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

-- LAB 09

--

-- Exercise 4

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

USE TSQL;

GO

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

-- Task 1

--

-- Write a SELECT statement to retrieve the top 10 customers by total sales amount that spent more than $10,000 in terms of sales amount. Display the custid column from the Orders table and a calculated column that contains the total sales amount based on the qty and unitprice columns from the Sales.OrderDetails table. Use the alias totalsalesamount for the calculated column.

--

-- Execute the written statement and compare the results that you got with the recommended result shown in the file 82 - Lab Exercise 4 - Task 1 Result.txt.

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

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

-- Detyra 1

--

-- Nxirrni 10 klientet sipas shumes totale te shitjes qe kane shpenzuar me shume se $10,000. Shfaq kolonen custid nga tabela Orders dhe nje kolone te perllogaritur qe permban shumen totale te shitjeve bazuar ne kolonat qty dhe unitprice nga tabela Sales.OrderDetails.Perdor alias totalsalesamount per kolonen e perllogaritur.

--

-- Ekzekutoni veprimet e mesiperme dhe krahasoni rezultatet qe ju moret me rezultatet e deshiruara te treguara ne dokumentin 82 - Lab Exercise 4 - Task 1 Result.txt.

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

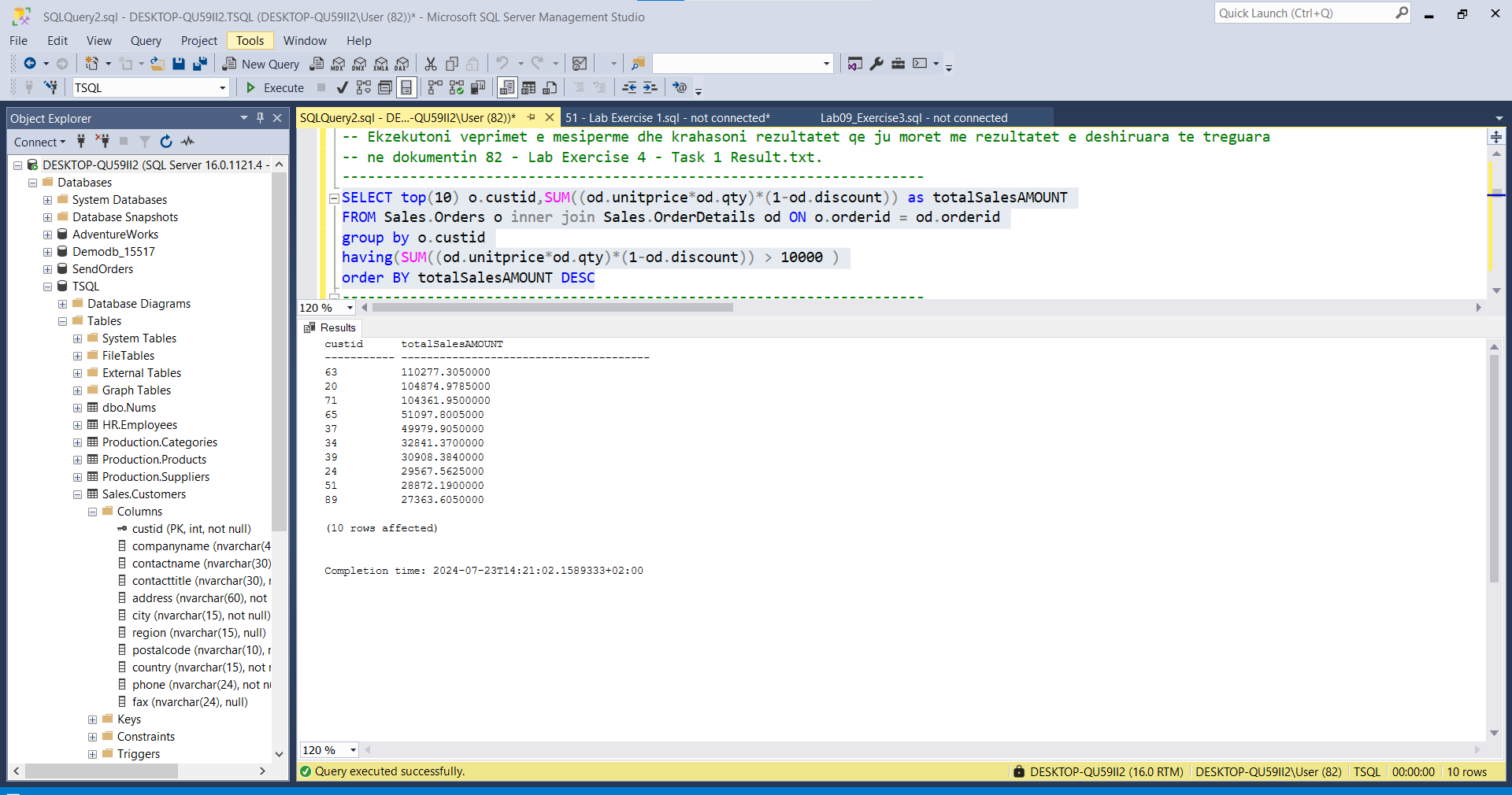
SELECT top(10) o.custid,SUM((od.unitprice\*od.qty)\*(1-od.discount)) as totalSalesAMOUNT

FROM Sales.Orders o inner join Sales.OrderDetails od ON o.orderid = od.orderid

group by o.custid

having(SUM((od.unitprice\*od.qty)\*(1-od.discount)) > 10000 )

order BY totalSalesAMOUNT DESC



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

-- Task 2

--

-- Write a SELECT statement against the Sales.Orders and Sales.OrderDetails tables and display the empid column and a calculated column representing the total sales amount. Filter the result to group only the rows with an order year 2008.

--

-- Execute the written statement and compare the results that you got with the recommended result shown in the file 83 - Lab Exercise 4 - Task 2 Result.txt.

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

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

-- Detyra 2

--

-- Selektoni kolonen empid dhe kolonen e perllogaritur qe perfaqeson shumen totale te shitjes nga tabelat Sales.Orders dhe Sales.OrderDetails. Filtroni rezultatin per te grupuar vetem rrjeshtat me vit porosie 2008.

--

-- Ekzekutoni veprimet e mesiperme dhe krahasoni rezultatet qe ju moret me rezultatet e deshiruara te treguara ne dokumentin 83 - Lab Exercise 4 - Task 2 Result.txt.

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

Select e.empid , SUM((od.qty \* od.unitprice)\*(1 - od.discount)) as totalAmountSales

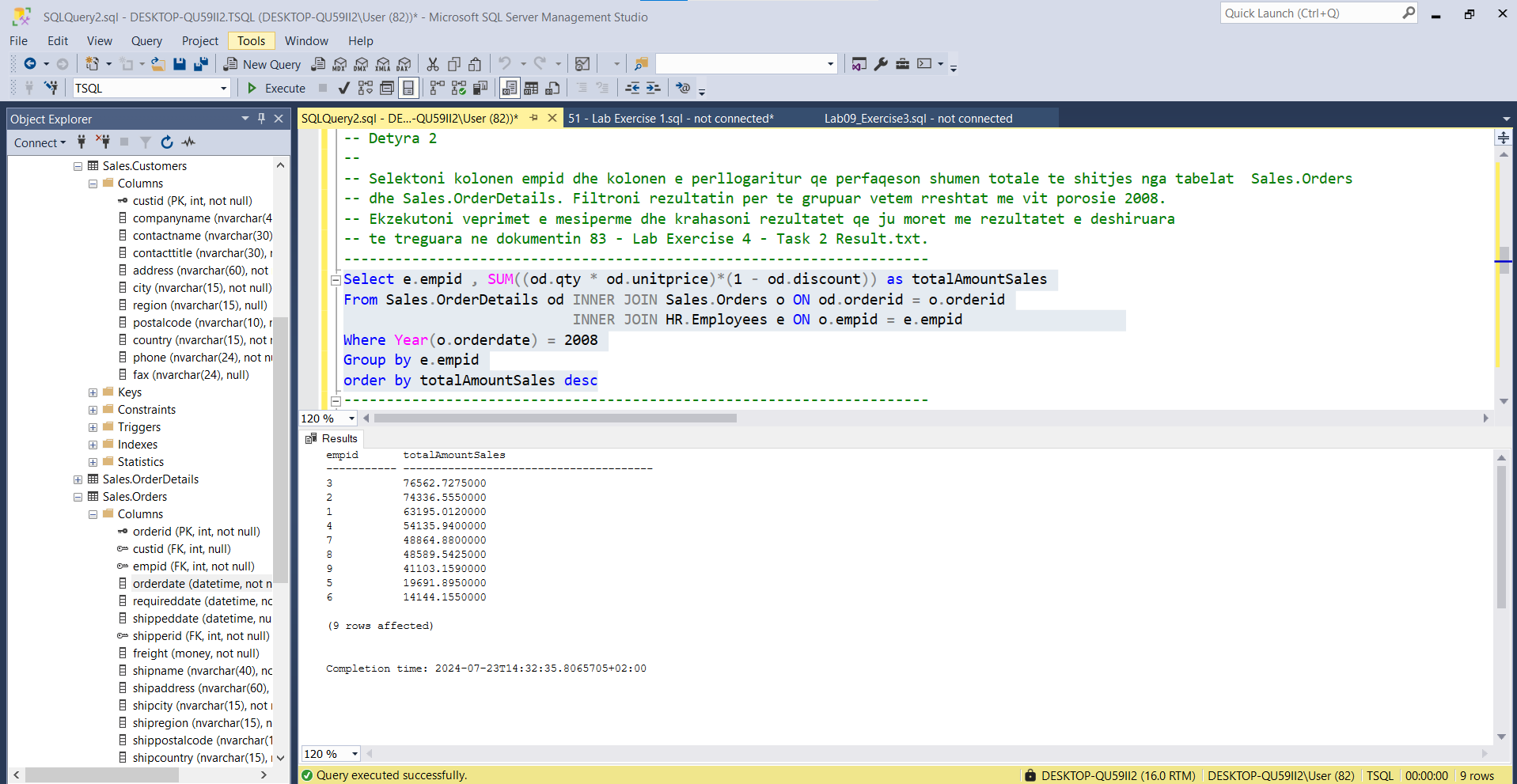
From Sales.OrderDetails od INNER JOIN Sales.Orders o ON od.orderid = o.orderid

INNER JOIN HR.Employees e ON o.empid = e.empid

Where Year(o.orderdate) = 2008

Group by e.empid

order by totalAmountSales desc



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

-- Task 3

--

--1-- Copy the T-SQL statement in task 2 and modify it to apply an additional filter to retrieve only the rows that have a sales amount higher than $10,000.

--

-- Execute the written statement and compare the results that you got with the recommended result shown in the file 84 - Lab Exercise 4 - Task 3\_1 Result.txt.

--

-- Apply an additional filter to show only employees with empid equal number 3.

--

-- Execute the written statement and compare the results that you got with the recommended result shown in the file 85 - Lab Exercise 4 - Task 3\_2 Result.txt.

--

-- Did you apply the predicate logic in the WHERE or in the HAVING clause? Which do you think is better? Why?

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

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

-- Task 3

--

--1-- Kopjoni veprimet T-SQL ne detyren 2 dhe ndryshojeni ate per te aplikuar nje filter per te nxjerre vetem rrjeshtat qe kane nje shume shitje me te larte se $10,000.

--

-- Ekzekutoni veprimet e mesiperme dhe krahasoni rezultatet qe ju moret me rezultatet e deshiruara te treguara ne dokumentin 84 - Lab Exercise 4 - Task 3\_1 Result.txt.

--

-- Aplikoni nje filtrim shtese per te treguar vetem punonjesit me empid te barabarte me numrin 3.

--

-- Ekzekutoni veprimet e mesiperme dhe krahasoni rezultatet qe ju moret me rezultatet e deshiruara te treguara ne dokumentin 85 - Lab Exercise 4 - Task 3\_2 Result.txt.

--

-- Keni aplikuar logjiken predicate ne klazulen WHERE apo HAVING? Cila mendoni qe eshte me e mire? Pse?

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

--1--

Select e.empid , SUM((od.qty \* od.unitprice)\*(1 - od.discount)) as totalAmountSales

From Sales.OrderDetails od INNER JOIN Sales.Orders o ON od.orderid = o.orderid

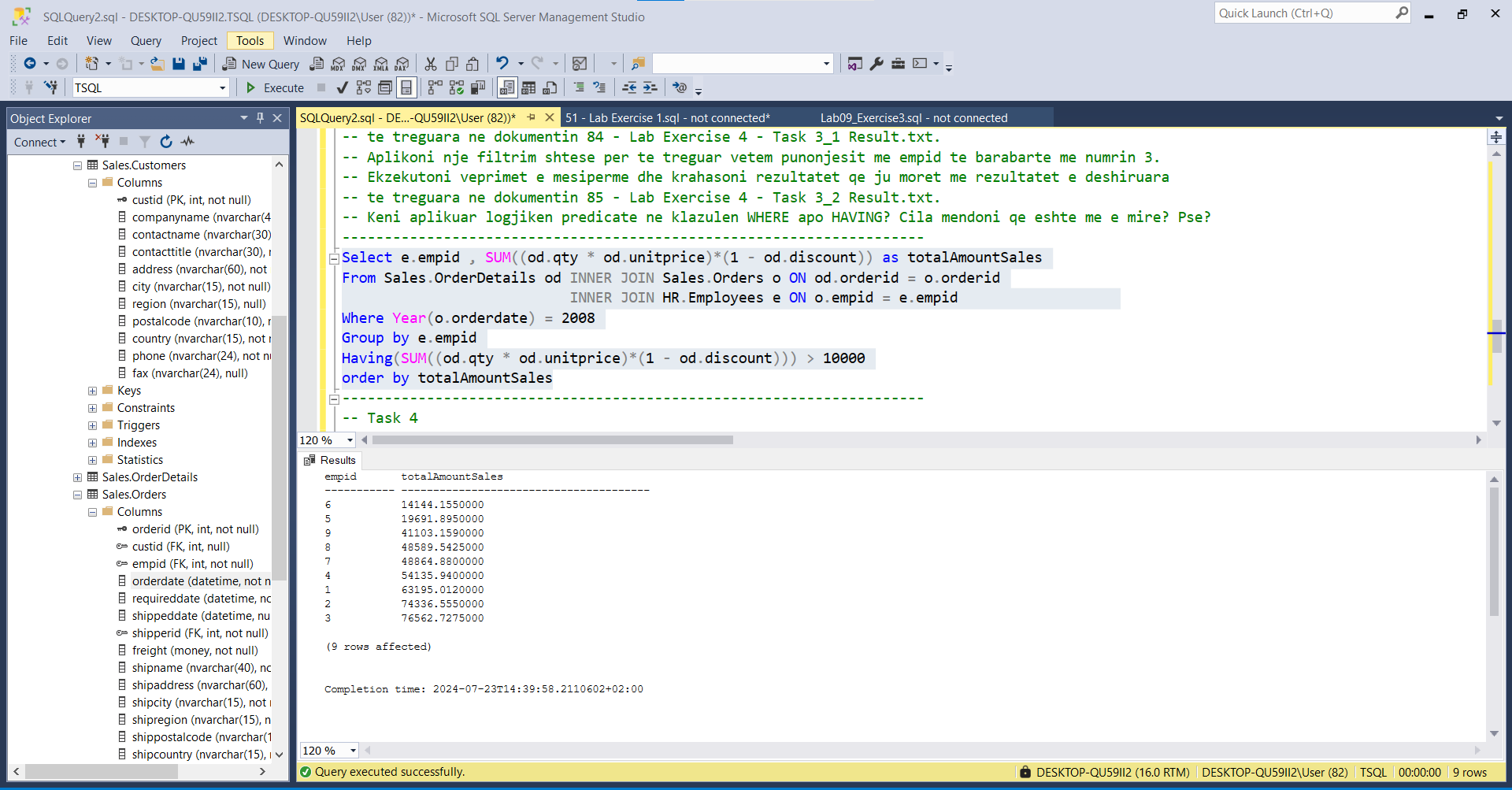
INNER JOIN HR.Employees e ON o.empid = e.empid

Where Year(o.orderdate) = 2008

Group by e.empid

Having(SUM((od.qty \* od.unitprice)\*(1 - od.discount))) > 10000

order by totalAmountSales



Krahasimi i rezultateve:

Task2 🡪

empid totalAmountSales

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

3 76562.7275000

2 74336.5550000

1 63195.0120000

4 54135.9400000

7 48864.8800000

8 48589.5425000

9 41103.1590000

5 19691.8950000

6 14144.1550000

(9 rows affected)

Task3 🡪

empid totalAmountSales

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

3 76562.7275000

2 74336.5550000

1 63195.0120000

4 54135.9400000

7 48864.8800000

8 48589.5425000

9 41103.1590000

5 19691.8950000

6 14144.1550000

\*\*\*Rezultatet e gjenaeruara në taskun\_2 dhe taskun\_3 jane te njejta pasi edhe ne rastin e mesiperm vlerat jane me te medha se 10’000 pavarsisht se ne taskun\_3 ekzekutojmë kushtin qe SUM(…….)>10’000

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

-- Task 4

--

-- Write a SELECT statement to retrieve all customers who placed more than 25 orders and add information about the date of the last order and the total sales amount. Display the custid column from the Sales.Orders table and two calculated columns: lastorderdate based on the orderdate column and totalsalesamount based on the qty and unitprice columns in the Sales.OrderDetails table.

--

-- Execute the written statement and compare the results that you got with the recommended result shown in the file 86 - Lab Exercise 4 - Task 4 Result.txt.

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

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

-- Task 4

--

-- Nxirrni te gjithe klientet qe kryen me shume se 25 porosi dhe shto informacionin rreth dates se porosise se fundit dhe shumes totale te shitjeve. Shfaqni kolonen custid nga tabela Sales.Orders dhe dy kolonat e perllogaritura: lastorderdate bazuar ne kolonen orderdate dhe totalsalesamount bazuar ne kolonat qty dhe unitprice ne tabelen Sales.OrderDetails.

--

-- Ekzekutoni veprimet e mesiperme dhe krahasoni rezultatet qe ju moret me rezultatet e deshiruara te treguara ne dokumentin 86 - Lab Exercise 4 - Task 4 Result.txt.

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

Select o.custid as CustomerId , SUM((od.unitprice\*od.qty)\*(1-od.discount)) as totalsaleamount ,MAX(o.orderdate) AS lastorderdate,

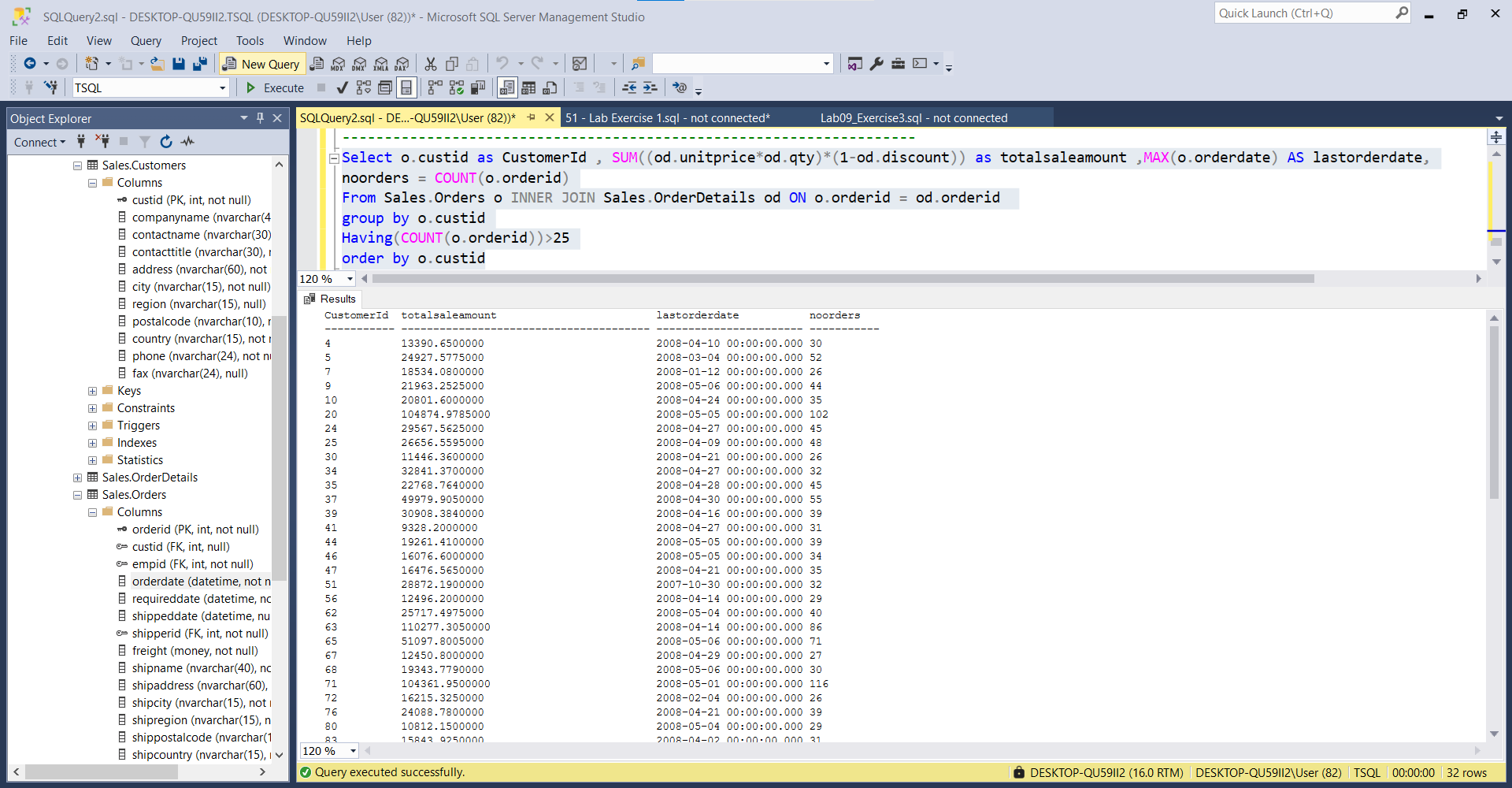
noorders = COUNT(o.orderid)

From Sales.Orders o INNER JOIN Sales.OrderDetails od ON o.orderid = od.orderid

group by o.custid

Having(COUNT(o.orderid))>25

order by o.custid



OUTPUT:

CustomerId totalsaleamount lastorderdate noorders

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

4 13390.6500000 2008-04-10 00:00:00.000 30

5 24927.5775000 2008-03-04 00:00:00.000 52

7 18534.0800000 2008-01-12 00:00:00.000 26

9 21963.2525000 2008-05-06 00:00:00.000 44

10 20801.6000000 2008-04-24 00:00:00.000 35

20 104874.9785000 2008-05-05 00:00:00.000 102

24 29567.5625000 2008-04-27 00:00:00.000 45

25 26656.5595000 2008-04-09 00:00:00.000 48

30 11446.3600000 2008-04-21 00:00:00.000 26

34 32841.3700000 2008-04-27 00:00:00.000 32

35 22768.7640000 2008-04-28 00:00:00.000 45

37 49979.9050000 2008-04-30 00:00:00.000 55

39 30908.3840000 2008-04-16 00:00:00.000 39

41 9328.2000000 2008-04-27 00:00:00.000 31

44 19261.4100000 2008-05-05 00:00:00.000 39

46 16076.6000000 2008-05-05 00:00:00.000 34

47 16476.5650000 2008-04-21 00:00:00.000 35

51 28872.1900000 2007-10-30 00:00:00.000 32

56 12496.2000000 2008-04-14 00:00:00.000 29

62 25717.4975000 2008-05-04 00:00:00.000 40

63 110277.3050000 2008-04-14 00:00:00.000 86

65 51097.8005000 2008-05-06 00:00:00.000 71

67 12450.8000000 2008-04-29 00:00:00.000 27

68 19343.7790000 2008-05-06 00:00:00.000 30

71 104361.9500000 2008-05-01 00:00:00.000 116

72 16215.3250000 2008-02-04 00:00:00.000 26

76 24088.7800000 2008-04-21 00:00:00.000 39

80 10812.1500000 2008-05-04 00:00:00.000 29

83 15843.9250000 2008-04-02 00:00:00.000 31

86 9588.4250000 2008-04-23 00:00:00.000 26

87 15648.7025000 2008-04-15 00:00:00.000 37

89 27363.6050000 2008-05-01 00:00:00.000 40

(32 rows affected)

Completion time: 2024-07-23T15:32:48.7887743+02:00