails

    GROUP BY OrderID

    HAVING SUM(Quantity) > 1

);
```



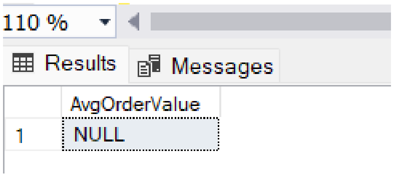
	CustomerID

```
-- Total revenue generated
SELECT SUM(p.Price * od.Quantity) AS TotalRevenue
FROM OrderDetails od
JOIN Products p ON od.ProductID = p.ProductID;
```



	TotalRevenue
1	NULL

```
-- Average order value
SELECT AVG(OrderValue) AS AvgOrderValue
FROM (
    SELECT o.OrderID, SUM(p.Price * od.Quantity) AS OrderValue
    FROM Orders o
    JOIN OrderDetails od ON o.OrderID = od.OrderID
    JOIN Products p ON od.ProductID = p.ProductID
    GROUP BY o.OrderID
) AS OrderValues;
```



	AvgOrderValue
1	NULL

```
IF OBJECT_ID('ProductSales', 'V') IS NOT NULL
```

```
    DROP VIEW ProductSales;
GO

CREATE VIEW ProductSales AS
SELECT p.ProductName, SUM(od.Quantity) AS TotalSold
FROM Products p
JOIN OrderDetails od ON p.ProductID = od.ProductID
GROUP BY p.ProductName;
GO

IF OBJECT_ID('CustomerOrderSummary', 'V') IS NOT NULL
    DROP VIEW CustomerOrderSummary;
GO

CREATE VIEW CustomerOrderSummary AS
SELECT c.Name, COUNT(o.OrderID) AS TotalOrders
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID
GROUP BY c.Name;
GO


-- Create index on foreign keys and search columns
CREATE INDEX idx_orders_customerid ON Orders(CustomerID);
CREATE INDEX idx_orderdetails_orderid ON OrderDetails(OrderID);
CREATE INDEX idx_orderdetails_productid ON OrderDetails(ProductID);
CREATE INDEX idx_products_category ON Products(Category);
```