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.

create proc insertorder

(

@OrderID int,

@CustomerID nvarchar(5)=null,

@EmployeeID int=null,

@OrderDate datetime=null,

@RequiredDate datetime=null,

@ShippedDate datetime=null,

@ShipVia int=null,

@Freight money=null,

@ShipName nvarchar(40)=null,

@ShipAddress nvarchar(60)=null,

@ShipCity nvarchar(15)=null,

@ShipRegion nvarchar(15)=null,

@ShipPostalCode nvarchar(10)=null,

@ShipCountry nvarchar(15)=null

)

as

begin

declare @avg decimal(10,2)

select @avg = avg(Freight) from Orders

if(@avg>@Freight)

begin

declare @temp int

set @temp=null

select @temp=OrderID from Orders where orderID = @OrderID

if(@temp is null)

begin

set identity\_insert "orders" on

insert into Orders(OrderID,CustomerID,EmployeeID,OrderDate,RequiredDate,ShippedDate,ShipVia,Freight,ShipName,ShipAddress,ShipCity,ShipRegion,ShipPostalCode,ShipCountry) values(@OrderID,@CustomerID,@EmployeeID,@OrderDate,@RequiredDate,@ShippedDate,@ShipVia,@Freight,@ShipName,@ShipAddress,@ShipCity,@ShipRegion,@ShipPostalCode,@ShipCountry)

end

else

begin

if @CustomerID is not null

update Orders set CustomerID=@CustomerID where OrderID=@OrderID

if @EmployeeID is not null

update Orders set EmployeeID=@EmployeeID where OrderID=@OrderID

if @OrderDate is not null

update Orders set OrderDate=@OrderDate where OrderID=@OrderID

if @RequiredDate is not null

update Orders set RequiredDate=@RequiredDate where OrderID=@OrderID

if @ShippedDate is not null

update Orders set ShippedDate=@ShippedDate where OrderID=@OrderID

if @ShipVia is not null

update Orders set ShipVia=@ShipVia where OrderID=@OrderID

if @Freight is not null

update Orders set Freight=@Freight where OrderID=@OrderID

if @ShipName is not null

update Orders set ShipName=@ShipName where OrderID=@OrderID

if @ShipAddress is not null

update Orders set ShipAddress=@ShipAddress where OrderID=@OrderID

if @ShipCity is not null

update Orders set ShipCity=@ShipCity where OrderID=@OrderID

if @ShipRegion is not null

update Orders set ShipRegion=@ShipRegion where OrderID=@OrderID

if @ShipPostalCode is not null

update Orders set ShipPostalCode=@ShipPostalCode where OrderID=@OrderID

if @ShipCountry is not null

update Orders set ShipCountry=@ShipCountry where OrderID=@OrderID

end

end

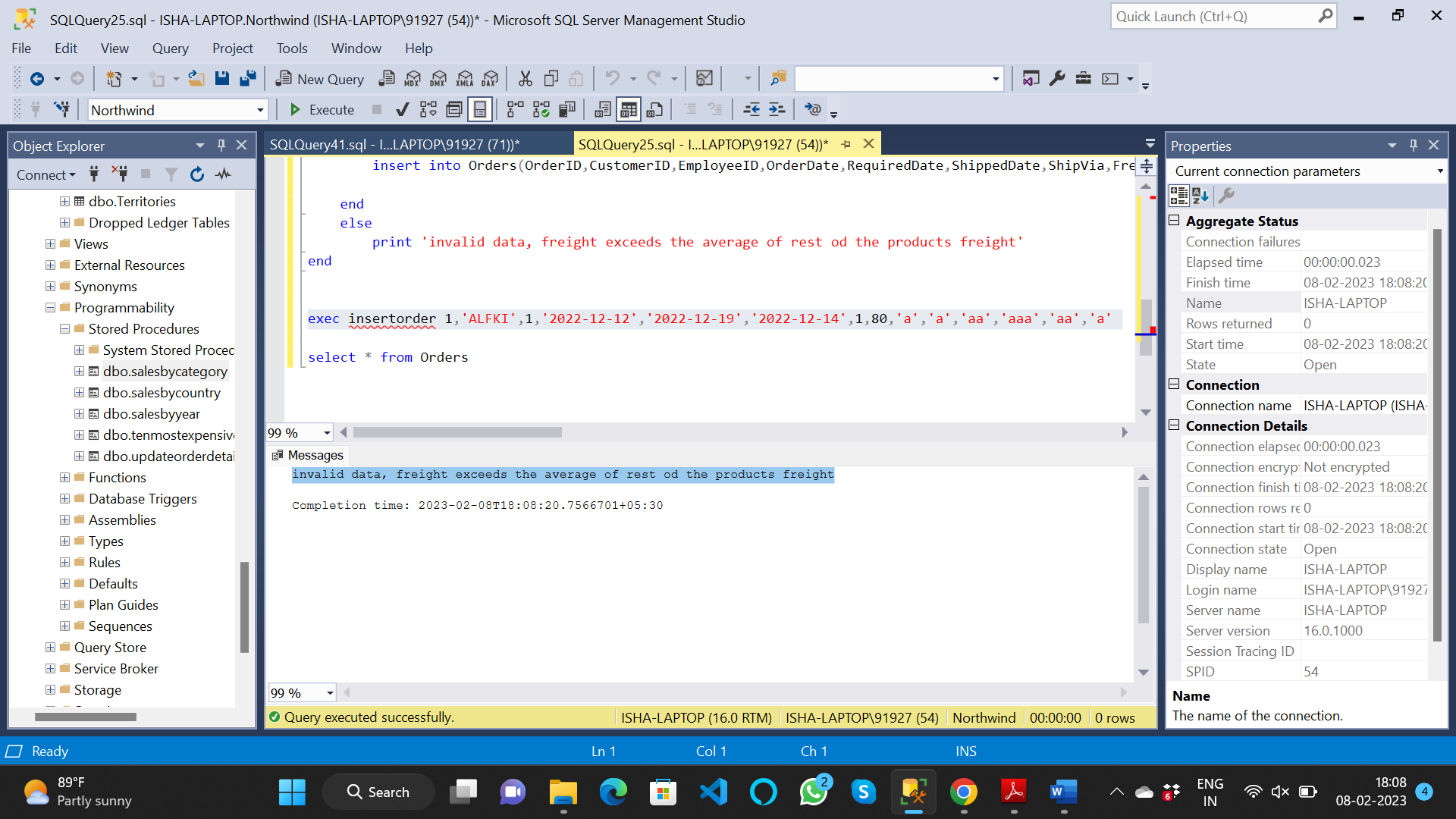
else

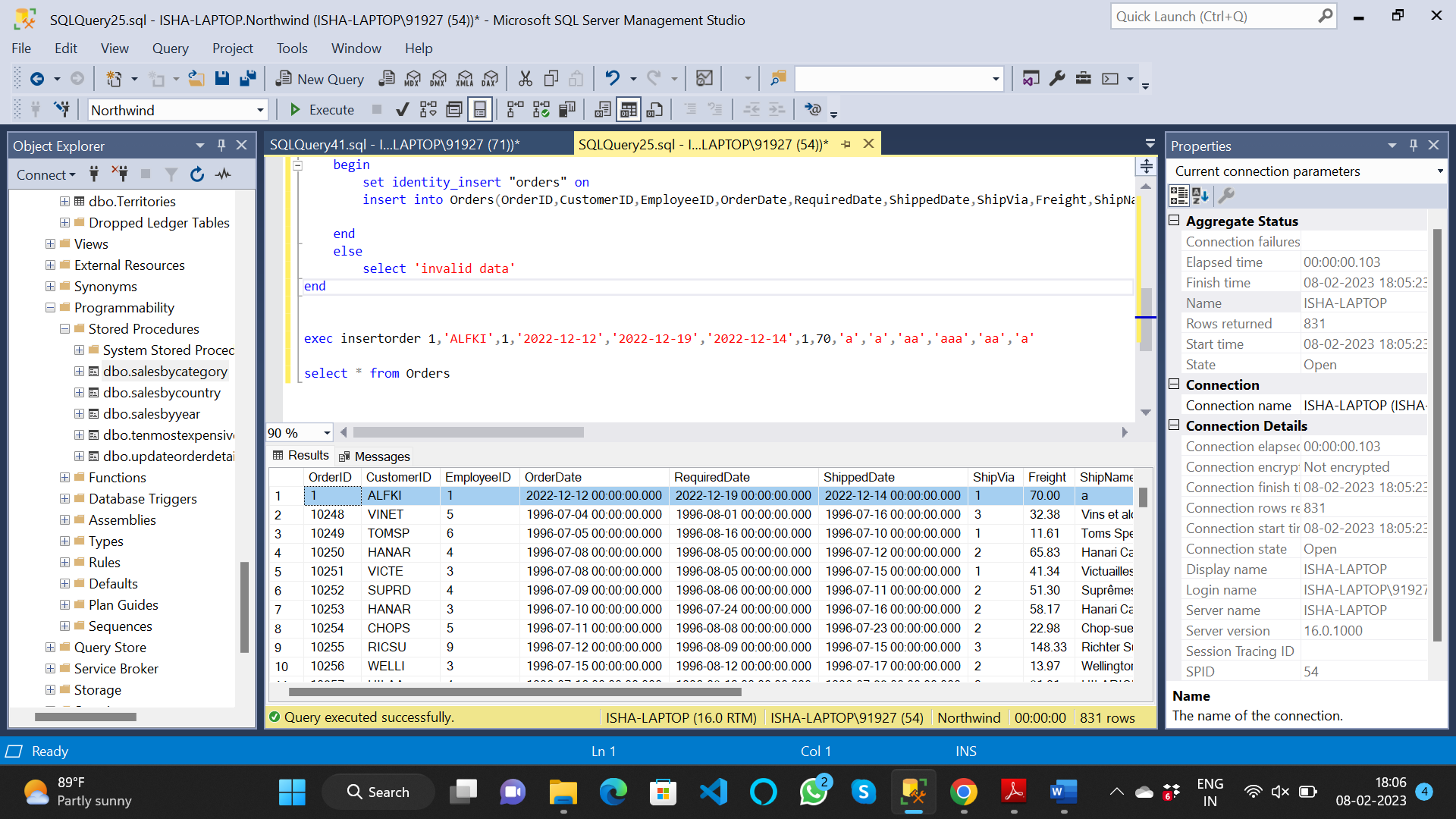
print 'invalid data, freight exceeds the average of rest od the products freight'

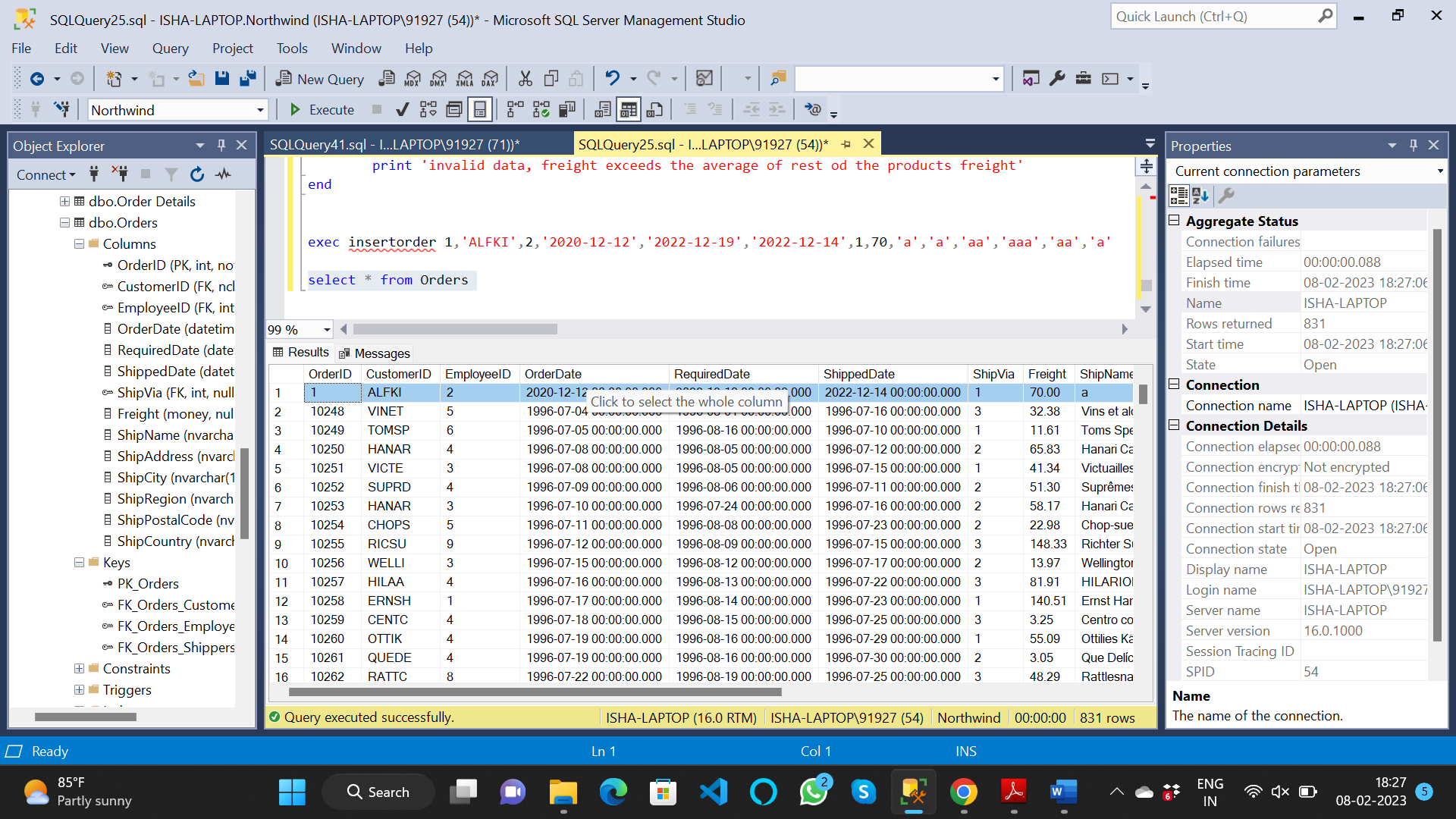
end

exec insertorder 1,'ALFKI',2,'2020-12-12','2022-12-19','2022-12-14',1,70,'a','a','aa','aaa','aa','a'

select \* from Orders







2. write a SQL query to Create Stored procedure in the Northwind database to retrieve

Employee Sales by Country

create proc salesbycountry

@country nvarchar(50)

as

begin

select Employees.EmployeeID, FirstName+' '+LastName as [Emp Name], Employees.Country, orders.OrderID, ProductID,

convert(date,Orders.OrderDate) as [order date],

convert(date,Orders.ShippedDate) as [shipped date],

[Order Details].UnitPrice, [Order Details].Quantity, [Order Details].Discount,

UnitPrice\*Quantity\*(1-Discount) as [total price],

Orders.Freight,

UnitPrice\*Quantity\*(1-Discount)+Freight as [total]

from Employees join Orders

on Orders.EmployeeID=Employees.EmployeeID

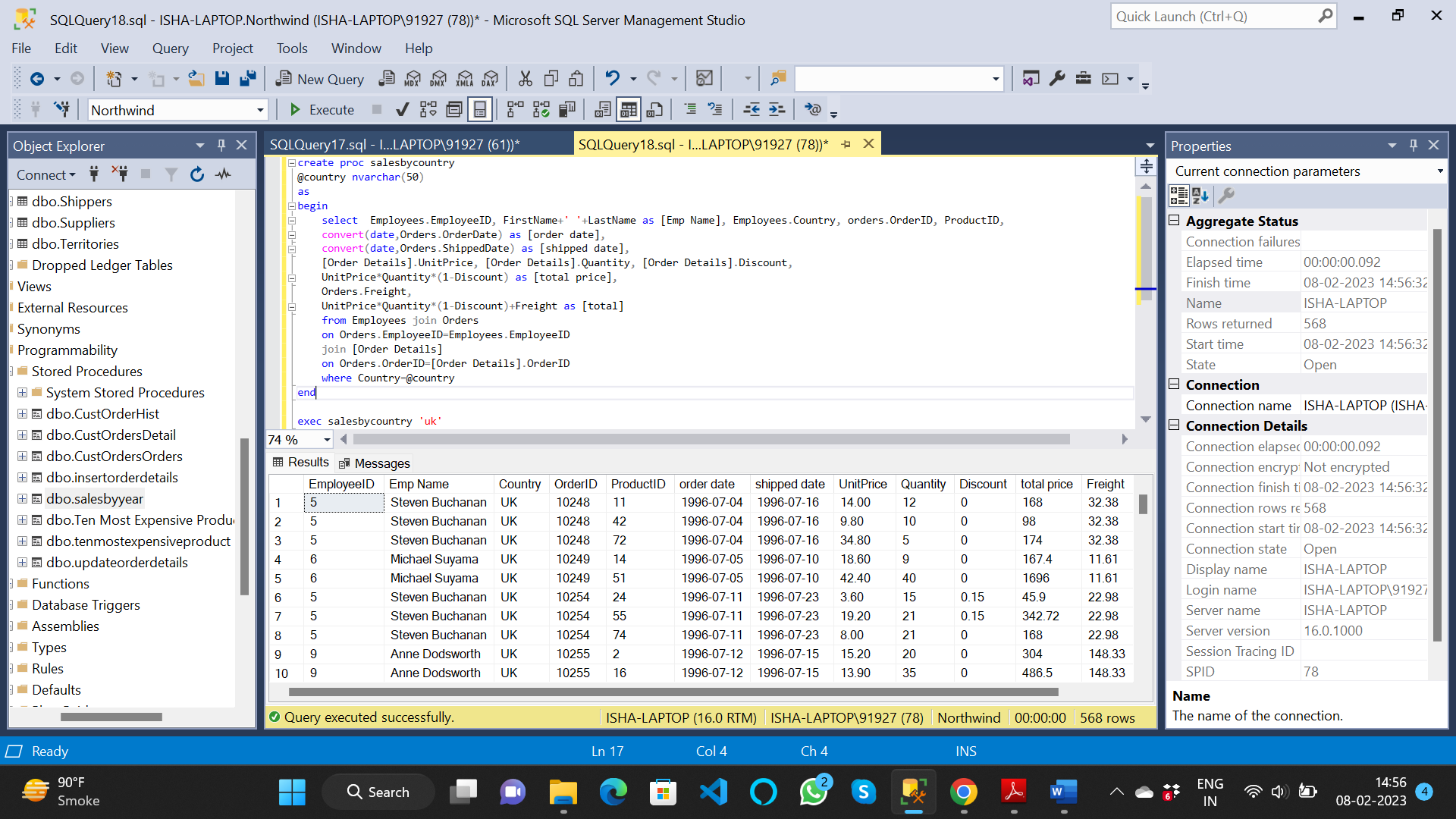
join [Order Details]

on Orders.OrderID=[Order Details].OrderID

where Country=@country

end

exec salesbycountry 'uk'



3. write a SQL query to Create Stored procedure in the Northwind database to retrieve

Sales by Year

create proc salesbyyear

@year int

as

begin

select orders.OrderID, ProductID, convert(date,Orders.OrderDate) as [order date],

convert(date,Orders.ShippedDate) as [shipped date],

[Order Details].UnitPrice, [Order Details].Quantity, [Order Details].Discount,

UnitPrice\*Quantity\*(1-Discount) as [total price],

Orders.Freight,

UnitPrice\*Quantity\*(1-Discount)+Freight as [total]

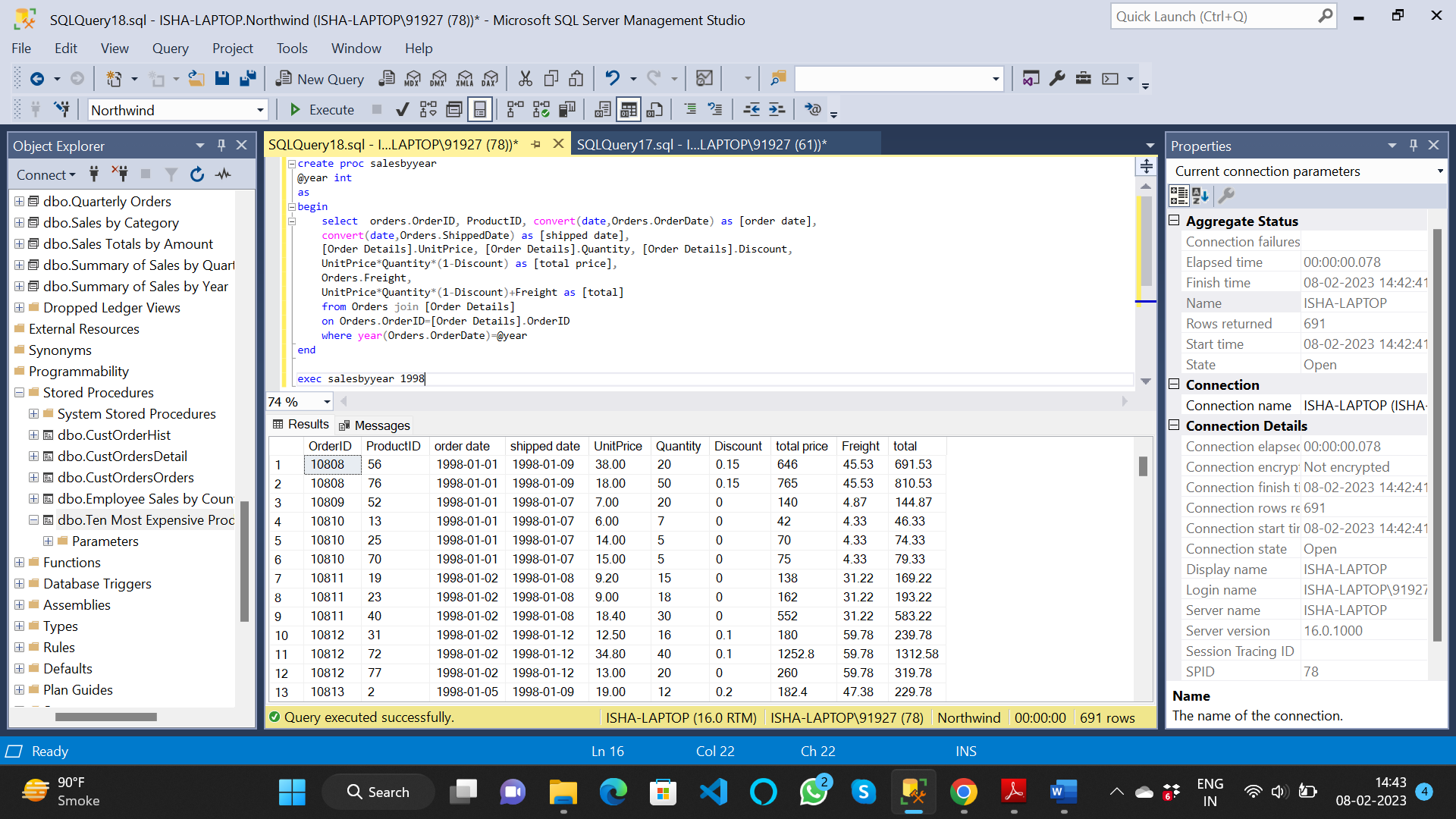
from Orders join [Order Details]

on Orders.OrderID=[Order Details].OrderID

where year(Orders.OrderDate)=@year

end

exec salesbyyear 1998



4. write a SQL query to Create Stored procedure in the Northwind database to retrieve

Sales By Category

create proc salesbycategory

@category nvarchar(50)

as

begin

select Categories.CategoryID, CategoryName, ProductID, ProductName, QuantityPerUnit,

UnitPrice, UnitsInStock, UnitsOnOrder, Discontinued

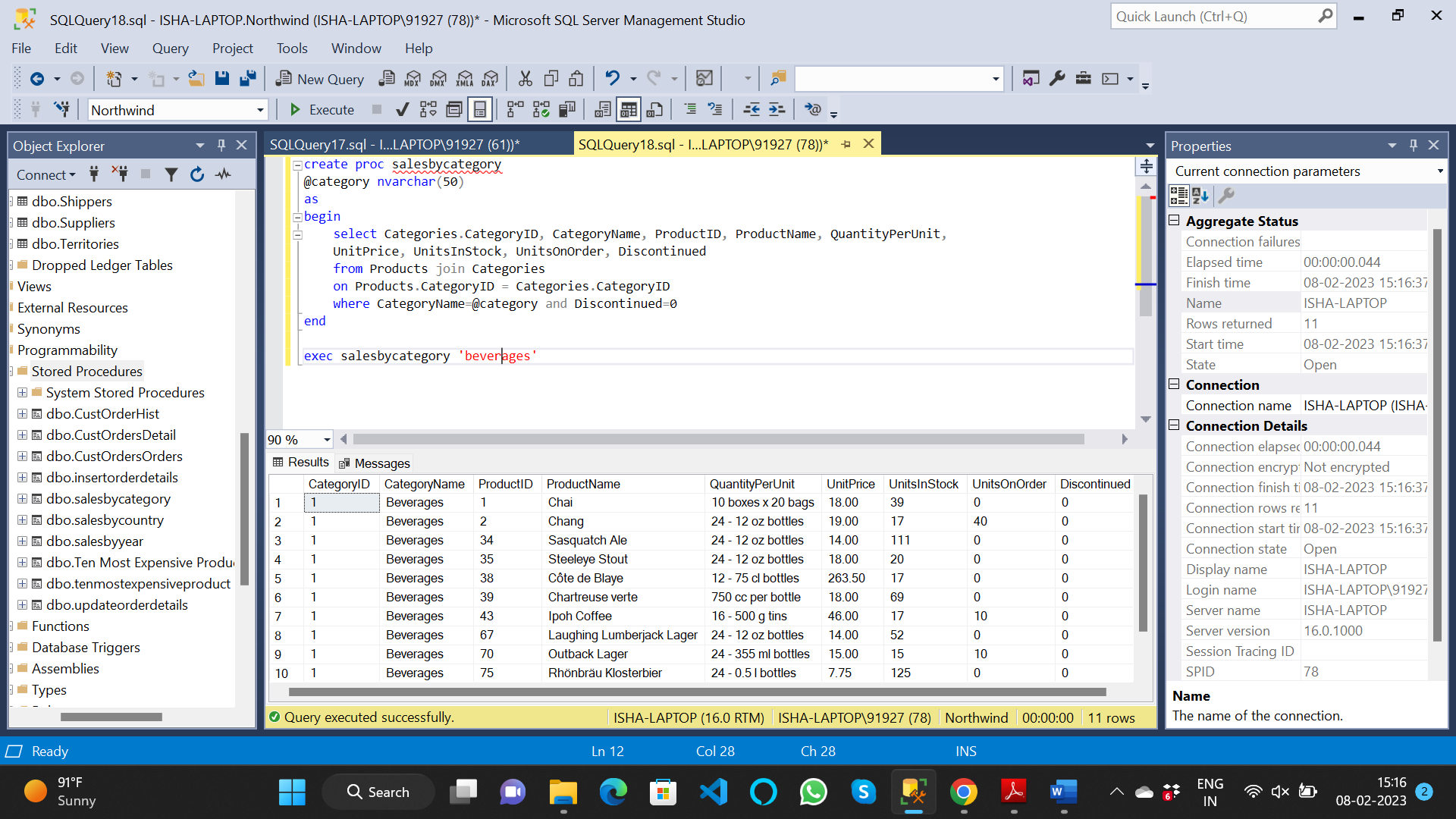
from Products join Categories

on Products.CategoryID = Categories.CategoryID

where CategoryName=@category and Discontinued=0

end

exec salesbycategory 'beverages'



5. write a SQL query to Create Stored procedure in the Northwind database to retrieve

Ten Most Expensive Products

create proc tenmostexpensiveproduct

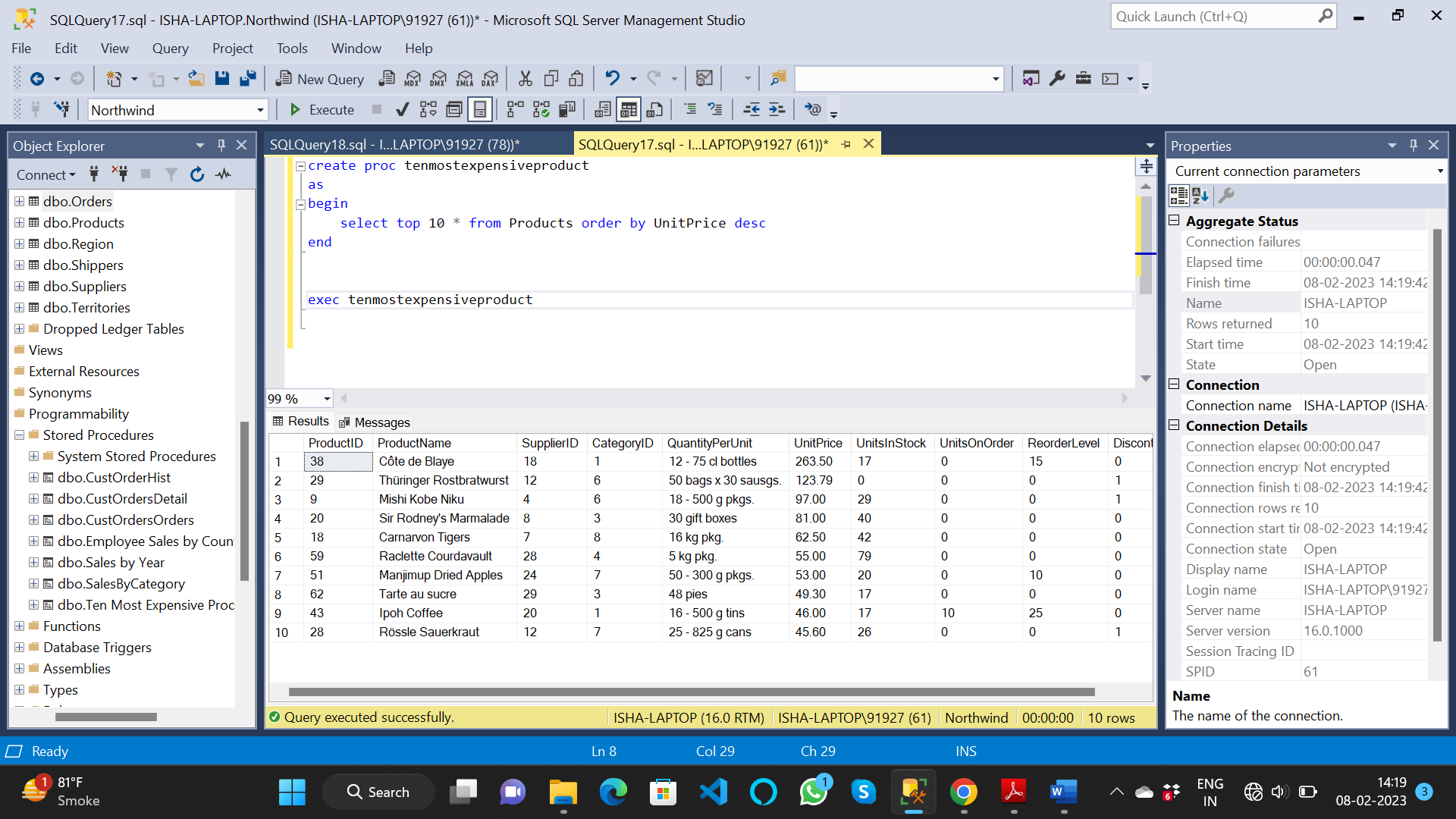
as

begin

select top 10 \* from Products order by UnitPrice desc

end

exec tenmostexpensiveproduct



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

Customer Order Details

create proc insertorderdetails

(

@order\_id int,

@product\_id int,

@unit\_price money,

@quantity smallint,

@discount real

)

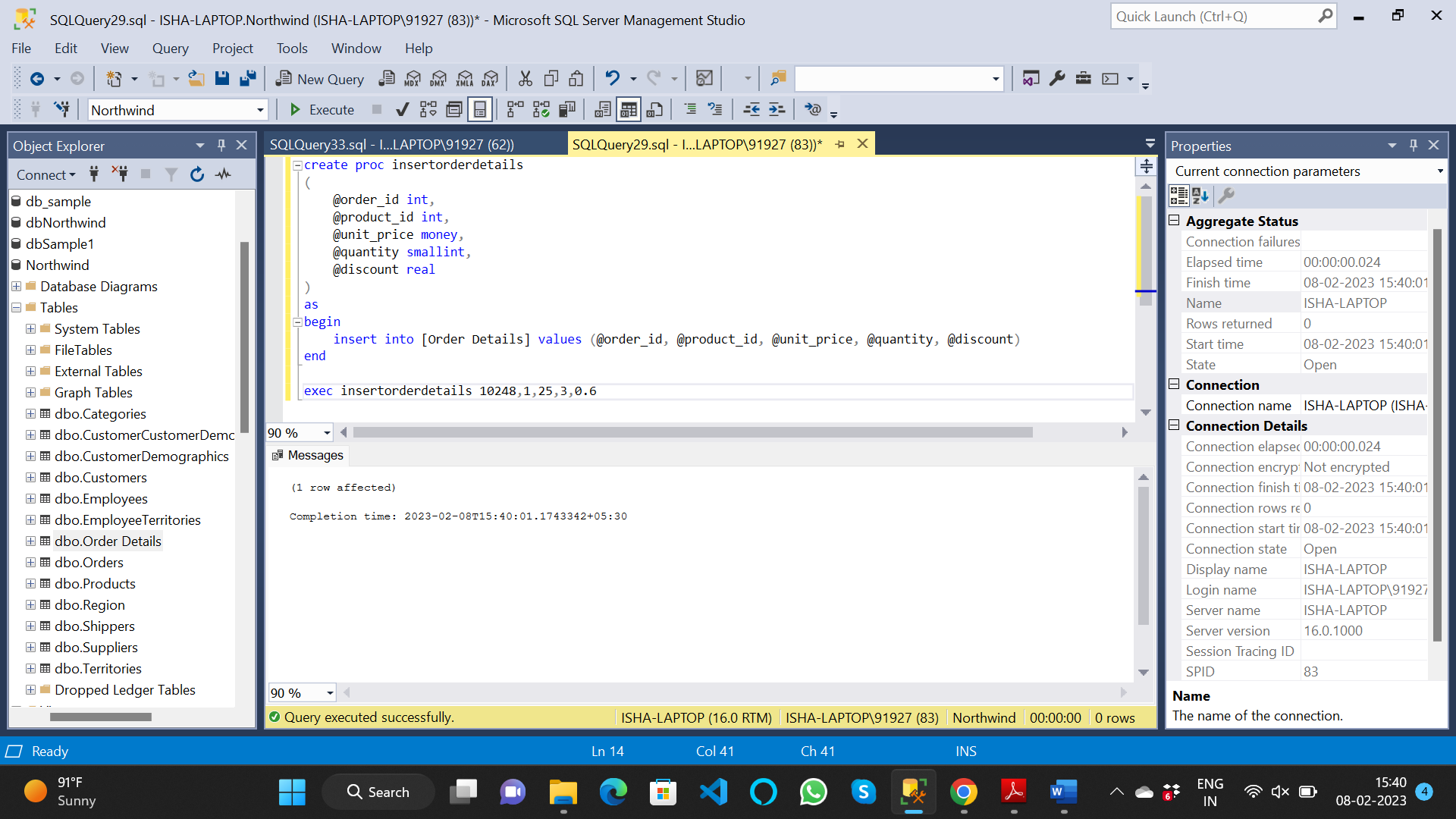
as

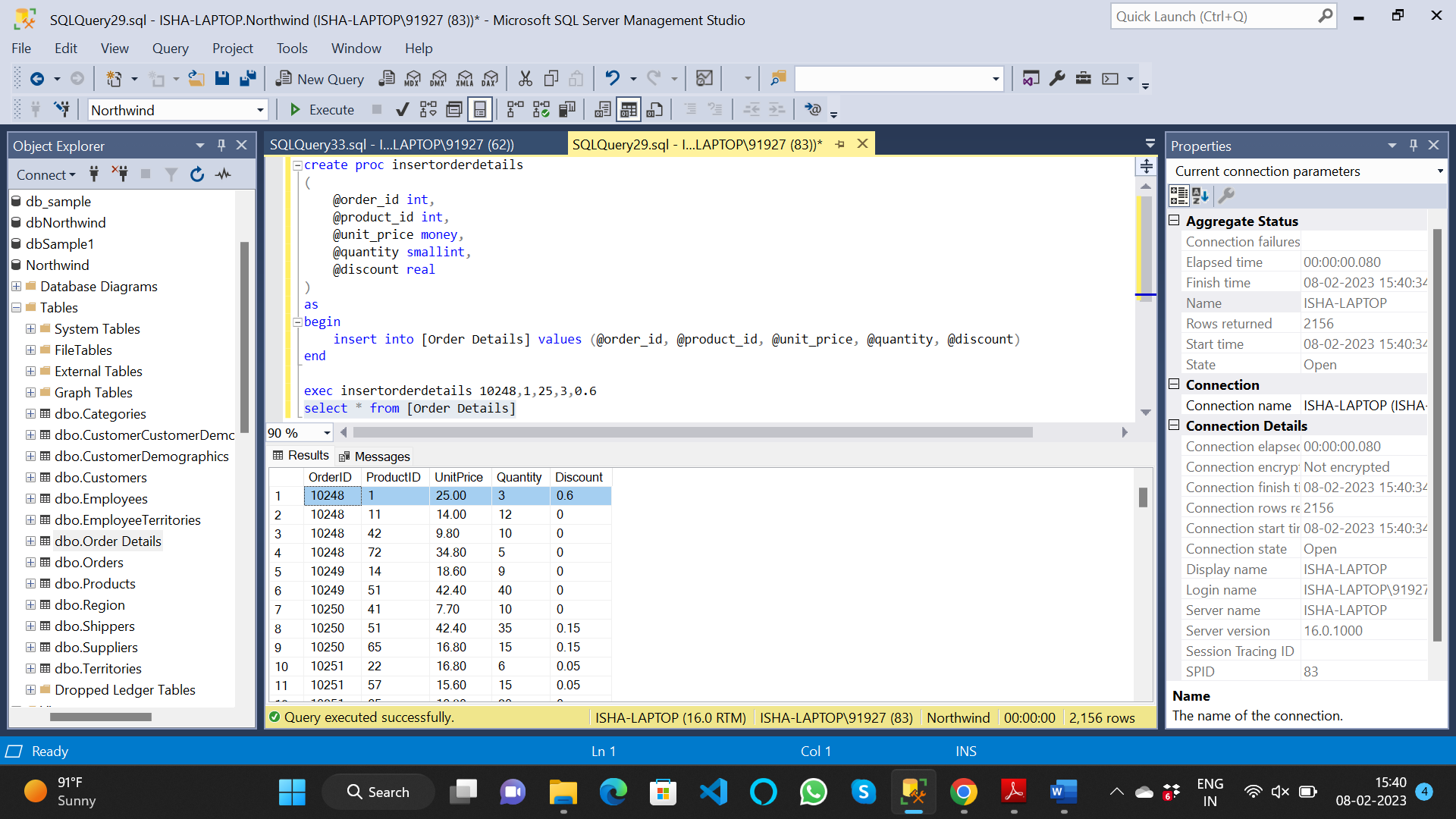
begin

insert into [Order Details] values (@order\_id, @product\_id, @unit\_price, @quantity, @discount)

end

exec insertorderdetails 10248,1,25,3,0.6





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

Customer Order Details

create proc updateorderdetails

(

@order\_id int,

@product\_id int,

@unit\_price money = null,

@quantity smallint = null,

@discount real = null

)

as

begin

if @unit\_price is not null

update [Order Details] set UnitPrice=@unit\_price where OrderID=@order\_id and ProductID=@product\_id

if @quantity is not null

update [Order Details] set Quantity=@quantity where OrderID=@order\_id and ProductID=@product\_id

if @discount is not null

update [Order Details] set Discount=@discount where OrderID=@order\_id and ProductID=@product\_id

end

exec updateorderdetails 10248,11,null,12

