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

-- LAB 13

--

-- Exercise 1

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

USE TSQL;

GO

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

-- Task 1

--

-- Write a SELECT statement to retrieve the orderid, orderdate, and val columns as well as a calculated column named rowno from

-- the view Sales.OrderValues. Use the ROW\_NUMBER function to return rowno. Order the row numbers by the orderdate column.

-- Execute the written statement and compare the results that you got with the desired results

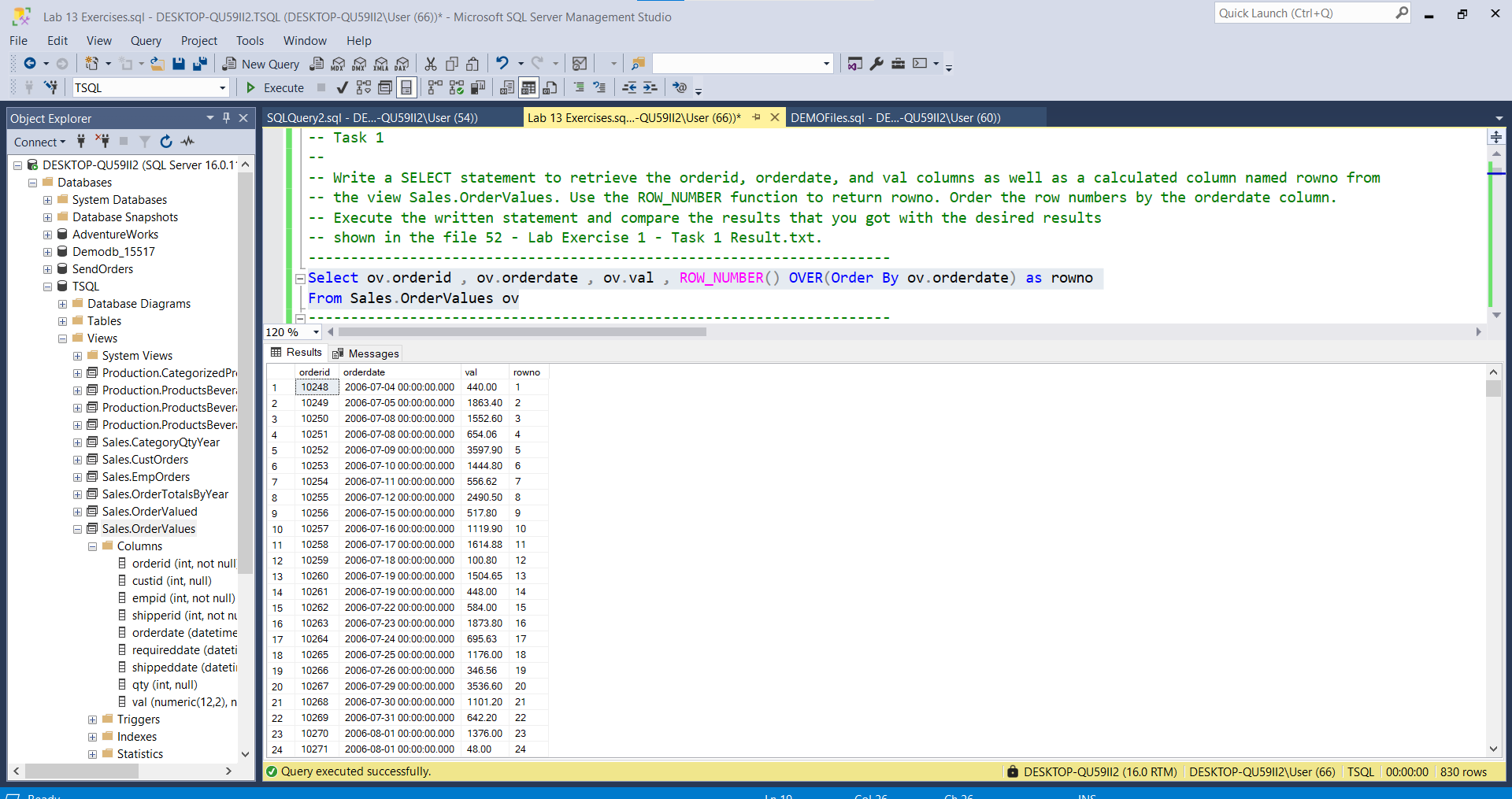
-- shown in the file 52 - Lab Exercise 1 - Task 1 Result.txt.

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

Select ov.orderid , ov.orderdate , ov.val , ROW\_NUMBER() OVER(Order By ov.orderdate) as rowno

From Sales.OrderValues ov

--Ekzekutimi si më poshtë:



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

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

-- Task 2

-- Copy the previous T-SQL statement and modify it by including an additional column named rankno.

-- To create rankno, use the RANK function, with the rank order based on the orderdate column.

-- Execute the modified statement and compare the results that you got with

-- the desired results shown in the file 53 - Lab Exercise 1 - Task 2 Result.txt.

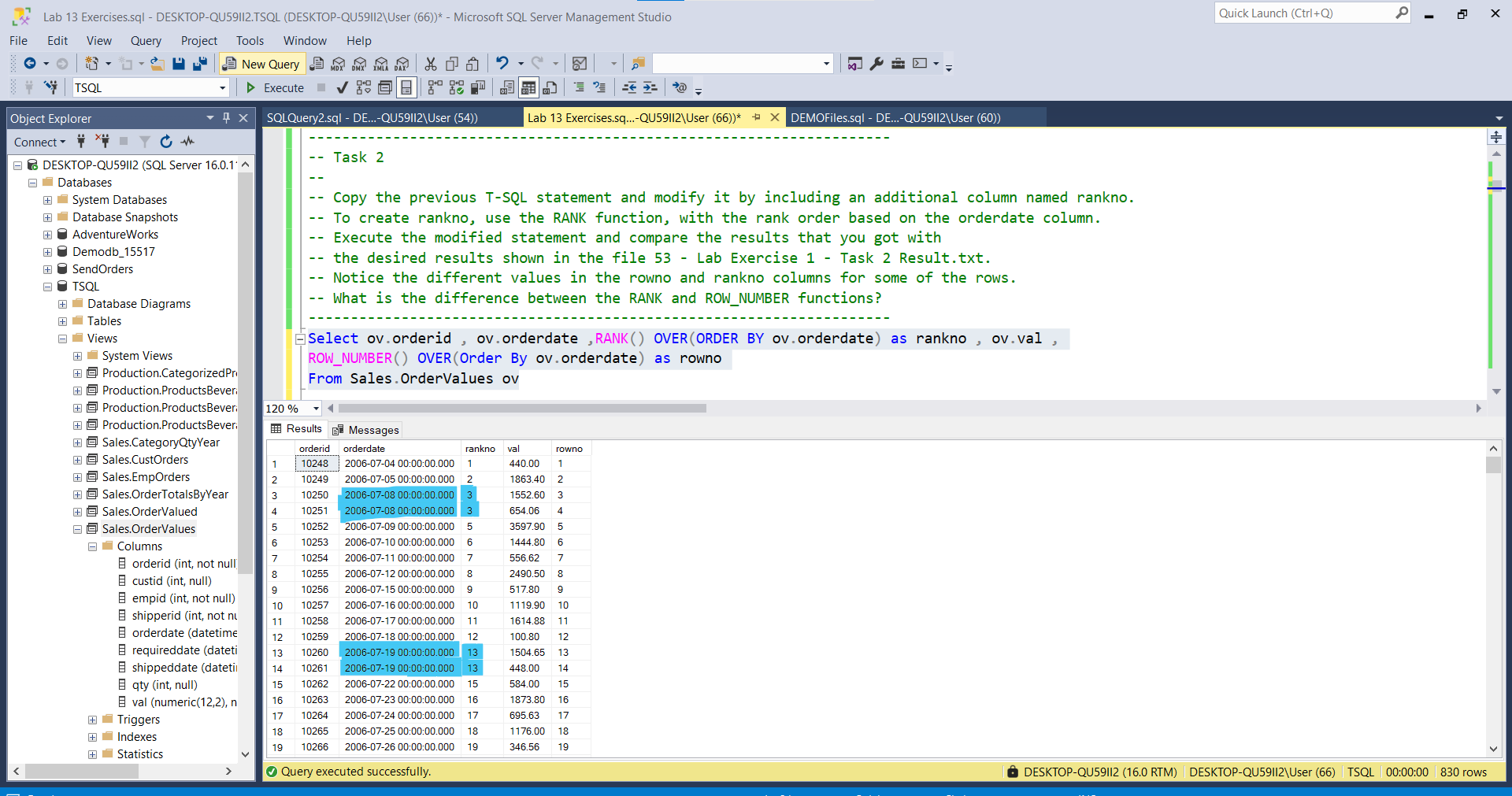
-- Notice the different values in the rowno and rankno columns for some of the rows.

-- What is the difference between the RANK and ROW\_NUMBER functions?

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

Select ov.orderid , ov.orderdate ,RANK() OVER(ORDER BY ov.orderdate) as rankno , ov.val , ROW\_NUMBER() OVER(Order By ov.orderdate) as rowno

From Sales.OrderValues ov



--ROW\_NUMBER()🡪Gjeneron një numër unik në mënyrë sekuencuale për çdo rresht , zakonisht

default\_value është 1

Ndërsa RANK() 🡪Shënjon vlerat per cdo rresht po ashtu ne menyre sekuenciale… por kur në renditjen (sipas ‘dates’ ne rastin ne fjale ) e vlerave per cdo rresht hasen dy vlera te njejta njera pas tjetes atehere ‘rankno ’ shenjon te njejten vlere edhe ne rreshtin pasardhës

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

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

-- Task 3

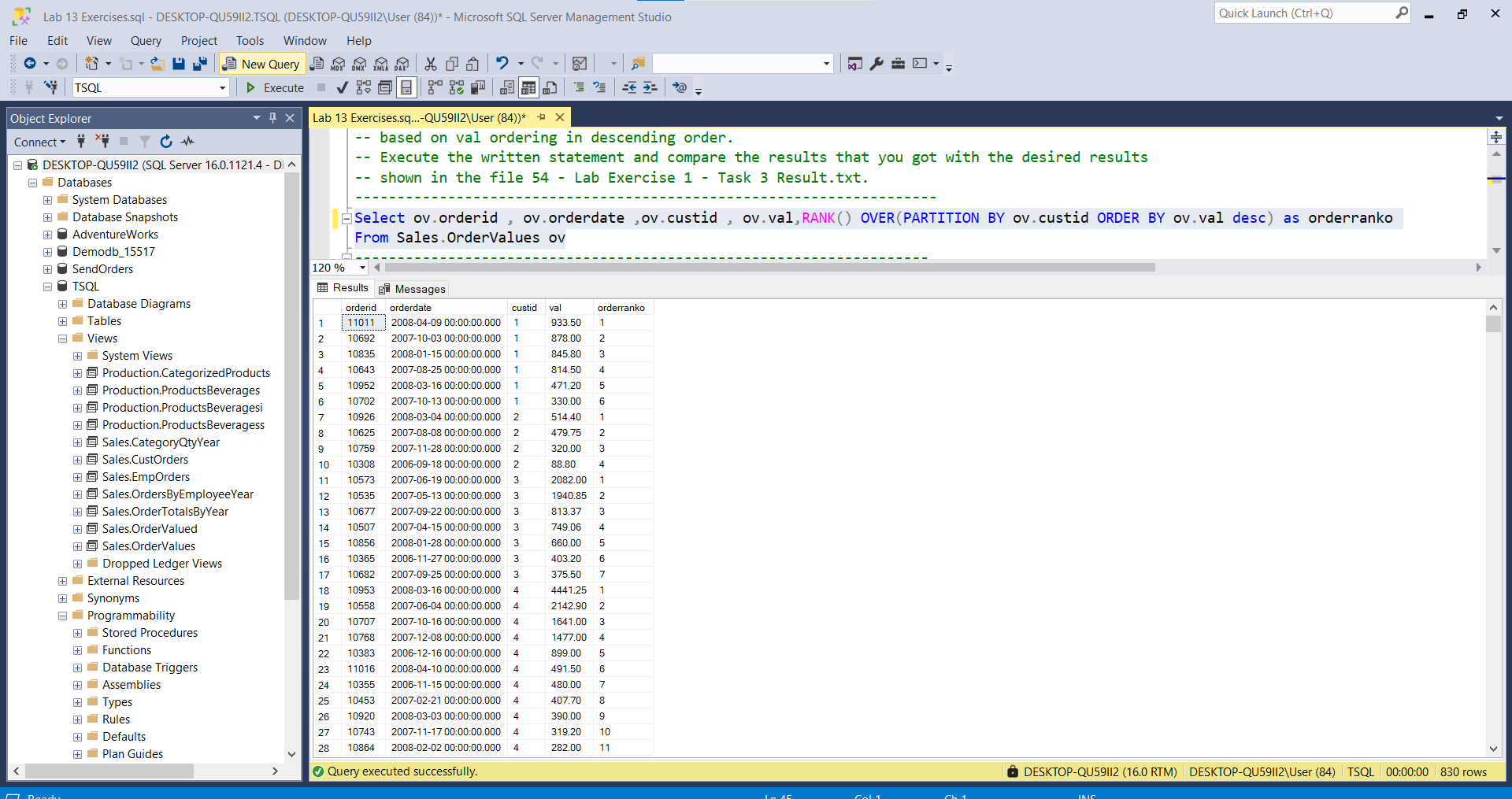
--

-- Write a SELECT statement to retrieve the orderid, orderdate, custid, and val columns as well as a calculated column named orderrankno from the Sales.OrderValues view. The orderrankno column should display the rank per each customer independently, based on val ordering in descending order.

Select ov.orderid , ov.orderdate ,ov.custid , ov.val,RANK() OVER(PARTITION BY ov.custid ORDER BY ov.val desc) as orderranko

From Sales.OrderValues ov

-- Execute the written statement and compare the results that you got with the desired results shown in the file 54 - Lab Exercise 1 - Task 3 Result.txt.



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

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

-- Task 4

--

-- Write a SELECT statement to retrieve the custid and val columns from the Sales.OrderValues view. Add two calculated columns:

-- orderyear as a year of the orderdate column

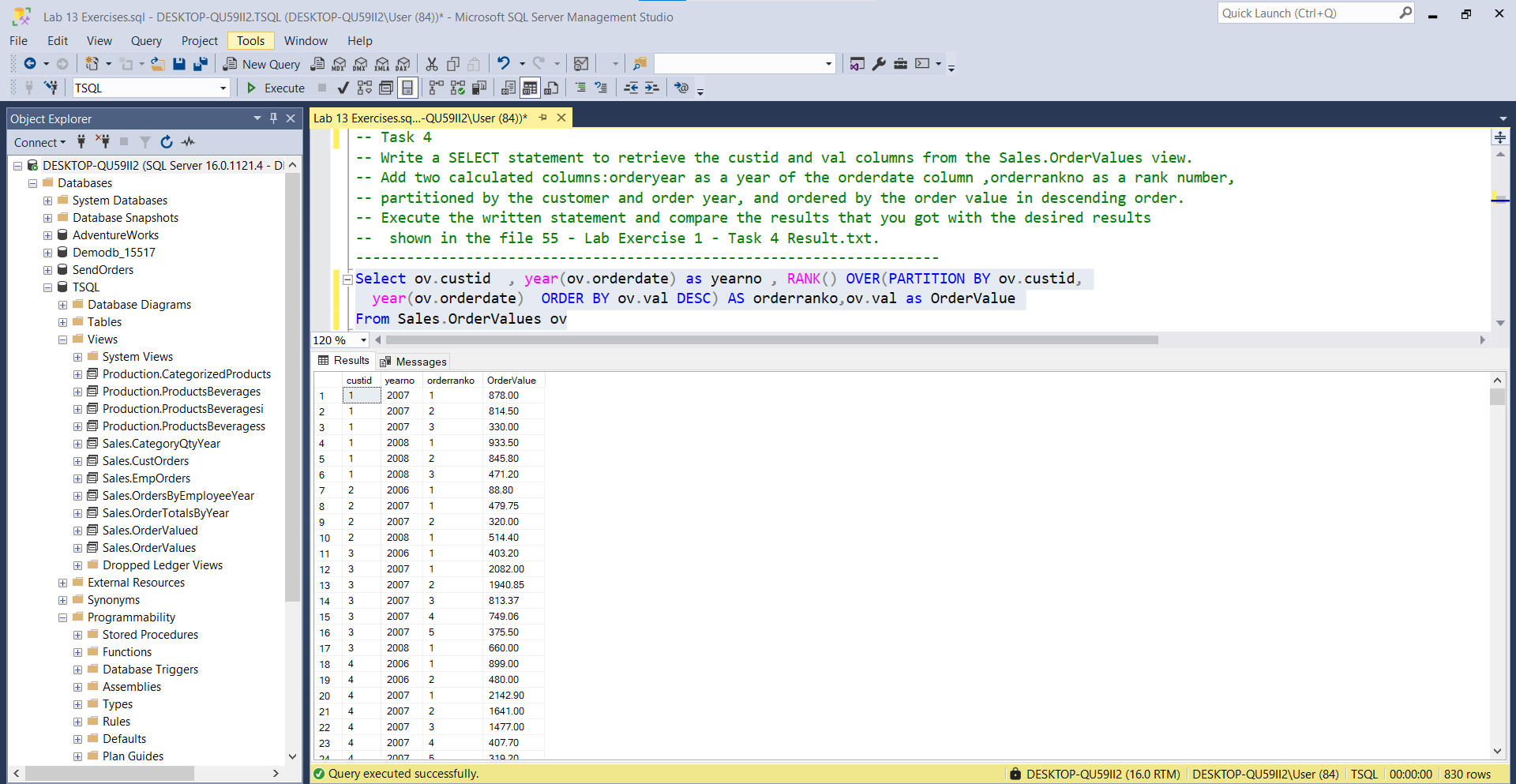
-- orderrankno as a rank number, partitioned by the customer and order year, and ordered by the order value in descending order.

Select ov.custid , year(ov.orderdate) as yearno , RANK() OVER(PARTITION BY ov.custid, year(ov.orderdate) ORDER BY ov.val DESC)

AS orderranko,ov.val as OrderValue

From Sales.OrderValues ov

-- Execute the written statement and compare the results that you got with the desired results shown in the file 55 - Lab Exercise 1 - Task 4 Result.txt.



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

-- Task 5

--

-- Copy the previous query and modify it to filter only orders with the first two ranks based on the orderrankno column.

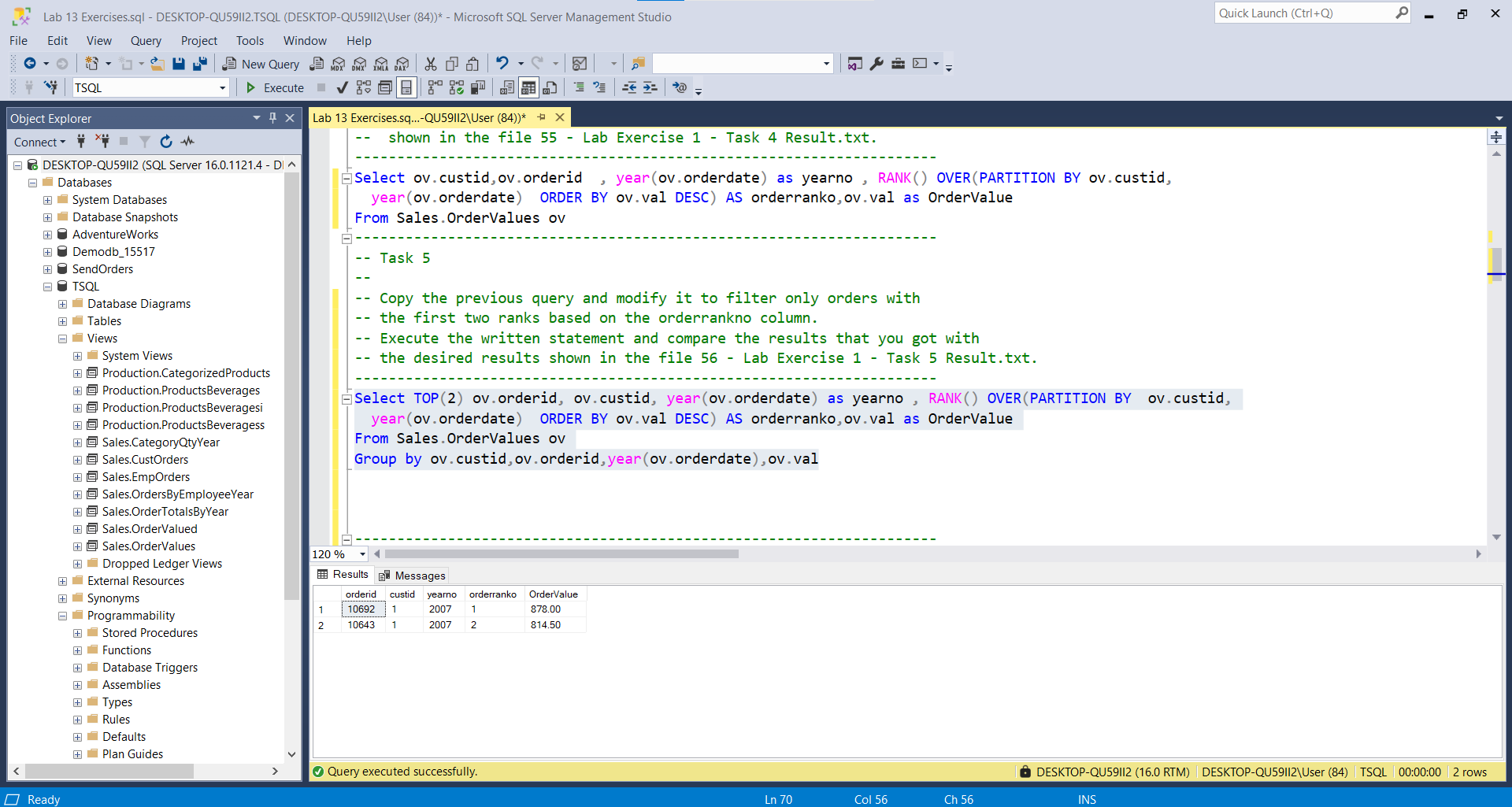
Select TOP(2) ov.orderid, ov.custid, year(ov.orderdate) as yearno , RANK() OVER(PARTITION BY ov.custid,

year(ov.orderdate) ORDER BY ov.val DESC) AS orderranko,ov.val as OrderValue

From Sales.OrderValues ov

Group by ov.custid,ov.orderid,year(ov.orderdate),ov.val

-- Execute the written statement and compare the results that you got with the desired results shown in the file 56 - Lab Exercise 1 - Task 5 Result.txt.

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

-- LAB 13

--

-- Exercise 2

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

USE TSQL;

GO

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

-- Task 1

--

-- Define a CTE named OrderRows based on a query that retrieves the orderid, orderdate, and val columns from the Sales.OrderValues view. Add a calculated column named rowno using the ROW\_NUMBER function, ordering by the orderdate and orderid columns.

--

-- Write a SELECT statement against the CTE and use the LEFT JOIN with the same CTE to retrieve the current row and the previous row based on the rowno column. Return the orderid, orderdate, and val columns for the current row and the val column from the previous row as prevval. Add a calculated column named diffprev to show the difference between the current val and previous val.

WITH CTE\_OrderRows

as

(SELECT o.orderid,o.orderdate,o.val,ROW\_NUMBER() OVER(ORDER BY o.orderid,o.orderdate) AS rowno

FROM Sales.OrderValues as o

), CTE\_PREVIOUSOrderRows

as(

SELECT o.orderid,o.orderdate,o.val,LAG(o.val,1,0) OVER(ORDER BY o.orderid,o.orderdate) AS preVal

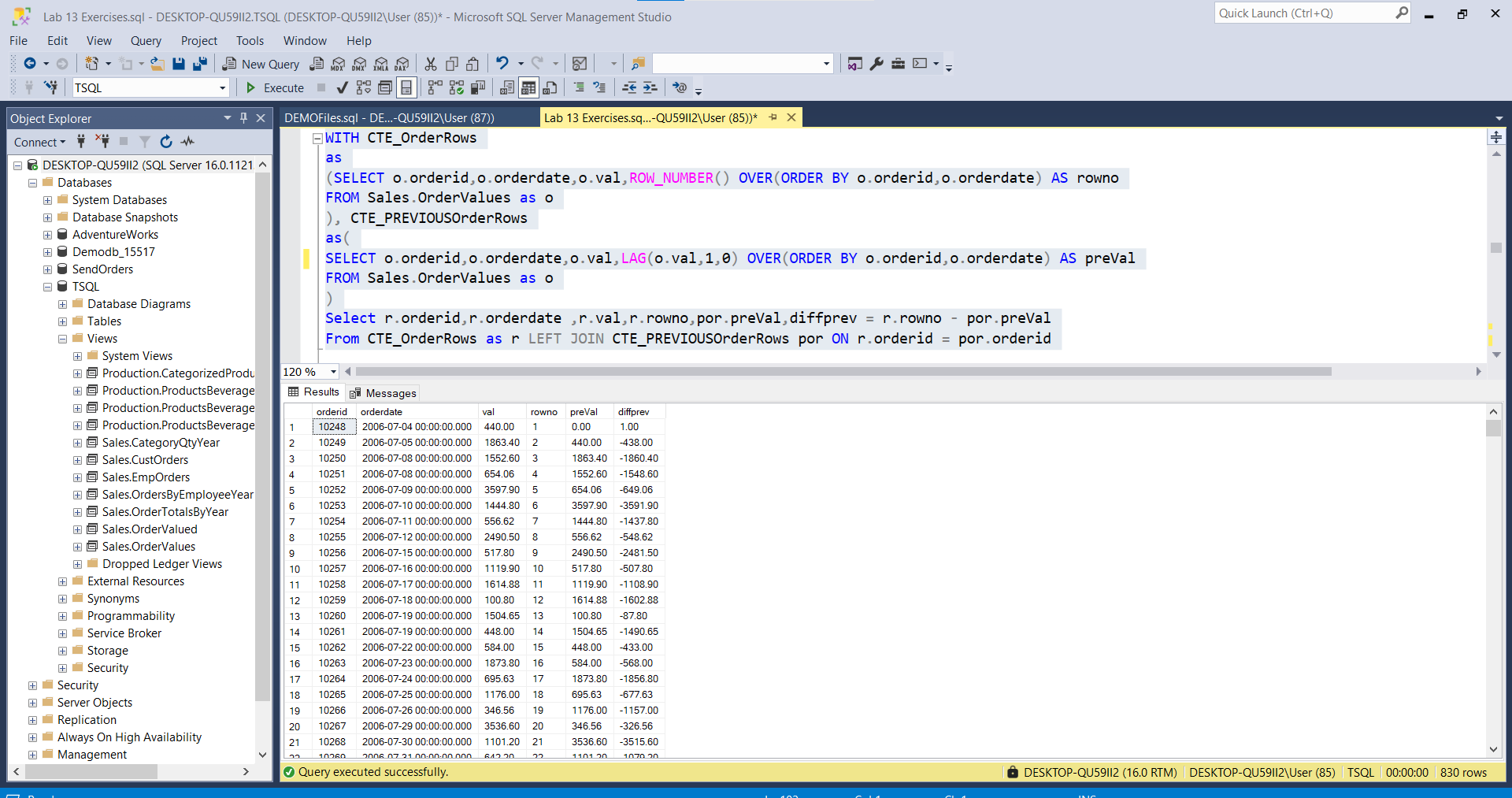
FROM Sales.OrderValues as o

)

Select r.orderid,r.orderdate ,r.val,r.rowno,por.preVal,diffprev = r.rowno - por.preVal

From CTE\_OrderRows as r LEFT JOIN CTE\_PREVIOUSOrderRows por ON r.orderid = por.orderid

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



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

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

-- Task 2

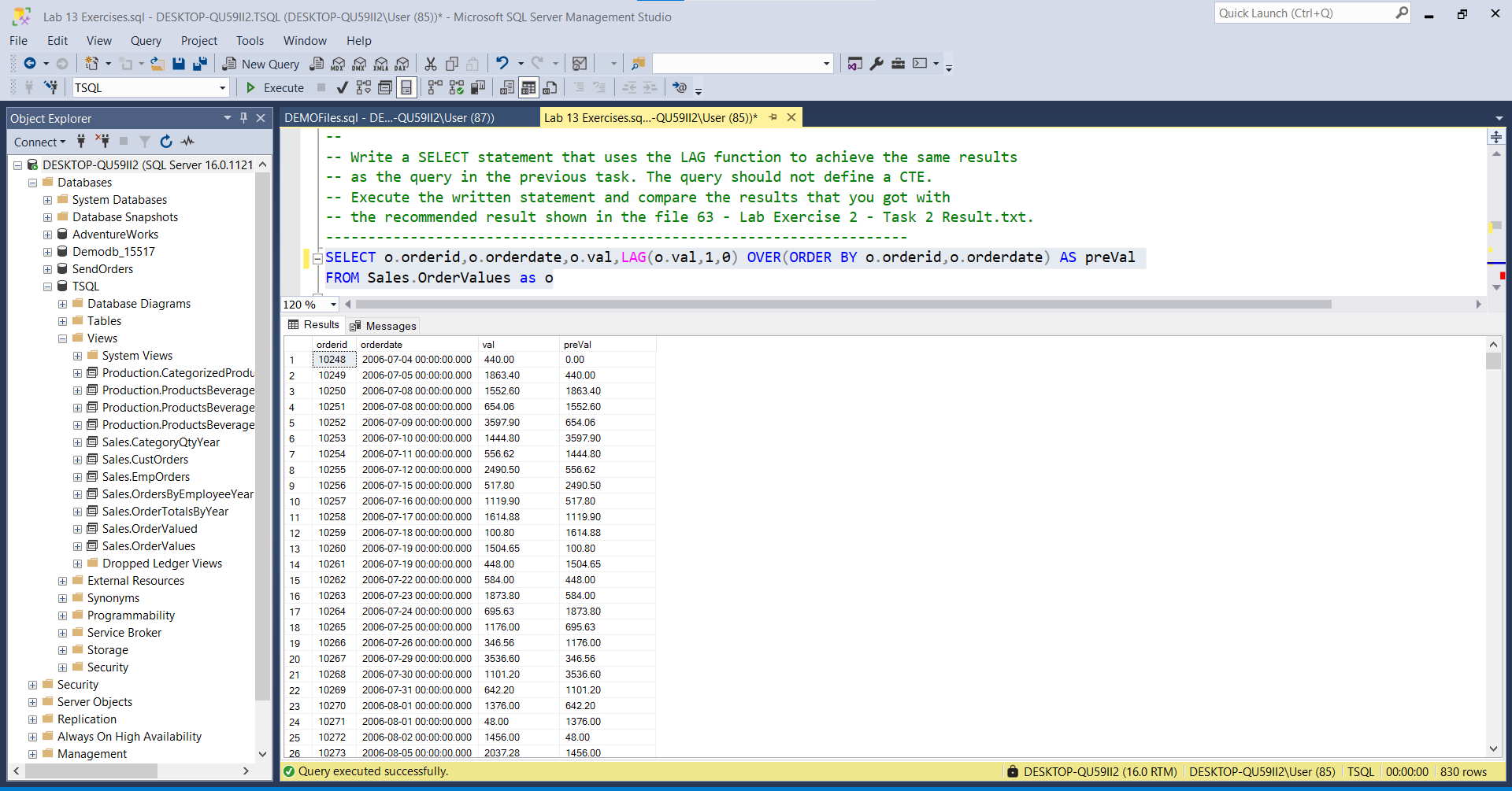
--

-- Write a SELECT statement that uses the LAG function to achieve the same results as the query in the previous task. The query should not define a CTE.

SELECT o.orderid,o.orderdate,o.val,LAG(o.val,1,0) OVER(ORDER BY o.orderid,o.orderdate) AS preVal

FROM Sales.OrderValues as o

-- Execute the written statement and compare the results that you got with the recommended result shown in the file 63 - Lab Exercise 2 - Task 2 Result.txt.



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

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

-- Task 3

--

--1-- Define a CTE named SalesMonth2007 that creates two columns: monthno (the month number of the orderdate column) and val (aggregated val column). Filter the results to include only the order year 2007 and group by monthno.

--2-- Write a SELECT statement that retrieves the monthno and val columns from the CTE and adds three calculated columns:

-- avglast3months. This column should contain the average sales amount for last three months before the current month. (Use multiple LAG functions and divide the sum by three.) You can assume that there’s a row for each month in the CTE.

-- diffjanuary. This column should contain the difference between the current val and the January val. (Use the FIRST\_VALUE function.)

-- nextval. This column should contain the next month value of the val column.

--1--

WITH CTE\_SalesMonth2007

AS

(Select v.custid,v.orderid as OrderId,month(v.orderdate) as monthno ,v.val as SalesPerMonth

From Sales.OrderValues v

Where year(v.orderdate) = 2007

Group by v.custid,v.orderid,month(v.orderdate),v.val

)

Select \* From CTE\_SalesMonth2007

go

--2--

WITH CTE\_AverageSales2007

AS

( Select TOP(12) m.monthForAverage ,m.CurrentAverageSales,m.RunningAVGValues , LAG(m.RunningAVGValues,1,0) OVER(Order By m.RunningAVGValues) AS avglast3months,

m.januaryValue,m.diffJanuary, LEAD(m.RunningAVGValues,1,0) OVER(Order By m.RunningAVGValues) AS nextVal

FROM(

Select month(o.orderdate) as monthForAverage , AVG(o.val) CurrentAverageSales , SUM(AVG(o.val)) OVER( Order By month(o.orderdate)) RunningAVGValues,

FIRST\_VALUE(AVG(o.val)) OVER (ORDER BY month(o.orderdate)) as januaryValue, diffJanuary = AVG(o.val) -FIRST\_VALUE(AVG(o.val)) OVER (ORDER BY month(o.orderdate))

From Sales.OrderValues o

Group by month(o.orderdate)

) as m

Group By m.monthForAverage,m.CurrentAverageSales,m.RunningAVGValues,m.januaryValue,diffJanuary

ORDER BY m.monthForAverage

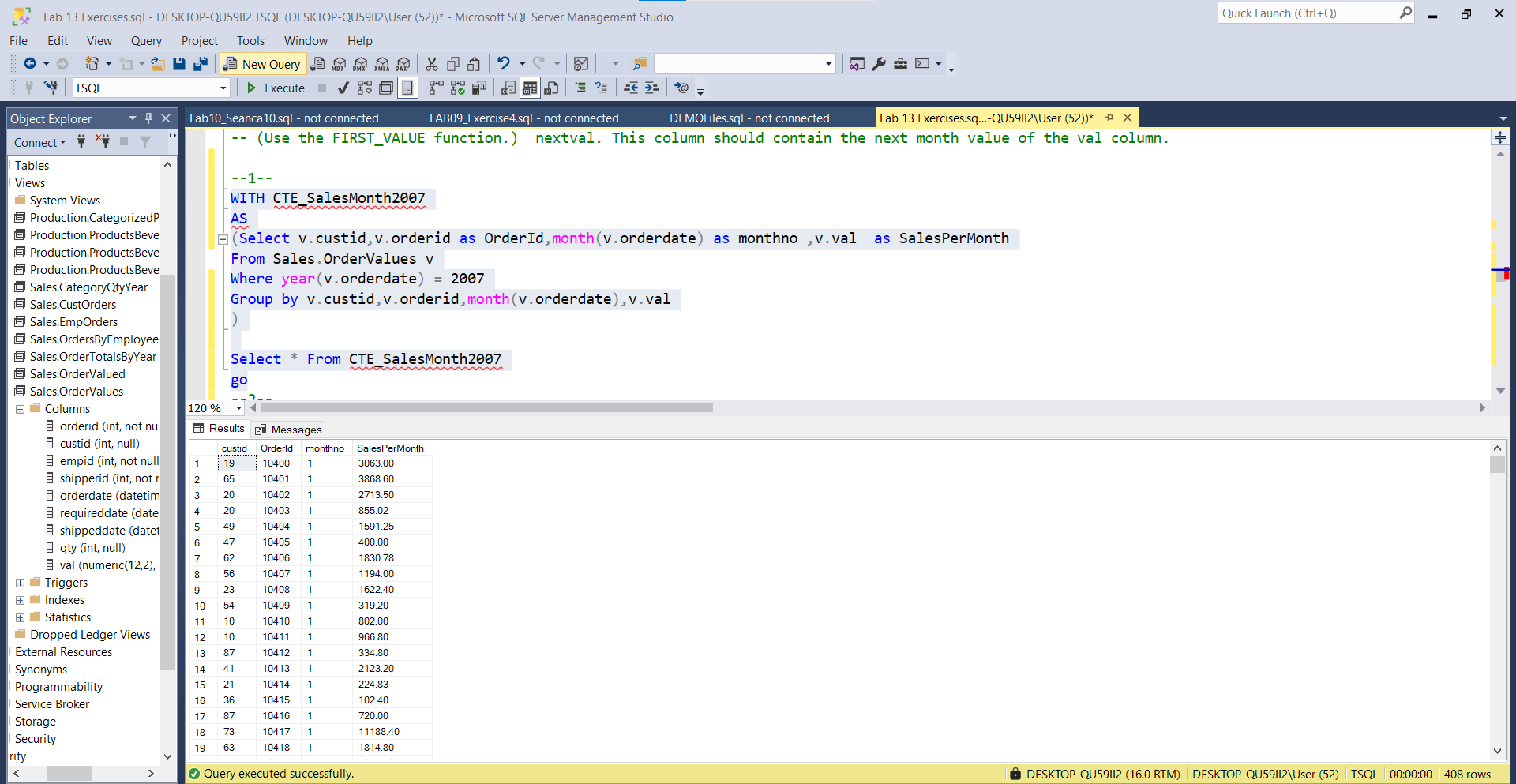
)

Select \*

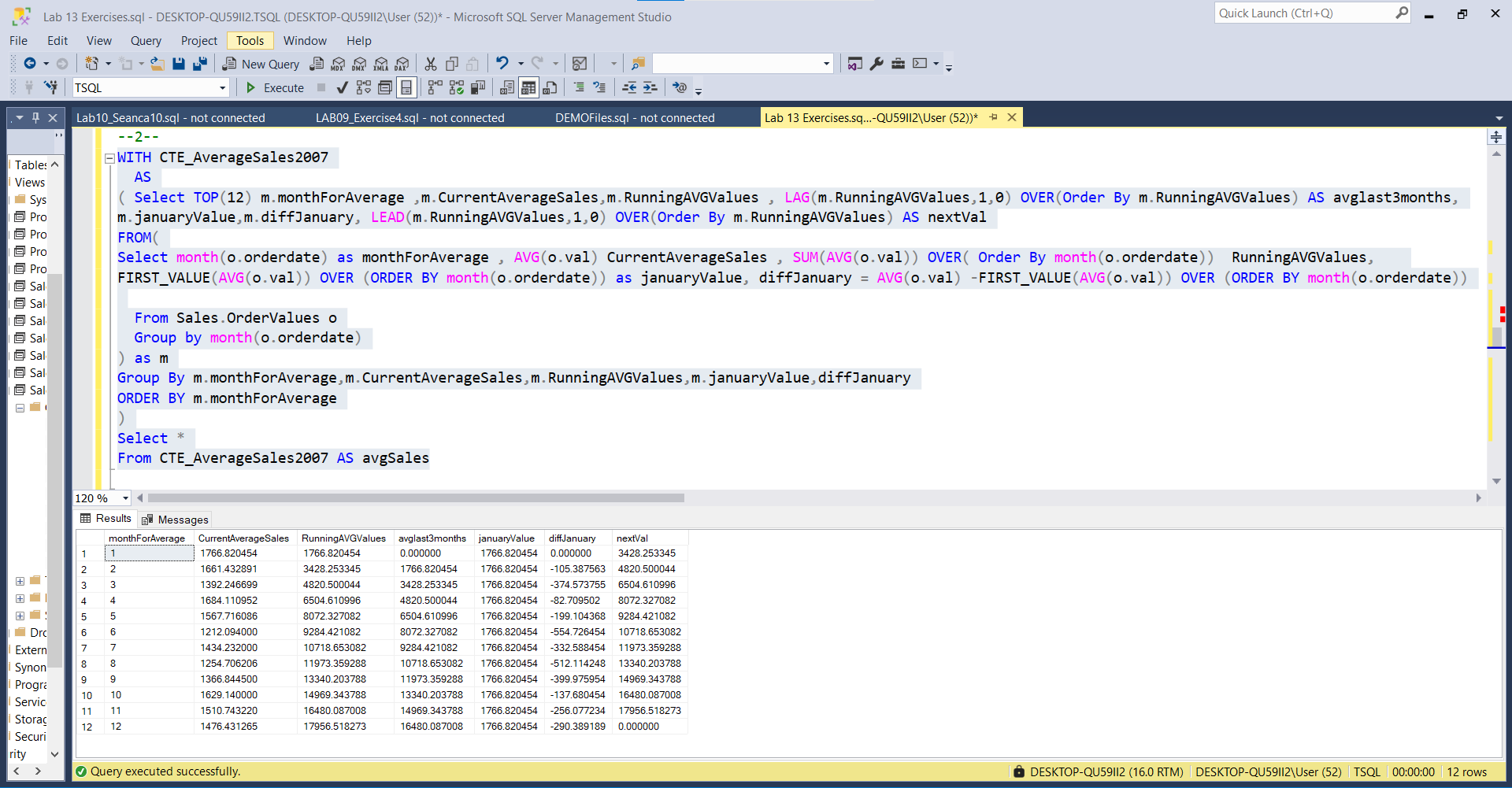
From CTE\_AverageSales2007 AS avgSales

Go Execute the written statement and compare the results that you got with the recommended result shown in the file 64 - Lab Exercise 2 - Task 3 Result.txt. Notice that the average amount for last three months is not correctly computed because the total amount for the first two months is divided by three. You will practice how to do this correctly in the next exercise.

--1--



--2--



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

-- LAB 13

--

-- Exercise 3

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

USE TSQL;

GO

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

-- Task 1

--

-- Write a SELECT statement to retrieve the custid, orderid, orderdate, and val columns from the Sales.OrderValues view. Add a calculated column named percoftotalcust that contains a percentage value of each order sales amount compared to the total sales amount for that customer.

--1--

Select os.custid , os.orderid , os.orderdate , os.val

From Sales.OrderValues os

Group By os.custid , os.orderid , os.orderdate , os.val

Order By os.custid

--2--

Select os.custid , os.orderid , os.orderdate , os.val ,os.percoftotalcust

From (

Select o.custid,o.orderid , o.orderdate , o.val ,

percoftotalcust = Concat(cast((o.val/(SUM(o.val) OVER(PARTITION BY o.custid Order By o.val))\*100) as numeric(8,2)),'%')

From Sales.OrderValues o

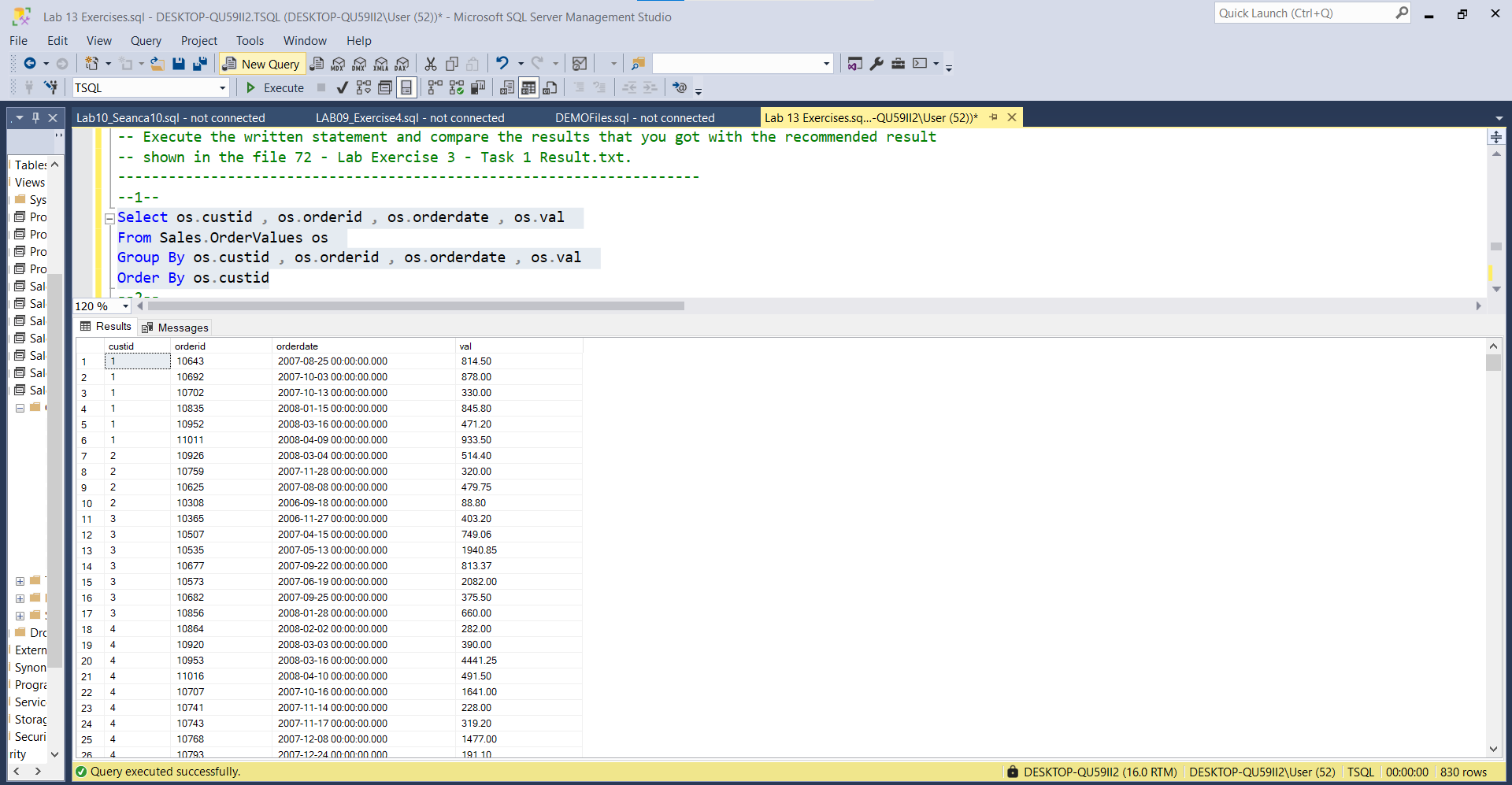
) as os

Group By os.custid , os.orderid , os.orderdate , os.val,os.percoftotalcust

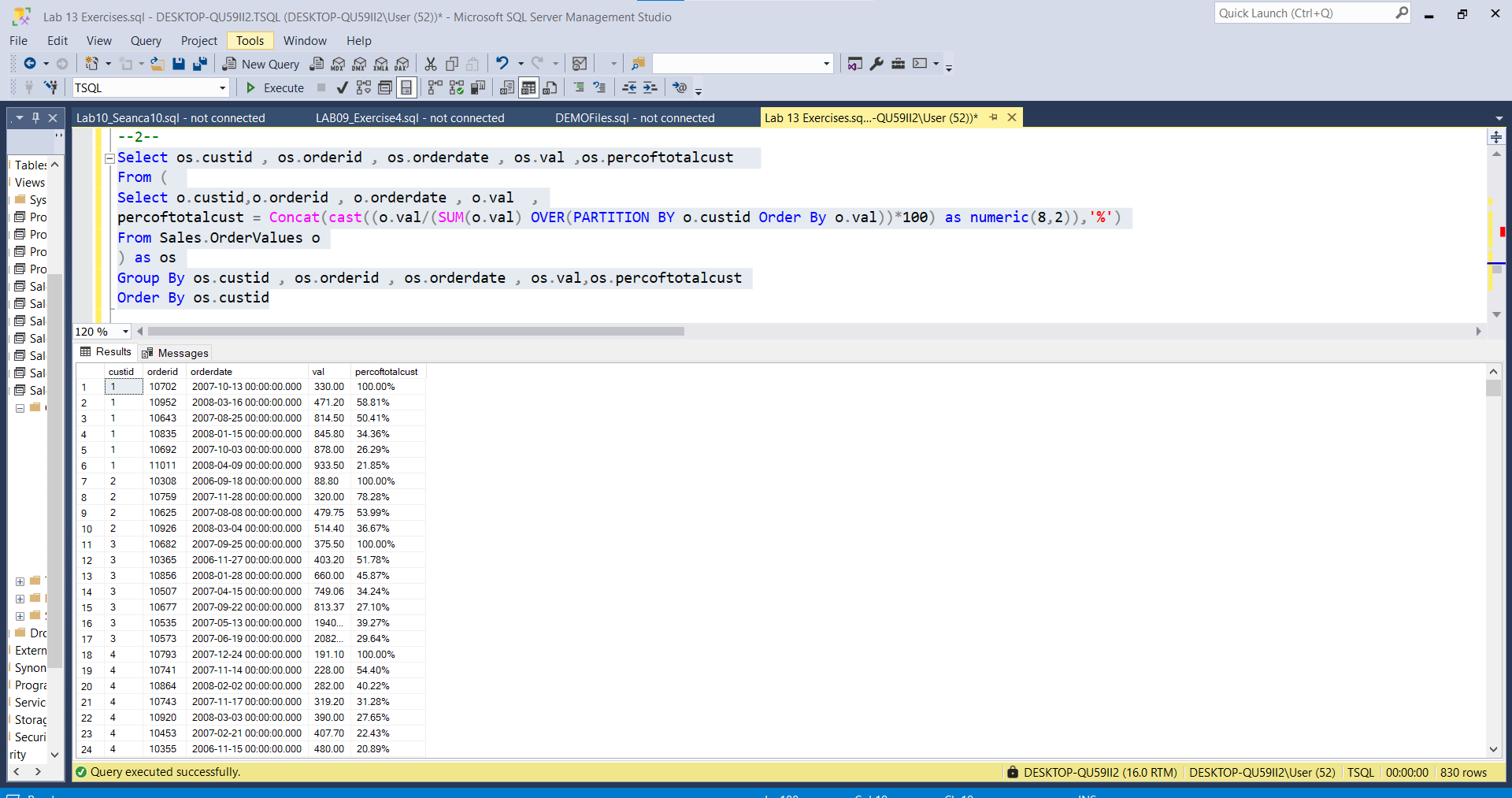
Order By os.custid

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

--1--



--2--



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

-- Task 2

--

-- Copy the previous SELECT statement and modify it by adding a new calculated column named runval. This column should contain a running sales total for each customer based on order date, using orderid as the tiebreaker.

Select os.customerId,os.orderid,os.orderdate, os.runval AS RunVal

From (

Select LAST\_VALUE(o.custid) OVER (ORDER BY o.custid ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW

) as customerId,o.orderid,o.orderdate ,runval = (SUM(o.val) OVER(PARTITION BY o.custid Order By o.val))

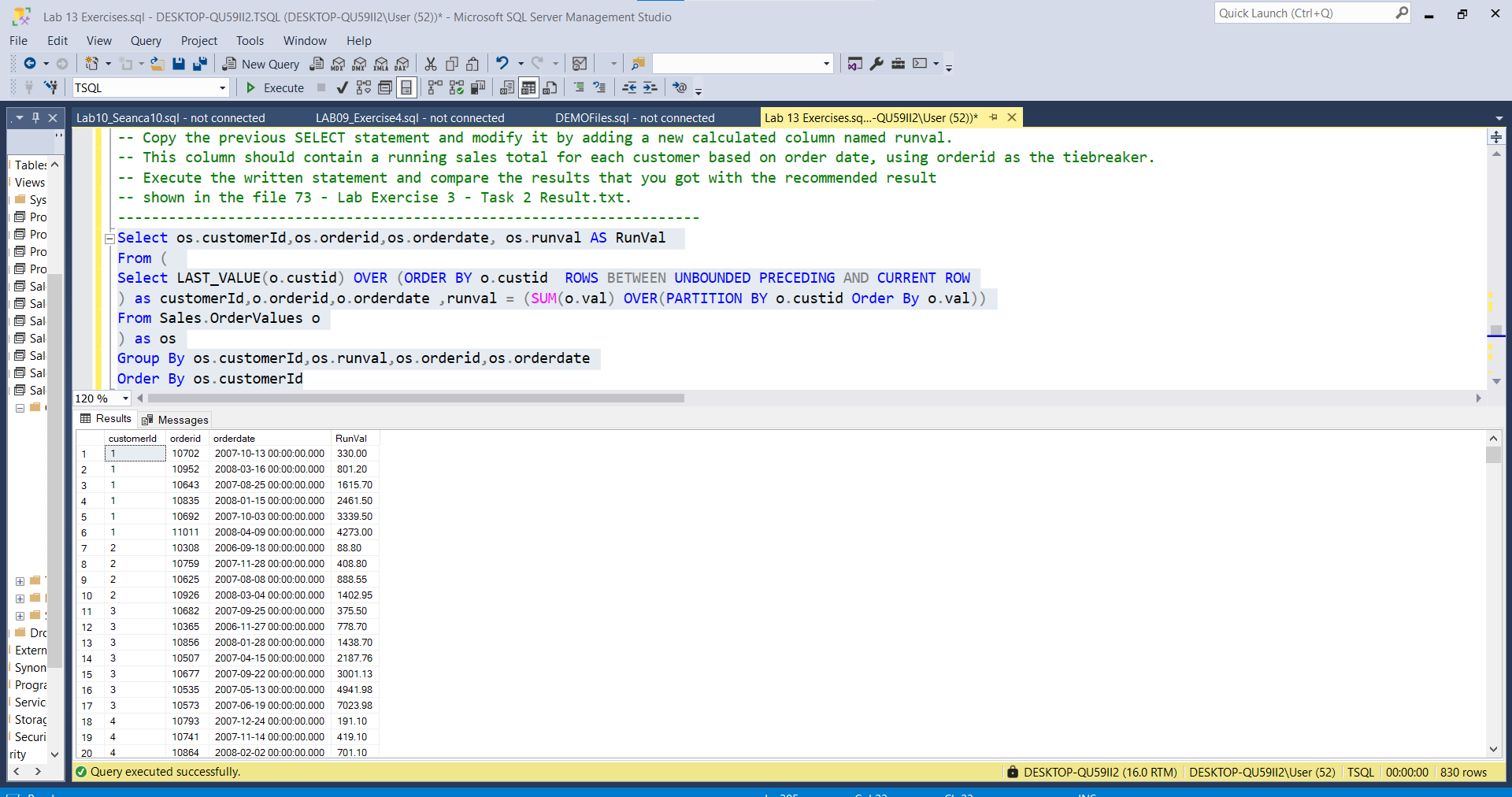
From Sales.OrderValues o

) as os

Group By os.customerId,os.runval,os.orderid,os.orderdate

Order By os.customerId

-- Execute the written statement and compare the results that you got with the recommended result shown in the file 73 - Lab Exercise 3 - Task 2 Result.txt.



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

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

-- Task 3

--

-- Copy the SalesMonth2007 CTE in the last task in exercise 2. Write a SELECT statement to retrieve the monthno and val columns. Add two calculated columns:

-- avglast3months. This column should contain the average sales amount for last three months before the current month using a window aggregate function. You can assume that there are no missing months.

-- ytdval. This column should contain the cumulative sales value up to the current month.

WITH CTE\_AverageSales2007

AS

( Select TOP(12) m.monthForAverage ,m.CurrentAverageSales,m.ytdval , LAG(m.ytdval,1,0) OVER(Order By m.ytdval) AS avglast3months,

m.januaryValue,m.diffJanuary, LEAD(m.ytdval,1,0) OVER(Order By m.ytdval) AS nextVal

FROM(

Select month(o.orderdate) as monthForAverage , AVG(o.val) CurrentAverageSales , SUM(AVG(o.val)) OVER( Order By month(o.orderdate)) ytdval,

FIRST\_VALUE(AVG(o.val)) OVER (ORDER BY month(o.orderdate)) as januaryValue, diffJanuary = AVG(o.val) -FIRST\_VALUE(AVG(o.val)) OVER (ORDER BY month(o.orderdate))

From Sales.OrderValues o

Group by month(o.orderdate)

) as m

Group By m.monthForAverage,m.CurrentAverageSales,m.ytdval,m.januaryValue,diffJanuary

ORDER BY m.monthForAverage

)

Select \*

From CTE\_AverageSales2007 AS avgSales

go

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

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

