**Assignment 4**

1. **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.**

**Query:**

-- Create Procedure to Calculate Freight Average

CREATE PROCEDURE spCalAvgFreight

@CustomerID NVARCHAR(5),

@AverageFreight MONEY OUTPUT

AS

BEGIN

SELECT @AverageFreight = AVG(Freight)

FROM Orders

WHERE CustomerID = @CustomerID

END

GO

-- Create Trigger for Verifing Freight before Insert

CREATE TRIGGER tr\_VerifyFreightForInsert

ON Orders

INSTEAD OF INSERT

AS

BEGIN

DECLARE @AvgFreightOfOrders MONEY

DECLARE @CustID NVARCHAR(5)

DECLARE @Freight MONEY

SELECT @CustId=CustomerID FROM inserted

SELECT @Freight=Freight FROM inserted

-- execute stored procedure

EXECUTE spCalAvgFreight @CustID,

@AverageFreight = @AvgFreightOfOrders OUTPUT

-- check the freight

IF @AvgFreightOfOrders IS NOT NULL

AND @AvgFreightOfOrders < @Freight

BEGIN

Raiserror('Invalid data as Freight value exceeds the average freight value',16,1)

RETURN

END

END

INSERT INTO Orders VALUES('VINET',null,null,null,null,null,23,null,null,null,null,null,null)

-- Create Trigger for Verifing Freight before Update

CREATE TRIGGER tr\_VerifyFreightForUpdate

ON Orders

INSTEAD OF UPDATE

AS

BEGIN

DECLARE @AvgFreightOfOrders MONEY

DECLARE @CustID NVARCHAR(5)

DECLARE @Freight MONEY

SELECT @CustId=CustomerID FROM inserted

SELECT @Freight=Freight FROM inserted

-- execute stored procedure

EXECUTE spCalAvgFreight @CustID,

@AverageFreight = @AvgFreightOfOrders OUTPUT

-- check the freight

IF @AvgFreightOfOrders IS NOT NULL

AND @AvgFreightOfOrders < @Freight

BEGIN

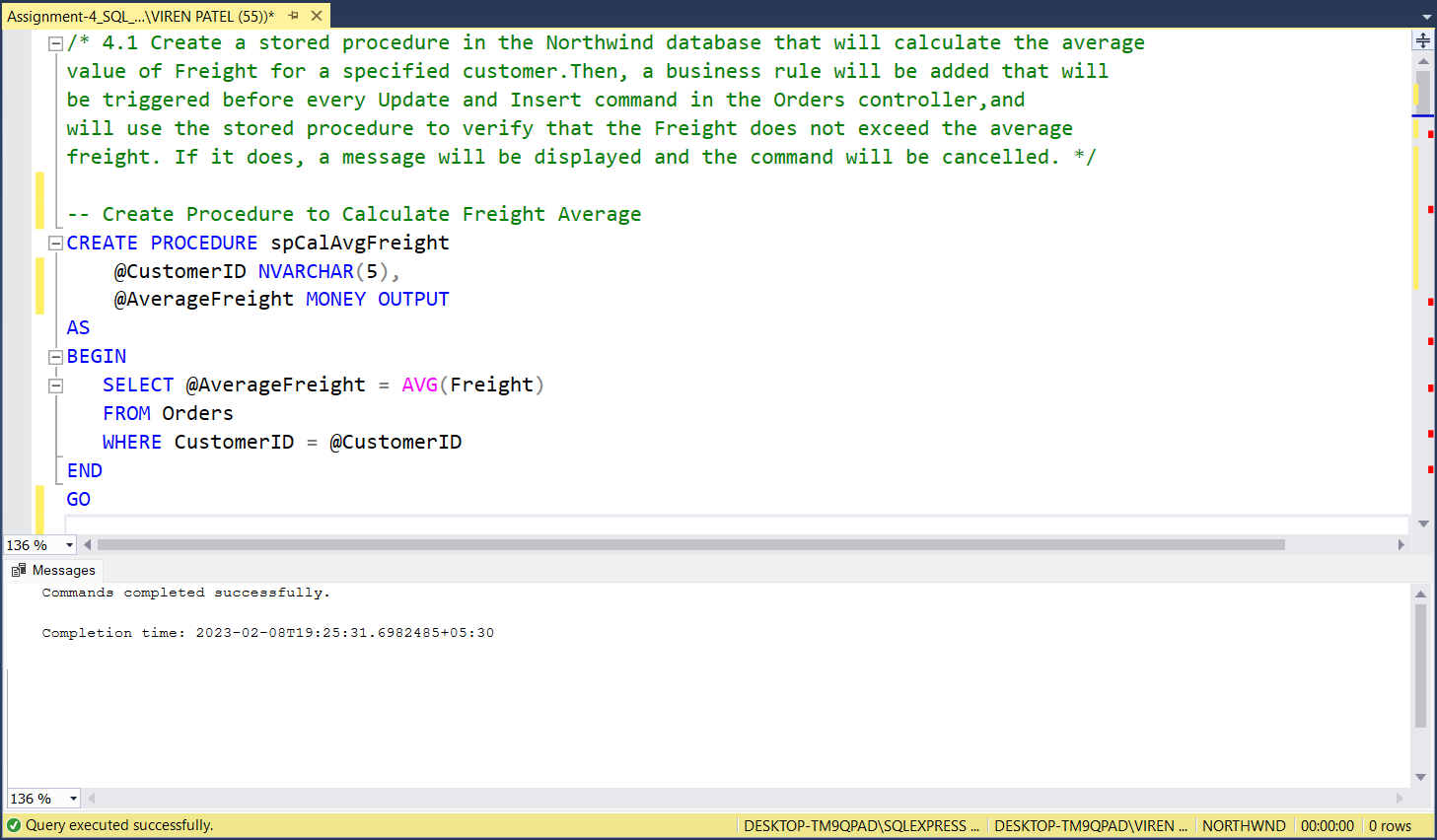
Raiserror('Invalid data as Freight value exceeds the average freight value',16,1)

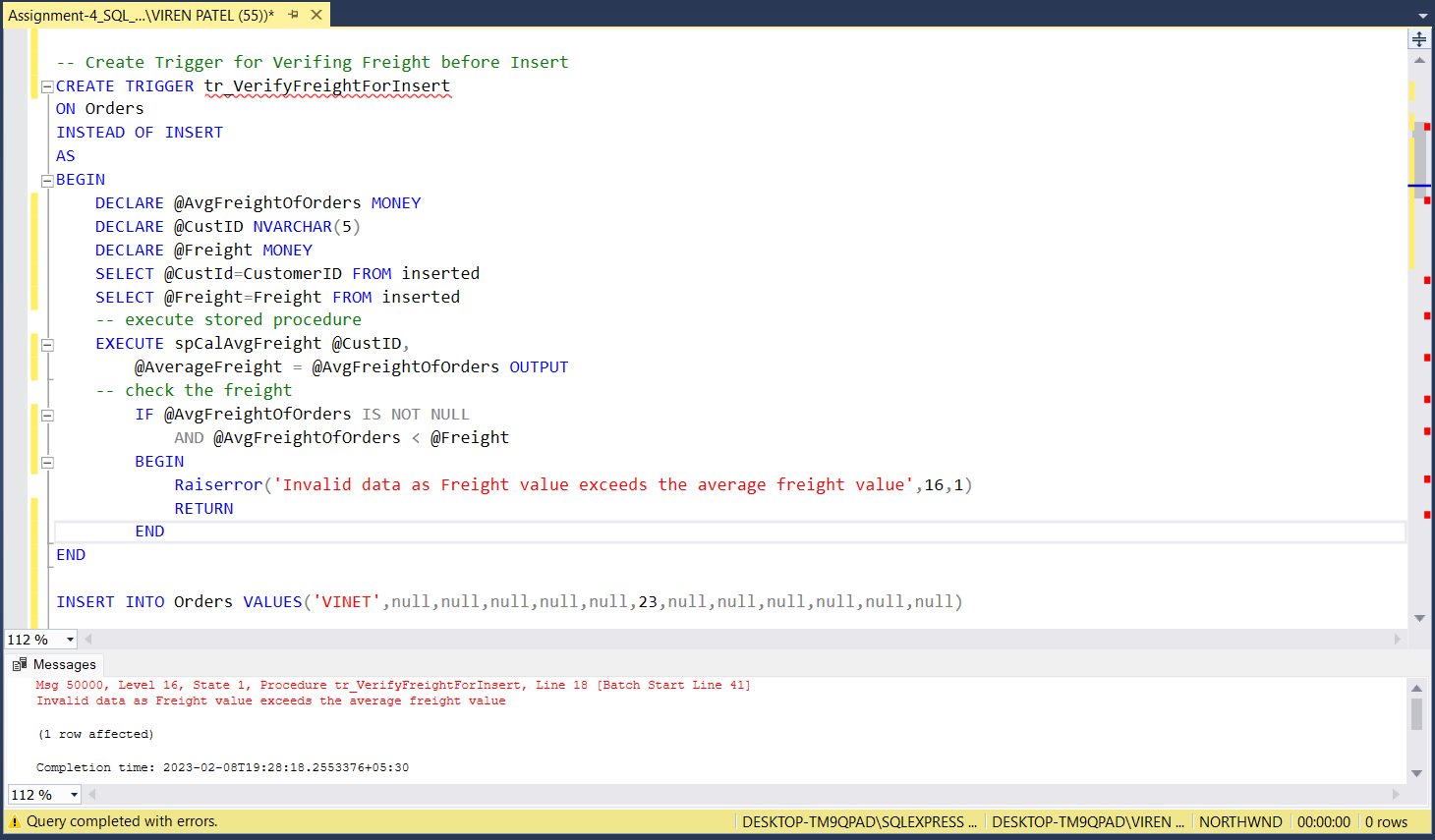
RETURN

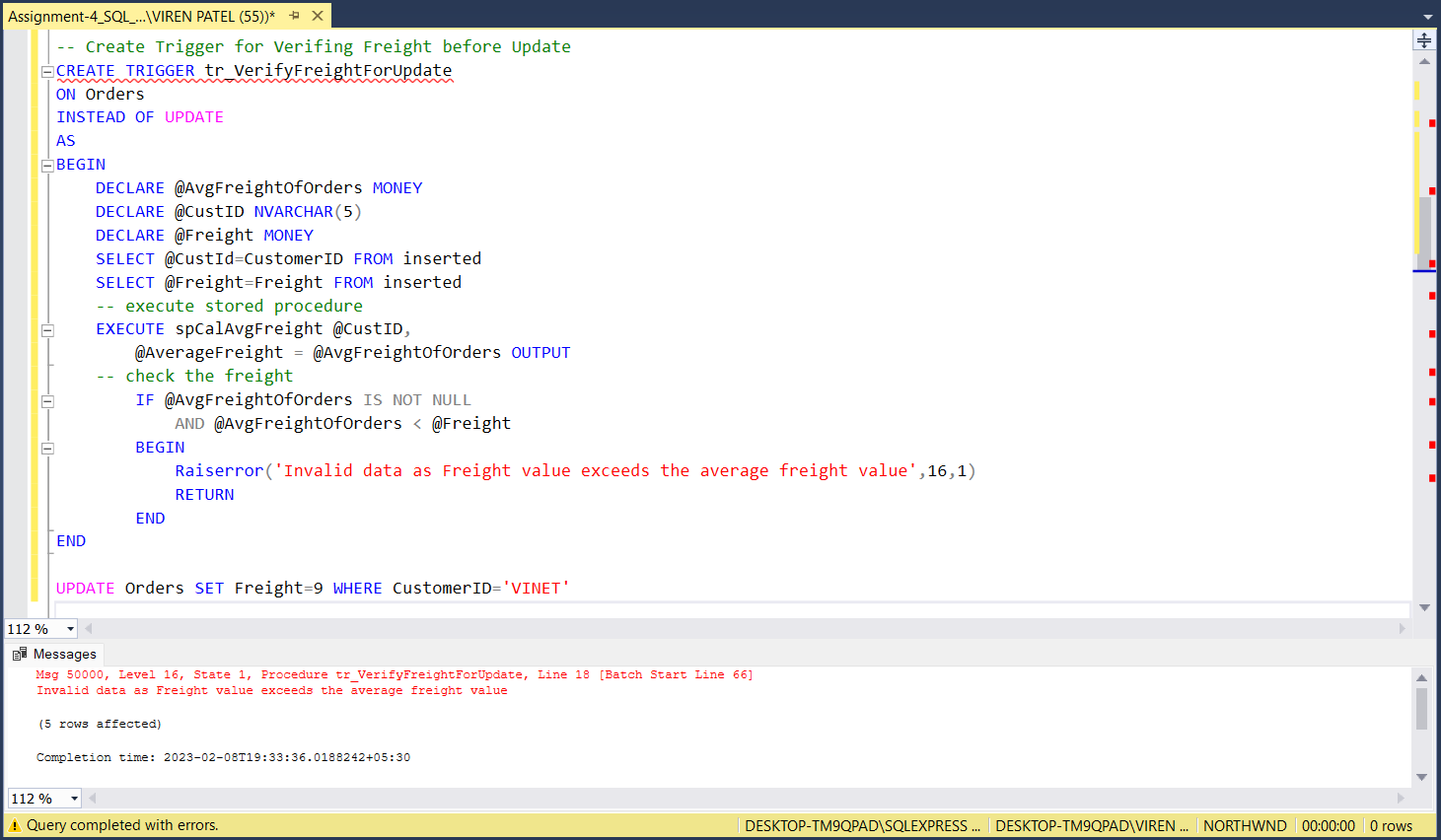
END

END

UPDATE Orders SET Freight=9 WHERE CustomerID='VINET'







1. **Write a SQL query to Create Stored procedure in the Northwind database to retrieve Employee Sales by Country**

**Query:**

CREATE PROCEDURE spEmployeeSalesByCountry

AS

SELECT o.ShipCountry AS [Order Country], e.FirstName, e.LastName, COUNT(o.OrderID) AS Orders

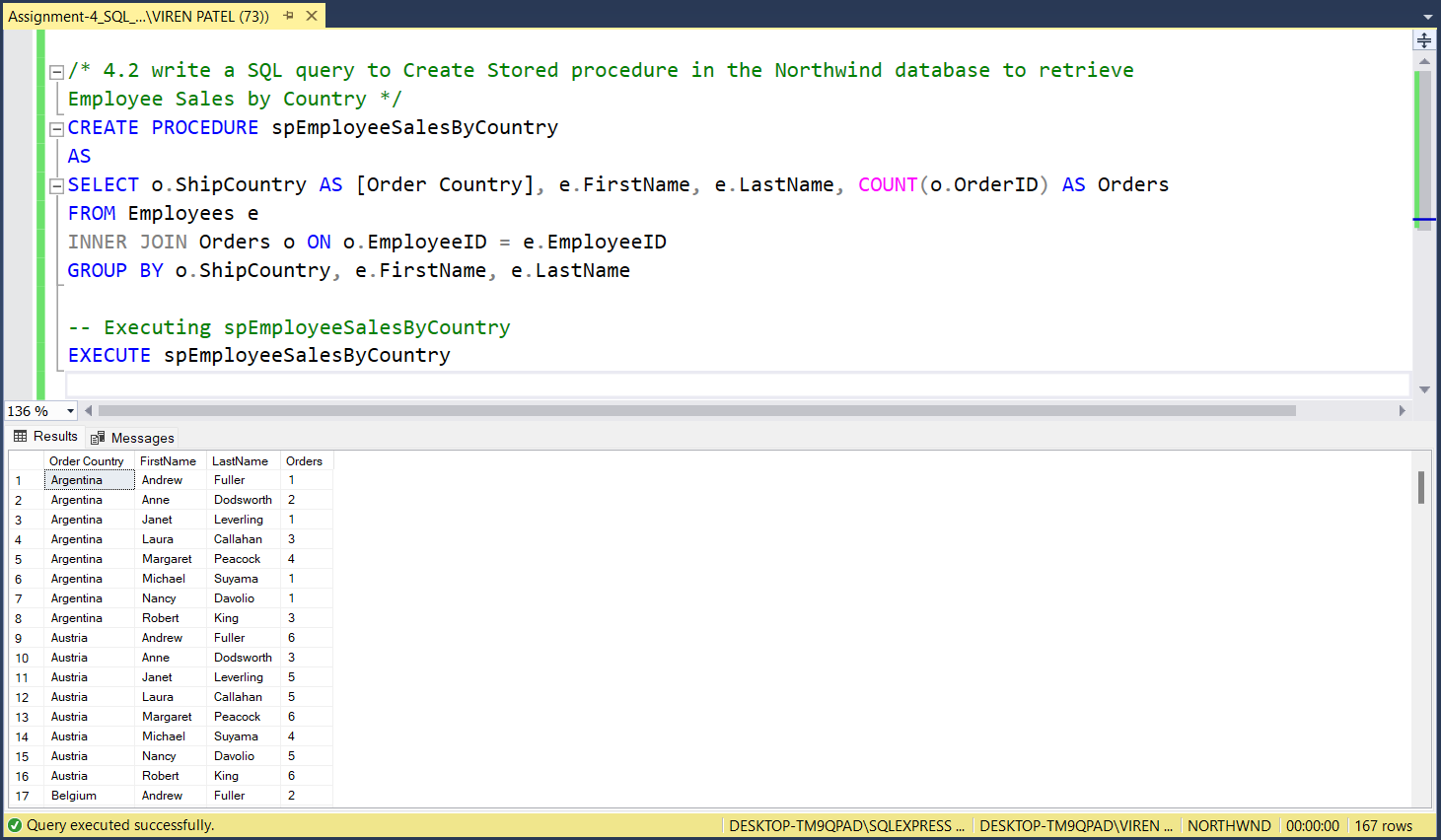
FROM Employees e

INNER JOIN Orders o ON o.EmployeeID = e.EmployeeID

GROUP BY o.ShipCountry, e.FirstName, e.LastName

-- Executing spEmployeeSalesByCountry

EXECUTE spEmployeeSalesByCountry



1. **Write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales by Year.**

**Query:**

CREATE PROCEDURE spEmployeeSalesByYear

@Year INT

AS

SELECT YEAR(o.OrderDate) AS Year, e.FirstName, e.LastName, COUNT(o.OrderID) AS Orders

FROM Employees e

INNER JOIN Orders o ON o.EmployeeID = e.EmployeeID

WHERE YEAR(o.OrderDate) = @Year

GROUP BY YEAR(o.OrderDate), e.FirstName, e.LastName

-- Executing spEmployeeSalesByCountry

EXECUTE spEmployeeSalesByYear @Year=1997

1. **Write a SQL query to Create Stored procedure in the Northwind database to retrieve Sales By Category**

**Query:**

CREATE PROCEDURE spSalesByCategory

@Category VARCHAR(30)

AS

SELECT ProductName, COUNT(od.OrderID) AS Orders

FROM [Order Details] od

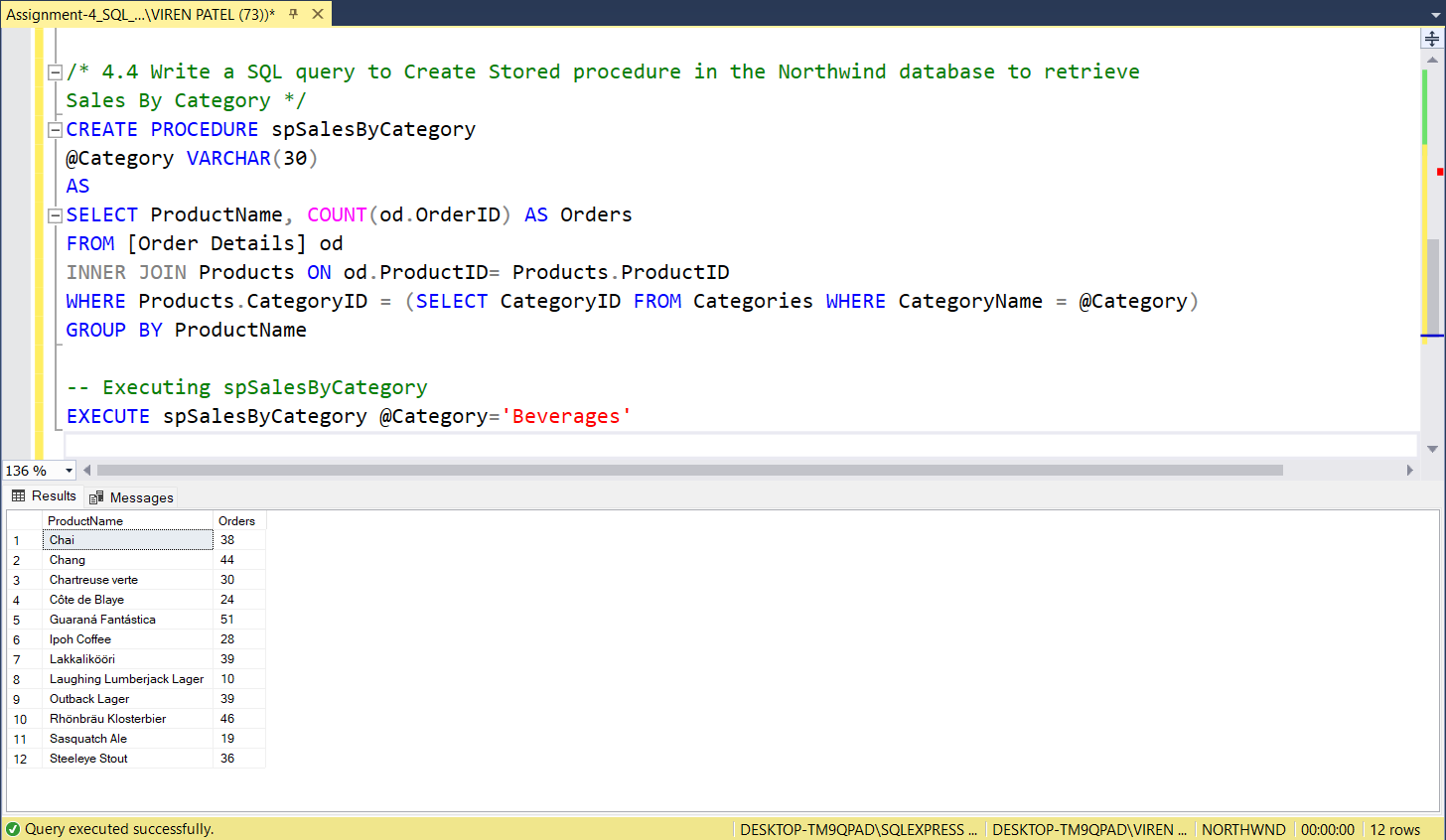
INNER JOIN Products ON od.ProductID= Products.ProductID

WHERE Products.CategoryID = (SELECT CategoryID FROM Categories WHERE CategoryName = @Category)

GROUP BY ProductName

-- Executing spSalesByCategory

EXECUTE spSalesByCategory @Category='Beverages'



1. **Write a SQL query to Create Stored procedure in the Northwind database to retrieve Ten Most Expensive Products**

**Query:**

CREATE PROCEDURE spTenMostExpensiveProds

AS

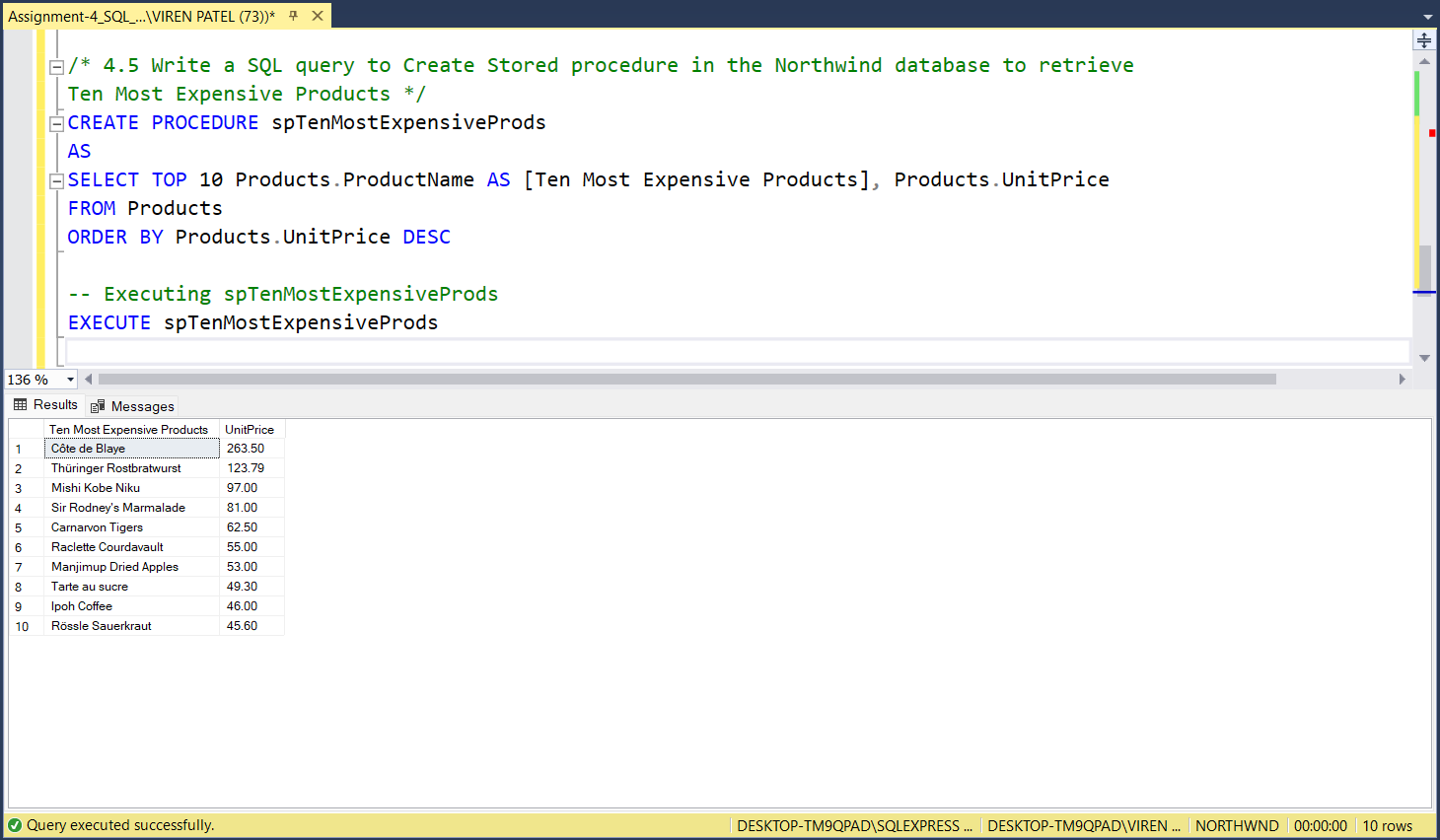
SELECT TOP 10 Products.ProductName AS [Ten Most Expensive Products], Products.UnitPrice

FROM Products

ORDER BY Products.UnitPrice DESC

-- Executing spTenMostExpensiveProds

EXECUTE spTenMostExpensiveProds



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

**Query:**

CREATE PROCEDURE spInsertOrderDetails

@OrderID INT,

@ProductID INT,

@UnitPrice MONEY,

@Quantity SMALLINT,

@Discount REAL

AS

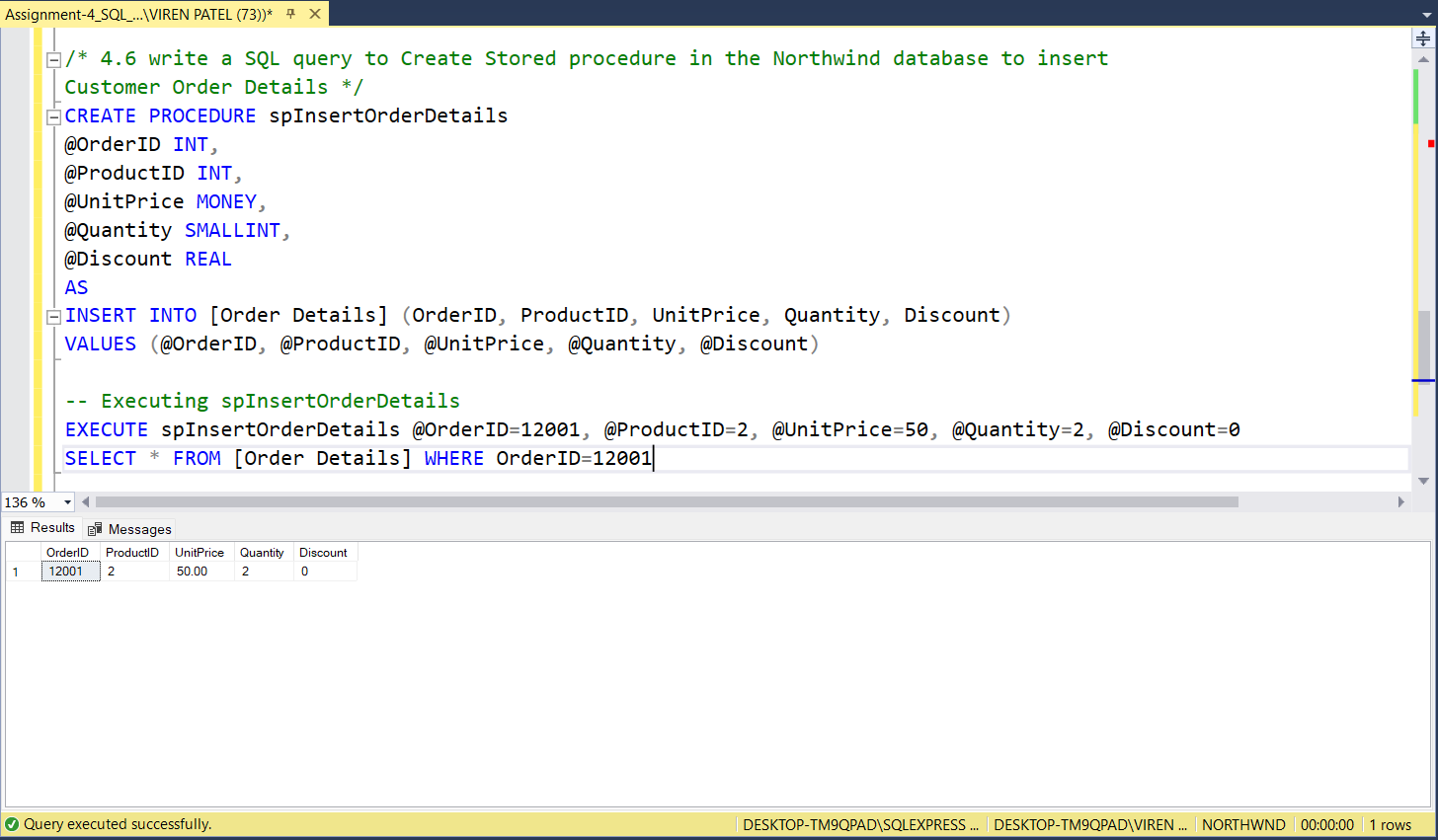
INSERT INTO [Order Details] (OrderID, ProductID, UnitPrice, Quantity, Discount)

VALUES (@OrderID, @ProductID, @UnitPrice, @Quantity, @Discount)

-- Executing spInsertOrderDetails

EXECUTE spInsertOrderDetails @OrderID=12001, @ProductID=2, @UnitPrice=50, @Quantity=2, @Discount=0

SELECT \* FROM [Order Details] WHERE OrderID=12001



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

**Query:**

CREATE PROCEDURE spUpdateOrderDetails

@OrderID INT,

@ProductID INT,

@UnitPrice MONEY,

@Quantity SMALLINT,

@Discount REAL

AS

UPDATE [Order Details]

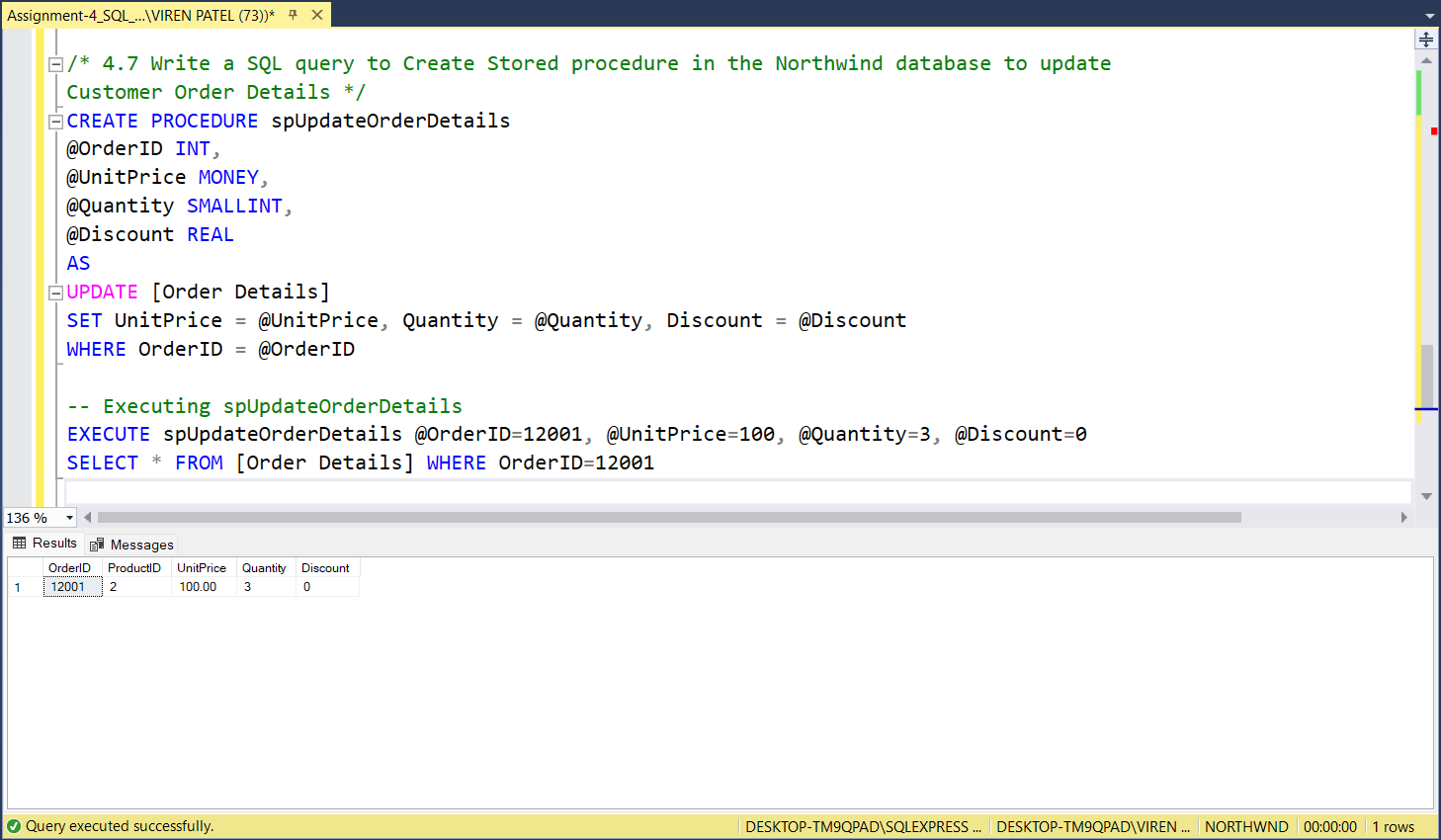
SET UnitPrice = @UnitPrice, Quantity = @Quantity, Discount = @Discount

WHERE OrderID = @OrderID AND ProductID = @ProductID

-- Executing spUpdateOrderDetails

EXECUTE spUpdateOrderDetails @OrderID=12001, @ProductID=2, @UnitPrice=100, @Quantity=3, @Discount=0

SELECT \* FROM [Order Details] WHERE OrderID=12001

****