**ASSIGNMENT 4: Create Stored procedure in Northwind database to insert or update a record in a table**

**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.

**Solution:**

-- create stored procedure

alter PROC spOrdersFreightAverage

@customer\_id varchar(20),

@average float(5) output

AS

BEGIN

select @average = avg(Freight) from orders where CustomerID = @customer\_id

END

--create trigger

alter trigger tr\_orders\_compareFreight

on orders

for insert, update

as

begin

declare @cust\_id varchar(50)

declare @freight float(5)

declare @average float(5)

select @cust\_id = CustomerID from inserted

select @freight = freight from inserted

exec spOrdersFreightAverage @cust\_id, @average output

if(@freight > @average)

begin

RAISERROR('Freight exceeds the average freight value', 16, 1)

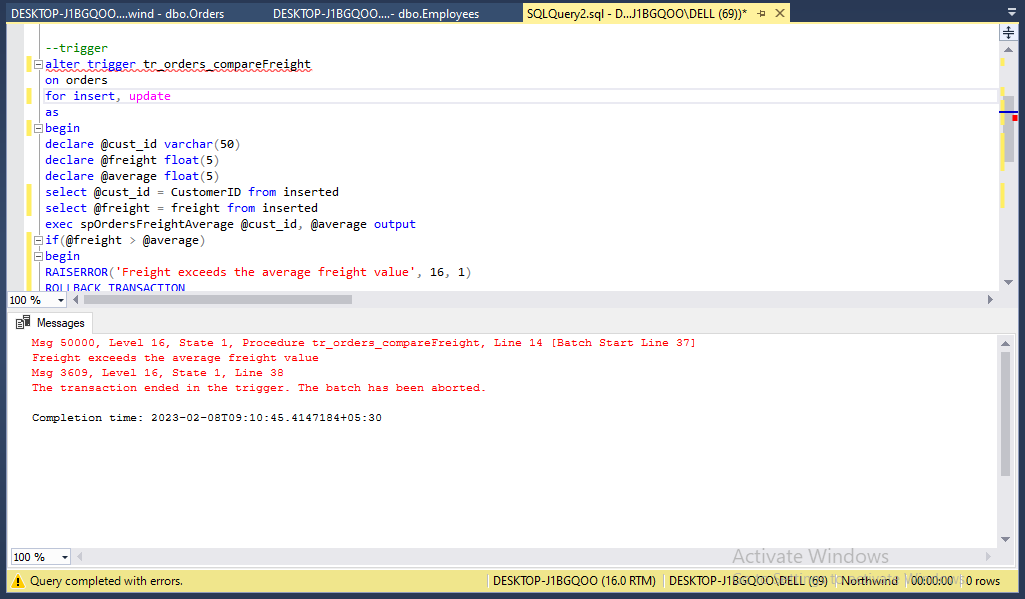
ROLLBACK TRANSACTION

end

end

insert into orders values ( 'VAFFE', 8,'1997-08-25 00:00:00.000', '1997-08-01 00:00:00.000','1997-01-01 00:00:00.000', 1,87, 'Wolski Zajazd', 'ul. Filtrowa 68', 'Warszawa', 'Tachira', 24100, 'brazil')

**Output:**



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

**Solution:**

-- create stored procedure

create proc spEmployeeSalesByCountry

@country varchar(50)

as

begin

select e.EmployeeID, e.FirstName, e.LastName, COUNT(o.OrderID) as orders, o.ShipCountry

from Employees e

left join Orders o

on e.EmployeeID = o.EmployeeID

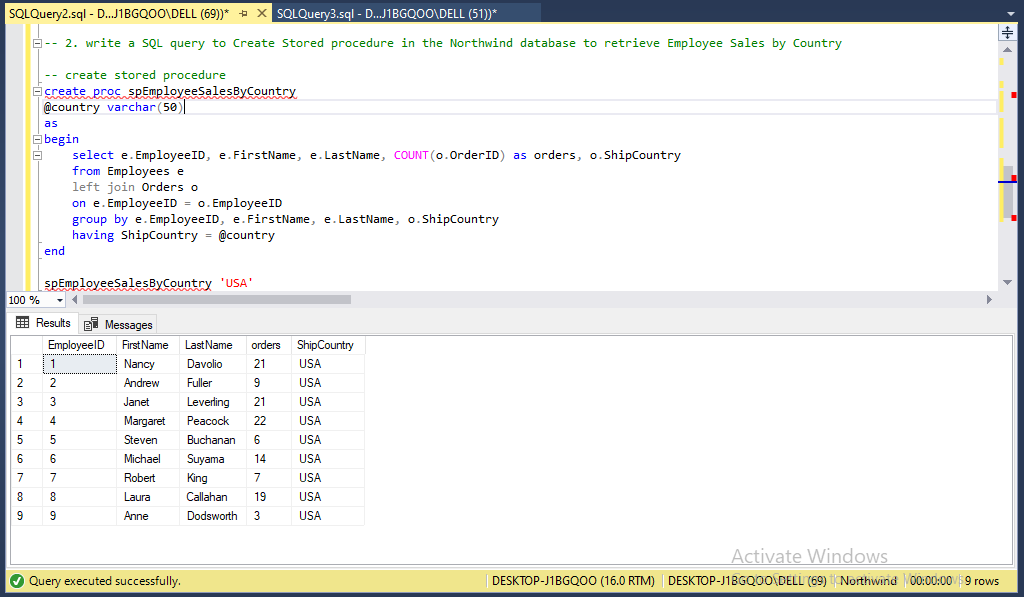
group by e.EmployeeID, e.FirstName, e.LastName, o.ShipCountry

having ShipCountry = @country

end

spEmployeeSalesByCountry 'USA'

**Output:**



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

**Solution:**

create proc spSalesByYear

@year varchar(4)

as

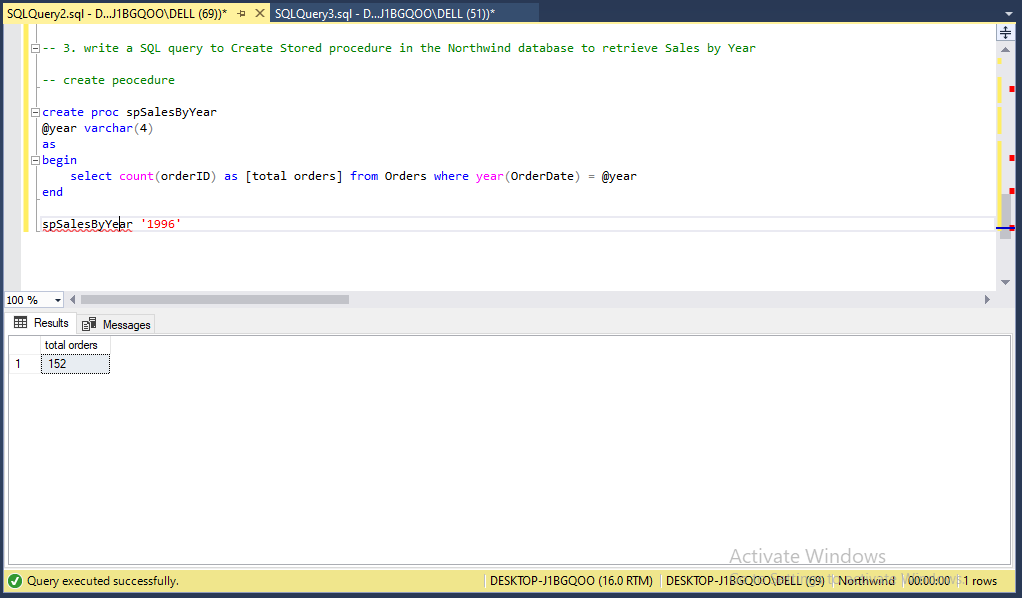
begin

select count(orderID) as [total orders] from Orders where year(OrderDate) = @year

end

spSalesByYear '1996'

**Output:**



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

**Solution:**

create proc spSalesByCategory

@category varchar(50)

as

begin

select c.CategoryID, c.CategoryName, count(od.OrderID) as sales

from Categories c

left join Products p

on c.CategoryID = p.CategoryID

left join [Order Details] od

on p.productID = od.ProductID

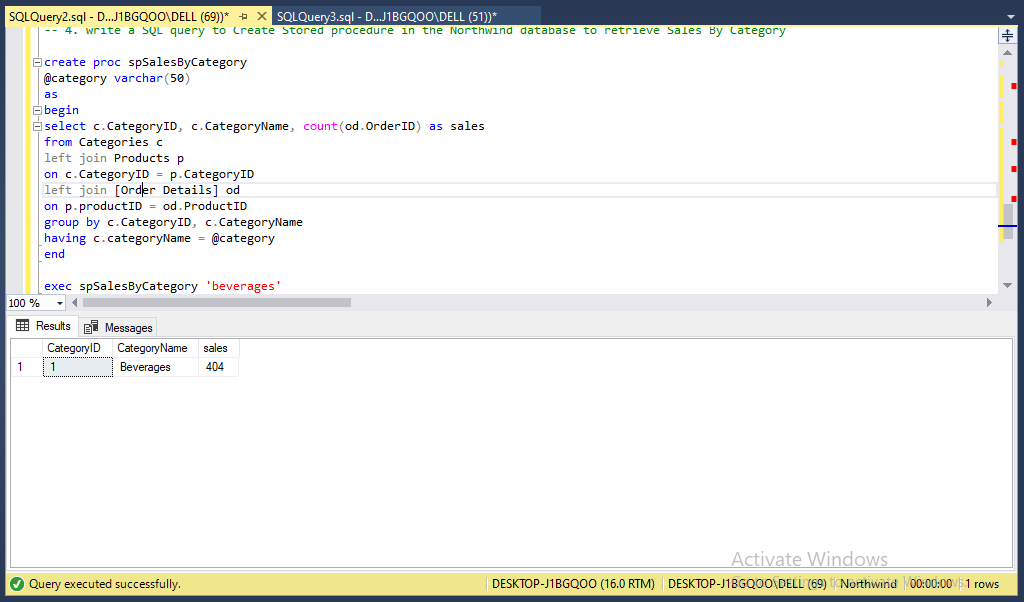
group by c.CategoryID, c.CategoryName

having c.categoryName = @category

end

exec spSalesByCategory 'beverages'

**Output:**



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

**Solution:**

create proc spTop10Products

as

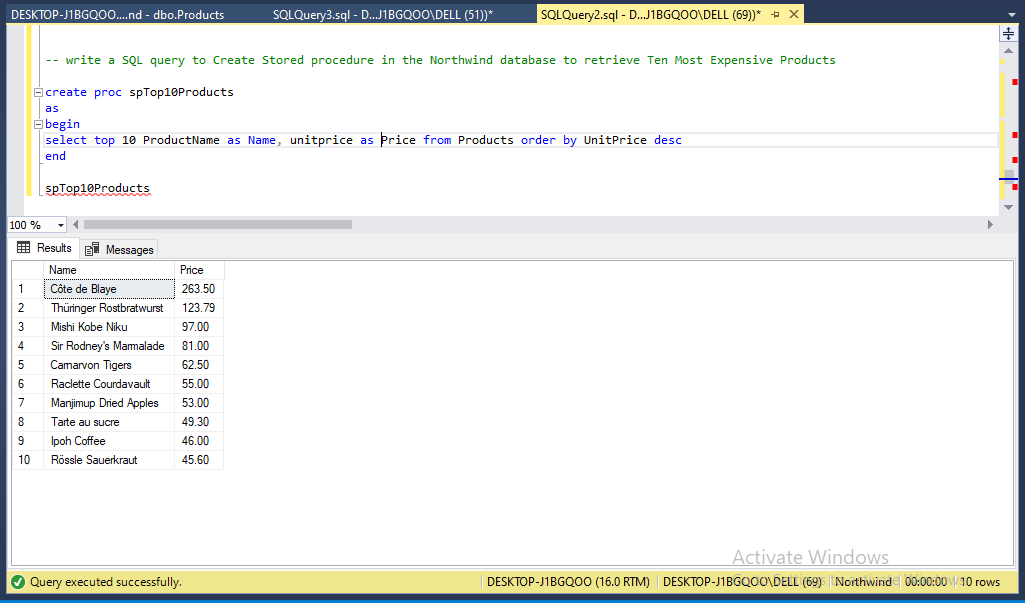
begin

select top 10 ProductName as Name, unitprice as Price from Products order by UnitPrice desc

end

spTop10Products

**Output:**



**6.** write a SQL query to Create Stored procedure in the Northwind database to insert

Customer Order Details

**Solution:**

create proc spOrderDetailsInsert

@orderid int,

@productid int,

@unitprice money,

@quantity smallint,

@discount real

as

begin

insert into [Order Details] (orderid, productid, unitprice, quantity, discount)

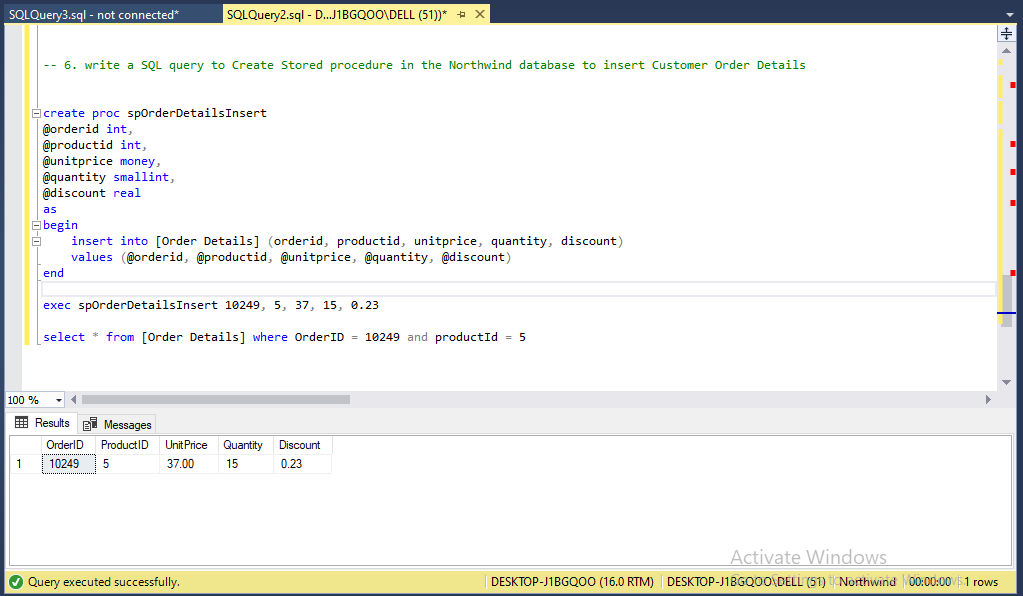
values (@orderid, @productid, @unitprice, @quantity, @discount)

end

exec spOrderDetailsInsert 10249, 5, 37, 15, 0.23

select \* from [Order Details] where OrderID = 10249 and productId = 5

**Output:**

****

**7.** write a SQL query to Create Stored procedure in the Northwind database to update

Customer Order Details

**Solution:**

CREATE PROC spOrderDetailsUpdate

@OrderID int,

@ProductID int,

@UnitPrice money,

@Quantity int,

@Discount real

AS

BEGIN

UPDATE [Order Details]

SET OrderID = @OrderID, ProductID = @ProductID, UnitPrice = @UnitPrice, Quantity = @Quantity, Discount = @Discount

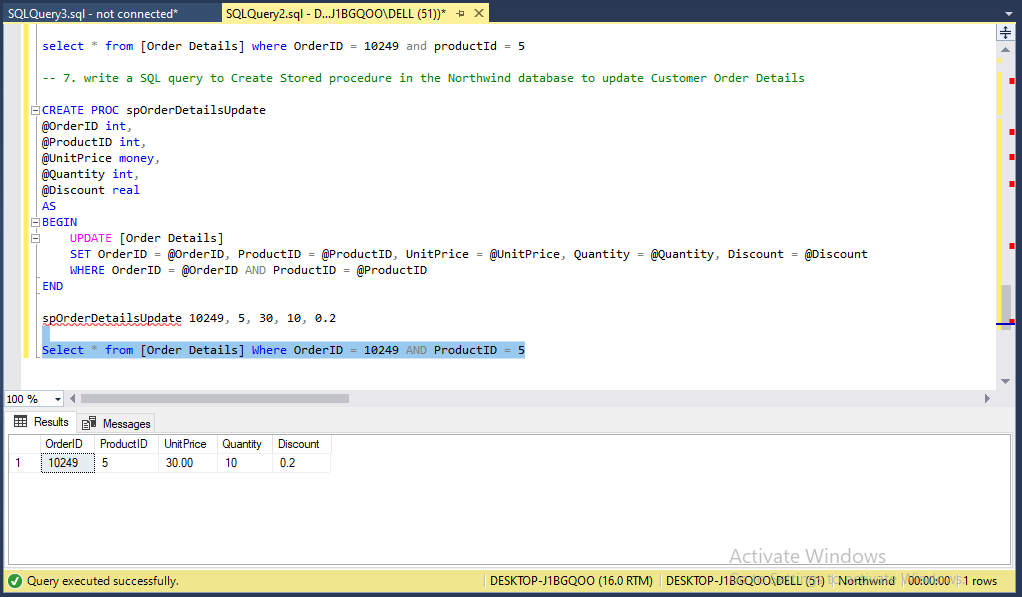
WHERE OrderID = @OrderID AND ProductID = @ProductID

END

spOrderDetailsUpdate 10249, 5, 30, 10, 0.2

Select \* from [Order Details] Where OrderID = 10249 AND ProductID = 5

**Output:**

****