---------------------------------------------------------------------

-- LAB 15

-- Exercise 1

---------------------------------------------------------------------

USE TSQL;

GO

---------------------------------------------------------------------

-- Task 1

-- Execute the provided T-SQL code to create the stored procedure Sales.GetTopCustomers.

CREATE PROCEDURE Sales.GetTopCustomers AS

SELECT TOP(10)

c.custid,

c.contactname,

SUM(o.val) AS salesvalue

FROM Sales.OrderValues AS o

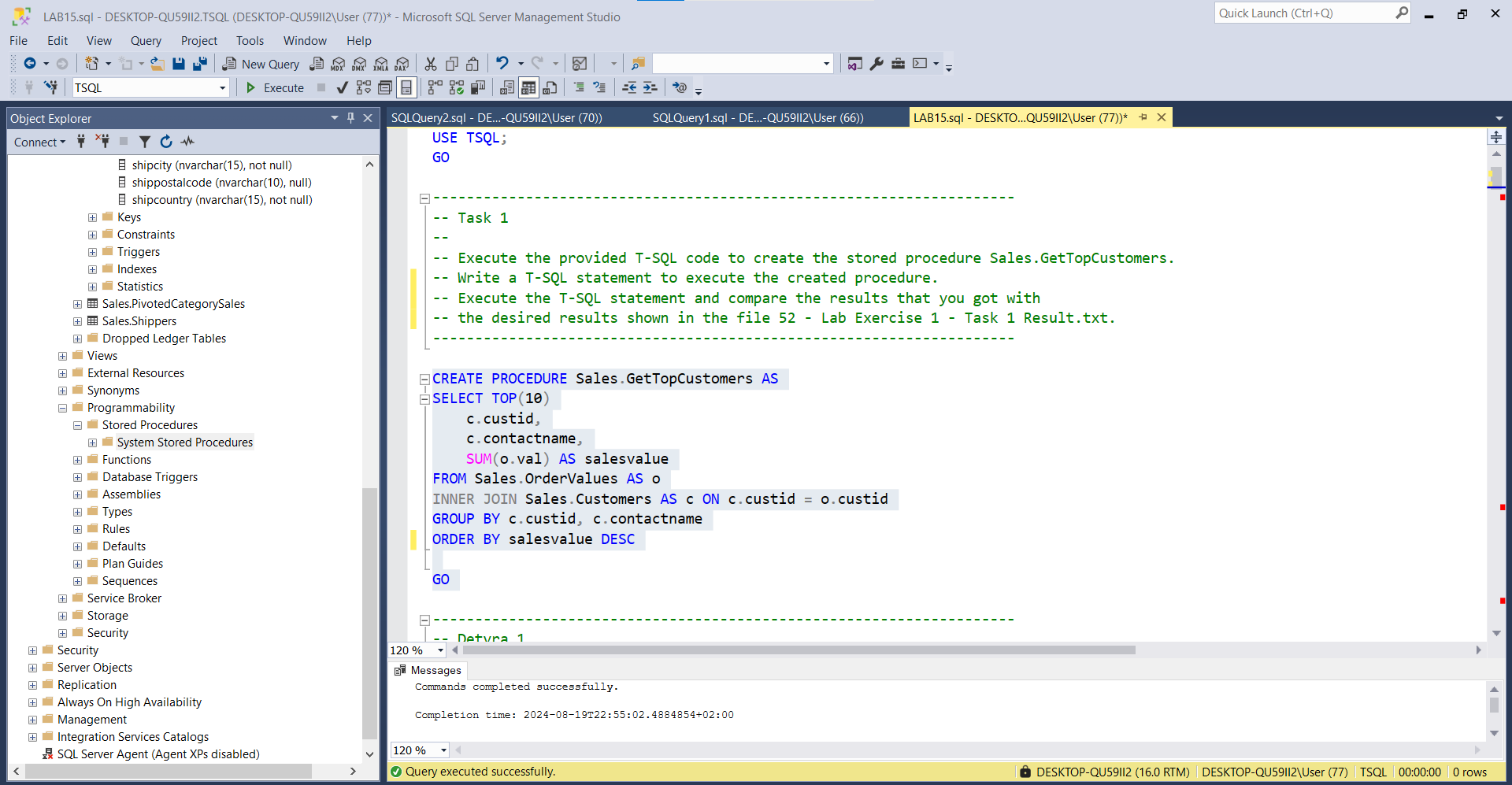
INNER JOIN Sales.Customers AS c ON c.custid = o.custid

GROUP BY c.custid, c.contactname

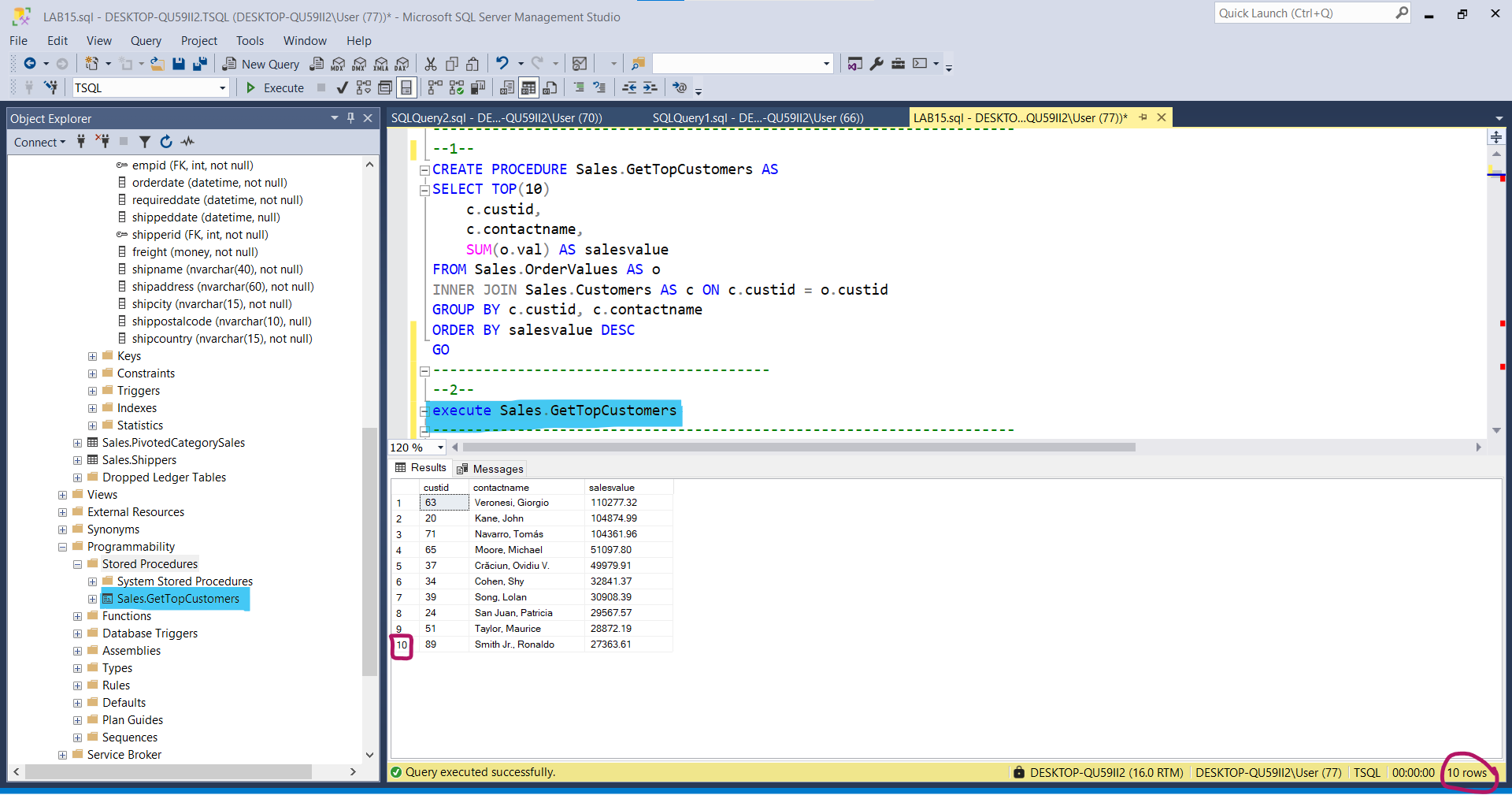
ORDER BY salesvalue DESC

GO

\*\*Procedure is created Successfully!!



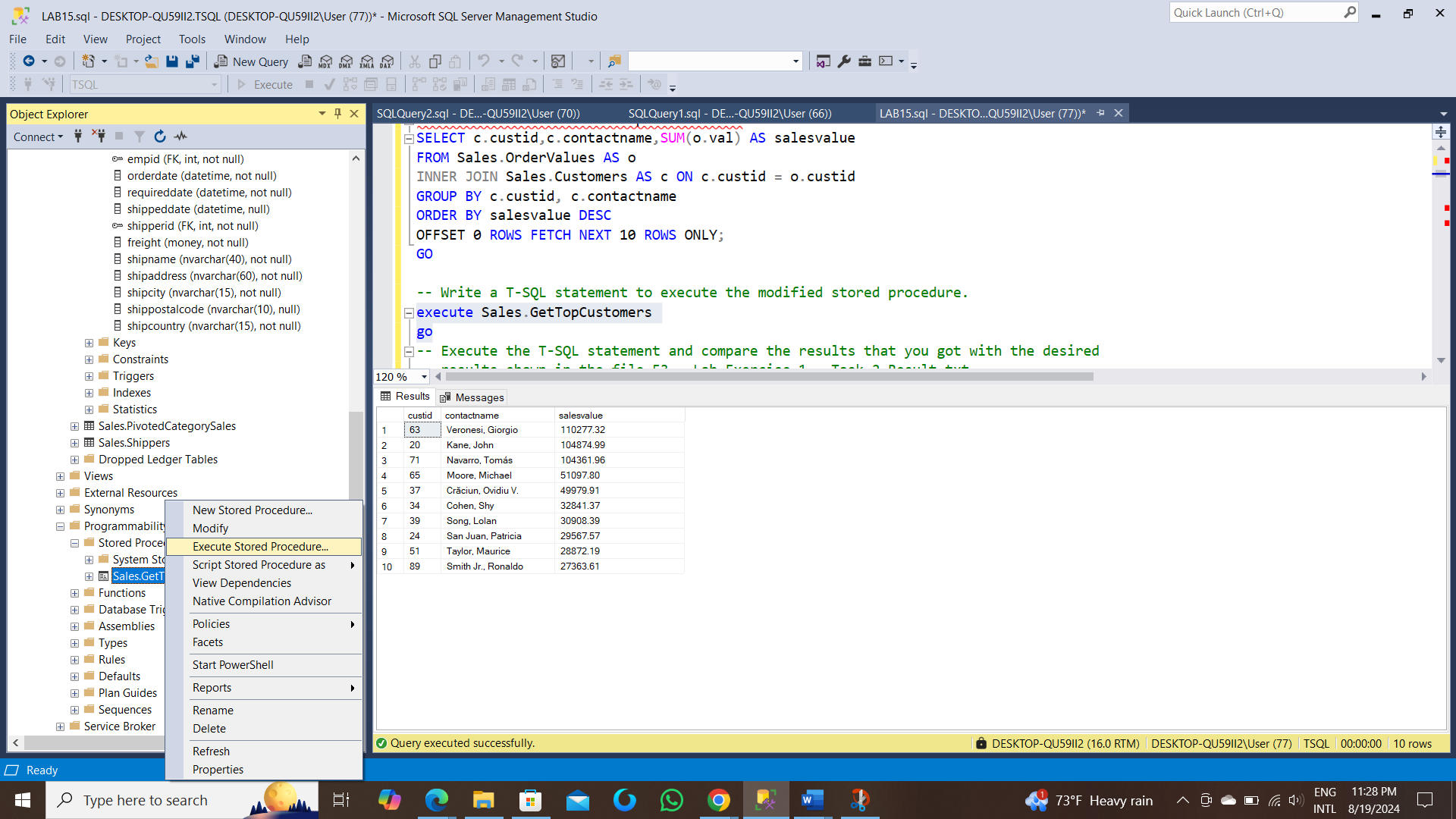
-- Write a T-SQL statement to execute the created procedure.

execute Sales.GetTopCustomers

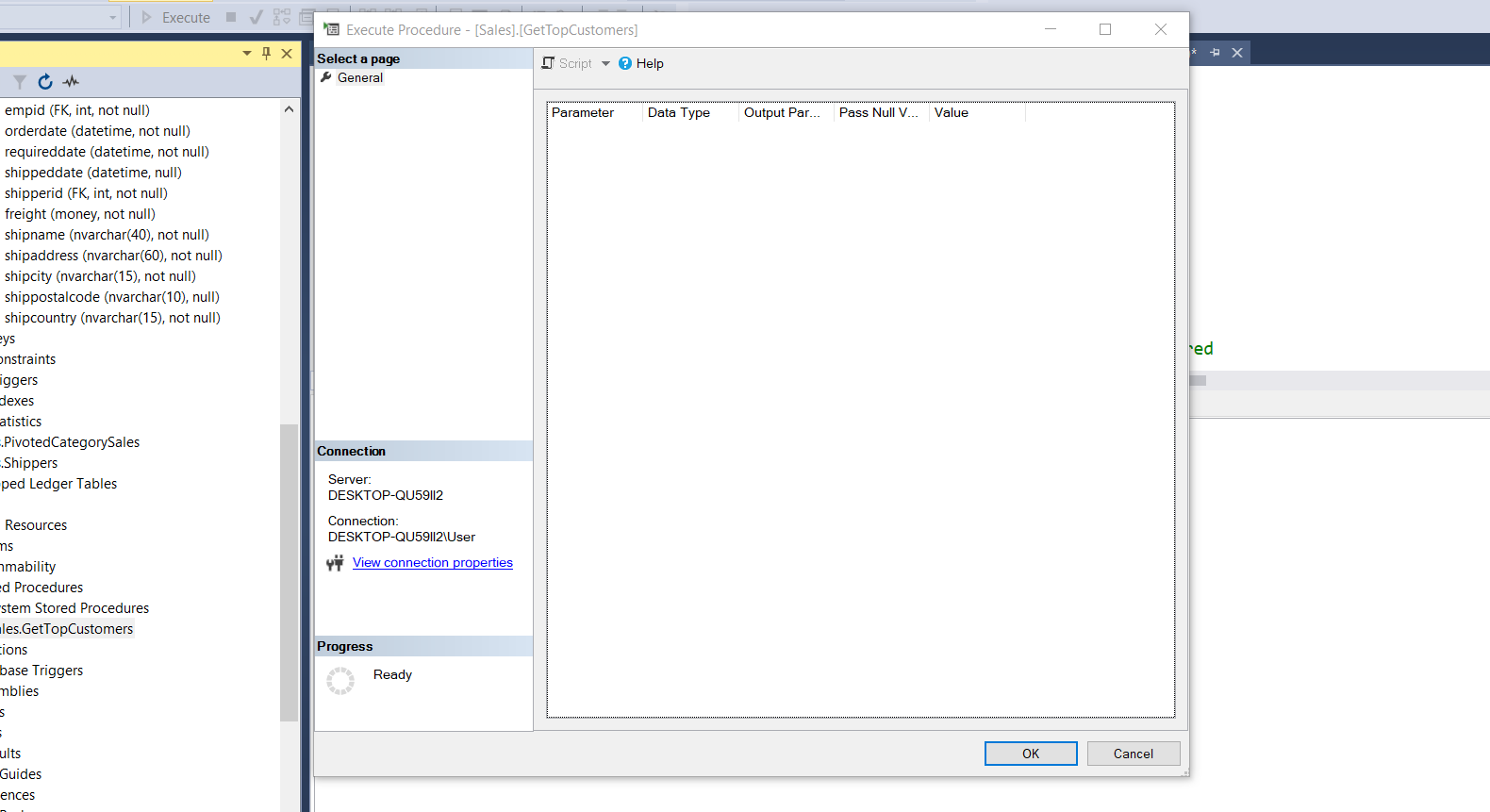
-- Execute the T-SQL statement and compare the results that you got with the desired results shown in the file 52 - Lab Exercise 1 - Task 1 Result.txt.

*Compare…*

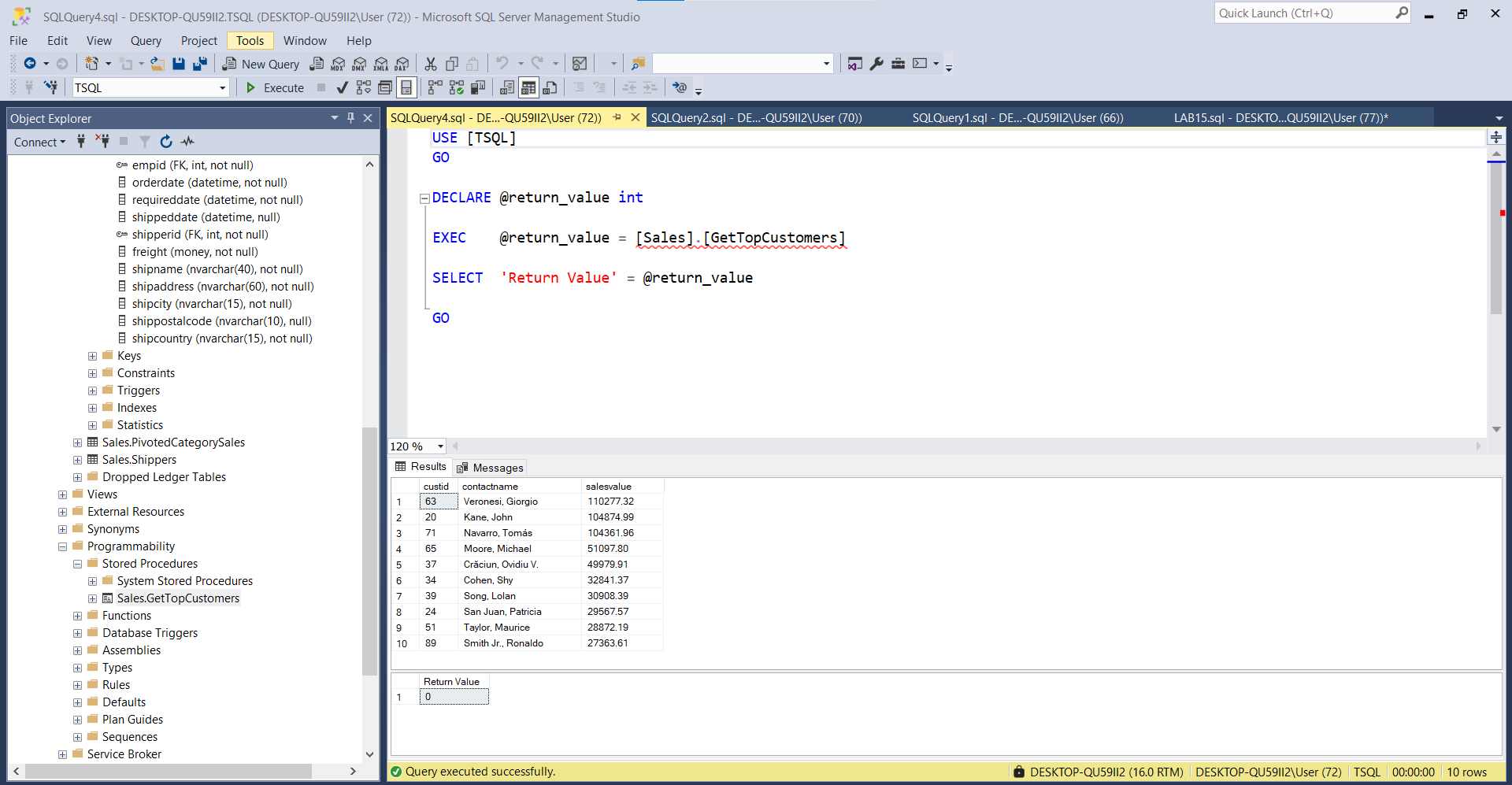
*--Execute manually*



--Step 2



--After pressing OK Button



---------------------------------------------------------------------

---------------------------------------------------------------------

-- Task 2

--

-- The IT department has changed the stored procedure from task 1 and has supplied you with T-SQL code to apply the needed changes. Execute the provided T-SQL code.

ALTER PROCEDURE Sales.GetTopCustomers AS

SELECT c.custid,c.contactname,SUM(o.val) AS salesvalue

FROM Sales.OrderValues AS o

INNER JOIN Sales.Customers AS c ON c.custid = o.custid

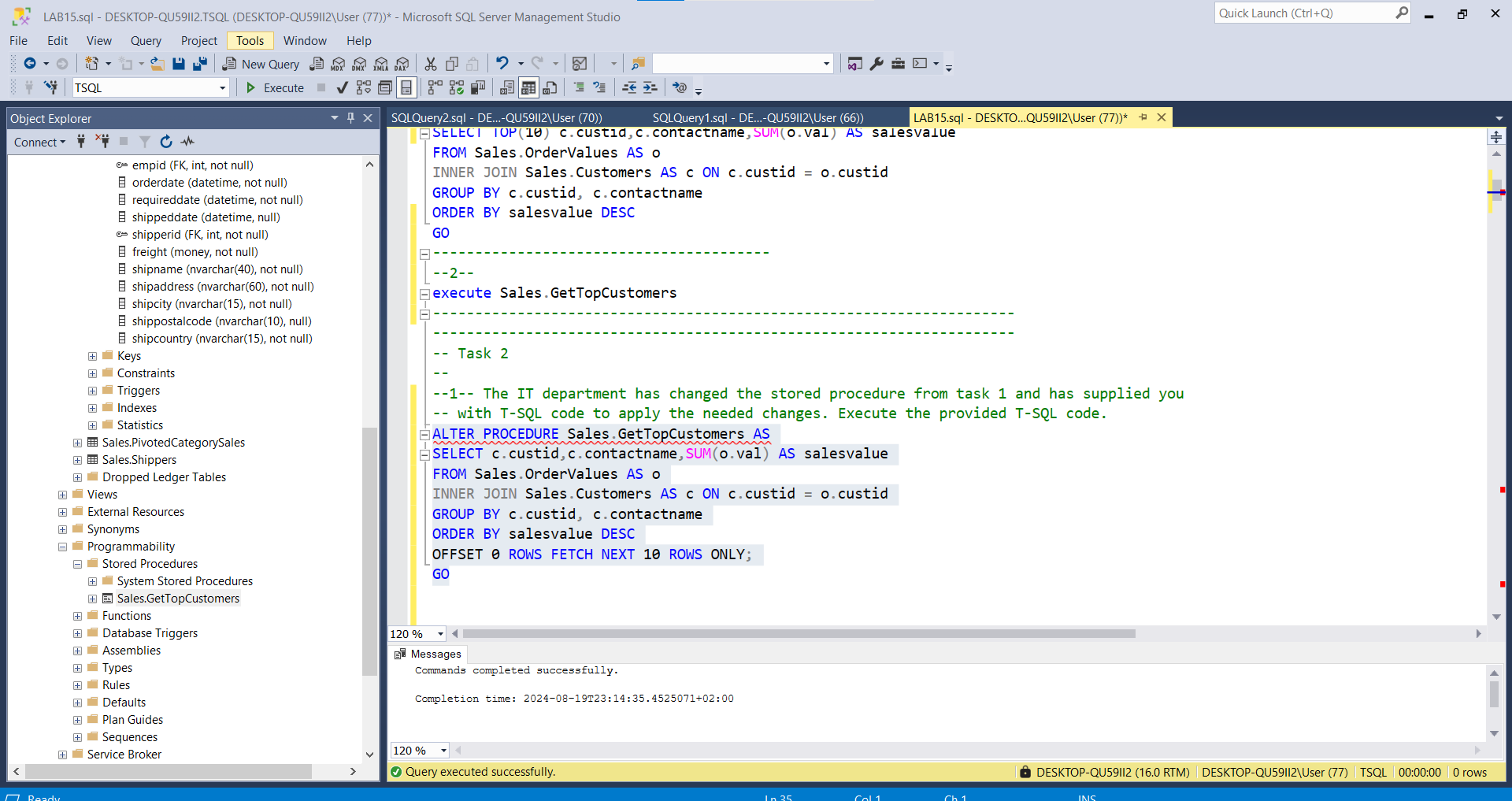
GROUP BY c.custid, c.contactname

ORDER BY salesvalue DESC

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

GO

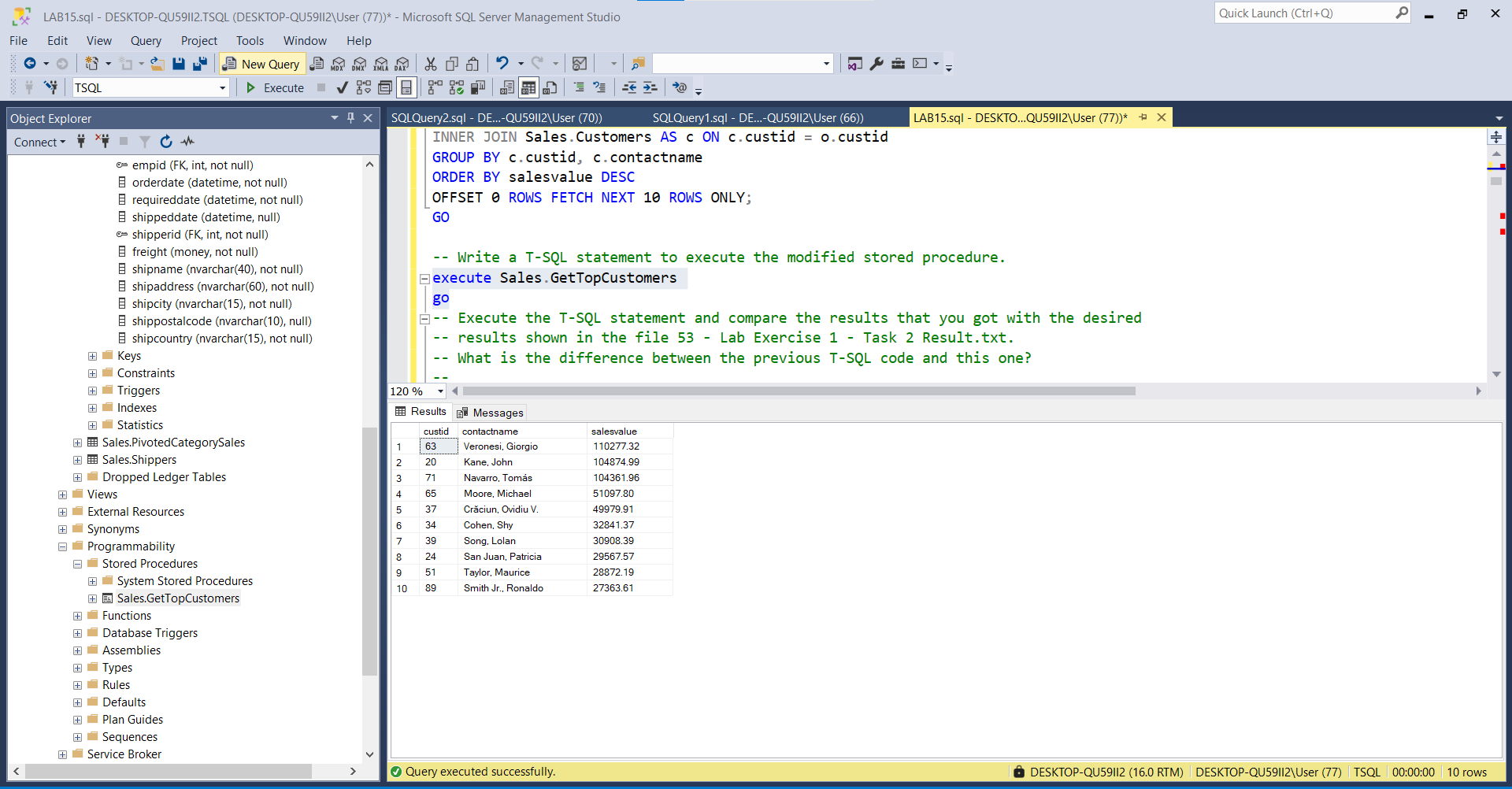
\*\*Stored Procedure is created Successfully



-- Write a T-SQL statement to execute the modified stored procedure.

execute Sales.GetTopCustomers

go The modified stored procedure IS EXECUTED :



-- Execute the T-SQL statement and compare the results that you got with the desired results shown in the file 53 - Lab Exercise 1 - Task 2 Result.txt.

--

-- What is the difference between the previous T-SQL code and this one?

execute Sales.GetTopCustomers

custid | contactname | salesvalue

---------------------------------------------------------------

63 Veronesi, Giorgio 110277.32

20 Kane, John 104874.99

71 Navarro, Tomás 104361.96

65 Moore, Michael 51097.80

37 Crăciun, Ovidiu V. 49979.91

34 Cohen, Shy 32841.37

39 Song, Lolan 30908.39

24 San Juan, Patricia 29567.57

51 Taylor, Maurice 28872.19

89 Smith Jr., Ronaldo 27363.61

execute Sales.GetTopCustomers

go



\*\*There is no difference!!!!!!!!!!!!

-- If some applications are using the stored procedure from task 1, would they still work properly after the changes you have applied in task 2? \*\*There is no difference!!!!!!!!!!!!

--------------------------------------------------------------------

---------------------------------------------------------------------

-- LAB 15

-- Exercise 2

---------------------------------------------------------------------

USE TSQL;

GO

---------------------------------------------------------------------

-- Task 1

--

-- Execute the provided T-SQL code to modify the Sales.GetTopCustomers stored procedure to include a parameter for order year (@orderyear).

CREATE PROCEDURE Sales.GetTopCustomersI(@orderyear int)

AS

SELECT TOP(10) c.custid,c.contactname,SUM(o.val) AS salesvalue,year(o.orderdate) AS orderyear

FROM Sales.OrderValues AS o

INNER JOIN Sales.Customers AS c ON c.custid = o.custid

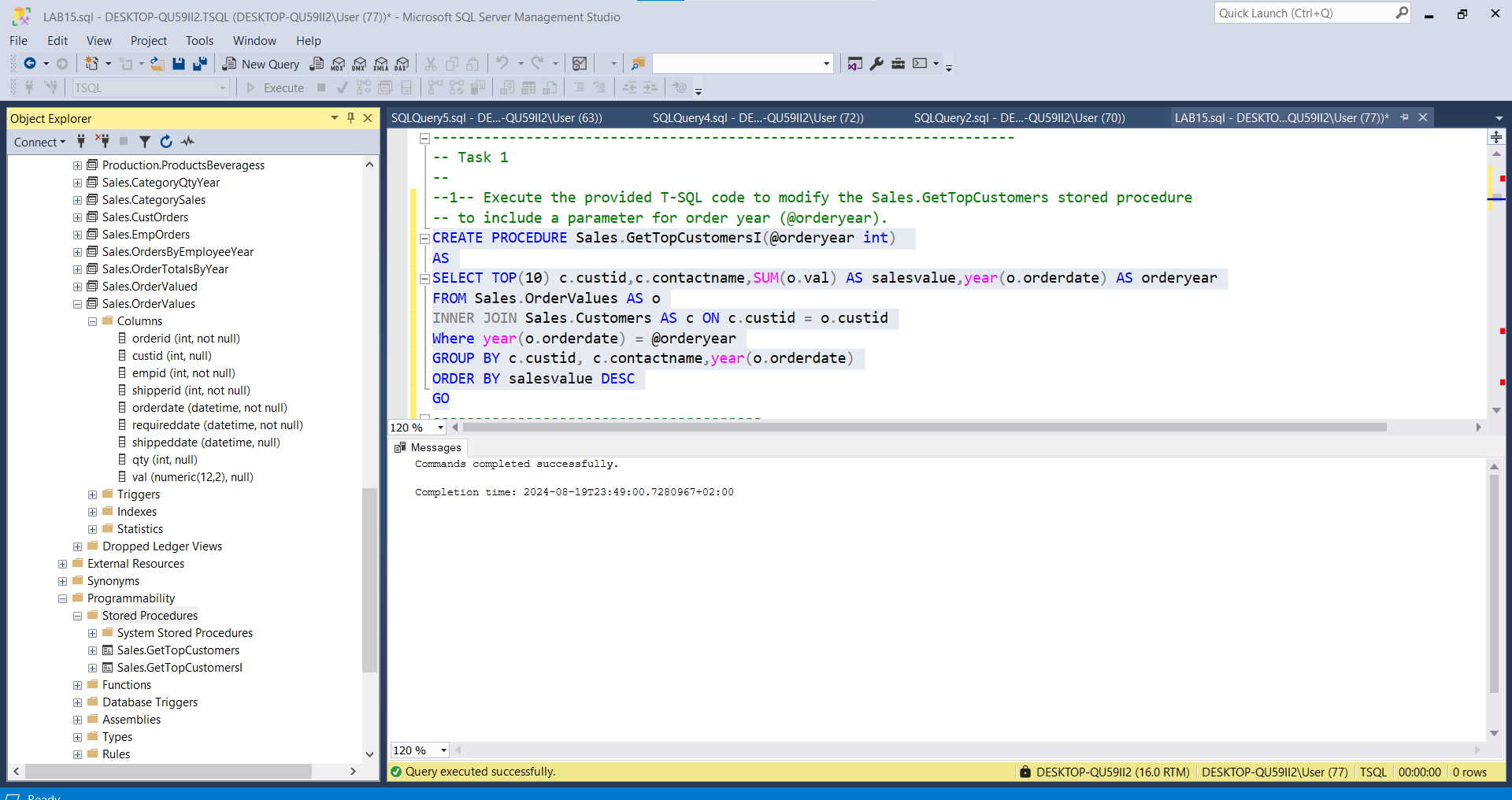
Where year(o.orderdate) = @orderyear

GROUP BY c.custid, c.contactname,year(o.orderdate)

ORDER BY salesvalue DESC

GO

--Stored Procedure ‘Sales.GetTopCustomersI’ is created successfully

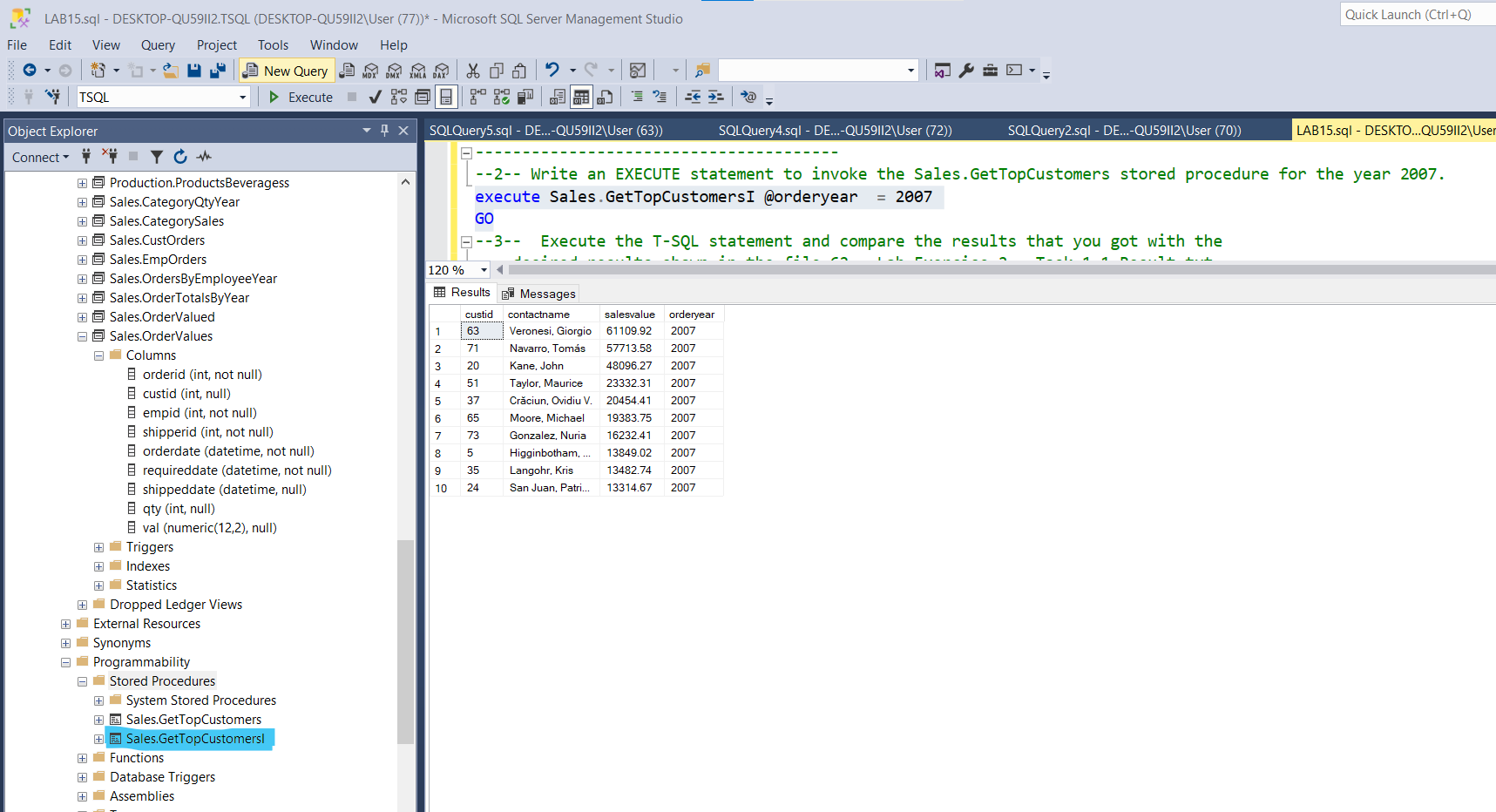


-- Write an EXECUTE statement to invoke the Sales.GetTopCustomers stored procedure for the year 2007.

execute Sales.GetTopCustomersI @orderyear = 2007

GO

-- Execute the T-SQL statement and compare the results that you got with the desired results shown in the file 62 - Lab Exercise 2 - Task 1\_1 Result.txt.

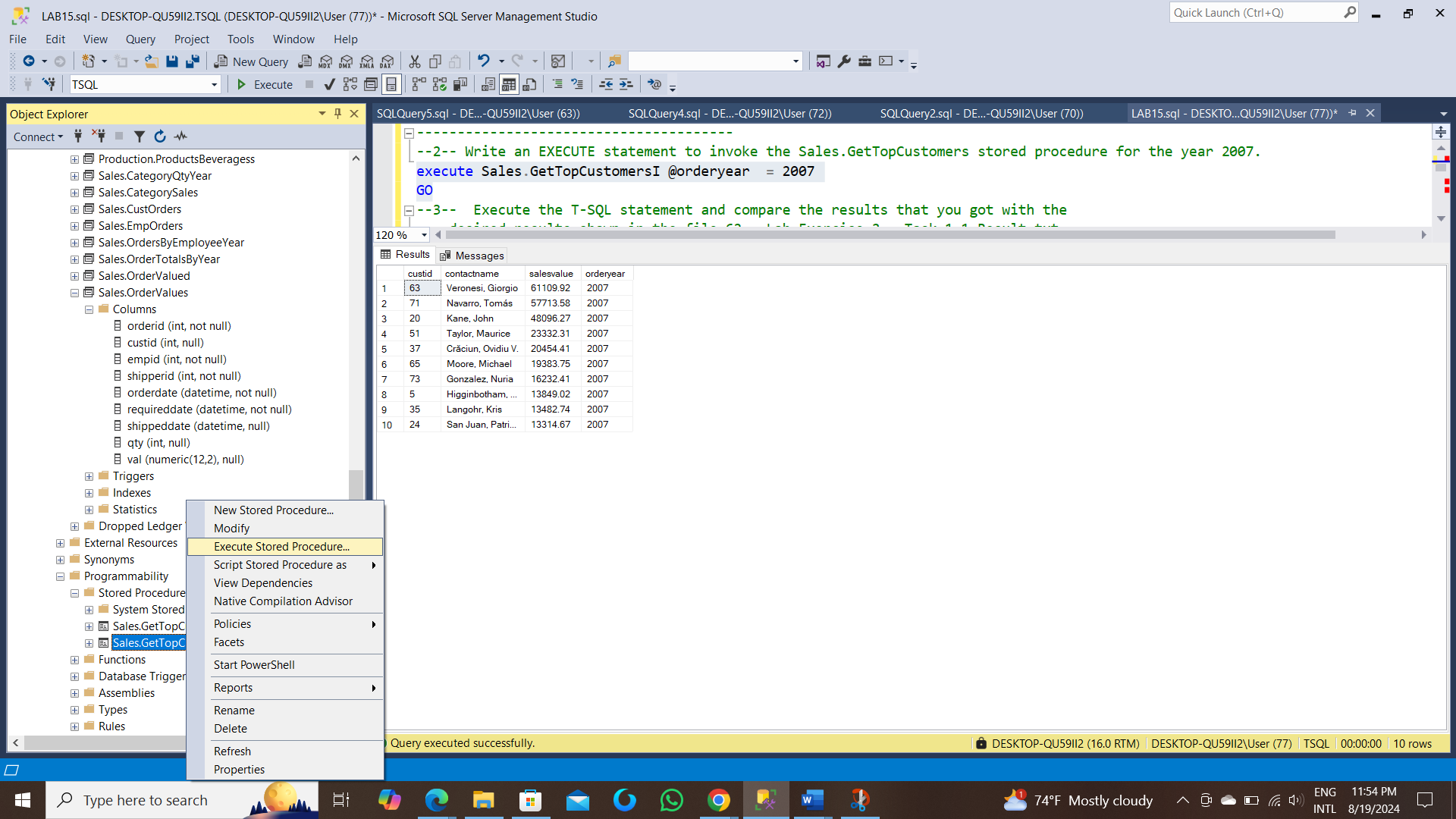




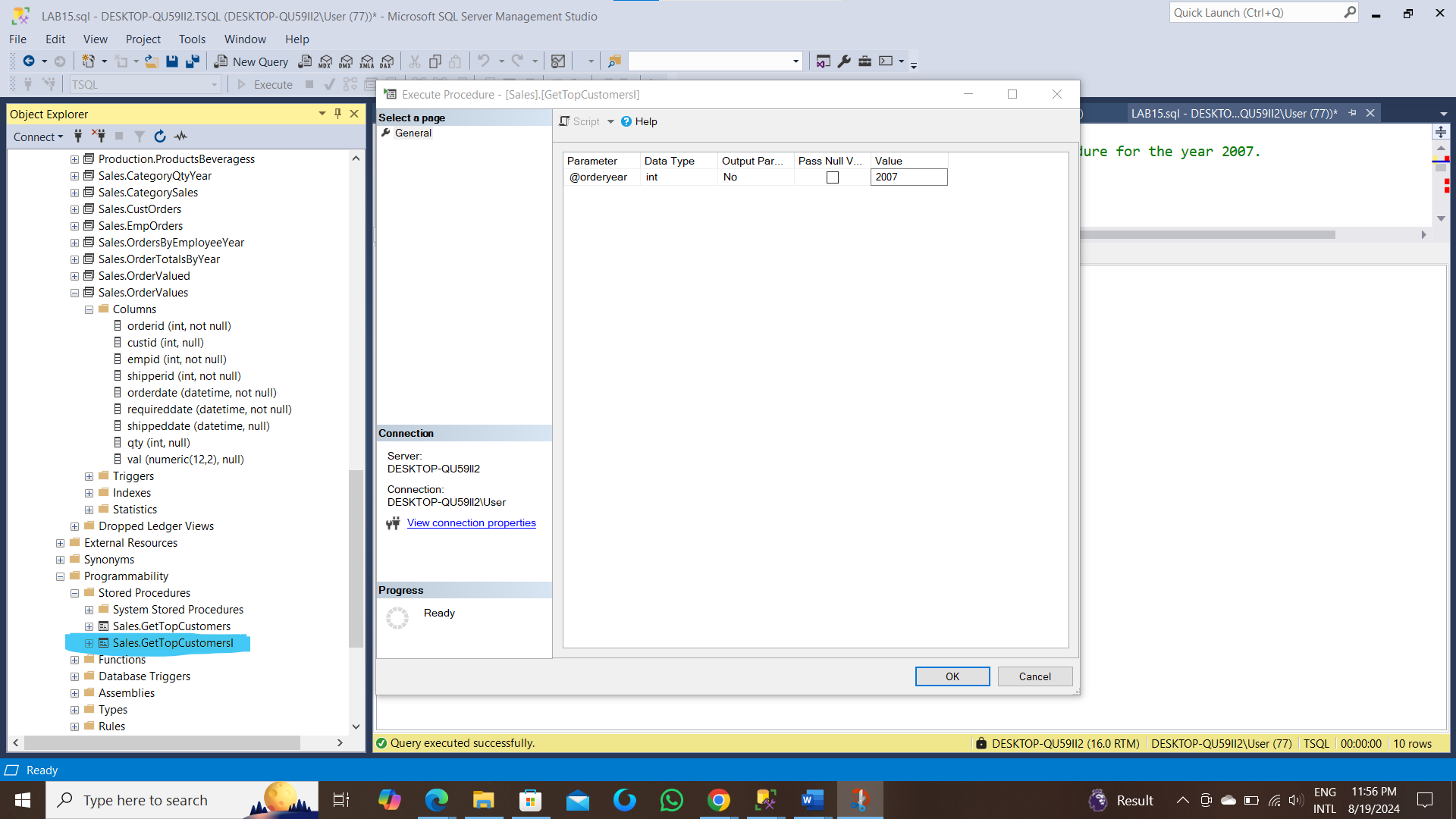
By Comparing…

--The selected Stored Procedure is executed manually:

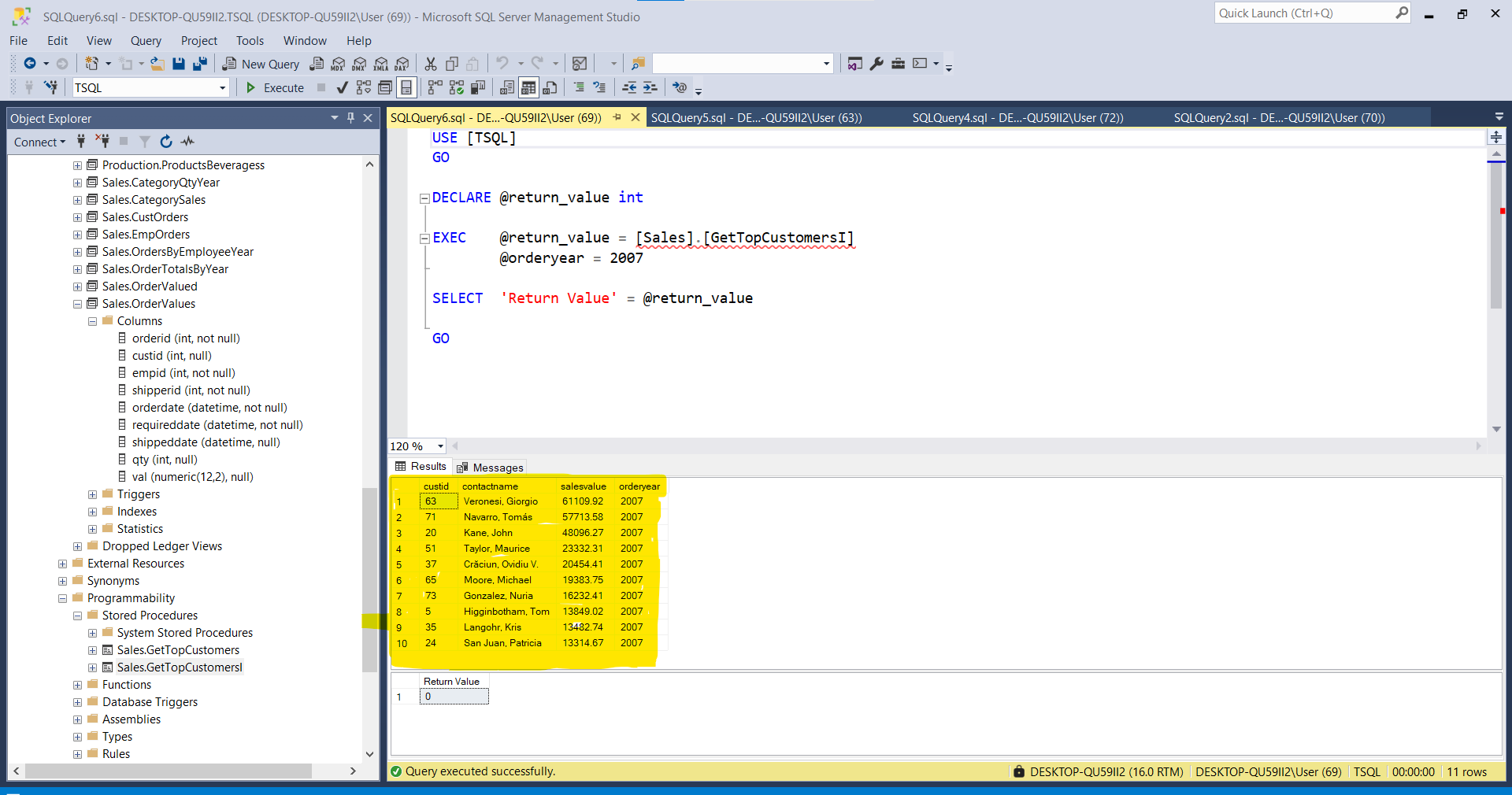
--1--



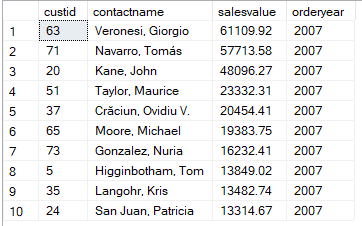
--2--



--3—After click the button ‘OK’



The Rows executed:



-- Write an EXECUTE statement to invoke the Sales.GetTopCustomers stored procedure for the year 2008.

execute Sales.GetTopCustomersI @orderyear = 2008

GO

-- Execute the T-SQL statement and compare the results that you got with the desired results shown in the file 63 - Lab Exercise 2 - Task 1\_2 Result.txt.

custid contactname salesvalue orderyear

-------------------------------------------------------------------------------

20 Kane, John 41210.65 2008

63 Veronesi, Giorgio 37217.32 2008

71 Navarro, Tomás 36310.11 2008

34 Cohen, Shy 23821.20 2008

65 Moore, Michael 21238.27 2008

37 Crăciun, Ovidiu V. 20402.12 2008

39 Song, Lolan 19582.78 2008

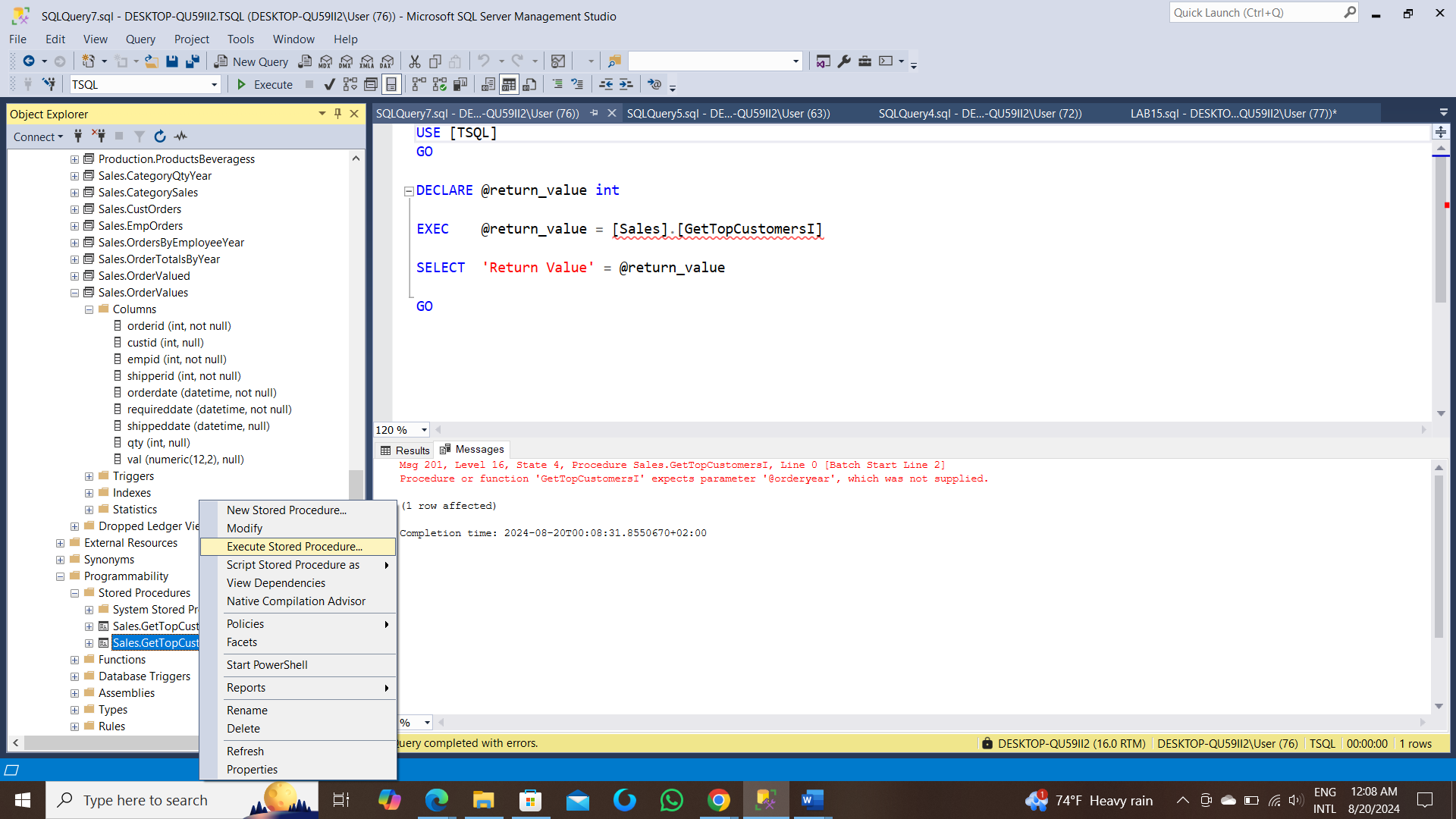
89 Smith Jr., Ronaldo 15278.90 2008

24 San Juan, Patricia 13644.07 2008

76 Gulbis, Katrin 11644.60 2008

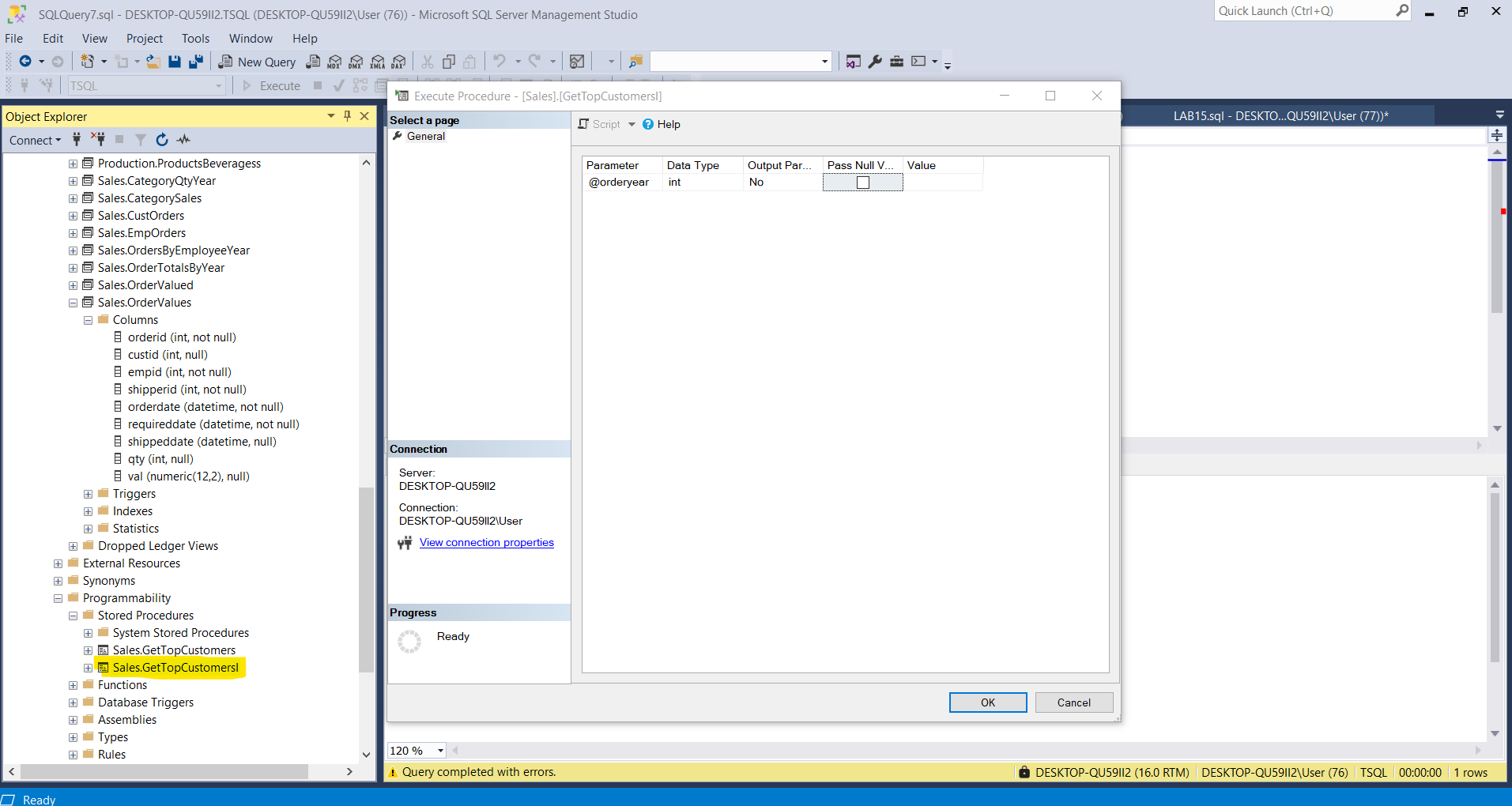
-- Write an EXECUTE statement to invoke the Sales.GetTopCustomers stored procedure without a parameter.

--Step1



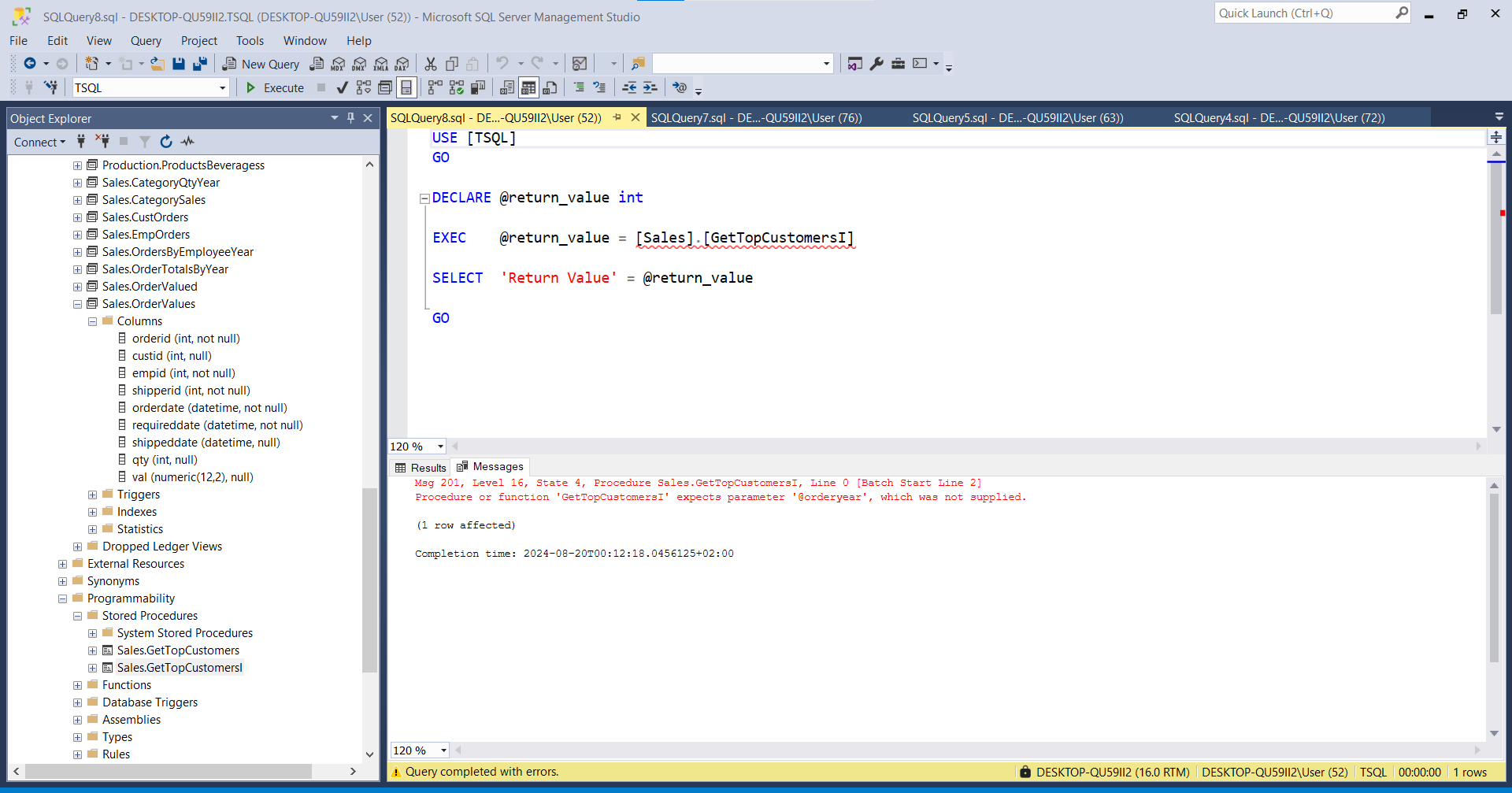
-- Execute the T-SQL statement. What happened? What is the error message?

--Step 2(manually)

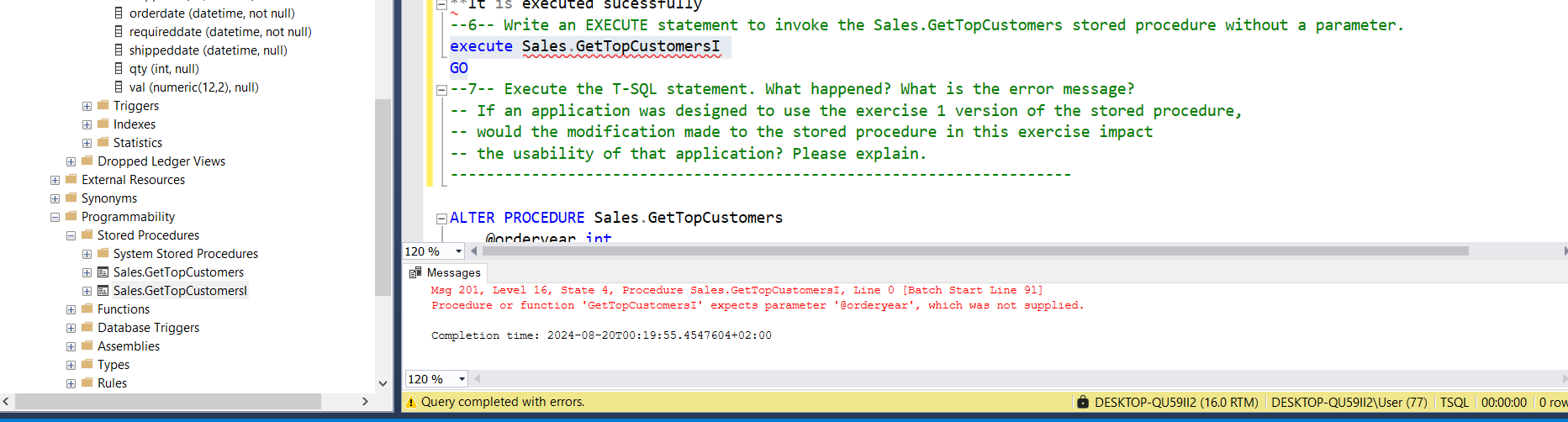


--Step 3

After clicked ‘OK’ Button:



By Writing a T-SQL Statement:



-- If an application was designed to use the exercise 1 version of the stored procedure, would the modification made to the stored procedure in this exercise impact the usability of that application? Please explain.

Yes,it will affect rows by the corresponding year

---------------------------------------------------------------------

---------------------------------------------------------------------

-- Task 2

--

-- Execute the provided T-SQL code to modify the Sales.GetTopCustomers stored procedure:

ALTER PROCEDURE Sales.GetTopCustomers

@orderyear int = NULL

AS

SELECT c.custid, c.contactname, SUM(o.val) AS salesvalue

FROM Sales.OrderValues AS o

INNER JOIN Sales.Customers AS c ON c.custid = o.custid

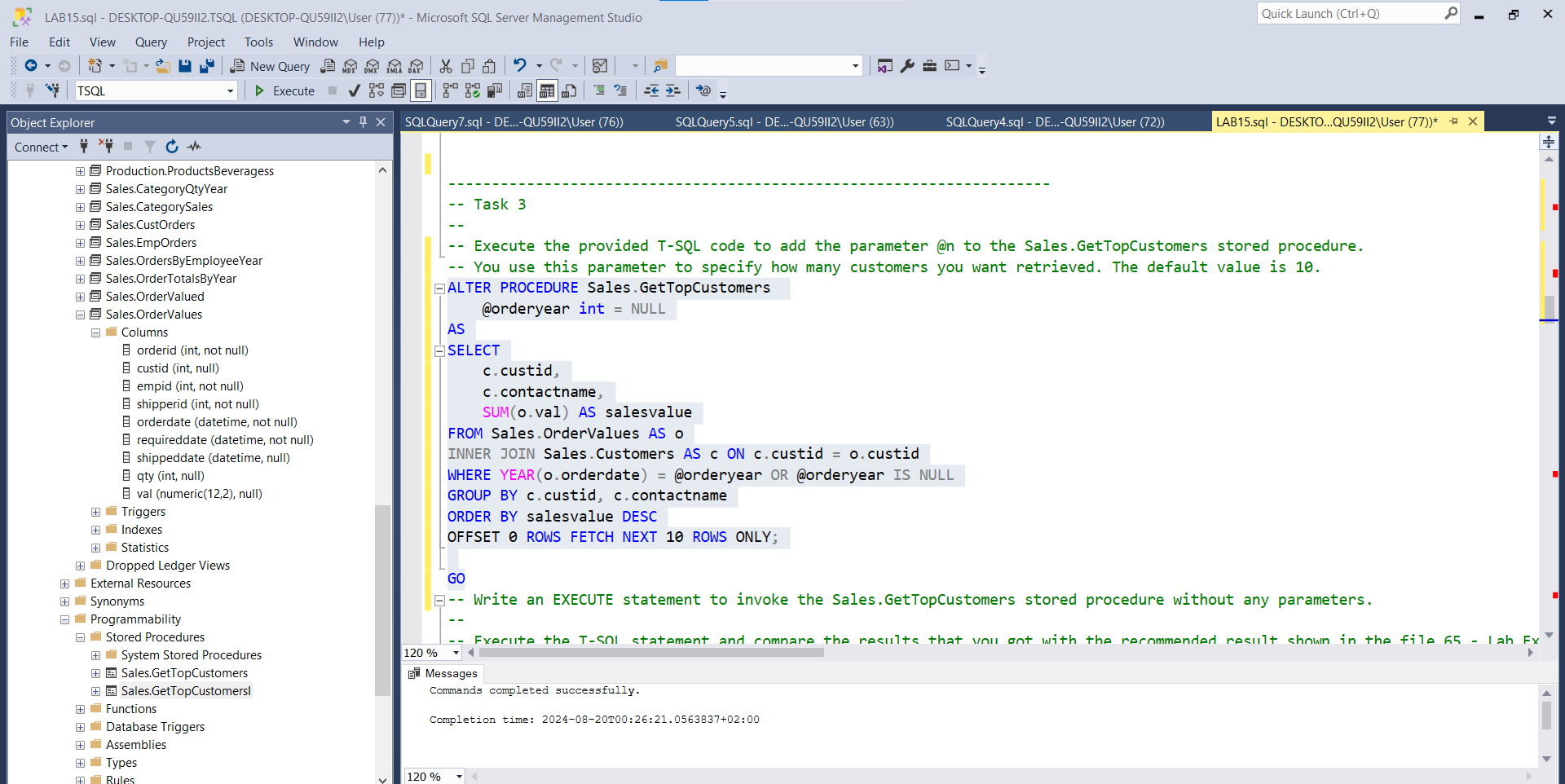
WHERE YEAR(o.orderdate) = @orderyear OR @orderyear IS NULL

GROUP BY c.custid, c.contactname

ORDER BY salesvalue DESC

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

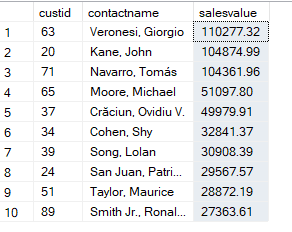
GO



-- Write an EXECUTE statement to invoke the Sales.GetTopCustomers stored procedure without a parameter.

execute Sales.GetTopCustomers

-- Execute the T-SQL statement and compare the results that you got with the recommended result shown in the file 64 - Lab Exercise 2 - Task 2 Result.txt.



-- If an application was designed to use the exercise 1 version of the stored procedure, would the change made to the stored procedure in this task impact the usability of that application? How does this change influence the design of future applications?

--------------------------------------------------------------------

---------------------------------------------------------------------

-- Task 3

--

-- Execute the provided T-SQL code to add the parameter @n to the Sales.GetTopCustomers stored procedure. You use this parameter to specify how many customers you want retrieved. The default value is 10.

ALTER PROCEDURE Sales.GetTopCustomers(@orderyear int = NULL,@n int = 10)

AS

SELECT

c.custid,

c.contactname,

SUM(o.val) AS salesvalue

FROM Sales.OrderValues AS o

INNER JOIN Sales.Customers AS c ON c.custid = o.custid

WHERE YEAR(o.orderdate) = @orderyear OR @orderyear IS NULL

GROUP BY c.custid, c.contactname

ORDER BY salesvalue DESC

OFFSET 0 ROWS FETCH NEXT @n ROWS ONLY

-- Write an EXECUTE statement to invoke the Sales.GetTopCustomers stored procedure without any parameters.

execute Sales.GetTopCustomers

-- Execute the T-SQL statement and compare the results that you got with the recommended result shown in the file 65 - Lab Exercise 2 - Task 3\_1 Result.txt.

custid contactname salesvalue

------------------------------------------------

63 Veronesi, Giorgio 110277.32

20 Kane, John 104874.99

71 Navarro, Tomás 104361.96

65 Moore, Michael 51097.80

37 Crăciun, Ovidiu V. 49979.91

34 Cohen, Shy 32841.37

39 Song, Lolan 30908.39

24 San Juan, Patricia 29567.57

51 Taylor, Maurice 28872.19

89 Smith Jr., Ronaldo 27363.61

-- Write an EXECUTE statement to invoke the Sales.GetTopCustomers stored procedure for order year 2008 and five customers.

ALTER PROCEDURE Sales.GetTopCustomersI(@orderyear int,@noCust int)

AS

SELECT c.custid,c.contactname,SUM(o.val) AS salesvalue,year(o.orderdate) AS orderyear

FROM Sales.OrderValues AS o

INNER JOIN Sales.Customers AS c ON c.custid = o.custid

Where year(o.orderdate) = @orderyear

GROUP BY c.custid, c.contactname,year(o.orderdate)

Having count(c.custid)=@noCust

ORDER BY salesvalue DESC

GO

EXECUTE Sales.GetTopCustomersI @orderyear = 2008 , @noCust = 5

go

--OSE

ALTER PROCEDURE Sales.GetTopCustomersI(@orderyear int = 2008,@noCust int=5)

AS

SELECT c.custid,c.contactname,SUM(o.val) AS salesvalue,year(o.orderdate) AS orderyear

FROM Sales.OrderValues AS o

INNER JOIN Sales.Customers AS c ON c.custid = o.custid

Where year(o.orderdate) = @orderyear

GROUP BY c.custid, c.contactname,year(o.orderdate)

Having count(c.custid)=@noCust

ORDER BY salesvalue DESC

GO

EXECUTE Sales.GetTopCustomersI

go

-- Execute the T-SQL statement and compare the results that you got with the recommended result shown in the file 66 - Lab Exercise 2 - Task 3\_2 Result.txt.

custid contactname salesvalue orderyear

-------------------------------------------------------

32 Krishnan, Venky 9942.14 2008

5 Higginbotham, Tom 6754.16 2008

62 Misiec, Anna 6373.83 2008

46 Dressler, Marlies 5507.32 2008

66 Voss, Florian 3967.30 2008

-- Write an EXECUTE statement to invoke the Sales.GetTopCustomers stored procedure for the order year 2007.

ALTER PROCEDURE [Sales].[GetTopCustomers](@orderyear int = 2007,@n int = 5)

AS

SELECT

c.custid,

c.contactname,

SUM(o.val) AS salesvalue

FROM Sales.OrderValues AS o

INNER JOIN Sales.Customers AS c ON c.custid = o.custid

WHERE YEAR(o.orderdate) = @orderyear OR @orderyear IS NULL

GROUP BY c.custid, c.contactname

ORDER BY salesvalue DESC

OFFSET 0 ROWS FETCH NEXT @n ROWS ONLY

GO

-- Execute the T-SQL statement and compare the results that you got with the recommended result shown in the file 67 - Lab Exercise 2 - Task 3\_3 Result.txt.

EXECUTE [Sales].[GetTopCustomers]

custid contactname salesvalue

--------------------------------------------------

63 Veronesi, Giorgio 61109.92

71 Navarro, Tomás 57713.58

20 Kane, John 48096.27

51 Taylor, Maurice 23332.31

37 Crăciun, Ovidiu V. 20454.41

-- Write an EXECUTE statement to invoke the Sales.GetTopCustomers stored procedure to retrieve 20 customers.

ALTER PROCEDURE [Sales].[GetTopCustomers](@orderyear int = 2007,@n int = 20)

AS

SELECT

c.custid,

c.contactname,

SUM(o.val) AS salesvalue

FROM Sales.OrderValues AS o

INNER JOIN Sales.Customers AS c ON c.custid = o.custid

WHERE YEAR(o.orderdate) = @orderyear OR @orderyear IS NULL

GROUP BY c.custid, c.contactname

ORDER BY salesvalue DESC

OFFSET 0 ROWS FETCH NEXT @n ROWS ONLY

GO

-- Execute the T-SQL statement and compare the results that you got with the recommended result shown in the file 68 - Lab Exercise 2 - Task 3\_4 Result.txt.

EXECUTE [Sales].[GetTopCustomers]



-- Do the applications using the stored procedure need to be changed because another parameter was added? Yes ,I altered or updated the procedure.

---------------------------------------------------------------------

---------------------------------------------------------------------

-- Task 4

--

-- Execute the provided T-SQL code to modify the Sales.GetTopCustomers stored procedure to return the customer contact name based on a specified position in a ranking of total sales, which is provided by the parameter @customerpos. The procedure also includes a new parameter named @customername, which has an output option.

ALTER PROCEDURE Sales.GetTopCustomers

(@customerpos int = 1 ,

@customername nvarchar(30) OUTPUT)

AS

SET @customername = (

SELECT

c.contactname

FROM Sales.OrderValues AS o

INNER JOIN Sales.Customers AS c ON c.custid = o.custid

--Where c.contactname = @customername

GROUP BY c.custid, c.contactname

ORDER BY SUM(o.val) DESC

OFFSET @customerpos - 1 ROWS FETCH NEXT 1 ROW ONLY

);

GO

-- The IT department also supplied you with T-SQL code to declare the new variable @outcustomername. You will use this variable as an output parameter for the stored procedure.

-- DECLARE @outcustomername nvarchar(30);

Declare @noRow INT = 10 , @outcustomername nvarchar(30) ;

execute Sales.GetTopCustomers @customerpos = @noRow ,@customername = @outcustomername OUTPUT;

Select @noRow as customerpos , @outcustomername as contactname;

go

-- Write an EXECUTE statement to invoke the Sales.GetTopCustomers stored procedure and retrieve the first customer.

Declare @firstRow INT = 1 ,@outcustomername nvarchar(30);

EXEC Sales.GetTopCustomers @customerpos = @firstRow , @customername = @outcustomername OUTPUT;

Select @firstRow as customerpos ,@outcustomername as contactname;

GO

-- Write a SELECT statement to retrieve the value of the output parameter @outcustomername.

Declare @firstRow INT = 1 ,@outcustomername nvarchar(30);

EXEC Sales.GetTopCustomers @customerpos = @firstRow , @customername = @outcustomername OUTPUT;

Select @outcustomername as contactname;

GO

-- Execute the batch of T-SQL code consisting of the provided DECLARE statement, the written EXECUTE statement, and the written SELECT statement.

----

customerpos | contactname

10 Smith Jr., Ronaldo

------------------------------------

-------------------------------------

customerpos| contactname

1 Veronesi, Giorgio

----------------------------------------------

----------------------------------------------

|contactname

---------------------

1| Veronesi, Giorgio

-- Observe and compare the results that you got with the recommended result shown in the file 69 - Lab Exercise 2 - Task 4 Result.txt.

---------------------------------------------------------------------

---------------------------------------------------------------------

-- LAB 15

--

-- Exercise 3

---------------------------------------------------------------------

USE TSQL;

GO

---------------------------------------------------------------------

-- Task 1

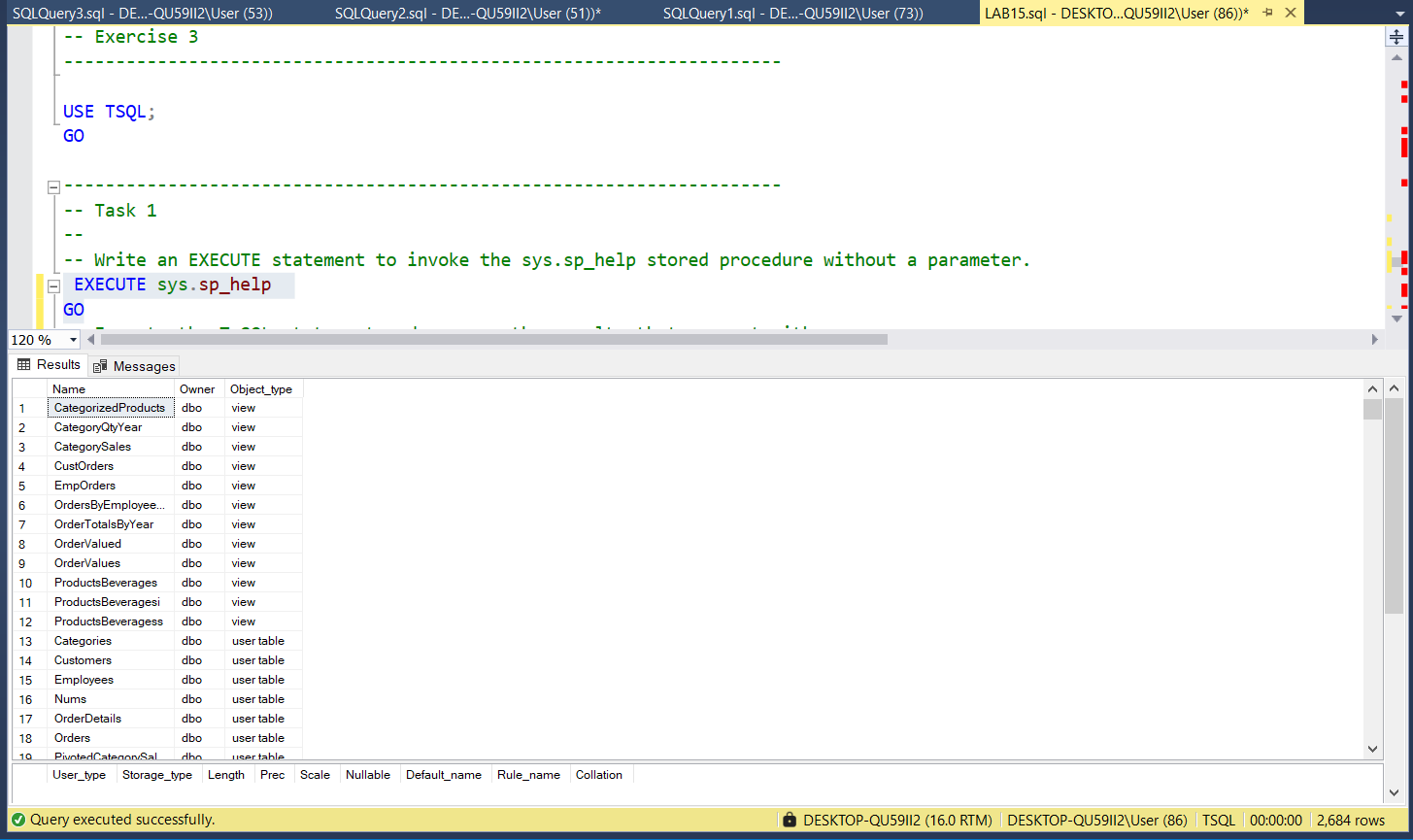
--

-- Write an EXECUTE statement to invoke the sys.sp\_help stored procedure without a parameter.

EXECUTE sys.sp\_help

GO

-- Execute the T-SQL statement and compare the results that you got with the recommended result shown in the file 72 - Lab Exercise 3 - Task 1\_1 Result.txt.



-- Write an EXECUTE statement to invoke the sys.sp\_help stored procedure for a specific table by passing the parameter Sales.Customers.

DECLARE @sqlcode as nvarchar(1000) = N'Select \* From Sales.Customers'

Execute sys.sp\_help @objname = @sqlcode

GO

-- Execute the T-SQL statement and compare the results that you got with the recommended result shown in the file 73 - Lab Exercise 3 - Task 1\_2 Result.txt.

---------------------------------------------------------------------

---------------------------------------------------------------------

-- Task 2

--

-- Write an EXECUTE statement to invoke the sys.sp\_helptext stored procedure, passing the Sales.GetTopCustomers stored procedure as a parameter.

DECLARE @sqlcode as nvarchar(1000) ;

SET @sqlcode = N'Select \* From Sales.Customers'

Execute sys.sp\_helptext @objname = @sqlcode

GO

-- Execute the T-SQL statement and compare the results that you got with the recommended result shown in the file 74 - Lab Exercise 3 - Task 2 Result.txt.

---------------------------------------------------------------------

---------------------------------------------------------------------

-- Task 3

--

-- Write an EXECUTE statement to invoke the sys.sp\_columns stored procedure for the table Sales.Customers. You will have to pass two parameters: @table\_name and @table\_owner.

--

-- Execute the T-SQL statement and compare the results that you got with the recommended result shown in the file 75 - Lab Exercise 3 - Task 3 Result.txt.

---------------------------------------------------------------------

---------------------------------------------------------------------

-- Task 4

-- Execute the provided T-SQL statement to remove the Sales.GetTopCustomers stored procedure.

---------------------------------------------------------------------

DROP PROCEDURE Sales.GetTopCustomers;