**Assignment 4**

--Assignment 4

1. 1st Create a stored procedure in the Northwind database that will calculate the average value of Freight for a specified customer.Then, a business rule will be added that will be triggered before every Update and Insert command in the Orders controller,and will use the stored procedure to verify that the Freight does not exceed the average freight. If it does, a message will be displayed and the command will be cancelled.

CREATE PROCEDURE spCheckFreightWithAvg

@customer\_id nvarchar(5),

@avg\_freight money output

AS

BEGIN

SELECT @avg\_freight = AVG(Freight)

FROM Orders

WHERE CustomerID = @customer\_id;

END

--trigger for insert data in Orders Table

CREATE TRIGGER trVerifyOnInsert ON Orders

INSTEAD OF INSERT

AS

BEGIN

DECLARE @cust\_id nvarchar(5)

DECLARE @freight money

DECLARE @avgFreight money

SELECT @cust\_id = CustomerID FROM INSERTED

SELECT @freight = Freight FROM INSERTED

EXEC spCheckFreightWithAvg @cust\_id, @avgFreight output

IF @freight IS NOT NULL AND @freight >= @avgFreight

BEGIN

RAISERROR ('Freight value exceeds the average freight value' ,10,1)

END

ELSE

BEGIN

INSERT INTO Orders("CustomerID","EmployeeID","OrderDate","RequiredDate",

"ShippedDate","ShipVia","Freight","ShipName","ShipAddress",

"ShipCity","ShipRegion","ShipPostalCode","ShipCountry")

VALUES((SELECT CustomerID FROM INSERTED),(SELECT EmployeeID FROM INSERTED),

(SELECT OrderDate FROM INSERTED),(SELECT RequiredDate FROM INSERTED),

(SELECT ShippedDate FROM INSERTED),(SELECT ShipVia FROM INSERTED),

(SELECT Freight FROM INSERTED),(SELECT ShipName FROM INSERTED),

(SELECT ShipAddress FROM INSERTED),(SELECT ShipCity FROM INSERTED),

(SELECT ShipRegion FROM INSERTED),(SELECT ShipPostalCode FROM INSERTED),

(SELECT ShipCountry FROM INSERTED));

END

END

--trigger for Update data in Orders Table

CREATE TRIGGER trVerifyOnUpdate ON Orders

INSTEAD OF UPDATE

AS

BEGIN

DECLARE @cust\_id nvarchar(5)

DECLARE @oldfreight money

DECLARE @newfreight money

DECLARE @avgFreight money

SELECT @cust\_id = CustomerID FROM INSERTED

SELECT @oldfreight = Freight FROM Orders

SELECT @newfreight = Freight FROM INSERTED

IF @oldfreight = @newfreight

BEGIN

return;

END

ELSE

BEGIN

EXEC spCheckFreightWithAvg @cust\_id, @avgFreight output

IF @newfreight IS NOT NULL AND @newfreight >= @avgFreight

BEGIN

RAISERROR ('Freight value exceeds the average freight value' ,10,1)

END

ELSE

BEGIN

UPDATE Orders SET Freight = (SELECT Freight FROM INSERTED) WHERE CustomerID = @cust\_id;

END

END

END

INSERT INTO Orders("CustomerID","EmployeeID","OrderDate","RequiredDate",

"ShippedDate","ShipVia","Freight","ShipName","ShipAddress",

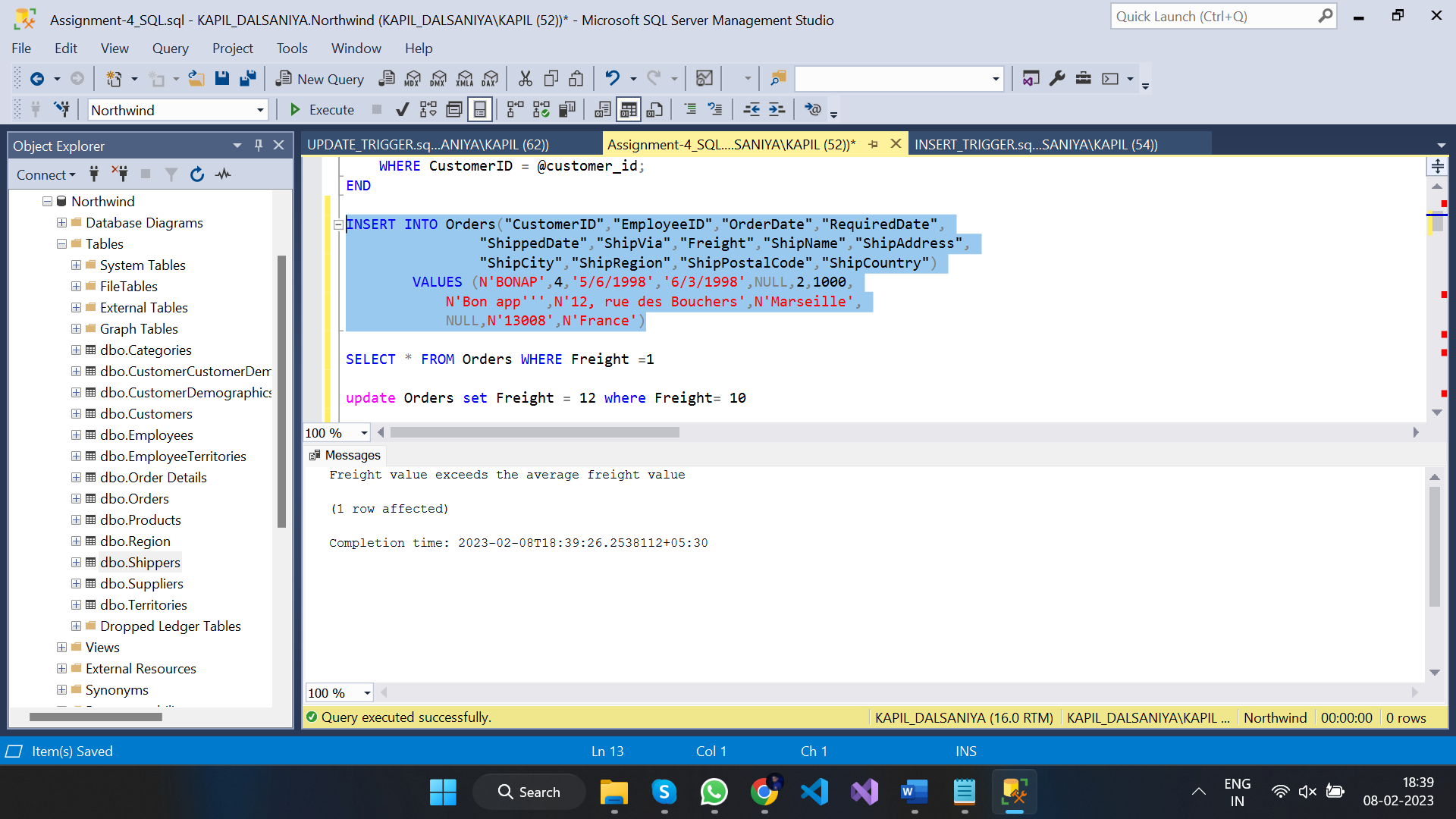
"ShipCity","ShipRegion","ShipPostalCode","ShipCountry")

VALUES (N'BONAP',4,'5/6/1998','6/3/1998',NULL,2,10,

N'Bon app''',N'12, rue des Bouchers',N'Marseille',

NULL,N'13008',N'France')

update Orders set Freight = 12 where Freight= 10



1. --2nd write a SQL query to Create Stored procedure in the Northwind database to retrieve Employee Sales by Country

CREATE PROCEDURE spGetEmployeeSalesByCountry

@country nvarchar(50)

AS

BEGIN

SELECT e.Country, e.LastName, e.FirstName, o.OrderID, od.Quantity, od.UnitPrice, od.Quantity\*od.UnitPrice AS [subtotal]

FROM Employees e

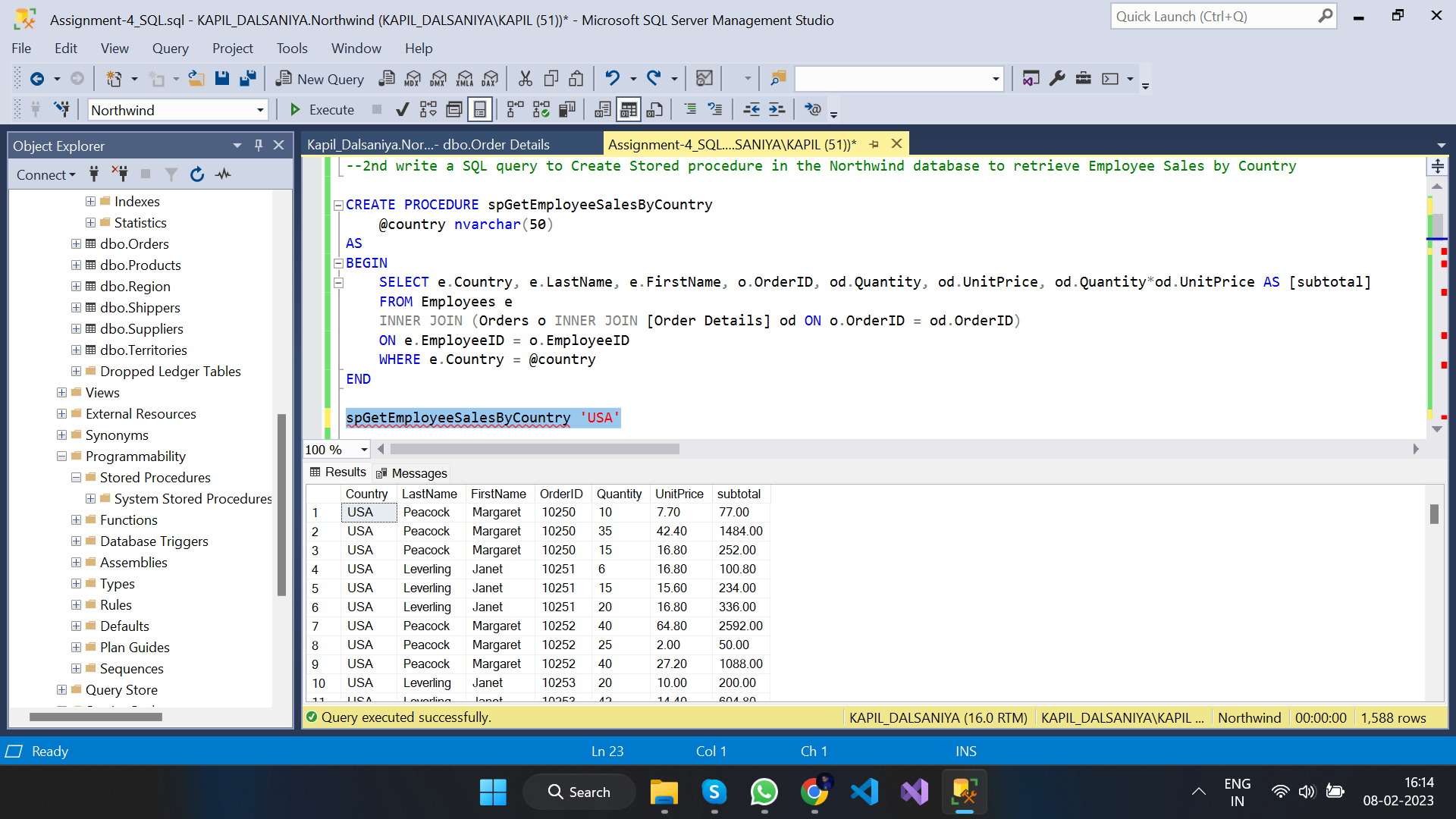
INNER JOIN (Orders o INNER JOIN [Order Details] od ON o.OrderID = od.OrderID)

ON e.EmployeeID = o.EmployeeID

WHERE e.Country = @country

END

spGetEmployeeSalesByCountry 'USA'



1. --3rd write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales by Year

CREATE PROCEDURE spGetSalesByYear

@year int

AS

BEGIN

SELECT o.ShippedDate, o.OrderID, od.Quantity, od.UnitPrice, od.Quantity\*od.UnitPrice AS [subtotal]

FROM Orders o

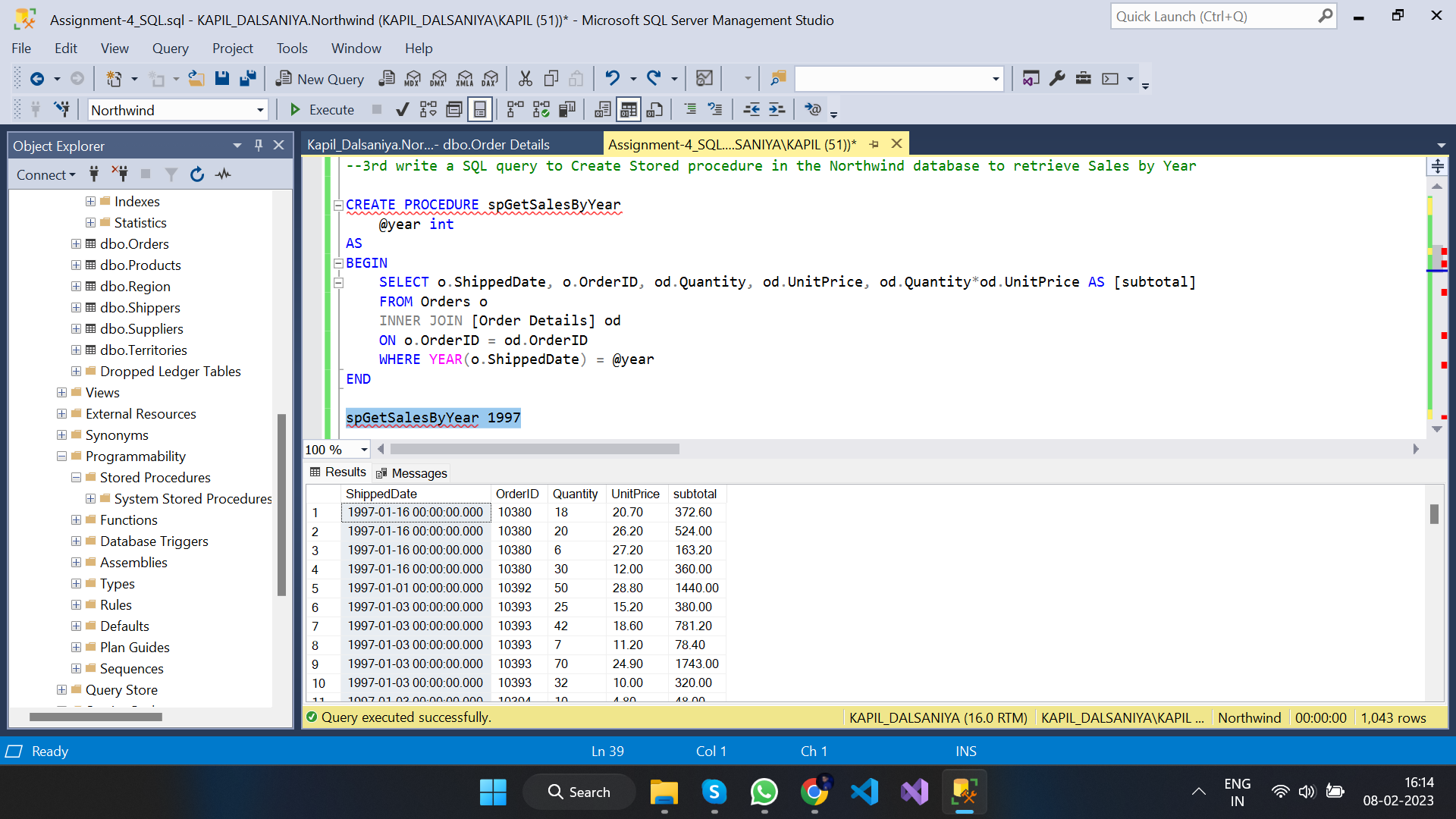
INNER JOIN [Order Details] od

ON o.OrderID = od.OrderID

WHERE YEAR(o.ShippedDate) = @year

END

spGetSalesByYear 1997



1. --4th write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales By Category

CREATE PROCEDURE spGetSalesByCategory

@category nvarchar(20)

AS

BEGIN

SELECT c.CategoryName, o.OrderID, od.Quantity, od.UnitPrice, od.Quantity\*od.UnitPrice AS [subtotal]

FROM Orders o

INNER JOIN ([Order Details] od INNER JOIN (Products p INNER JOIN Categories c ON p.CategoryID = c.CategoryID)

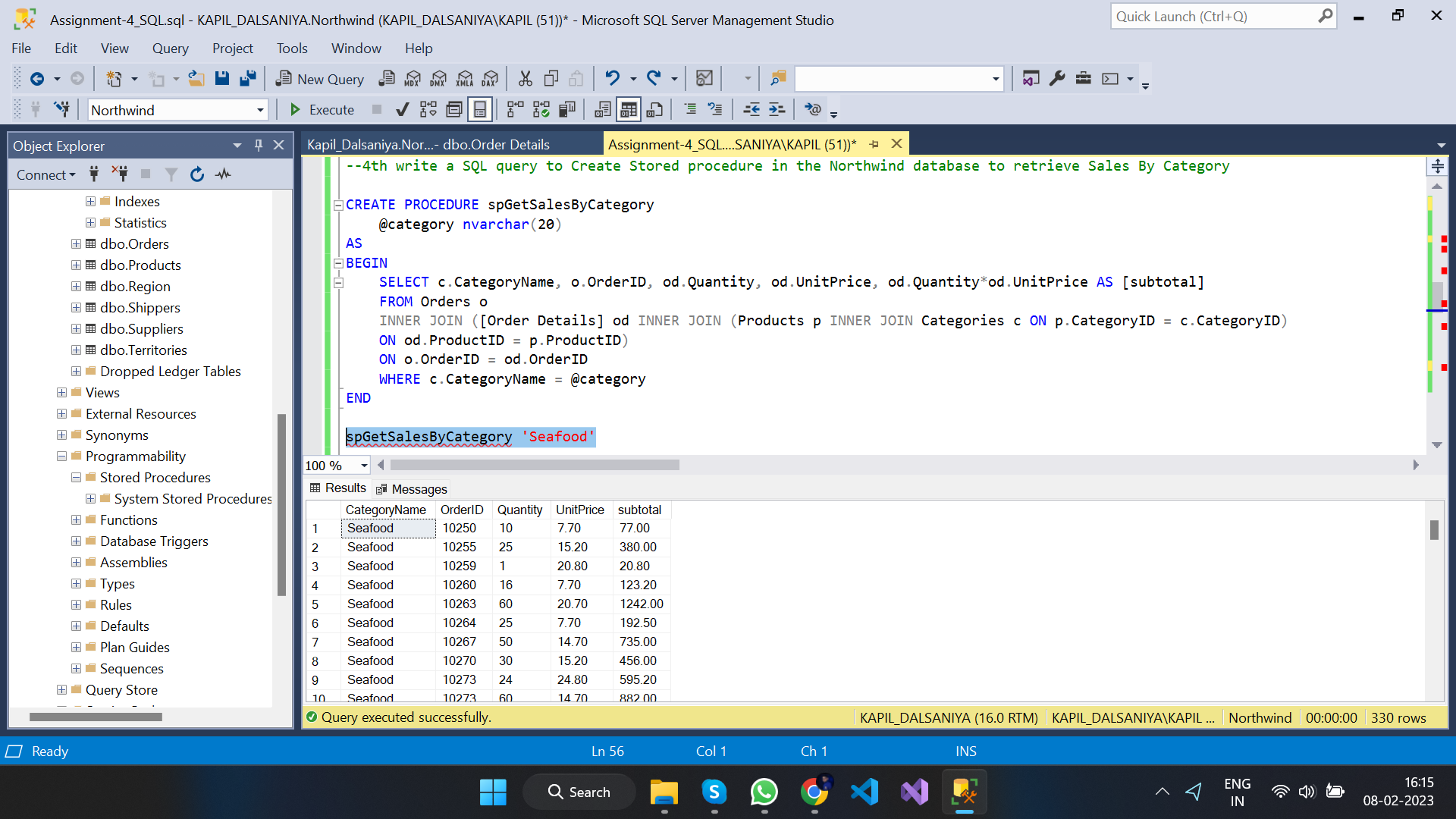
ON od.ProductID = p.ProductID)

ON o.OrderID = od.OrderID

WHERE c.CategoryName = @category

END

spGetSalesByCategory 'Seafood'



1. --5th write a SQL query to Create Stored procedure in the Northwind database to retrieve Ten Most Expensive Products

CREATE PROCEDURE spTenMostExpensiveProduct

AS

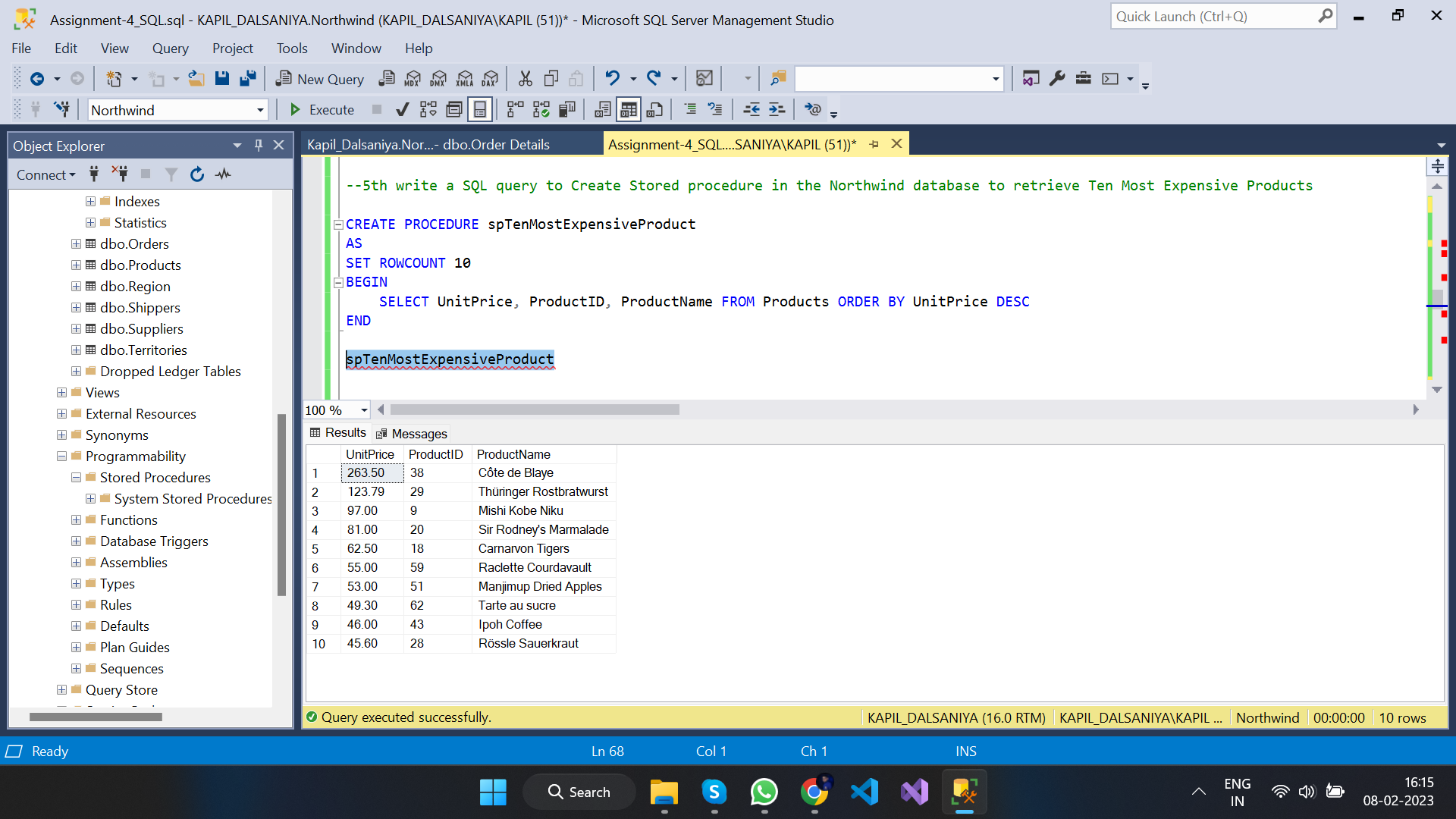
SET ROWCOUNT 10

BEGIN

SELECT UnitPrice, ProductID, ProductName FROM Products ORDER BY UnitPrice DESC

END

spTenMostExpensiveProduct



1. --6th write a SQL query to Create Stored procedure in the Northwind database to insert Customer Order Details

CREATE PROCEDURE spInsertCustomerOrderDetails

@order\_id int,

@product\_id int,

@unit\_price money,

@qty smallint,

@discount real

AS

BEGIN

INSERT INTO [Order Details] VALUES(@order\_id,@product\_id,@unit\_price,@qty,@discount);

SELECT \* FROM [Order Details] WHERE OrderID=@order\_id

AND ProductID=@product\_id

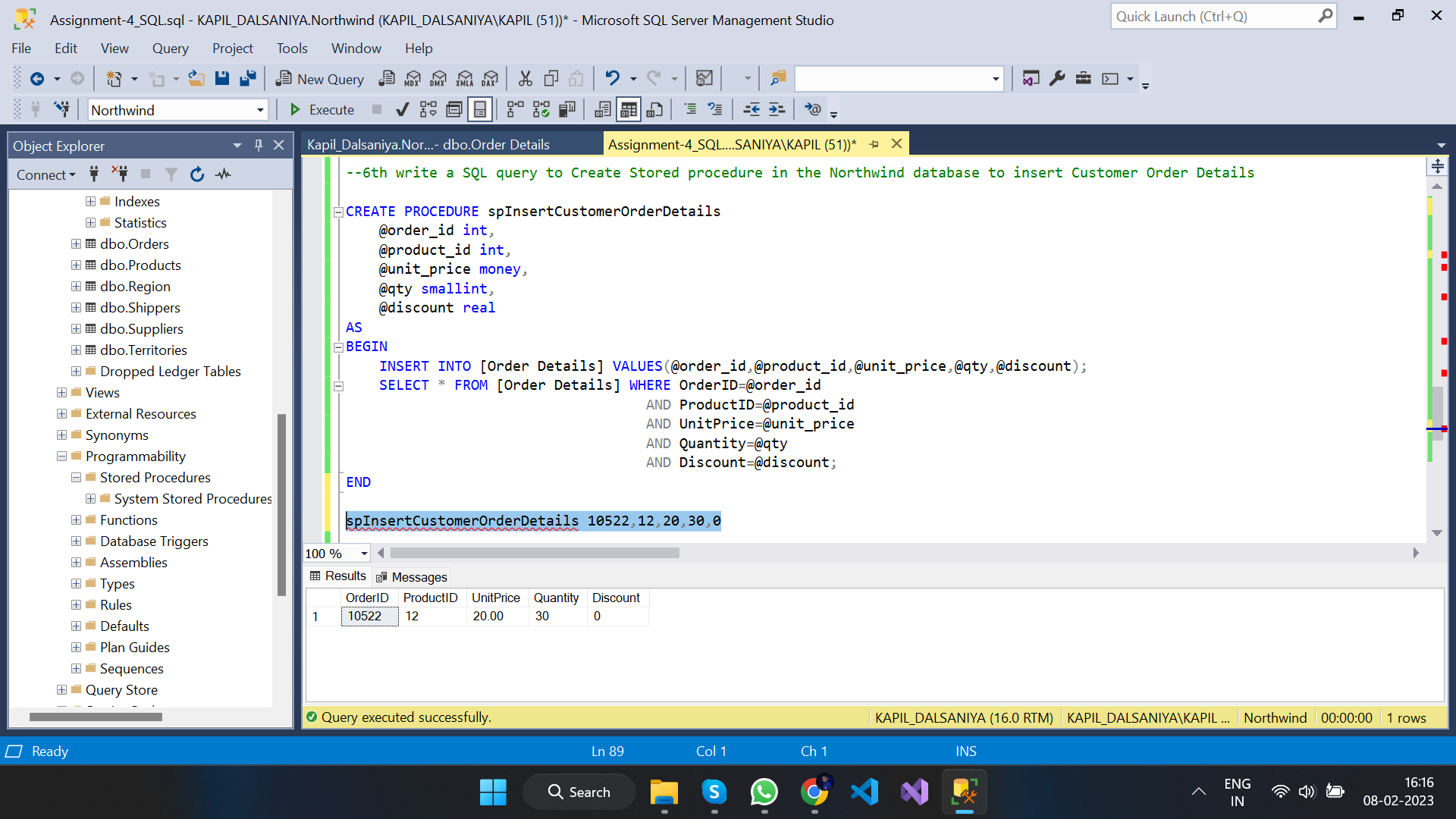
AND UnitPrice=@unit\_price

AND Quantity=@qty

AND Discount=@discount;

END

spInsertCustomerOrderDetails 10522,12,20,30,0



1. --7th write a SQL query to Create Stored procedure in the Northwind database to update Customer Order Details

CREATE PROCEDURE spUpdateCustomerOrderDetails

AS

BEGIN

SELECT \* FROM Employees

END