**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.

alter PROCEDURE spcustomers\_AvgFreight

@customer\_id varchar(10),

@avgFreight real output

AS

select @avgFreight = (select avg(o.freight) as Avg\_Freight from customers c

inner join orders o

on o.customer\_id = c.customer\_id

where c.customer\_id = @customer\_id

group by c.contact\_name)

return @avgFreight

GO

CREATE TRIGGER orders\_Check\_Freight

on dbo.orders

after INSERT, UPDATE

AS

BEGIN

declare @freight real, @avgFreight real, @cust\_id varchar(10)

select @freight = freight, @cust\_id = customer\_id from inserted

exec spcustomers\_AvgFreight @cust\_id, @avgFreight output

if(@freight > @avgFreight)

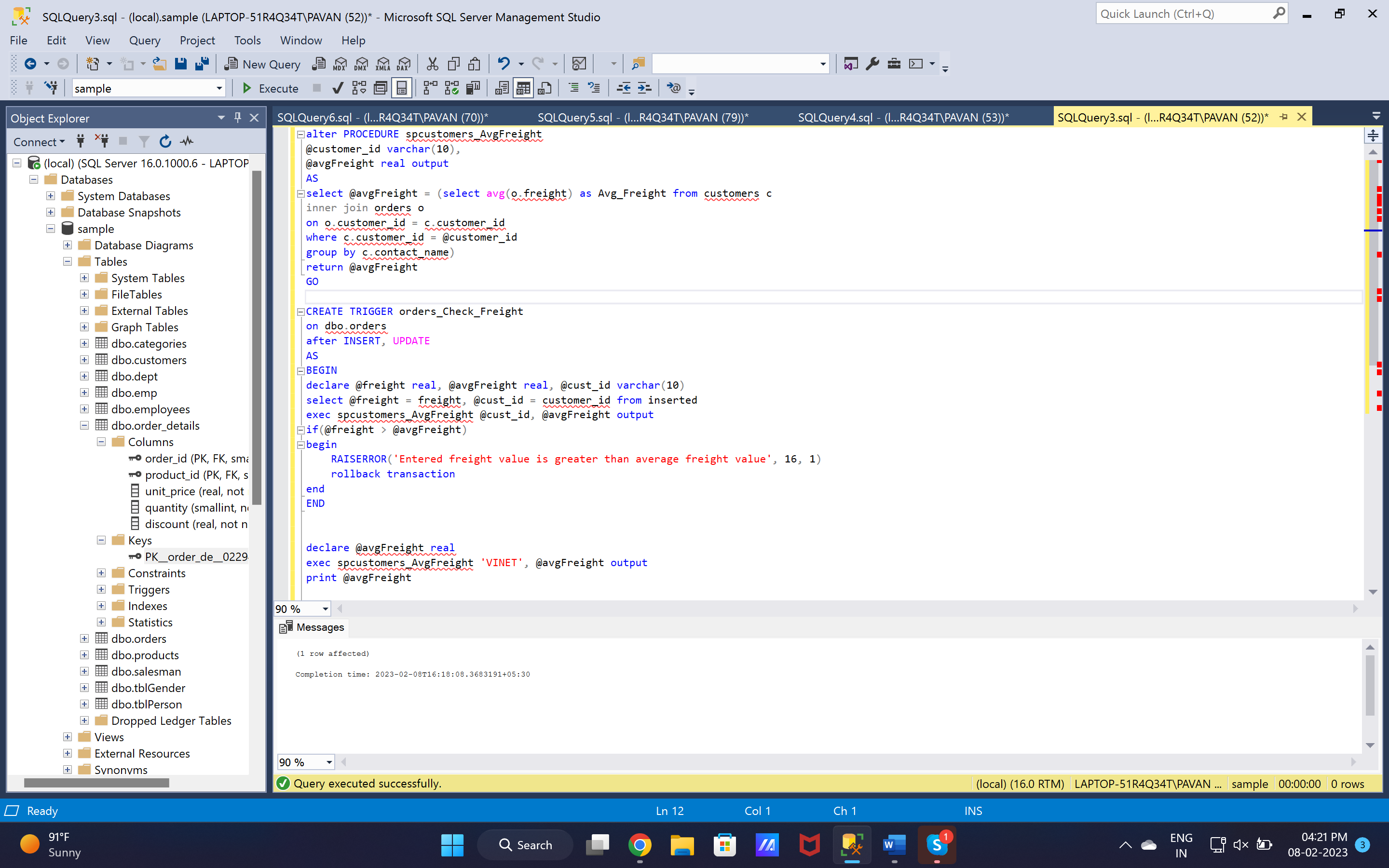
begin

RAISERROR('Entered freight value is greater than average freight value', 16, 1)

rollback transaction

end

END



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

create procedure spEmpSalesByCountry

@country varchar(15)

as

select e.first\_name, e.last\_name, count(o.customer\_id) as total\_sales, e.country from employees e

join orders o

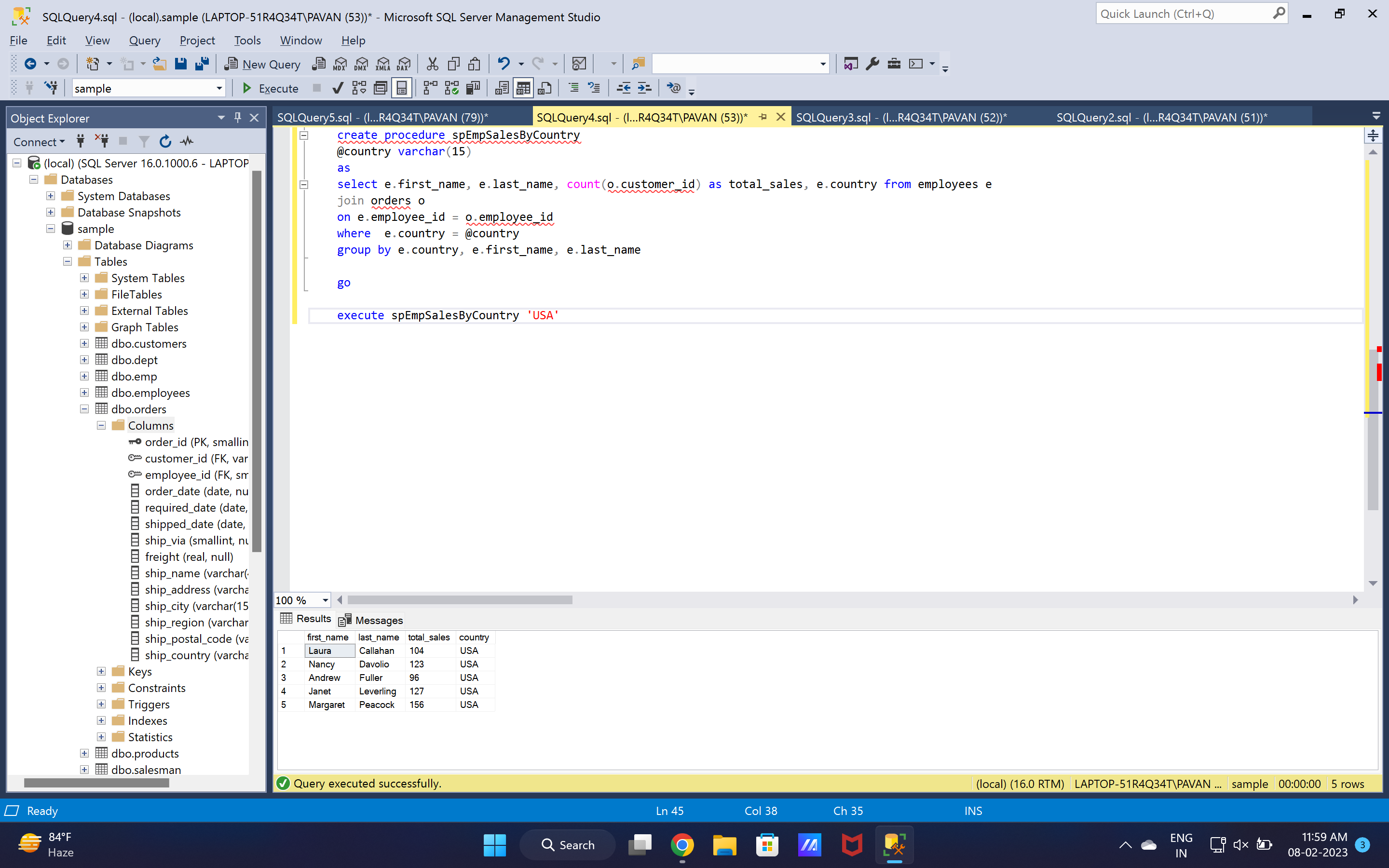
on e.employee\_id = o.employee\_id

where e.country = @country

group by e.country, e.first\_name, e.last\_name

go

execute spEmpSalesByCountry 'USA'



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

create procedure spSalesByYear

@year varchar(4)

as

select e.first\_name, e.last\_name, count(o.customer\_id) as total\_sales,

year(o.order\_date) as sales\_year from employees e

join orders o

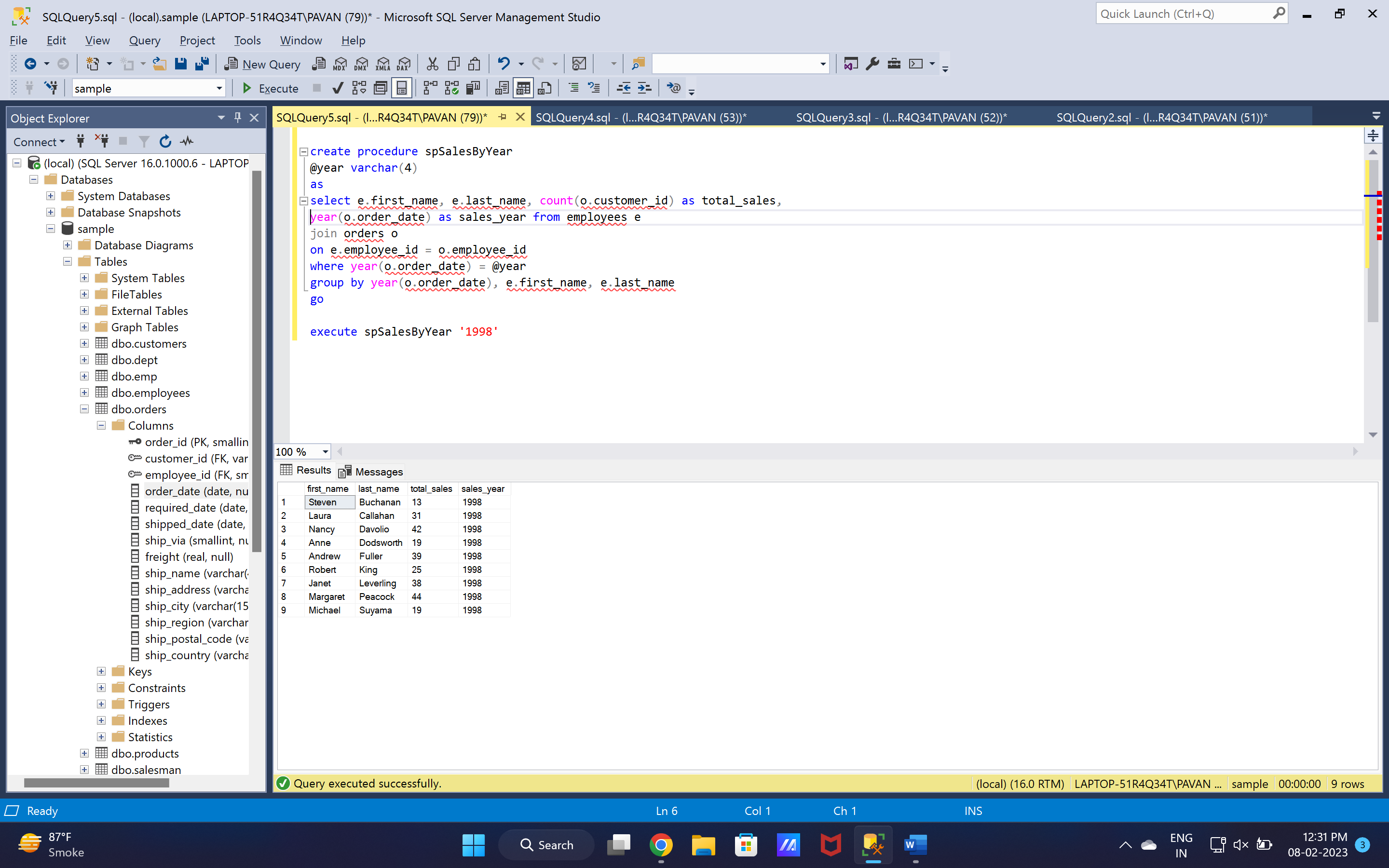
on e.employee\_id = o.employee\_id

where year(o.order\_date) = @year

group by year(o.order\_date), e.first\_name, e.last\_name

go

execute spSalesByYear '1998'



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

create procedure spSalesByCategory

@category varchar(20)

as

select count(o.order\_id) as total\_sales, c.category\_name from orders o

join order\_details od

on o.order\_id = od.order\_id

join products p

on od.product\_id = p.product\_id

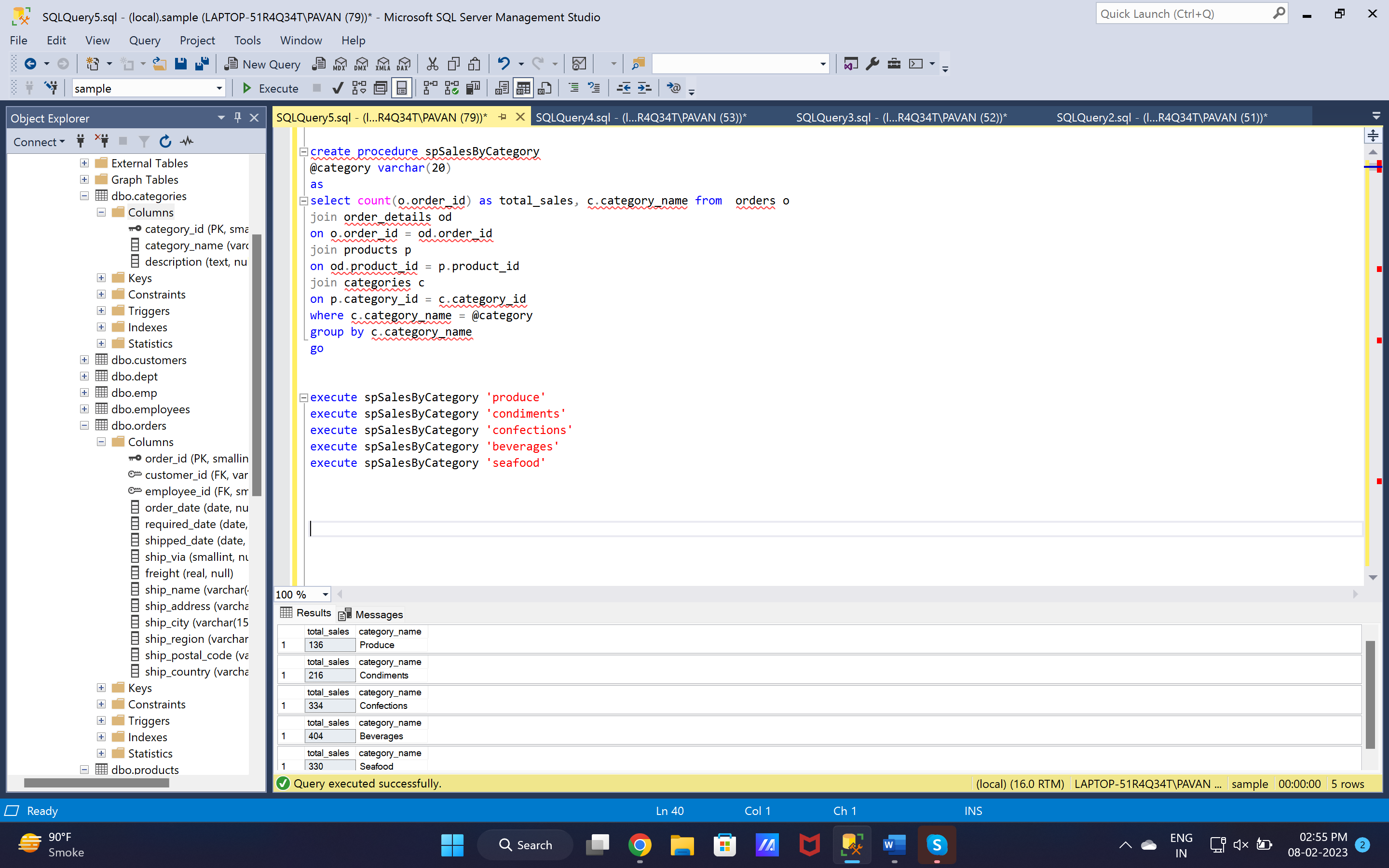
join categories c

on p.category\_id = c.category\_id

where c.category\_name = @category

group by c.category\_name

go



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

alter procedure spMostExpensive

as

select top 10 od.unit\_price as unitPrice, p.product\_name from order\_details od

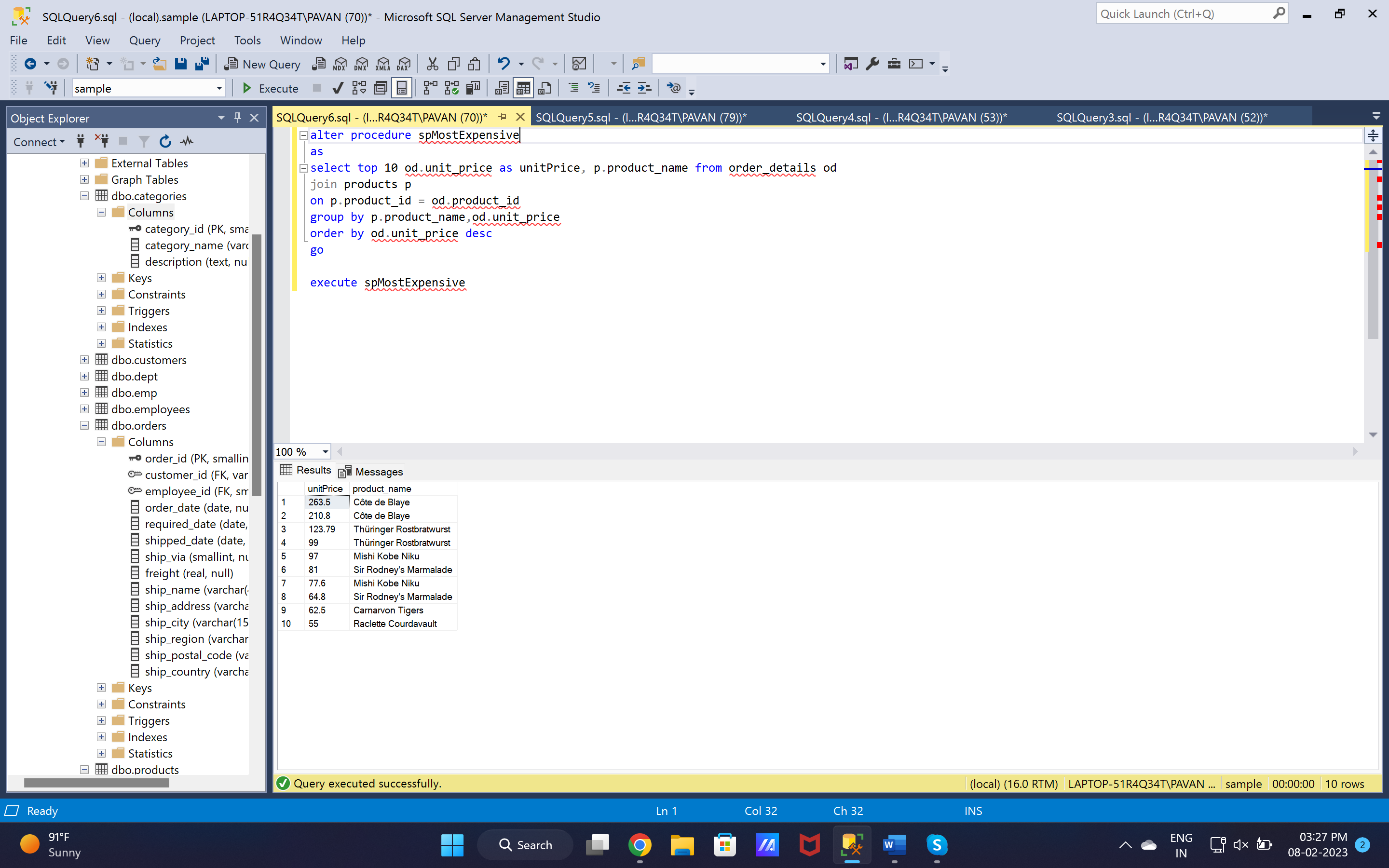
join products p

on p.product\_id = od.product\_id

group by p.product\_name,od.unit\_price

order by od.unit\_price desc

go



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

alter procedure spInsertOrderDetails

@order\_id smallint,

@product\_id smallint,

@unit\_price real,

@quantity smallint,

@discount real

as

begin

insert into order\_details(

order\_id,

product\_id,

unit\_price,

quantity,

discount

) values(

@order\_id,

@product\_id,

@unit\_price,

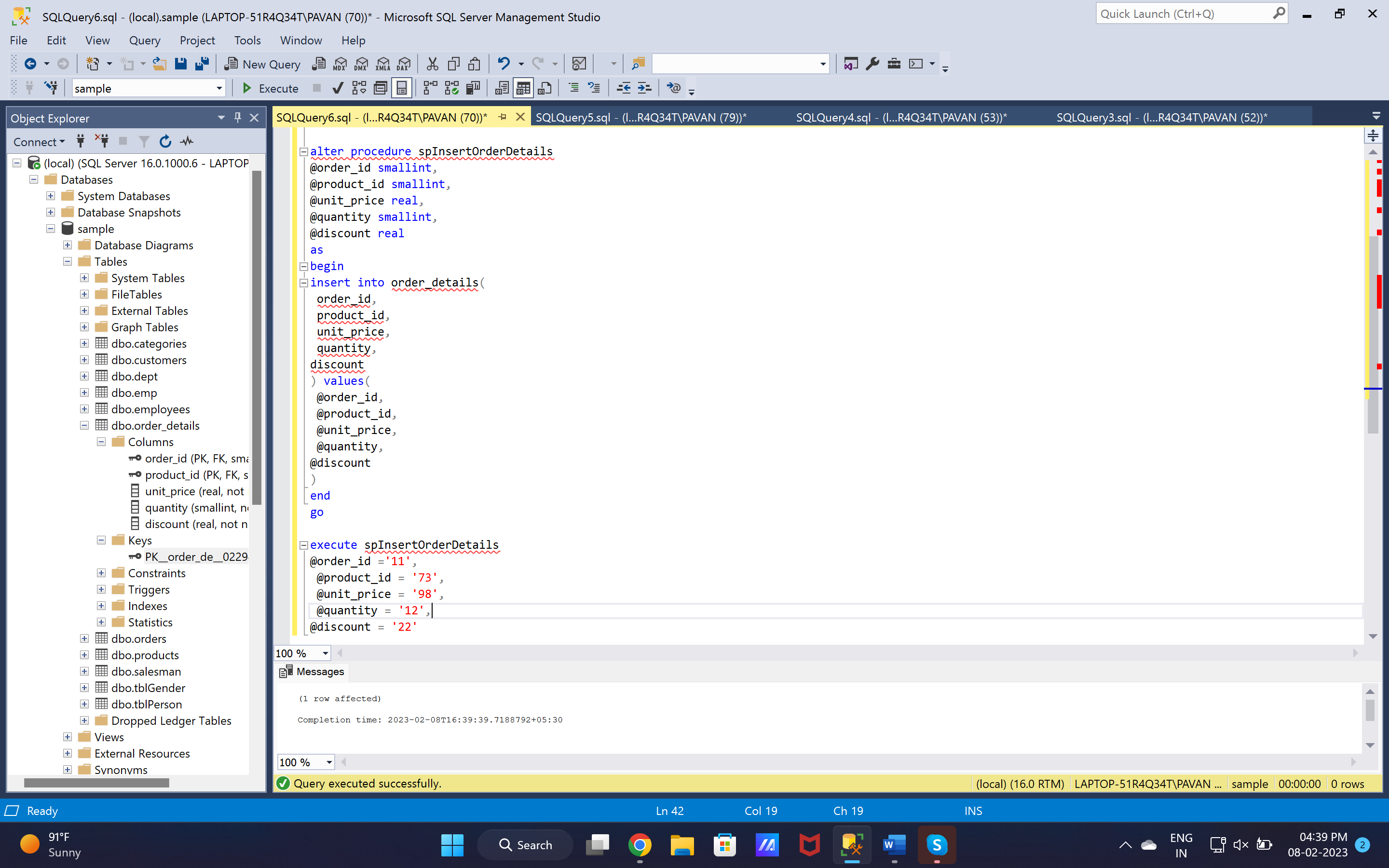
@quantity,

@discount

)

end

go



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

alter procedure spUpdateOrderDetails

@order\_id smallint,

@product\_id smallint,

@unit\_price real,

@quantity smallint,

@discount real

as

begin

update order\_details

set

unit\_price = @unit\_price,

quantity = @quantity,

discount =@discount

where order\_id = @order\_id AND product\_id = @product\_id

end

go

