Task:- Safertek backend

SQL QUERIES

create database sarftekbackend;

show databases;

use sarftekbackend;

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),

(2, 'Smartphone', 600),

(3, 'Headphones', 100);

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),

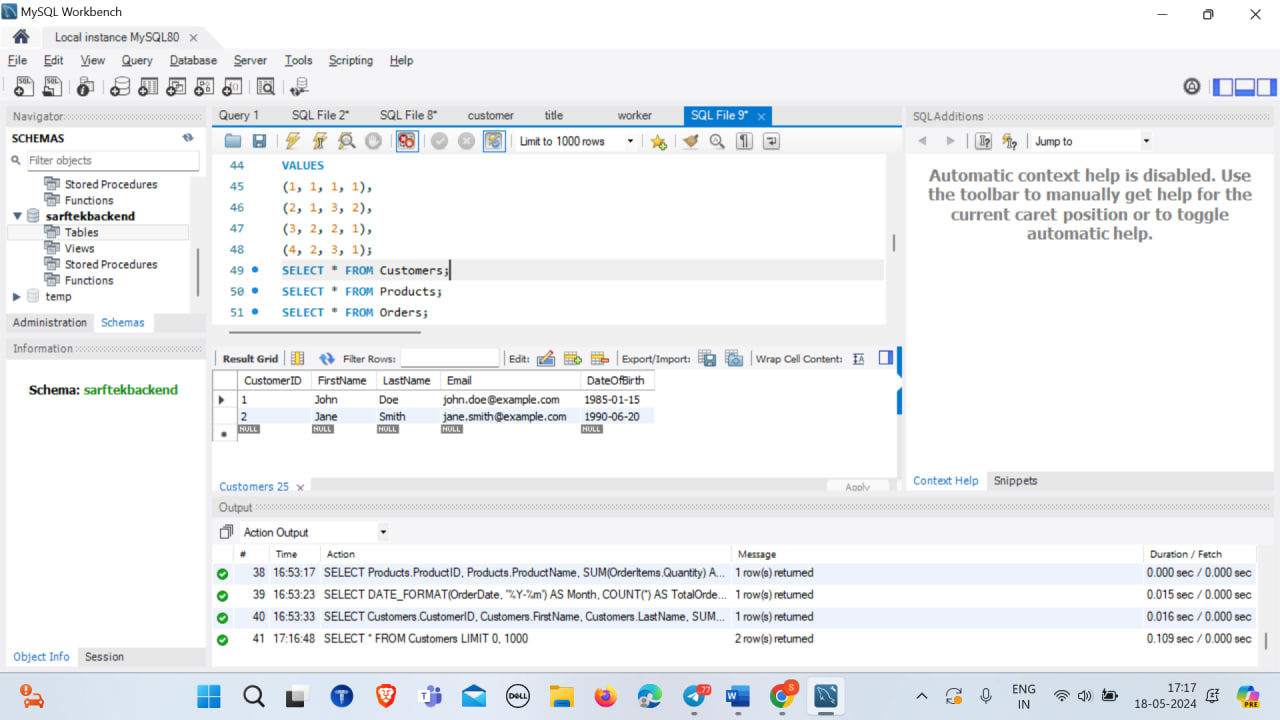
(2, 1, 3, 2),

(3, 2, 2, 1),

(4, 2, 3, 1);

SELECT \* FROM Customers;

Output of Customer Table:-



Query for Products table

SELECT \* FROM Products;

Output of Products Table:-

A screenshot of a computer

Description automatically generated

Query for Order Table

SELECT \* FROM Orders;

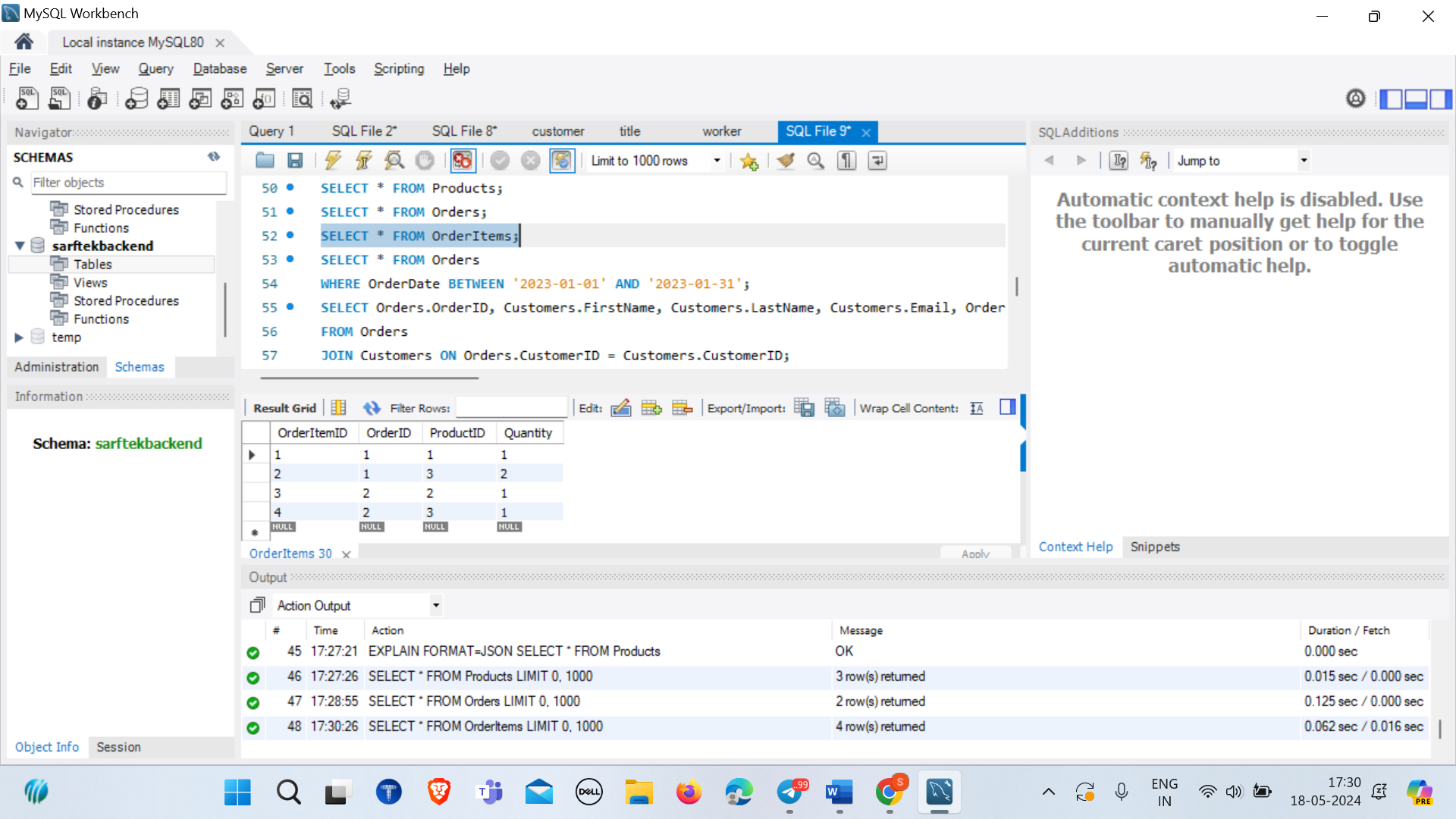
Output of Order Table

A screenshot of a computer

Description automatically generated

Query for Order items Table

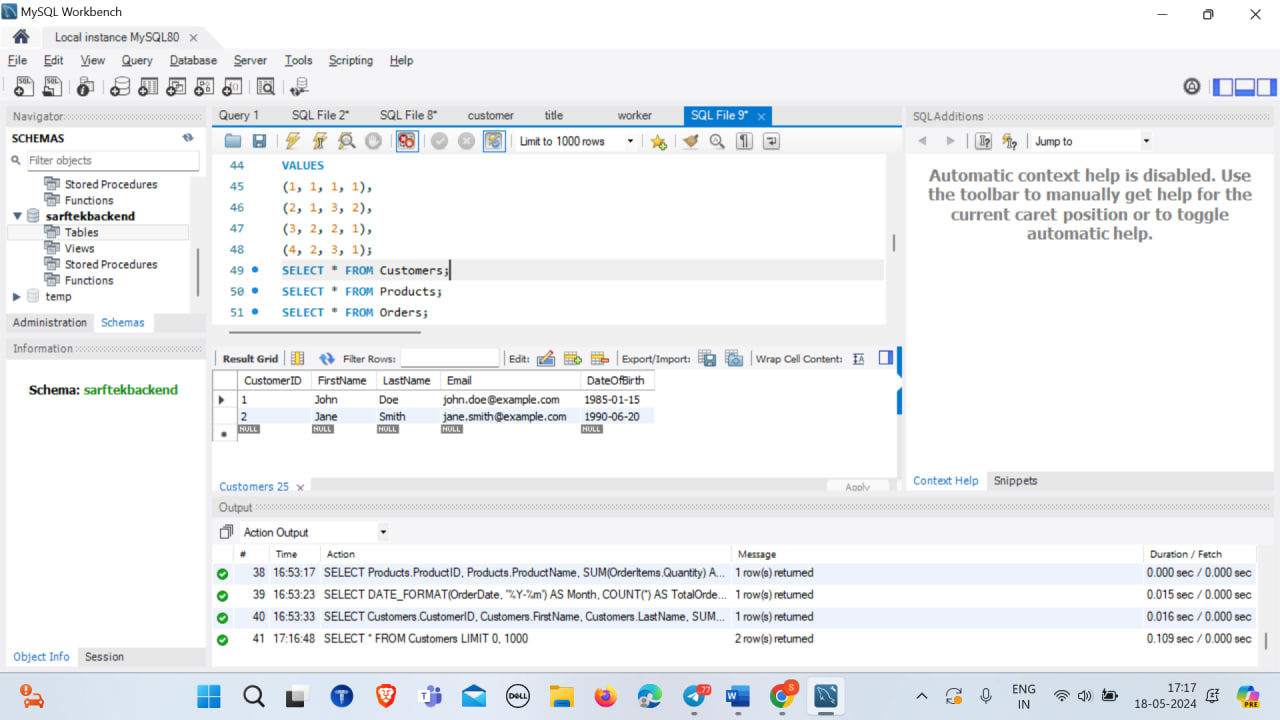
SELECT \* FROM OrderItems;



Sample Queries

1. List all customers.

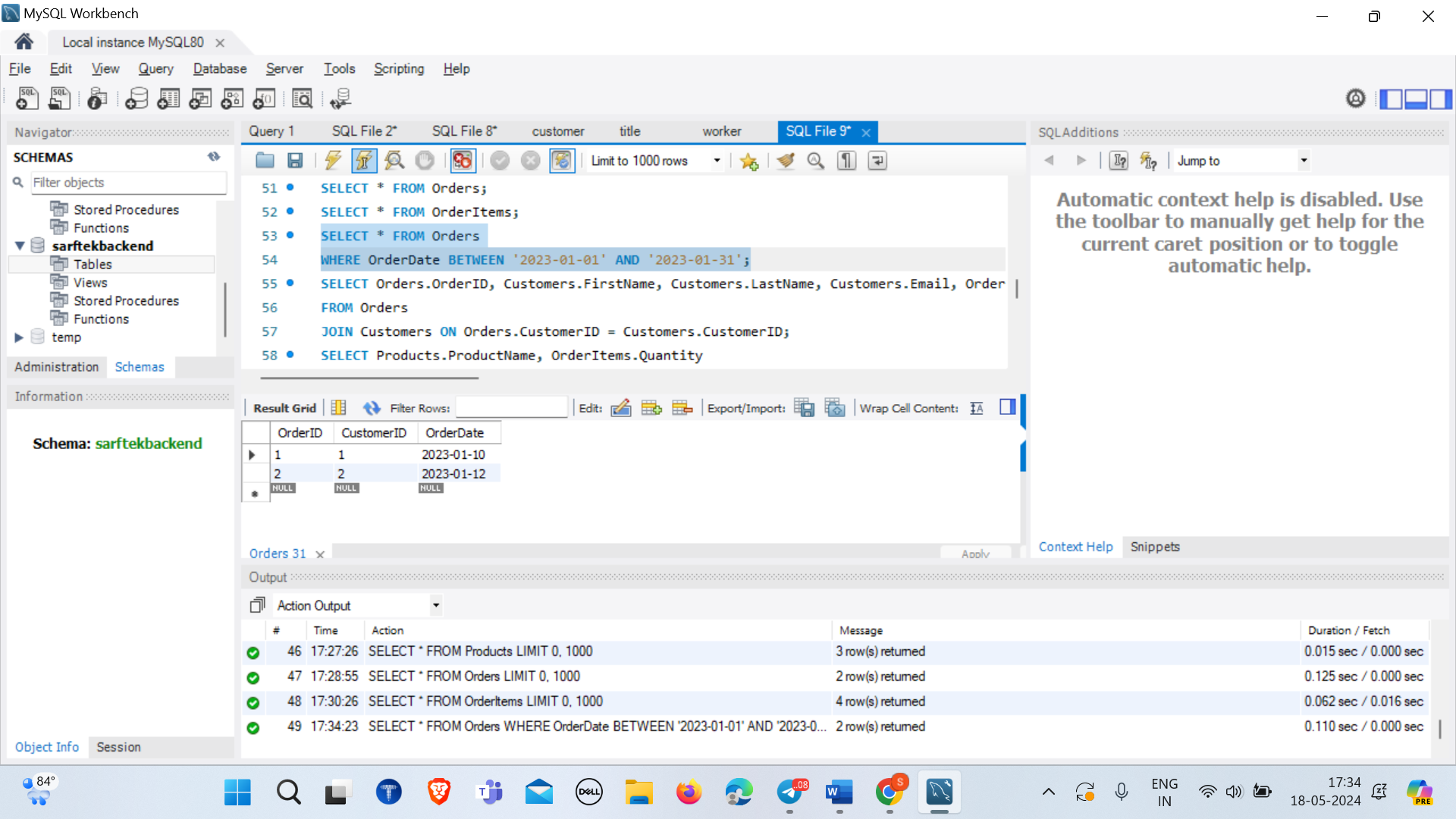
SELECT \* FROM Customers;



2. Find all orders placed in January 2023.

SELECT \* FROM Orders

WHERE OrderDate BETWEEN '2023-01-01' AND '2023-01-31';

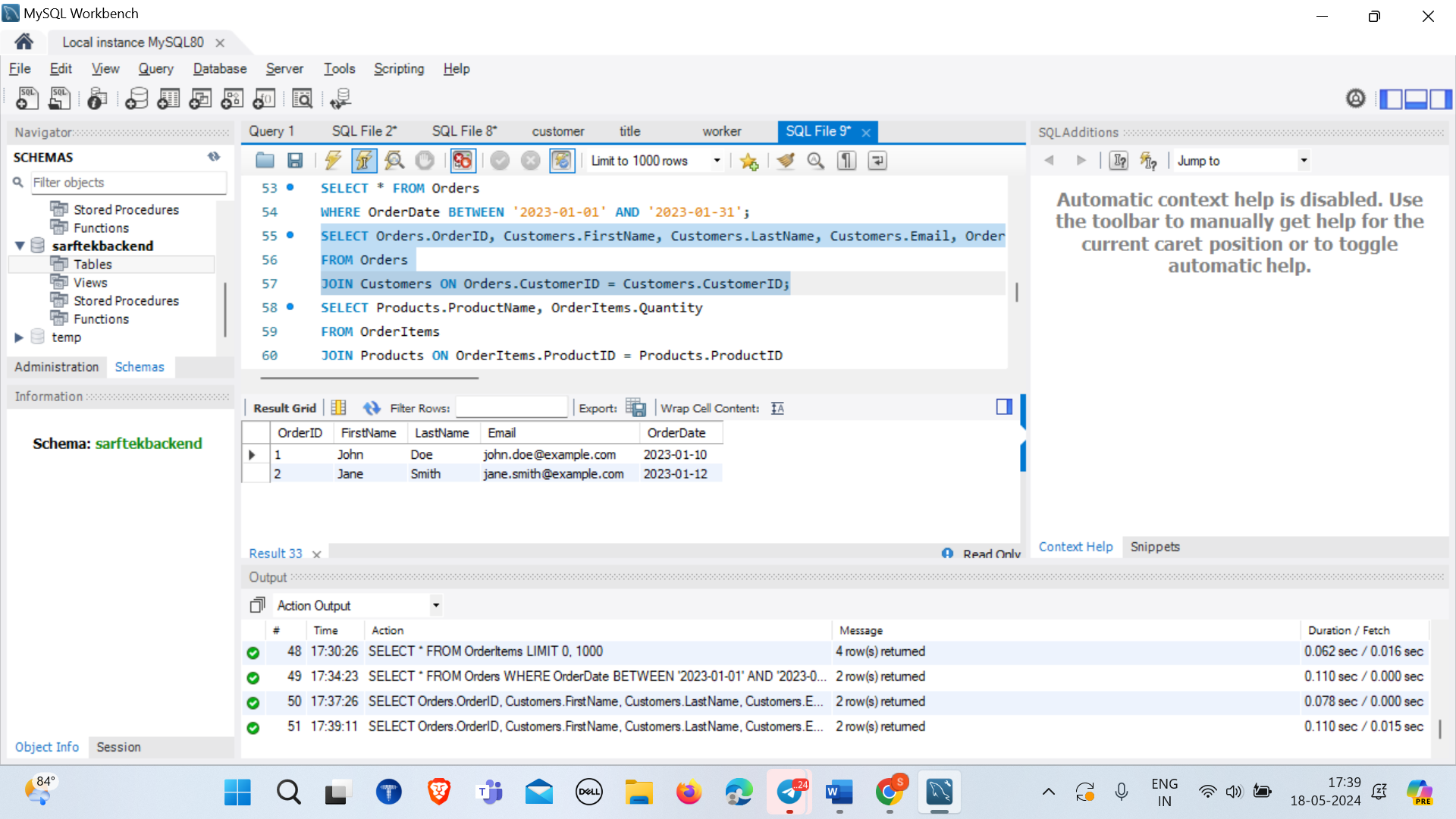


3. Get the details of each order, including the customer name and email.

SELECT Orders.OrderID, Customers.FirstName, Customers.LastName, Customers.Email, Orders.OrderDate

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID;



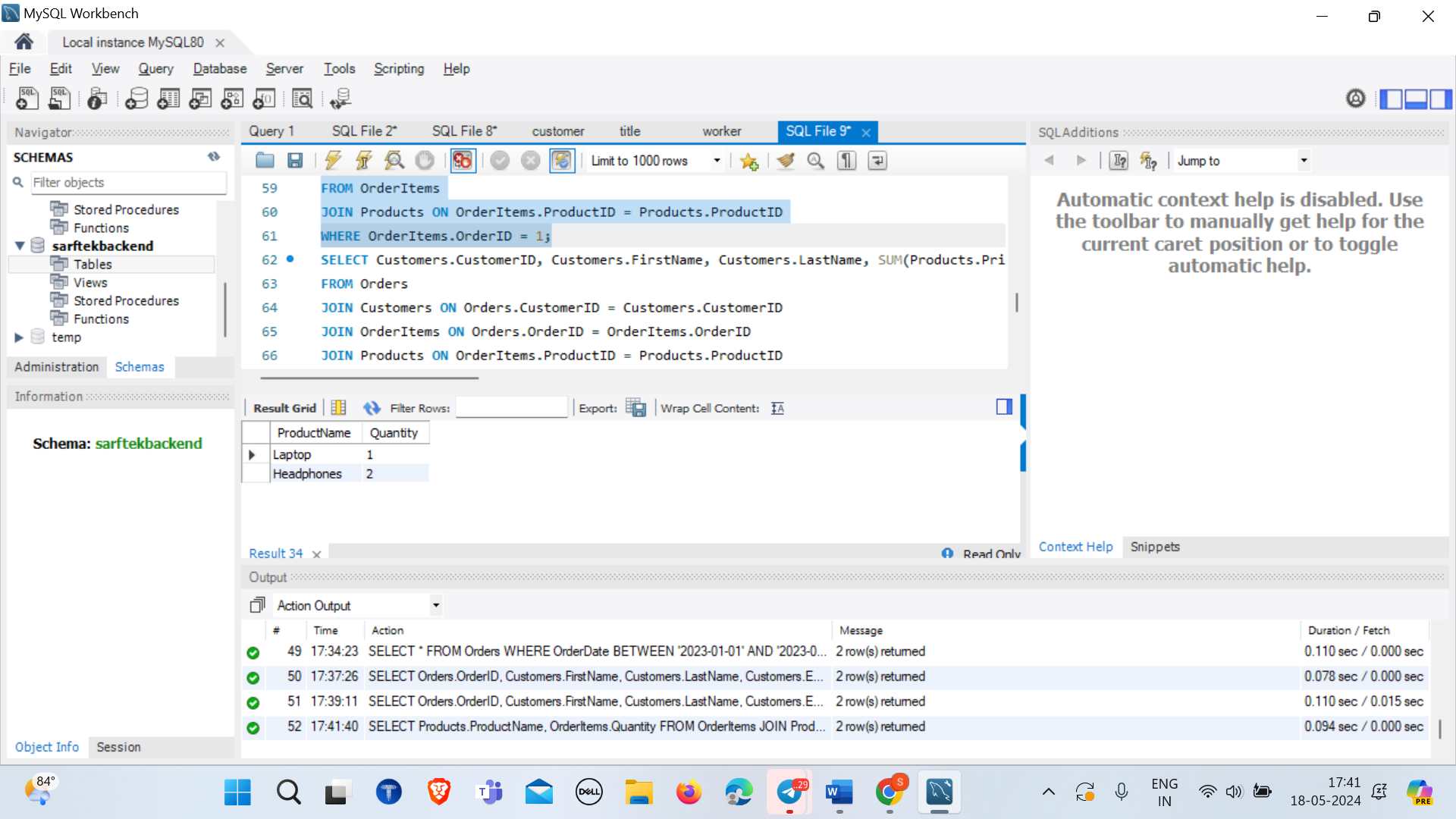
4. List the products purchased in a specific order (e.g., OrderID = 1).

SELECT Products.ProductName, OrderItems.Quantity

FROM OrderItems

JOIN Products ON OrderItems.ProductID = Products.ProductID

WHERE OrderItems.OrderID = 1;



5. Calculate the total amount spent by each customer.

SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName, SUM(Products.Price \* OrderItems.Quantity) AS TotalAmountSpent

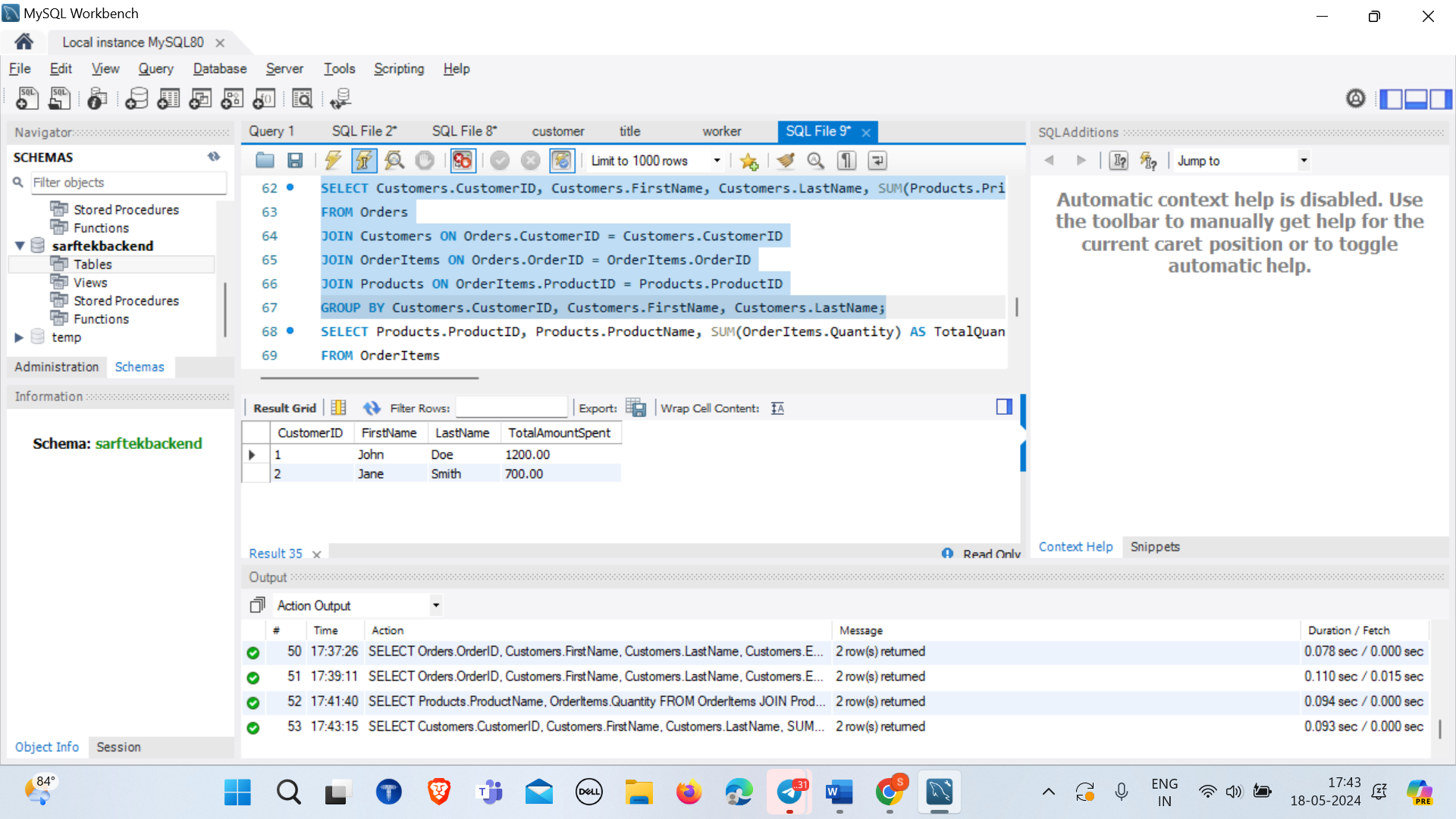
FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID

JOIN Products ON OrderItems.ProductID = Products.ProductID

GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName;



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

SELECT Products.ProductID, Products.ProductName, SUM(OrderItems.Quantity) AS TotalQuantityOrdered

FROM OrderItems

JOIN Products ON OrderItems.ProductID = Products.ProductID

GROUP BY Products.ProductID, Products.ProductName

ORDER BY TotalQuantityOrdered DESC

LIMIT 1;

A screenshot of a computer

Description automatically generated

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

SELECT DATE\_FORMAT(OrderDate, '%Y-%m') AS Month, COUNT(\*) AS TotalOrders, SUM(Products.Price \* OrderItems.Quantity) AS TotalSalesAmount

FROM Orders

JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID

JOIN Products ON OrderItems.ProductID = Products.ProductID

WHERE YEAR(OrderDate) = 2023

GROUP BY DATE\_FORMAT(OrderDate, '%Y-%m');

A screenshot of a computer

Description automatically generated

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

SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName, SUM(Products.Price \* OrderItems.Quantity) AS TotalAmountSpent

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

JOIN OrderItems ON Orders.OrderID = OrderItems.OrderID

JOIN Products ON OrderItems.ProductID = Products.ProductID

GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName

HAVING TotalAmountSpent > 1000;

A screenshot of a computer

Description automatically generated