**Customer\_Orders\_Products Database**

**By**

**Sachin Gunjal**

CREATE DATABASE Customers\_Orders\_Products

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

Name VARCHAR(50),

Email VARCHAR(100)

);

INSERT INTO Customers (CustomerID, Name, Email)

VALUES

(1, 'John Doe', 'johndoe@example.com'),

(2, 'Jane Smith', 'janesmith@example.com'),

(3, 'Robert Johnson', 'robertjohnson@example.com'),

(4, 'Emily Brown', 'emilybrown@example.com'),

(5, 'Michael Davis', 'michaeldavis@example.com'),

(6, 'Sarah Wilson', 'sarahwilson@example.com'),

(7, 'David Thompson', 'davidthompson@example.com'),

(8, 'Jessica Lee', 'jessicalee@example.com'),

(9, 'William Turner', 'williamturner@example.com'),

(10, 'Olivia Martinez', 'oliviamartinez@example.com'),

(11, 'James Anderson', 'jamesanderson@example.com'),

(12, 'Kelly Clarkson', 'kellyclarkson@example.com');

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

ProductName VARCHAR(50),

OrderDate DATE,

Quantity INT,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

INSERT INTO Orders (OrderID, CustomerID, ProductName, OrderDate, Quantity)

VALUES

(1, 1, 'Product A', '2023-07-01', 5),

(2, 2, 'Product B', '2023-07-02', 3),

(3, 3, 'Product C', '2023-07-03', 2),

(4, 4, 'Product A', '2023-07-04', 1),

(5, 5, 'Product B', '2023-07-05', 4),

(6, 6, 'Product C', '2023-07-06', 2),

(7, 7, 'Product A', '2023-07-07', 3),

(8, 8, 'Product B', '2023-07-08', 2),

(9, 9, 'Product C', '2023-07-09', 5),

(10, 10, 'Product A', '2023-07-10', 1),

(11, 11, 'Product D', '2023-07-10', 3),

(12, 12, 'Product E', '2023-07-11', 6),

(13, 5, 'Product G', '2023-07-12', 2),

(14, 4, 'Product H', '2023-07-13', 4),

(15, 6, 'Product I', '2023-07-14', 3);

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(50),

Price DECIMAL(10, 2)

);

INSERT INTO Products (ProductID, ProductName, Price)

VALUES

(1, 'Product A', 10.99),

(2, 'Product B', 8.99),

(3, 'Product C', 5.99),

(4, 'Product D', 12.99),

(5, 'Product E', 7.99),

(6, 'Product F', 6.99),

(7, 'Product G', 9.99),

(8, 'Product H', 11.99),

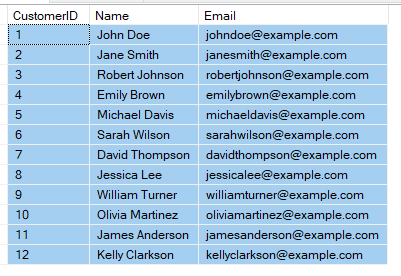
(9, 'Product I', 14.99),

(10, 'Product J', 4.99),

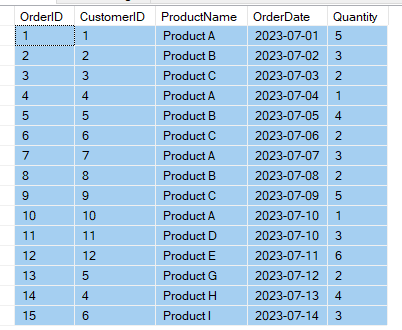
(11, 'Product K', 3.99),

(12, 'Product L', 15.99);

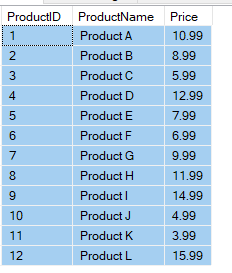
SELECT \* FROM Customers



SELECT \* FROM Orders



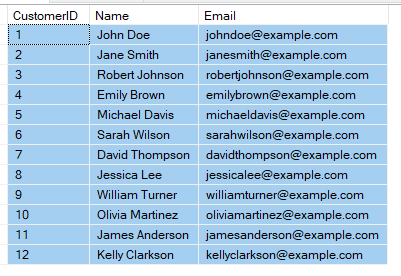
SELECT \* FROM Products



**Task 1**

**1. Write a query to retrieve all records from the Customers table**.

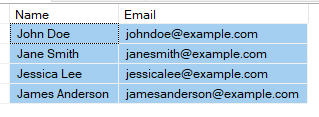
SELECT \* FROM Customers



**2. Write a query to retrieve the names and email addresses of customers whose names start with 'J'.**

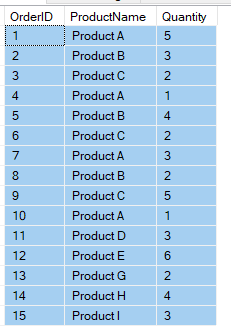
SELECT Name, Email FROM Customers

WHERE name LIKE 'J%';



**3. Write a query to retrieve the order details (OrderID, ProductName, Quantity) for all orders.**

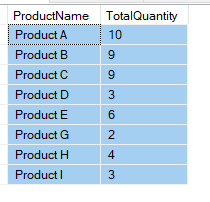
SELECT OrderID, ProductName,Quantity FROM Orders



**4.Write a query to calculate the total quantity of products ordered.**

SELECT ProductName, SUM(Quantity) AS TotalQuantity FROM Orders

GROUP BY ProductName



**5. Write a query to retrieve the names of customers who have placed an order.**

SELECT O.CustomerID, C.Name FROM Customers C

INNER JOIN

Orders O

ON

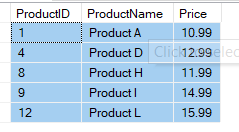
C.CustomerID = O.CustomerID



**6. Write a query to retrieve the products with a price greater than $10.00.**

SELECT \* FROM Products

WHERE Price > 10.00



**7. Write a query to retrieve the customer name and order date for all orders placed on or after '2023-07-05'.**

SELECT O.CustomerID, C.Name, O.OrderDate FROM Customers C

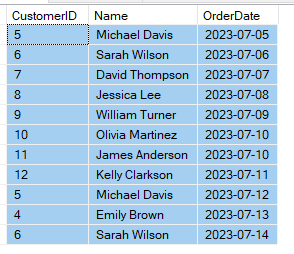
INNER JOIN

Orders O

ON

C.CustomerID = O.CustomerID

WHERE OrderDate >= '2023-07-05'



**8. Write a query to calculate the average price of all products.**

SELECT AVG(Price) AS AveragePrice

FROM Products



**9. Write a query to retrieve the customer names along with the total quantity of products they have ordered.**

SELECT C.Name, SUM(O.Quantity) as TotalQuantity FROM Customers C

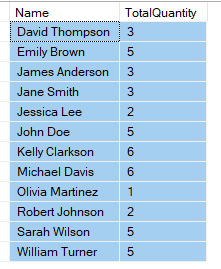
INNER JOIN

Orders O

ON

C.CustomerID = O.CustomerID

GROUP BY c.Name



**10. Write a query to retrieve the products that have not been ordered.**

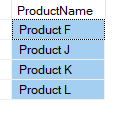
SELECT ProductName

FROM Products

WHERE ProductName NOT IN (

SELECT DISTINCT ProductName

FROM Orders );



**Task 2**

**1. Write a query to retrieve the top 5 customers who have placed the highest total quantity of orders.**

SELECT TOP 5 C.Name, SUM(O.Quantity) AS TotalQuantity

FROM Customers C

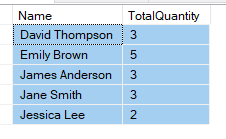
INNER JOIN

Orders O

ON

C.CustomerID = O.CustomerID

GROUP BY C.Name



**2. Write a query to calculate the average price of products for each product category.**

SELECT AVG(Price) AS AveragePrice FROM Products;



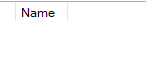
**3. Write a query to retrieve the customers who have not placed any orders.**

SELECT C.Name

FROM Customers C

LEFT JOIN Orders O ON C.CustomerID = O.CustomerID

WHERE O.OrderID IS NULL;



**4. Write a query to retrieve the order details (OrderID, ProductName, Quantity) for orders placed by customers whose names start with 'M'.**

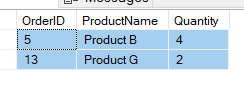
SELECT O.OrderID, O.ProductName, O.Quantity

FROM Customers C

INNER JOIN Orders O

ON C.CustomerID = O.CustomerID

WHERE C.Name LIKE 'M%';



**5. Write a query to calculate the total revenue generated from all orders.**

SELECT SUM(O.QUANTITY \* P.Price) AS TotalRevenue FROM ORDERS O

INNER JOIN

Products P

ON

O.ProductName = P.ProductName



**6. Write a query to retrieve the customer names along with the total revenue generated from their orders.**

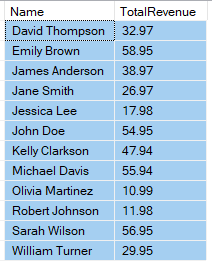
SELECT C.Name, SUM(O.Quantity \* P.Price) AS TotalRevenue

FROM Customers C

JOIN Orders O ON C.CustomerID = O.CustomerID

JOIN Products P ON O.ProductName = P.ProductName

GROUP BY C.Name;



**7. Write a query to retrieve the customers who have placed at least one order for each product category.**

SELECT C.CustomerID, C.Name

FROM Customers C

INNER JOIN Orders O

ON C.CustomerID = O.CustomerID

GROUP BY C.CustomerID, C.Name

HAVING COUNT(DISTINCT O.ProductName) = (SELECT COUNT(\*) FROM Products)



**8. Write a query to retrieve the customers who have placed orders on consecutive days.**

SELECT DISTINCT C.Name

FROM Orders O1

JOIN Orders O2 ON O1.CustomerID = O2.CustomerID

AND DATEDIFF(DAY, O1.OrderDate, O2.OrderDate) = 1

JOIN Customers C ON O1.CustomerID = C.CustomerID;

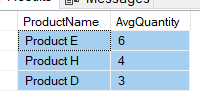


**9. Write a query to retrieve the top 3 products with the highest average quantity ordered.**

SELECT TOP 3 ProductName, AVG(QUANTITY) AS AvgQuantity FROM Orders

GROUP BY ProductName

ORDER BY AVG(QUANTITY) DESC



**10. Write a query to calculate the percentage of orders that have a quantity greater than the average quantity.**

SELECT ROUND(100.0 \* COUNT(\*) / (

SELECT COUNT(ProductName) FROM Orders), 2) AS TotalPercentage

FROM ORDERS

WHERE Quantity >

(SELECT AVG(QUANTITY) FROM Orders)

