**SAFERTEK-BACKEND SQL**

**Name:** Venkata Phani Charan Reddy Udumula

**ID:** 2100031346

CREATE DATABASE safertek;

USE safertek;

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

Email VARCHAR(100),

DateOfBirth DATE

);

CREATE TABLE Products (

ProductID INT PRIMARY KEY,

ProductName VARCHAR(100),

Price DECIMAL(10, 2)

);

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

OrderDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

CREATE TABLE OrderItems (

OrderItemID INT PRIMARY KEY,

OrderID INT,

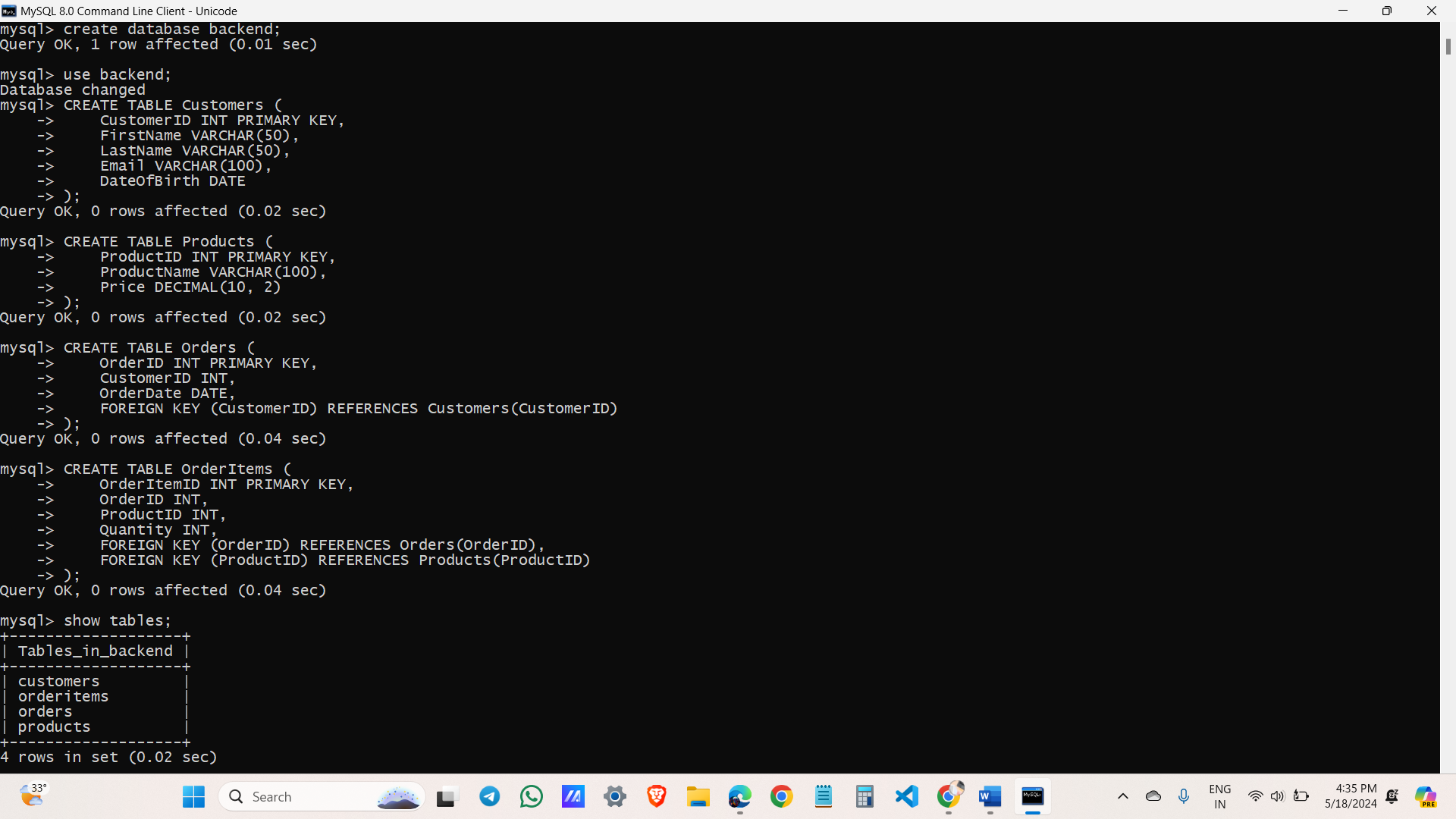
ProductID INT,

Quantity INT,

FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),

FOREIGN KEY (ProductID) REFERENCES Products(ProductID)

);



INSERT INTO Customers (CustomerID, FirstName, LastName, Email, DateOfBirth) VALUES

(1, 'John', 'Doe', 'john.doe@example.com', '1985-01-15'),

(2, 'Jane', 'Smith', 'jane.smith@example.com', '1990-06-20');

INSERT INTO Products (ProductID, ProductName, Price) VALUES

(1, 'Laptop', 1000.00),

(2, 'Smartphone', 600.00),

(3, 'Headphones', 100.00);

INSERT INTO Orders (OrderID, CustomerID, OrderDate) VALUES

(1, 1, '2023-01-10'),

(2, 2, '2023-01-12');

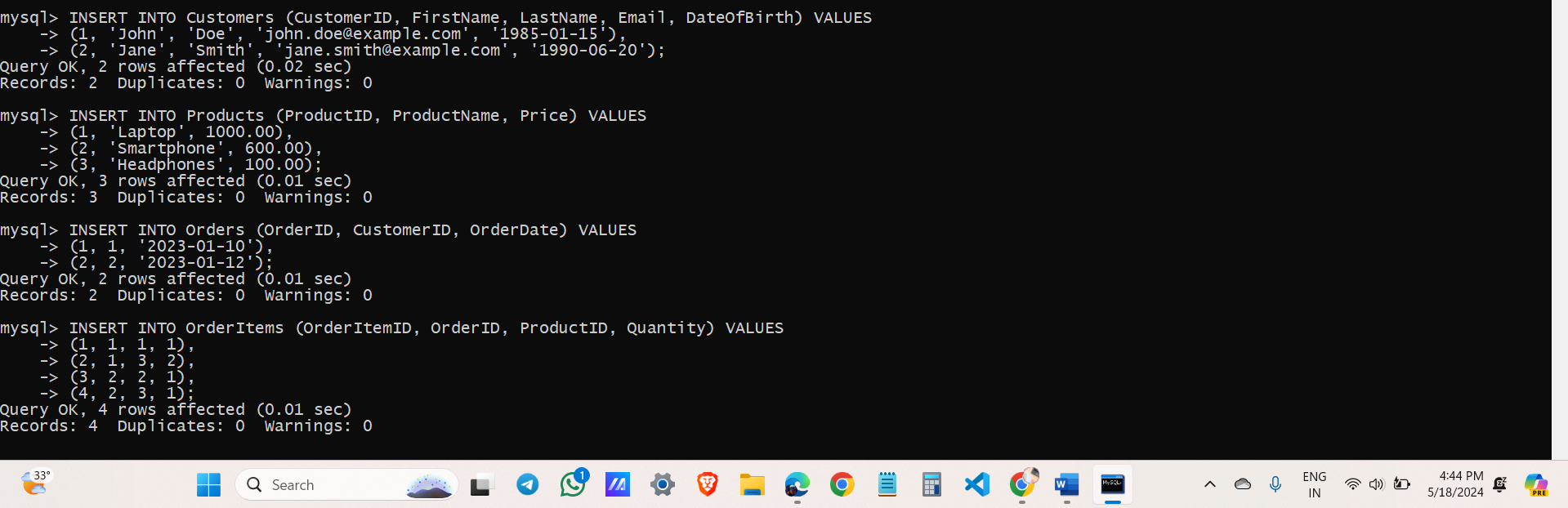
INSERT INTO OrderItems (OrderItemID, OrderID, ProductID, Quantity) VALUES

(1, 1, 1, 1),

(2, 1, 3, 2),

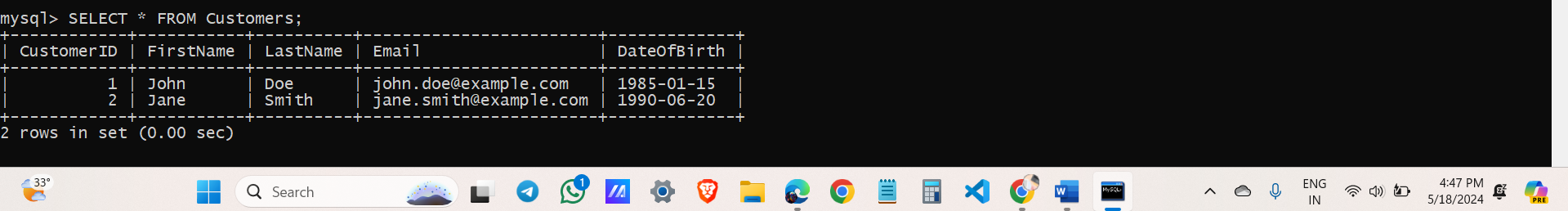
(3, 2, 2, 1),

(4, 2, 3, 1);



1. List all customers.

SELECT \* FROM Customers;



2. Find all orders placed in January 2023.

SELECT \* FROM Orders

WHERE OrderDate >= '2023-01-01' AND OrderDate <= '2023-01-31';

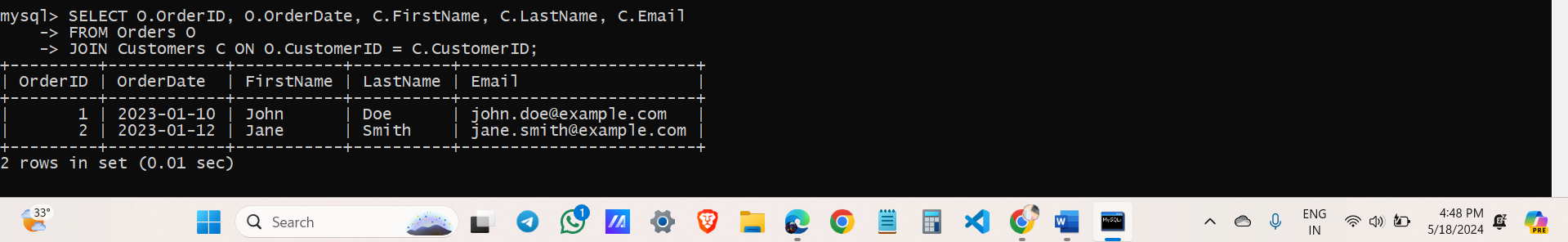


3. Get the details of each order, including the customer name and email.

SELECT O.OrderID, O.OrderDate, C.FirstName, C.LastName, C.Email

FROM Orders O

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



4. List the products purchased in a specific order (e.g., OrderID = 1).

SELECT P.ProductName, OI.Quantity

FROM OrderItems OI

JOIN Products P ON OI.ProductID = P.ProductID

WHERE OI.OrderID = 1;



5. Calculate the total amount spent by each customer.

SELECT C.CustomerID, C.FirstName, C.LastName, C.Email, SUM(P.Price \* OI.Quantity) AS TotalAmountSpent

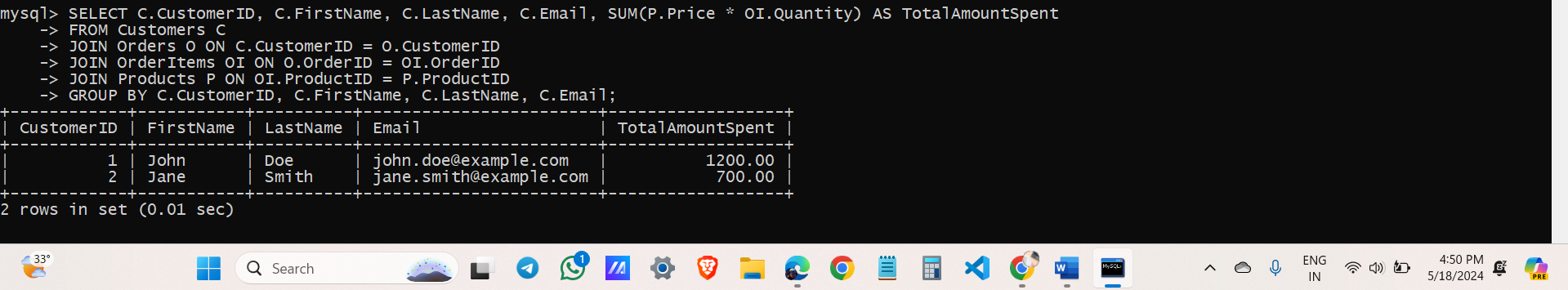
FROM Customers C

JOIN Orders O ON C.CustomerID = O.CustomerID

JOIN OrderItems OI ON O.OrderID = OI.OrderID

JOIN Products P ON OI.ProductID = P.ProductID

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



6. Find the most popular product (the one that has been ordered the most).

SELECT P.ProductID, P.ProductName, SUM(OI.Quantity) AS TotalQuantityOrdered

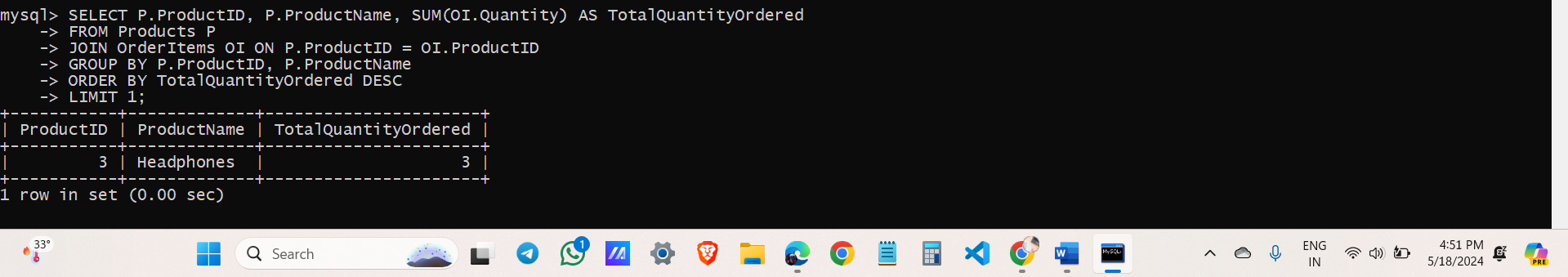
FROM Products P

JOIN OrderItems OI ON P.ProductID = OI.ProductID

GROUP BY P.ProductID, P.ProductName

ORDER BY TotalQuantityOrdered DESC

LIMIT 1;



7. Get the total number of orders and the total sales amount for each month in 2023.

SELECT

YEAR(OrderDate) AS Year,

MONTH(OrderDate) AS Month,

COUNT(\*) AS TotalOrders,

SUM(Price \* Quantity) AS TotalSalesAmount

FROM

Orders O

JOIN

OrderItems ON O.OrderID = OrderItems.OrderID

JOIN

Products ON OrderItems.ProductID = Products.ProductID

WHERE

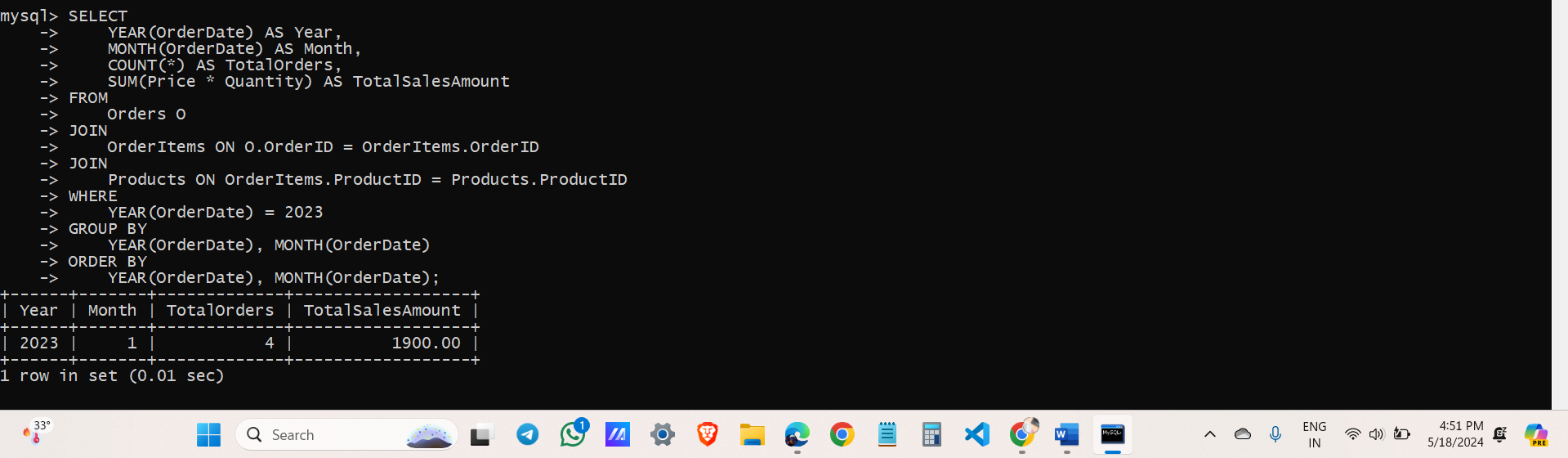
YEAR(OrderDate) = 2023

GROUP BY

YEAR(OrderDate), MONTH(OrderDate)

ORDER BY

YEAR(OrderDate), MONTH(OrderDate);



8. Find customers who have spent more than $1000.

SELECT

C.CustomerID,

C.FirstName,

C.LastName,

C.Email,

SUM(P.Price \* OI.Quantity) AS TotalSpent

FROM

Customers C

JOIN

Orders O ON C.CustomerID = O.CustomerID

JOIN

OrderItems OI ON O.OrderID = OI.OrderID

JOIN

Products P ON OI.ProductID = P.ProductID

GROUP BY

C.CustomerID, C.FirstName, C.LastName, C.Email

HAVING

TotalSpent > 1000;

