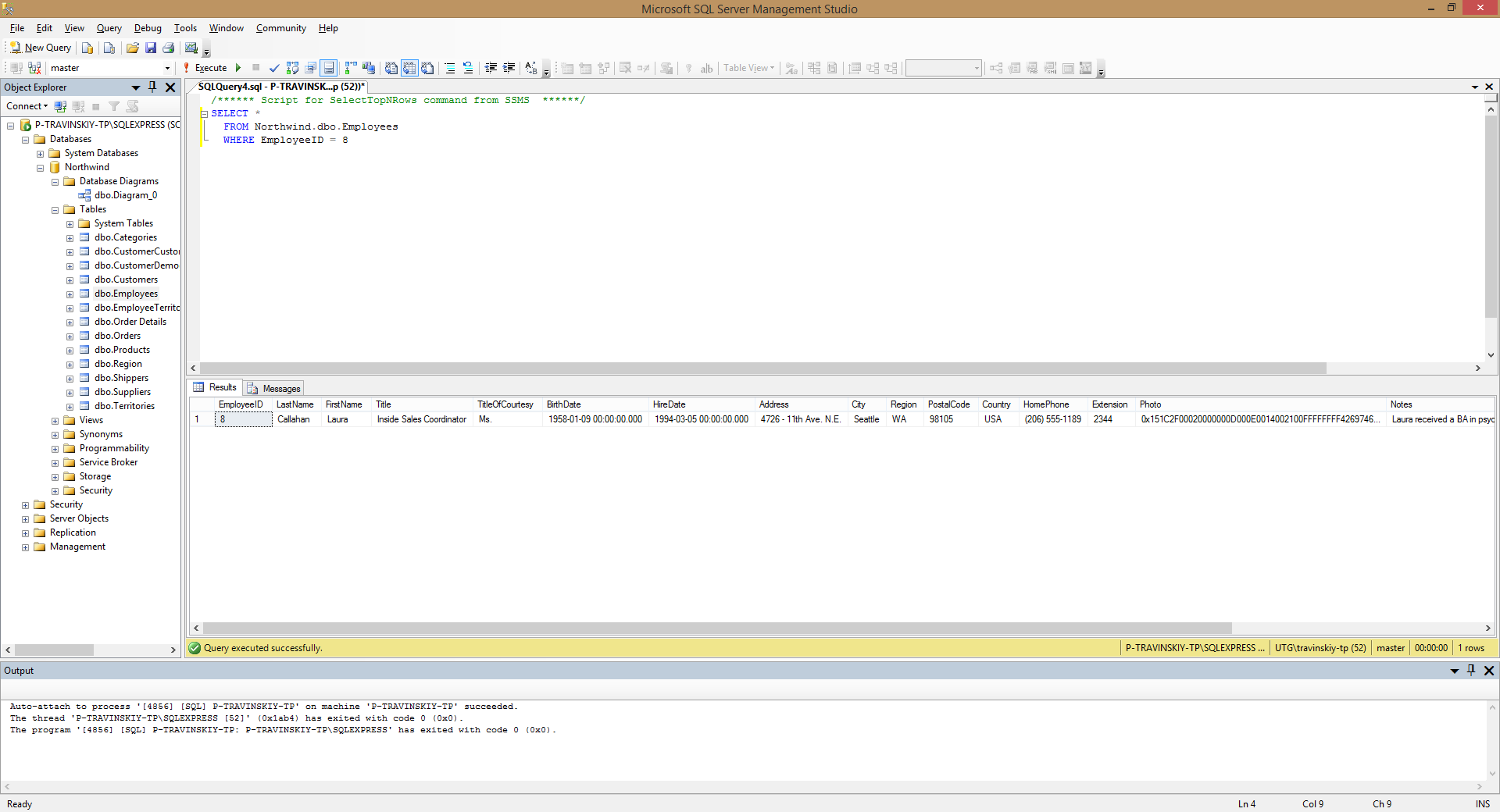
# Examples of tasks on SQL:

1. Show all info about the employee with ID 8.

SELECT \*

FROM Northwind.dbo.Employees

WHERE EmployeeID = 8

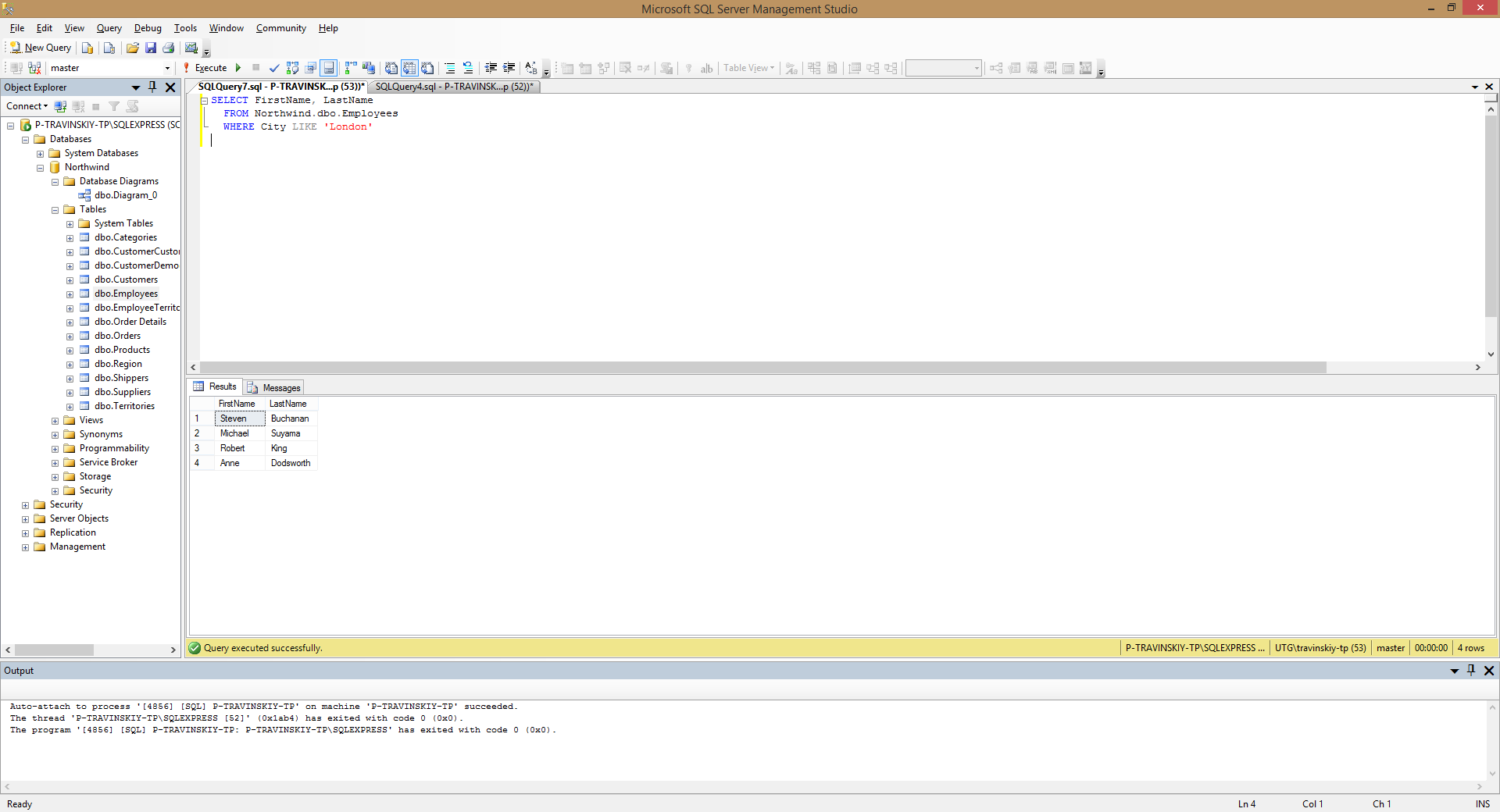


1. Show the list of first and last names of the employees from London.

SELECT FirstName, LastName

FROM Northwind.dbo.Employees

WHERE City LIKE 'London'

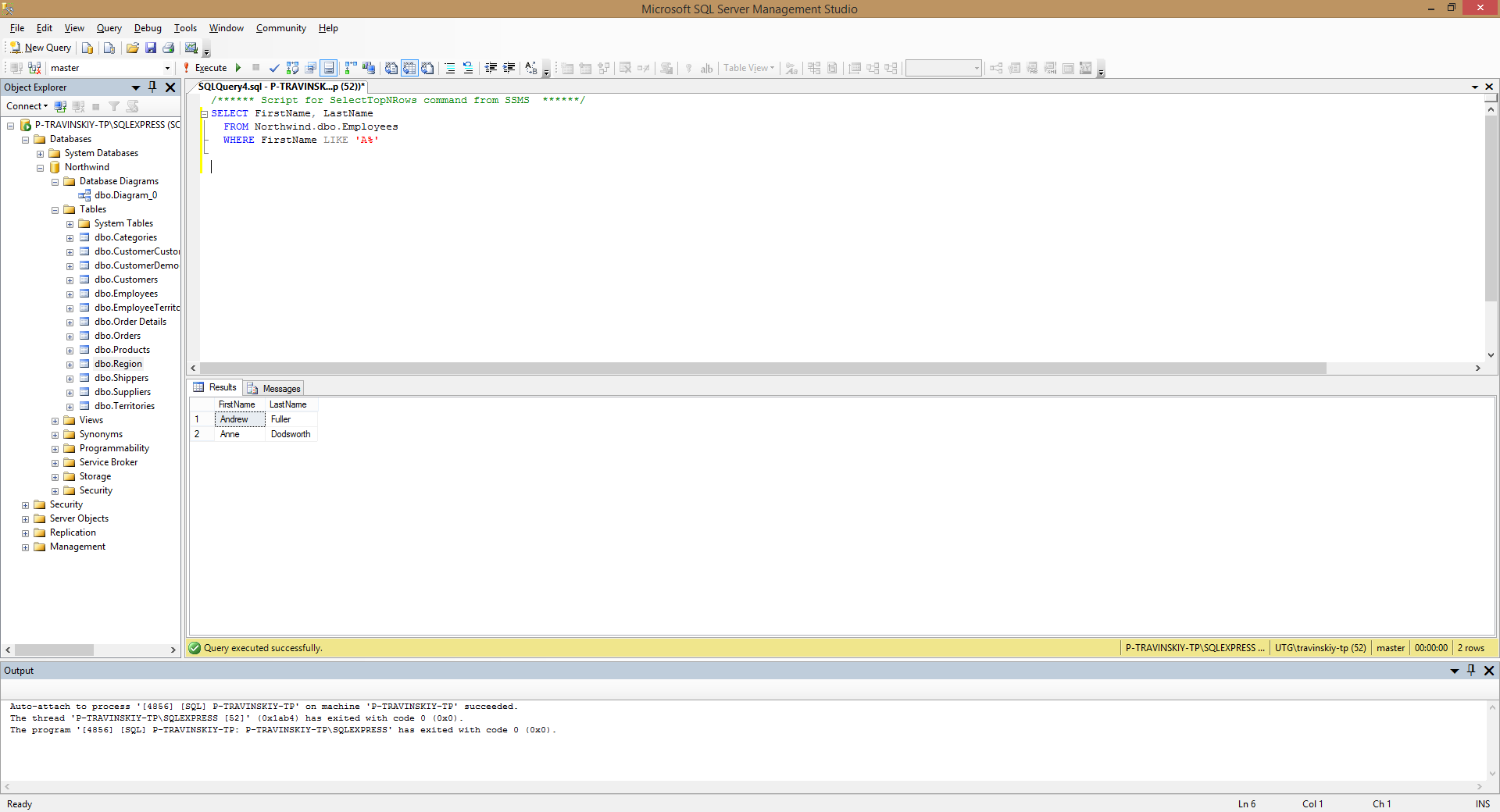


1. Show the list of first and last names of the employees whose first name begins with letter A.

SELECT FirstName, LastName

FROM Northwind.dbo.Employees

WHERE FirstName LIKE 'A%'



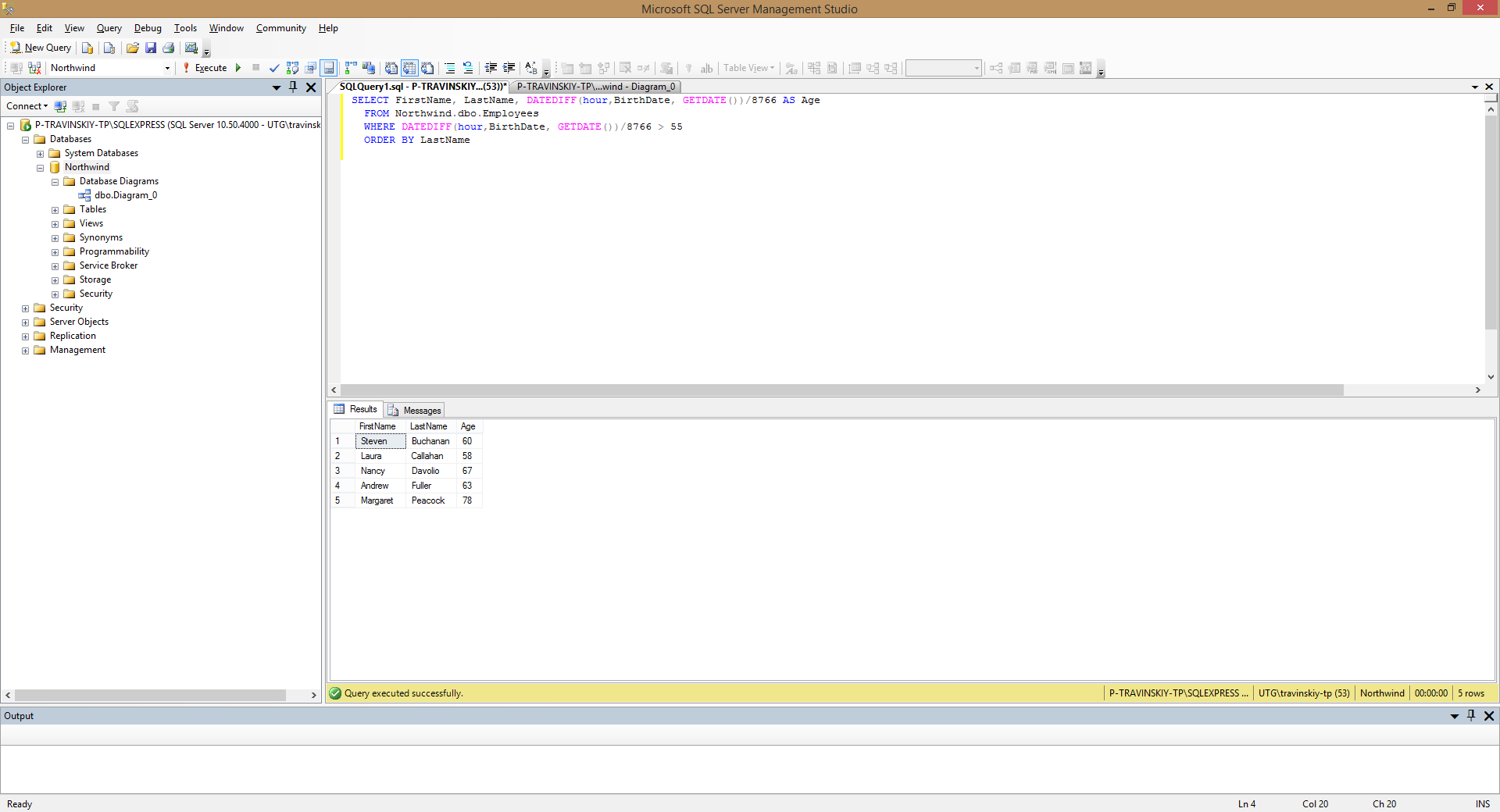
1. Show the list of first, last names and ages of the employees whose age is greater than 55. The result should be sorted by last name.

SELECT FirstName, LastName, DATEDIFF(hour,BirthDate, GETDATE())/8766 AS Age

FROM Northwind.dbo.Employees

WHERE DATEDIFF(hour,BirthDate, GETDATE())/8766 > 55

ORDER BY LastName

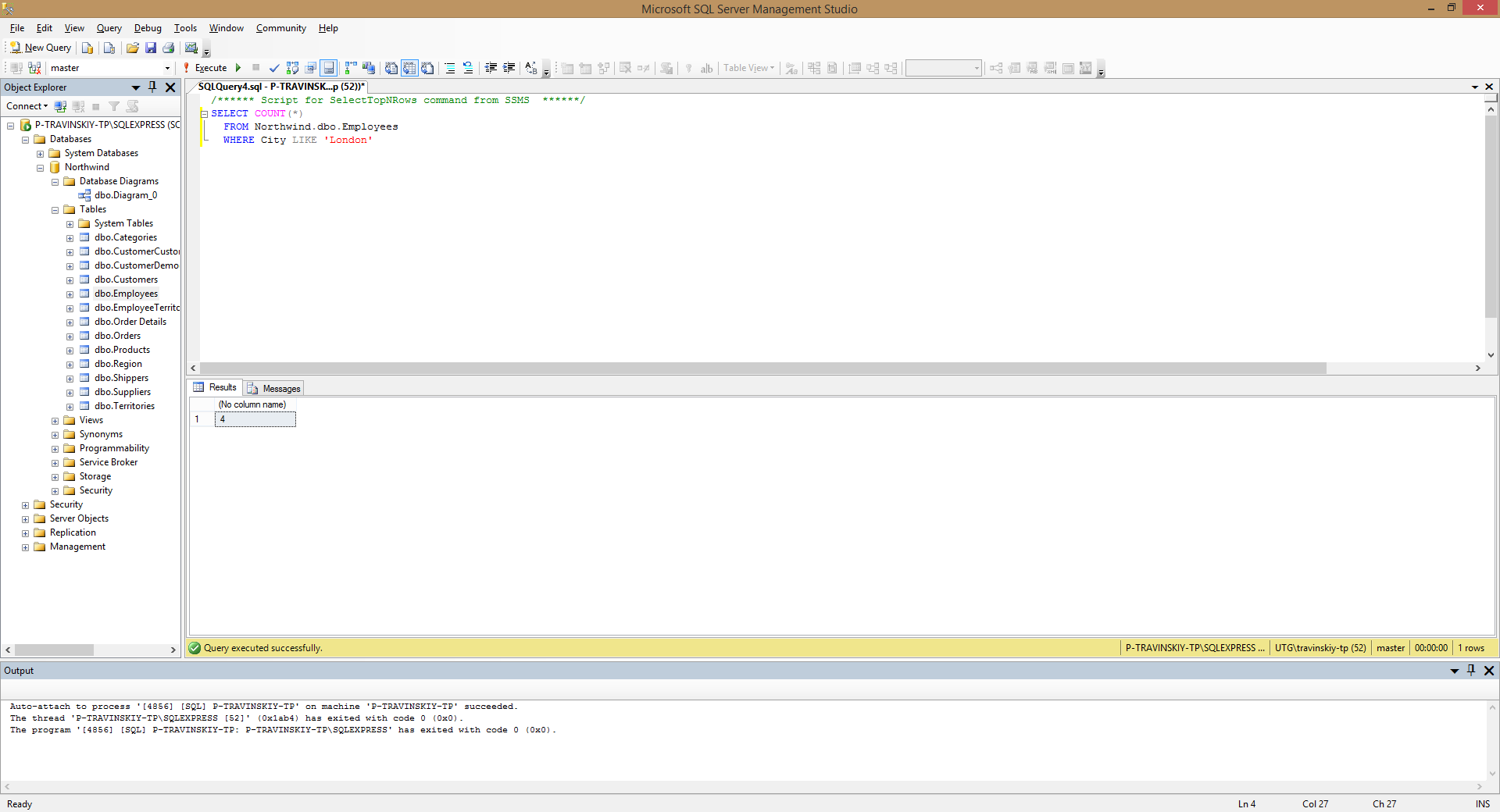


1. Calculate the count of employees from London.

SELECT COUNT(\*)

FROM Northwind.dbo.Employees

WHERE City LIKE 'London'

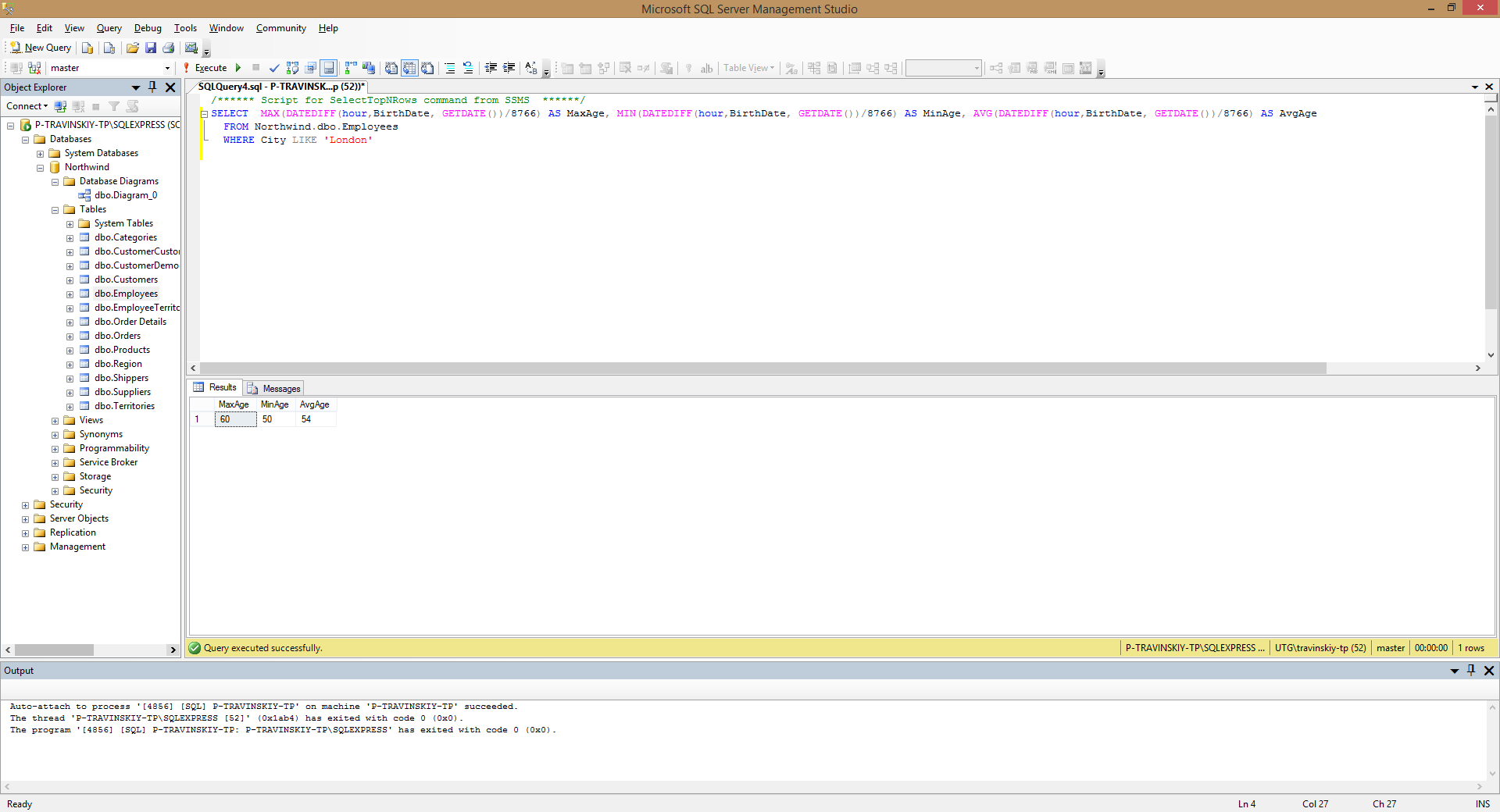


1. Calculate the greatest, the smallest and the average age among the employees from London.

SELECT MAX(DATEDIFF(hour,BirthDate, GETDATE())/8766) AS MaxAge, MIN(DATEDIFF(hour,BirthDate, GETDATE())/8766) AS MinAge, AVG(DATEDIFF(hour,BirthDate, GETDATE())/8766) AS AvgAge

FROM Northwind.dbo.Employees

WHERE City LIKE 'London'

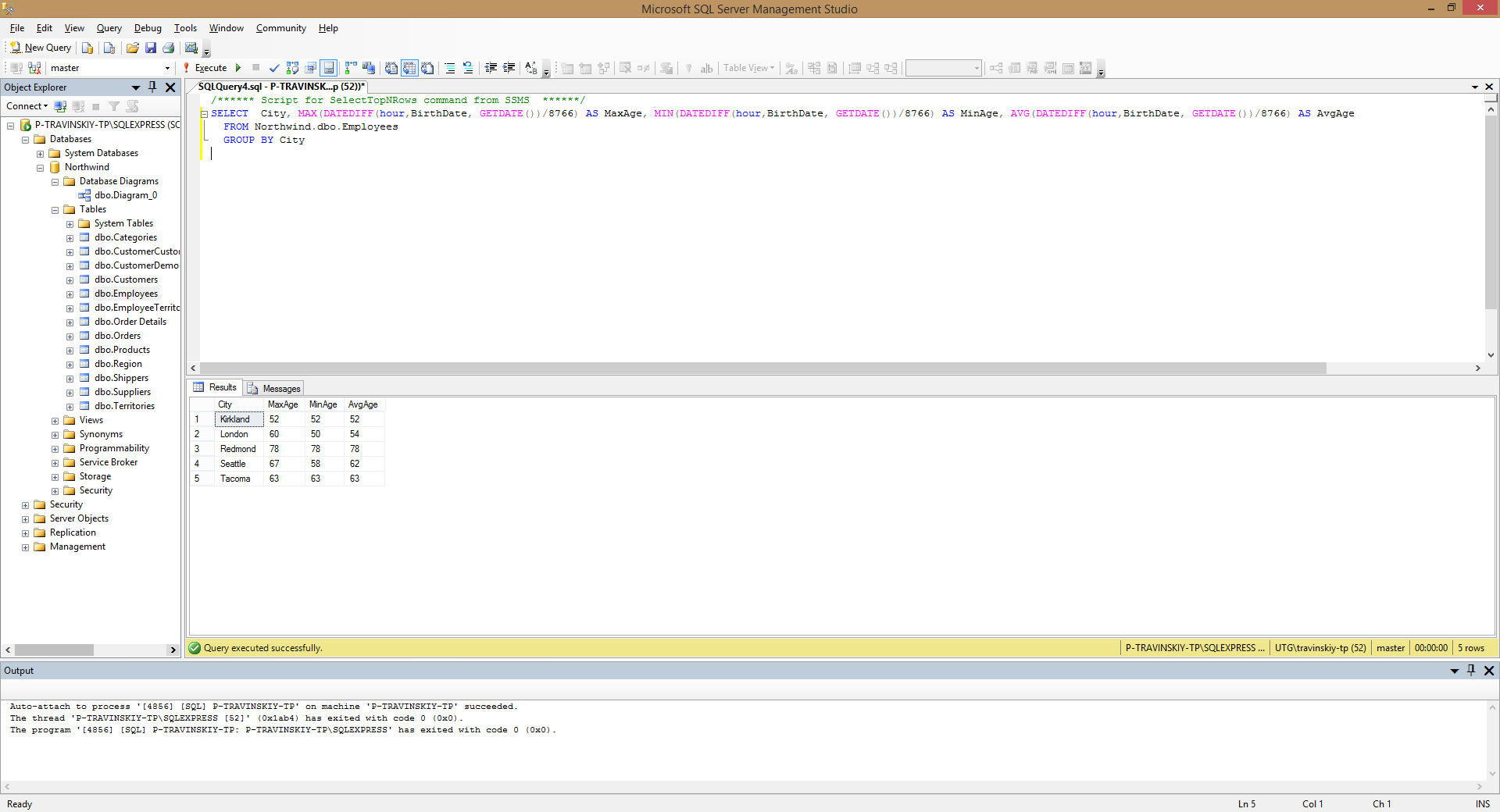


1. Calculate the greatest, the smallest and the average age of the employees for each city.

SELECT City, MAX(DATEDIFF(hour,BirthDate, GETDATE())/8766) AS MaxAge, MIN(DATEDIFF(hour,BirthDate, GETDATE())/8766) AS MinAge, AVG(DATEDIFF(hour,BirthDate, GETDATE())/8766) AS AvgAge

FROM Northwind.dbo.Employees

GROUP BY City



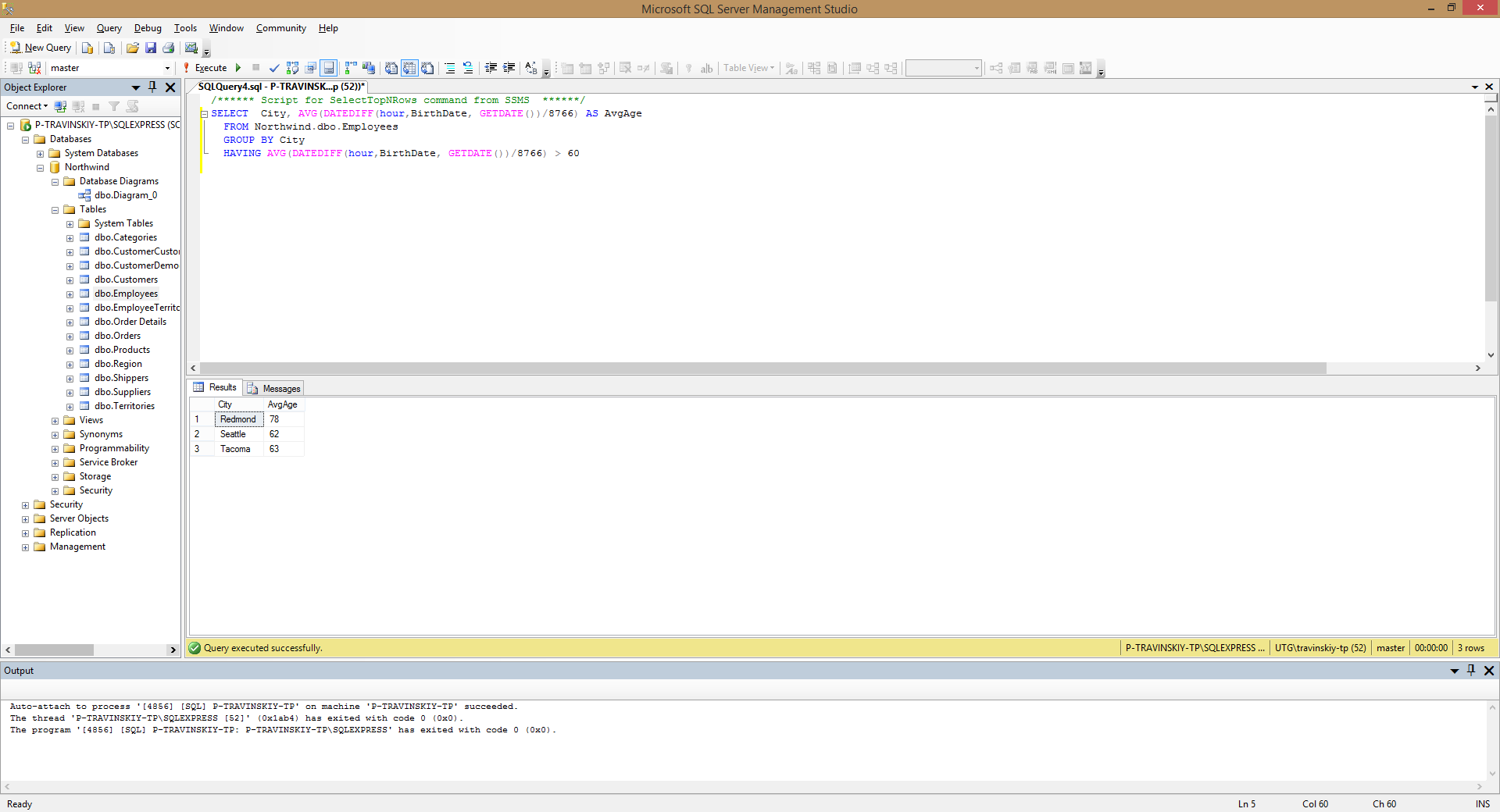
1. Show the list of cities in which the average age of employees is greater than 60 (the average age is also to be shown)

SELECT City, AVG(DATEDIFF(hour,BirthDate, GETDATE())/8766) AS AvgAge

FROM Northwind.dbo.Employees

GROUP BY City

HAVING AVG(DATEDIFF(hour,BirthDate, GETDATE())/8766) > 60



1. Show the first and last name(s) of the eldest employee(s). Use a subquery.

SELECT FirstName, LastName

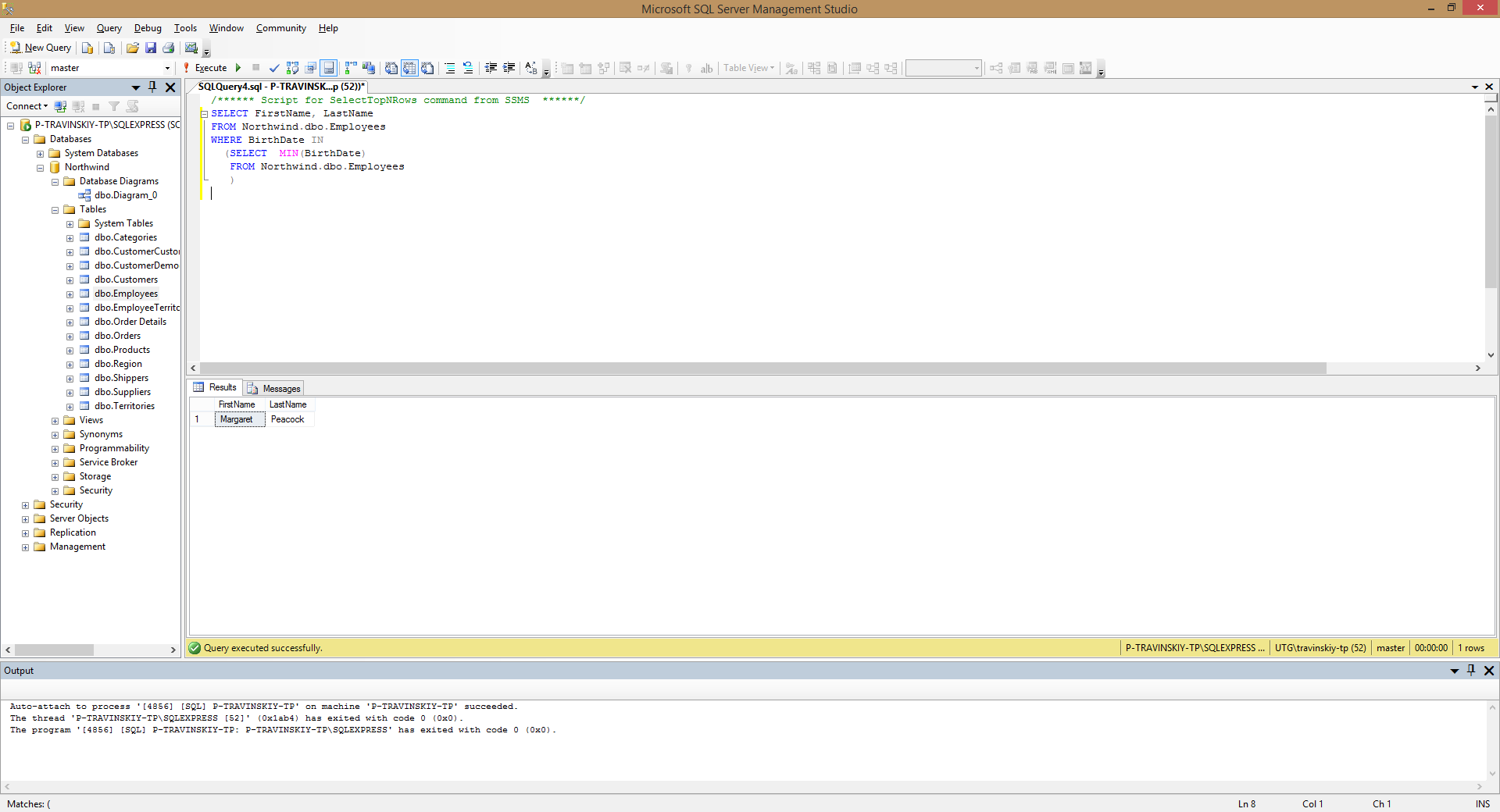
FROM Northwind.dbo.Employees

WHERE BirthDate IN

(SELECT MIN(BirthDate)

FROM Northwind.dbo.Employees

)

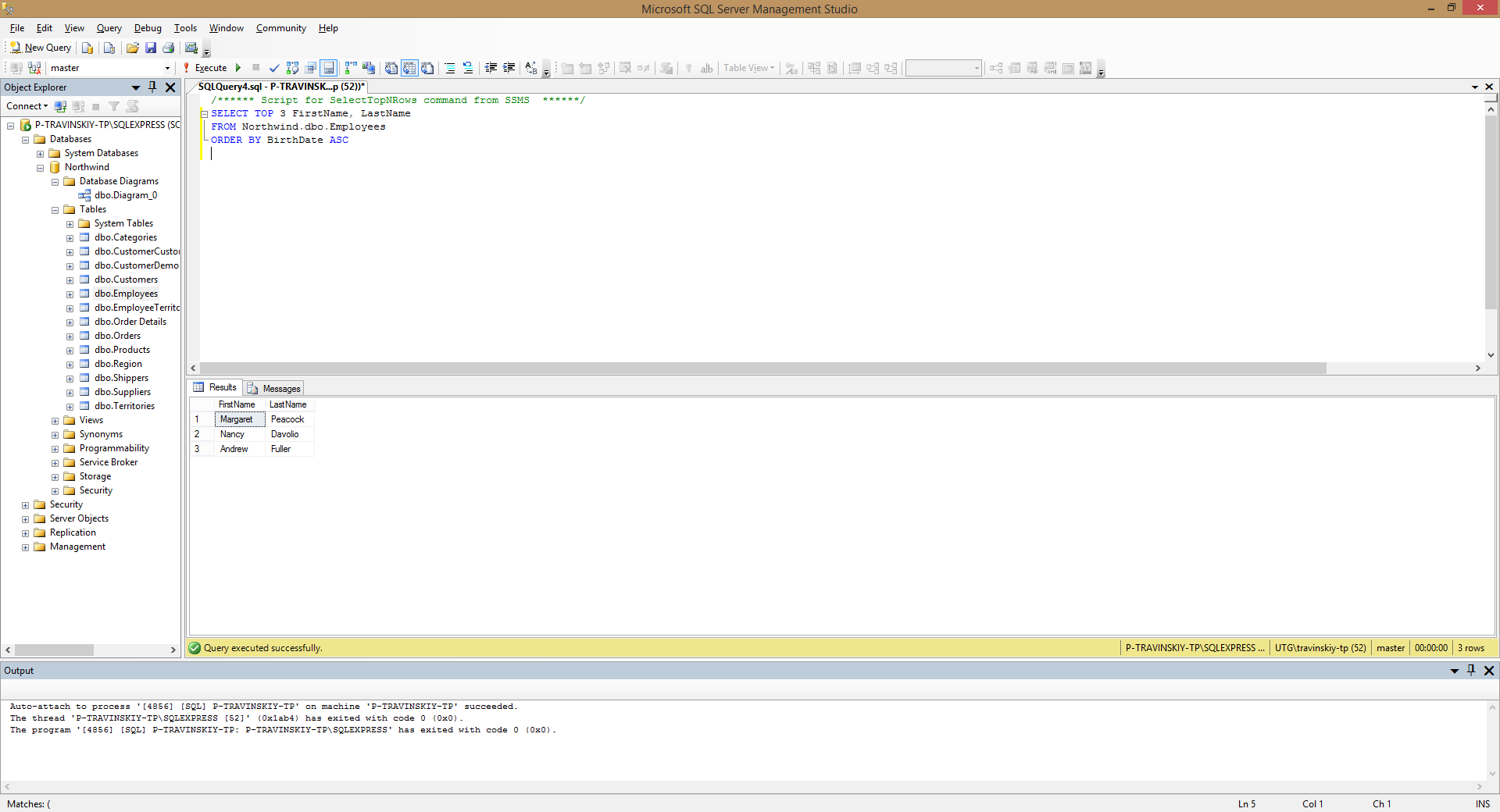


1. Show first, last names and ages of 3 eldest employees.

SELECT TOP 3 FirstName, LastName

FROM Northwind.dbo.Employees

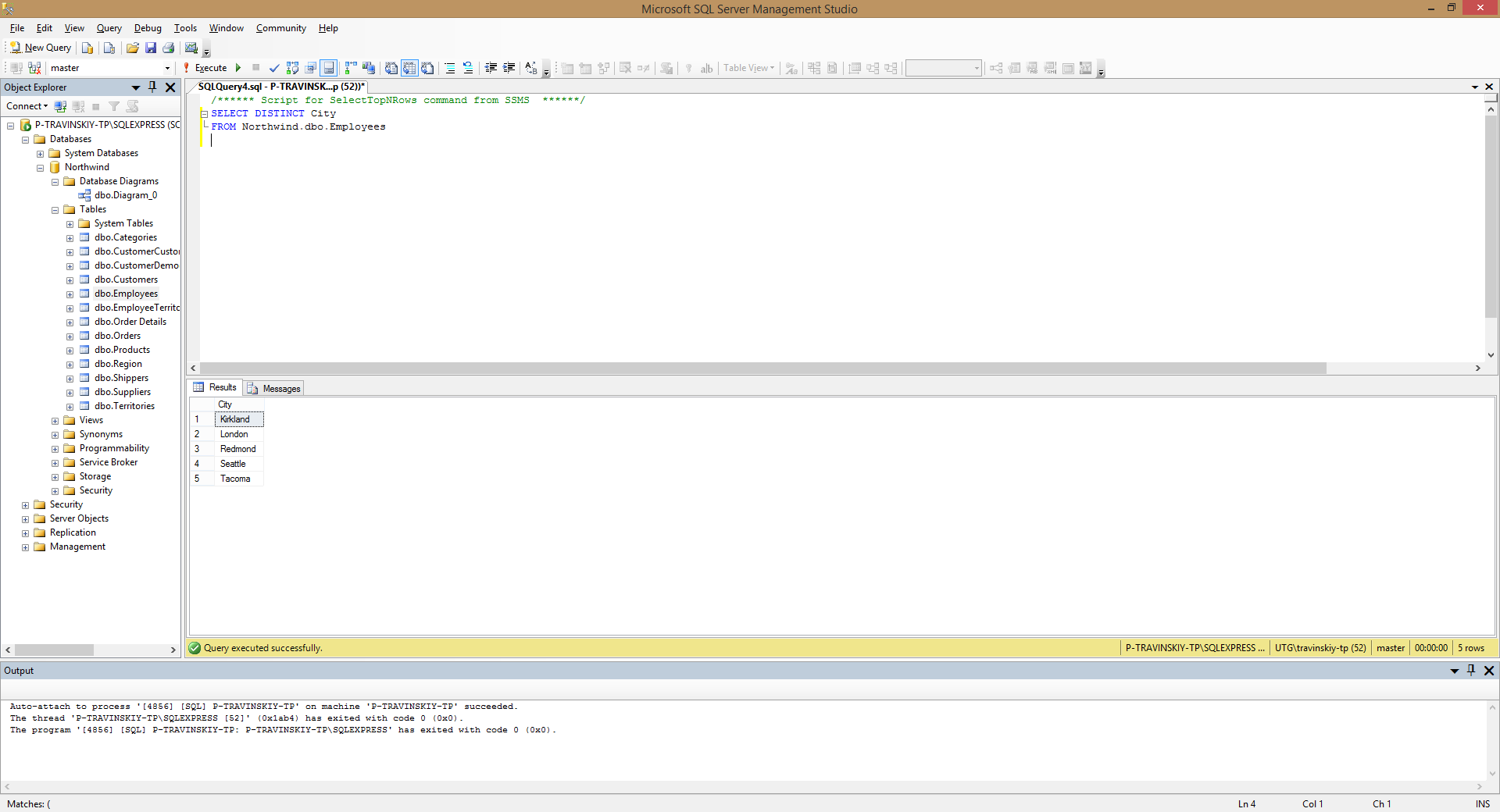
ORDER BY BirthDate ASC



1. Show the list of all cities where the employees are from.

SELECT DISTINCT City

FROM Northwind.dbo.Employees

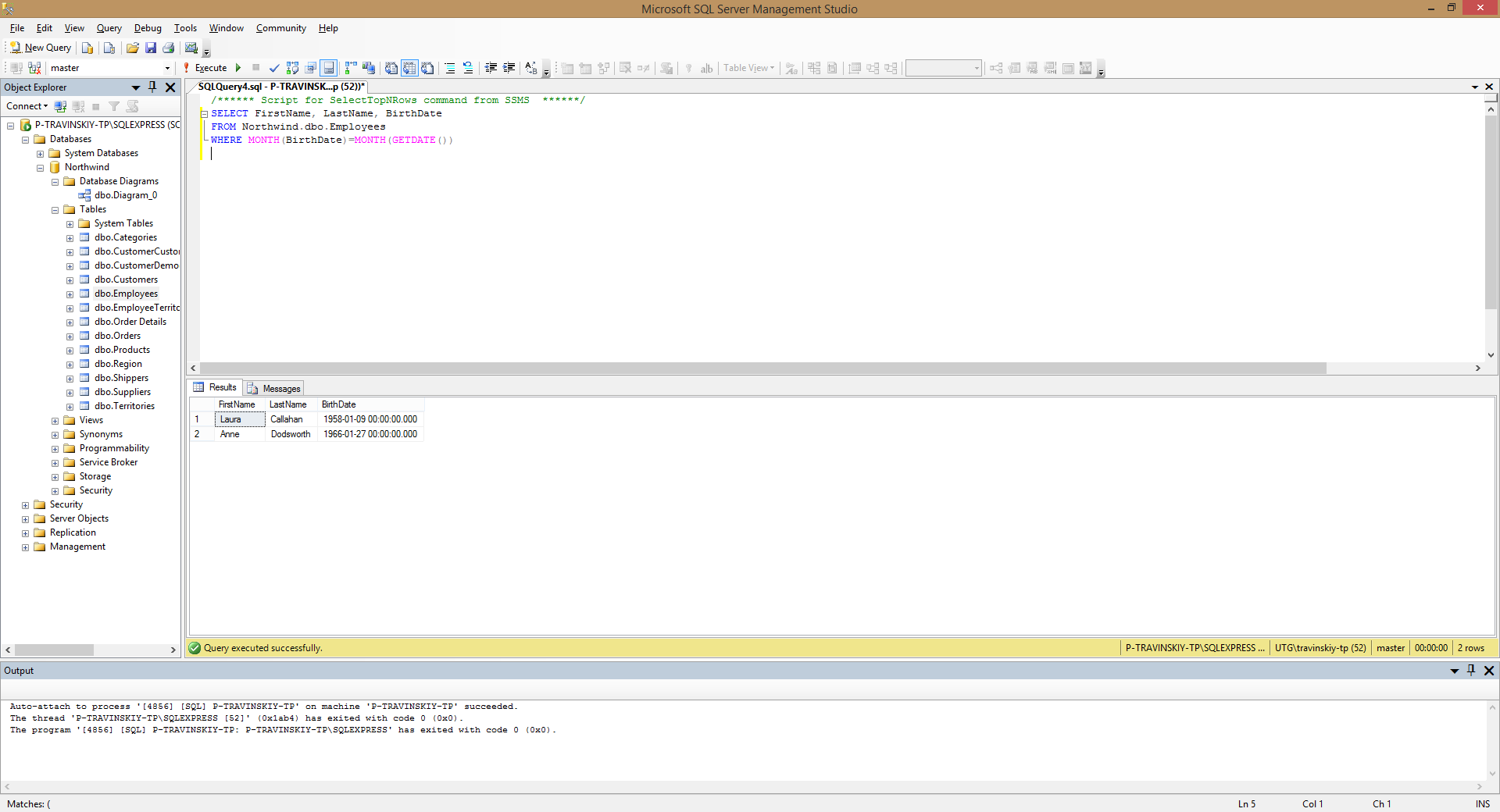


1. Show first, last names and dates of birth of the employees who celebrate their birthdays this month.

SELECT FirstName, LastName, BirthDate

FROM Northwind.dbo.Employees

WHERE MONTH(BirthDate)=MONTH(GETDATE())



1. Show first and last names of the employees who used to serve orders shipped to Madrid.

SELECT DISTINCT a.FirstName, a.LastName

FROM Northwind.dbo.Employees a, Northwind.dbo.Orders b

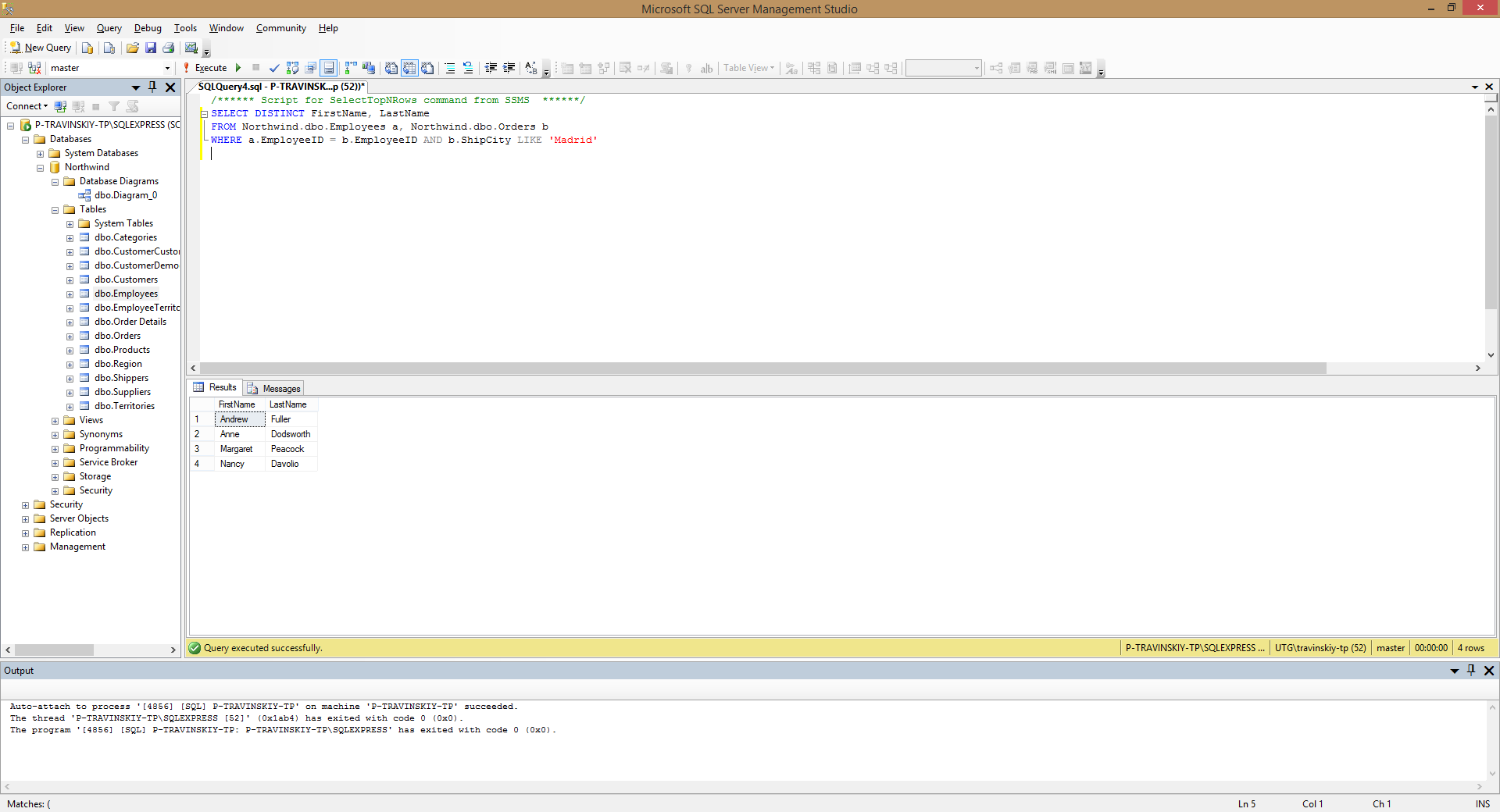
WHERE a.EmployeeID = b.EmployeeID AND b.ShipCity LIKE 'Madrid'

or else

SELECT DISTINCT a.FirstName, a.LastName

FROM Northwind.dbo.Employees a LEFT JOIN Northwind.dbo.Orders b ON a.EmployeeID = b.EmployeeID

WHERE b.ShipCity LIKE 'Madrid'



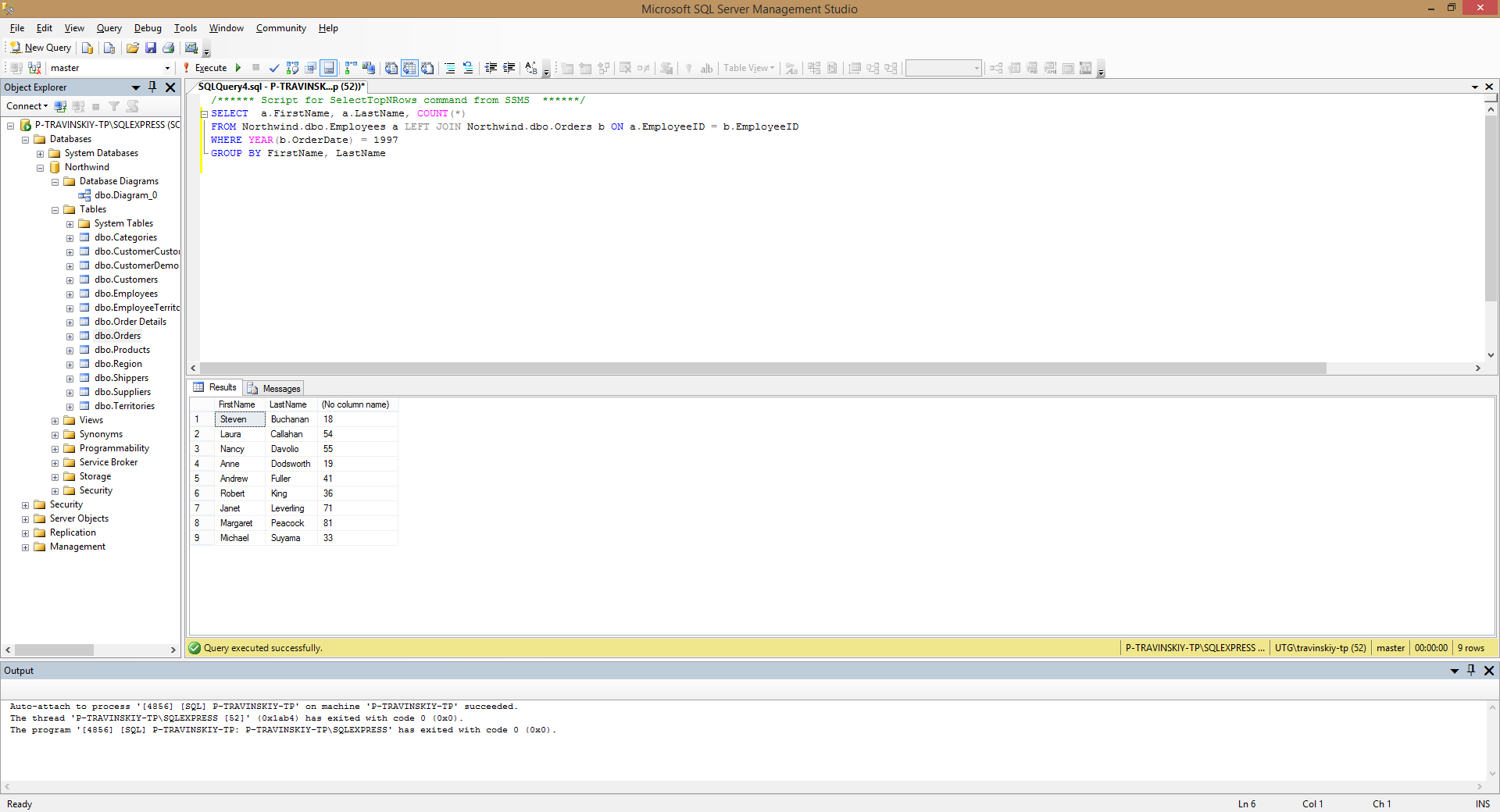
1. Show first and last names of the employees as well as the count of orders each of them have received during the year 1997 (use left join).

SELECT a.FirstName, a.LastName, COUNT(\*)

FROM Northwind.dbo.Employees a LEFT JOIN Northwind.dbo.Orders b ON a.EmployeeID = b.EmployeeID

WHERE YEAR(b.OrderDate) = 1997

GROUP BY FirstName, LastName



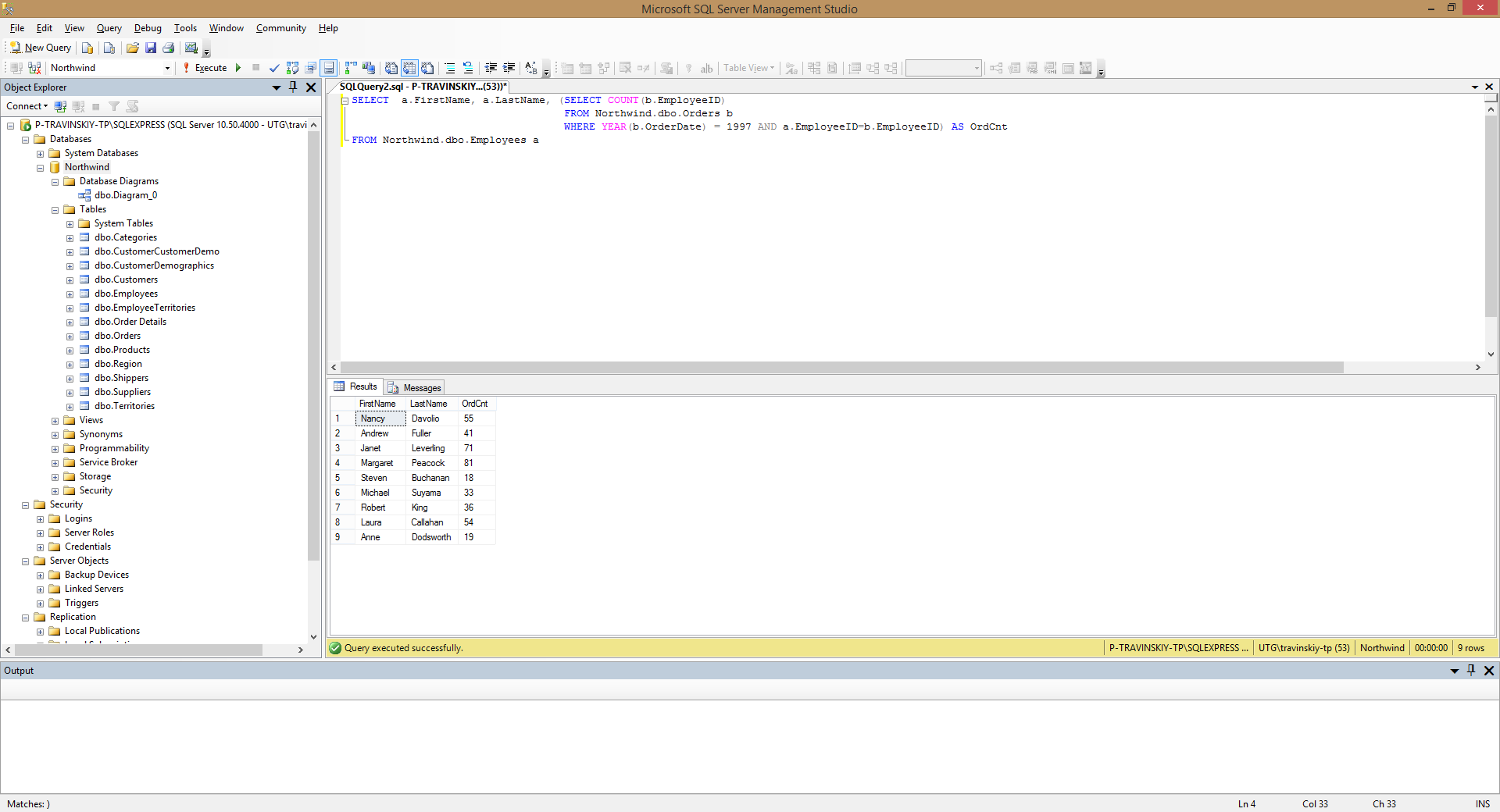
1. Show first and last names of the employees as well as the count of orders each of them have received during the year 1997 (use a subquery).

SELECT a.FirstName, a.LastName, (SELECT COUNT(b.EmployeeID)

FROM Northwind.dbo.Orders b

WHERE YEAR(b.OrderDate) = 1997 AND a.EmployeeID=b.EmployeeID) AS OrdCnt

FROM Northwind.dbo.Employees a



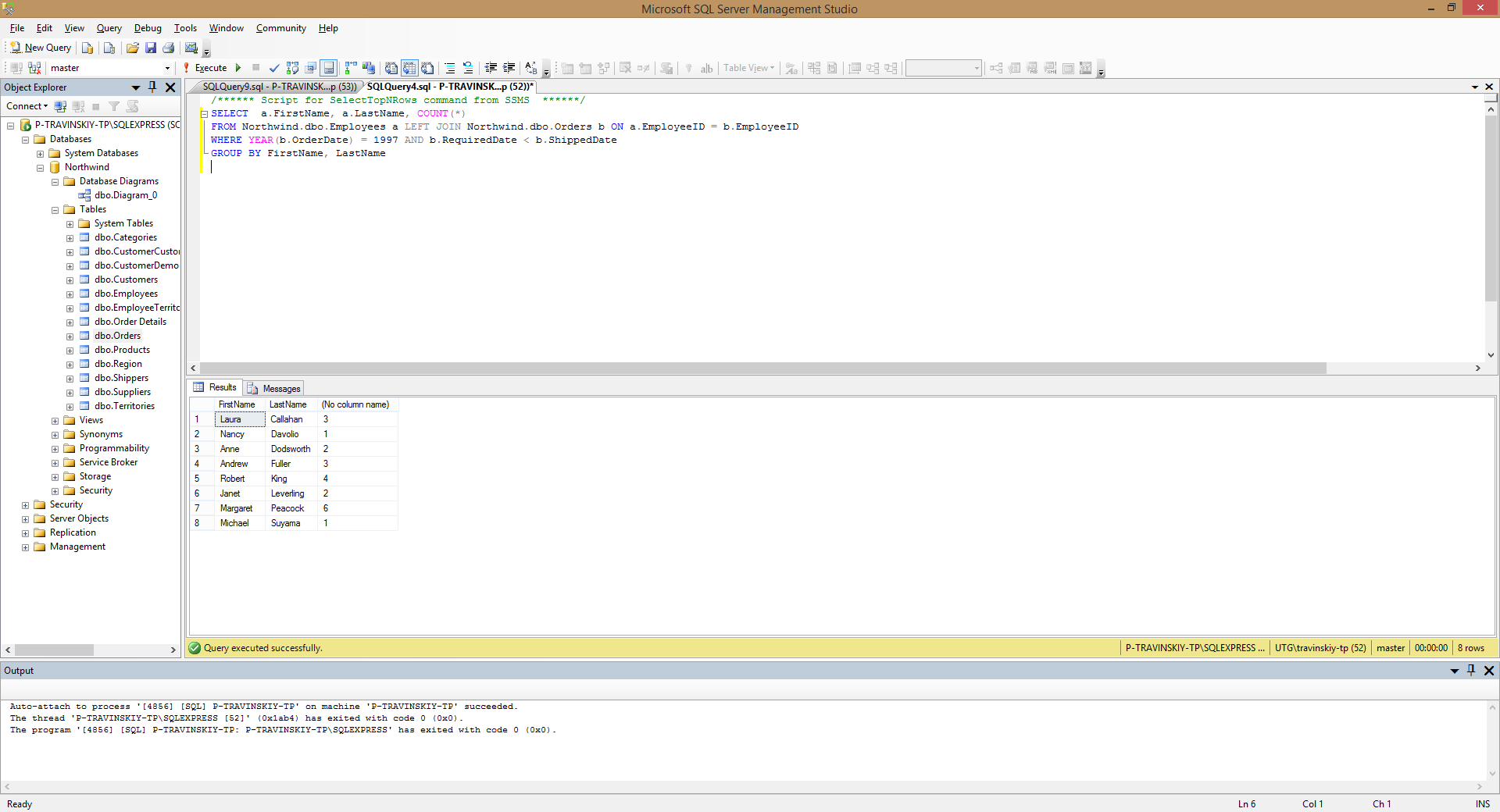
1. Show first and last names of the employees as well as the count of their orders shipped after required date during the year 1997 (use left join).

SELECT a.FirstName, a.LastName, COUNT(\*)

FROM Northwind.dbo.Employees a LEFT JOIN Northwind.dbo.Orders b ON a.EmployeeID = b.EmployeeID

WHERE YEAR(b.OrderDate) = 1997 AND b.RequiredDate < b.ShippedDate

GROUP BY FirstName, LastName



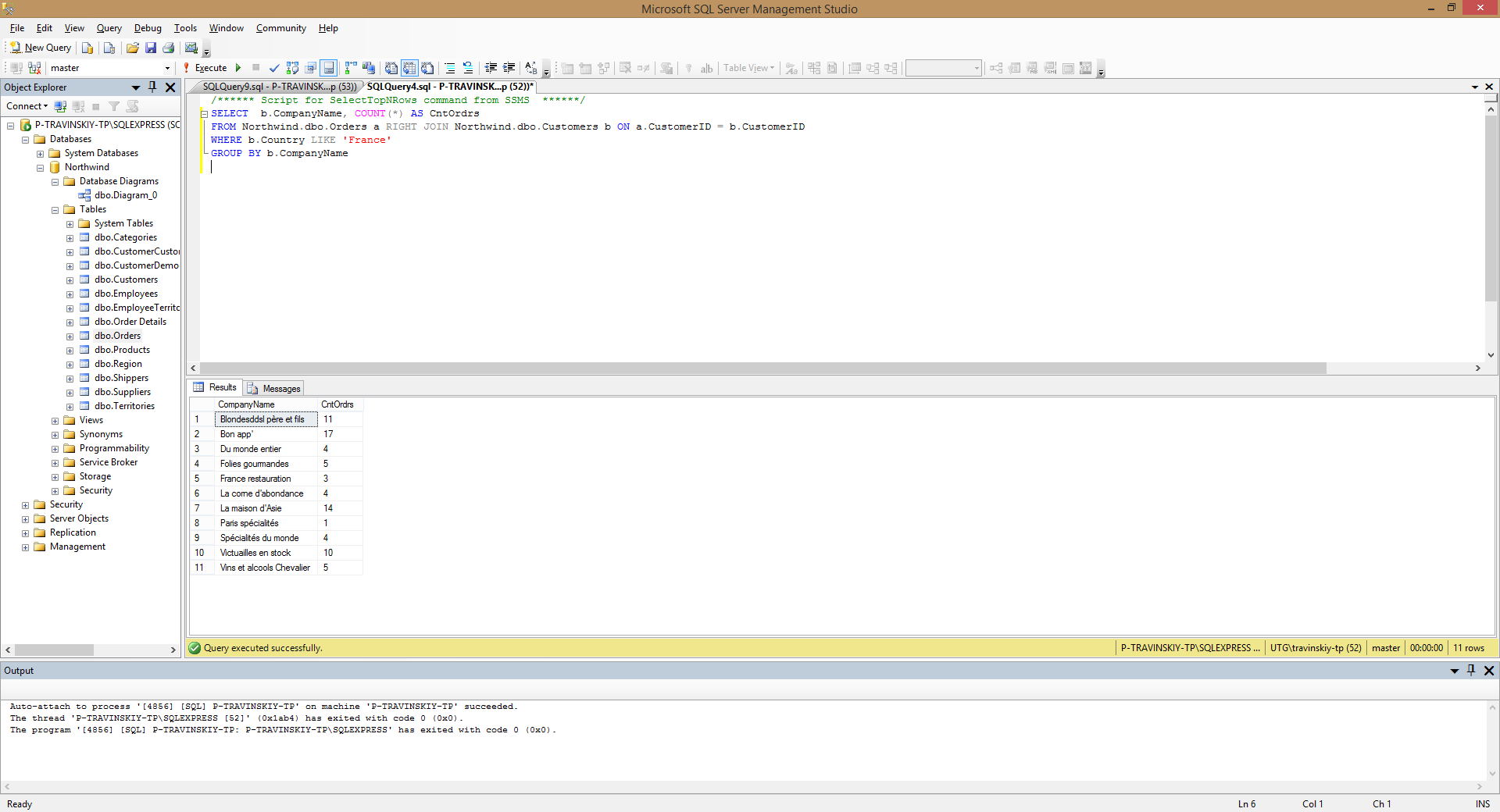
1. Show the count of orders made by each customer from France.

SELECT b.CompanyName, COUNT(\*) AS CntOrdrs

FROM Northwind.dbo.Orders a RIGHT JOIN Northwind.dbo.Customers b ON a.CustomerID = b.CustomerID

WHERE b.Country LIKE 'France'

GROUP BY b.CompanyName



1. Show the list of french customers’ names who have made more than one order (use grouping).

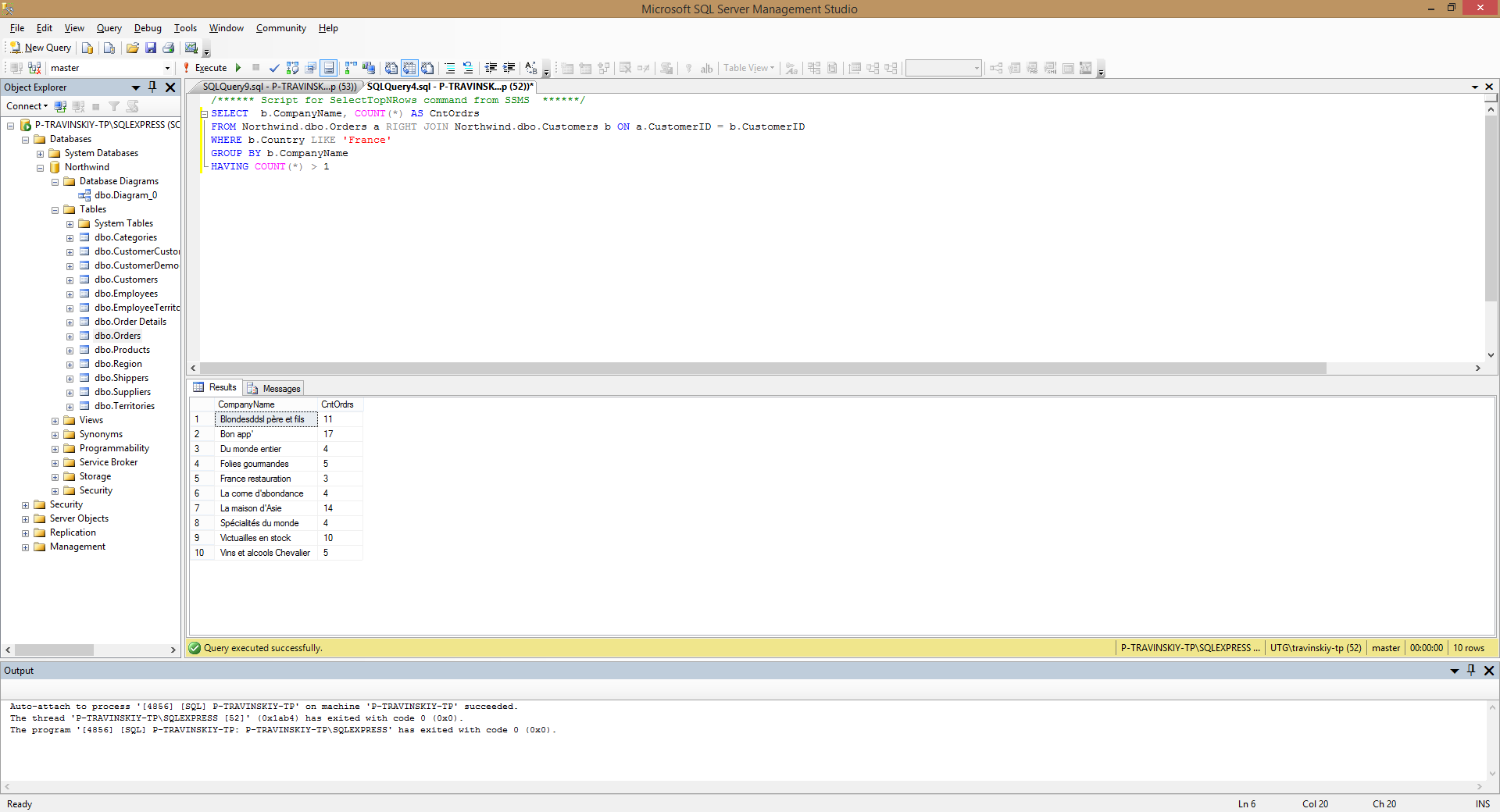
SELECT b.CompanyName, COUNT(\*) AS CntOrdrs

FROM Northwind.dbo.Orders a RIGHT JOIN Northwind.dbo.Customers b ON a.CustomerID = b.CustomerID

WHERE b.Country LIKE 'France'

GROUP BY b.CompanyName

HAVING COUNT(\*) > 1



1. Show the list of french customers’ names who have made more than one order (use a subquery).

SELECT a.CompanyName, (SELECT COUNT(b.CustomerID)

FROM Northwind.dbo.Orders b

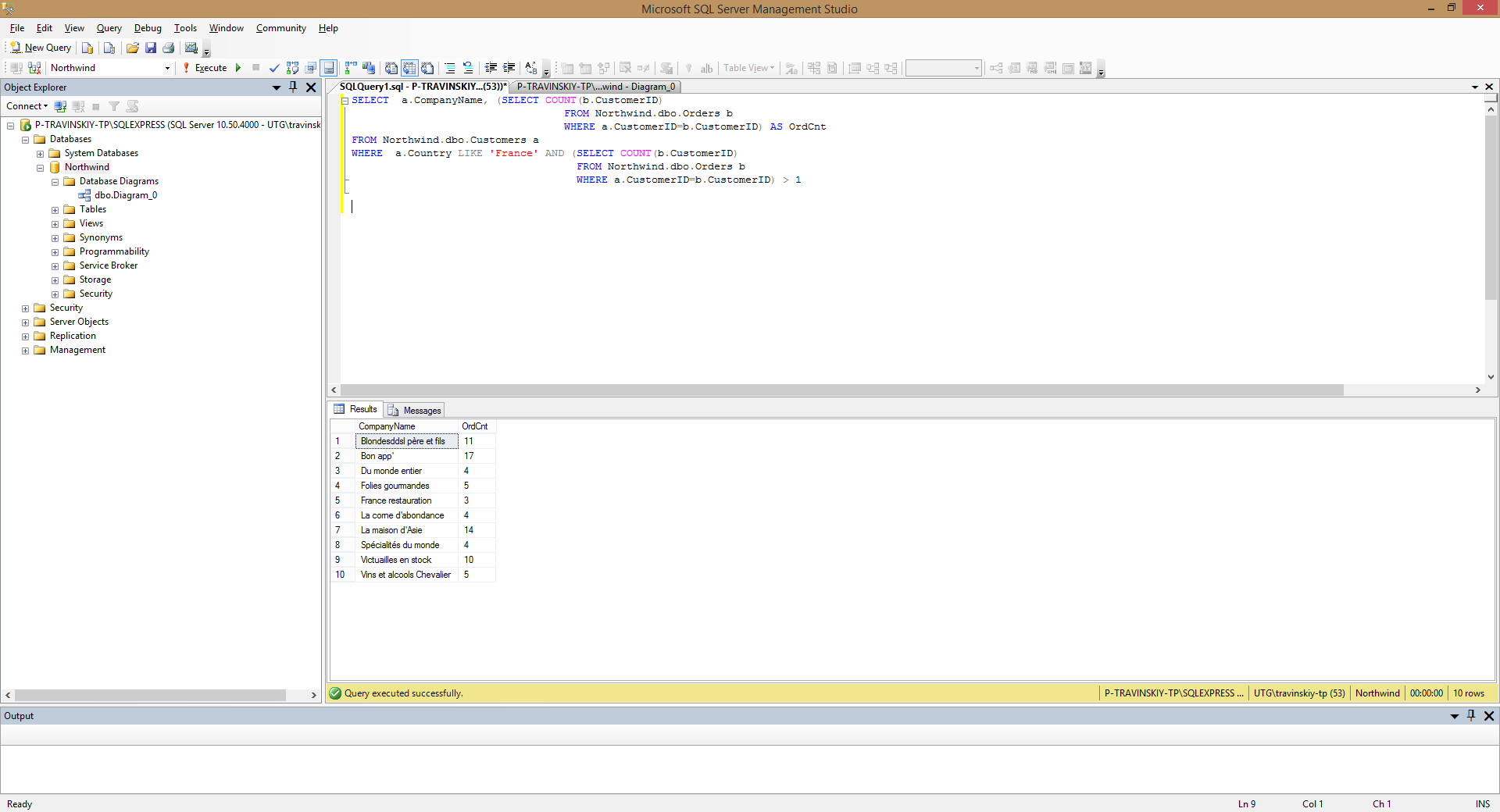
WHERE a.CustomerID=b.CustomerID) AS OrdCnt

FROM Northwind.dbo.Customers a

WHERE a.Country LIKE 'France' AND (SELECT COUNT(b.CustomerID)

FROM Northwind.dbo.Orders b

WHERE a.CustomerID=b.CustomerID) > 1



1. Show the list of customers’ names who used to order the ‘Tofu’ product (use a subquery).

SELECT DISTINCT CompanyName

FROM Northwind.dbo.Customers

WHERE Country LIKE 'France' AND CustomerID IN

(SELECT CustomerID

FROM Northwind.dbo.Orders

WHERE OrderID IN

(SELECT OrderID

FROM Northwind.dbo.[Order Details]

WHERE ProductID IN

(SELECT ProductID

FROM Northwind.dbo.Products

WHERE ProductName LIKE 'Tofu')

)

)

GROUP BY CompanyName

Or else

SELECT DISTINCT a.CompanyName

FROM Northwind.dbo.Customers a,

Northwind.dbo.Orders b,

Northwind.dbo.[Order Details] c,

Northwind.dbo.Products d

WHERE a.Country LIKE 'France'

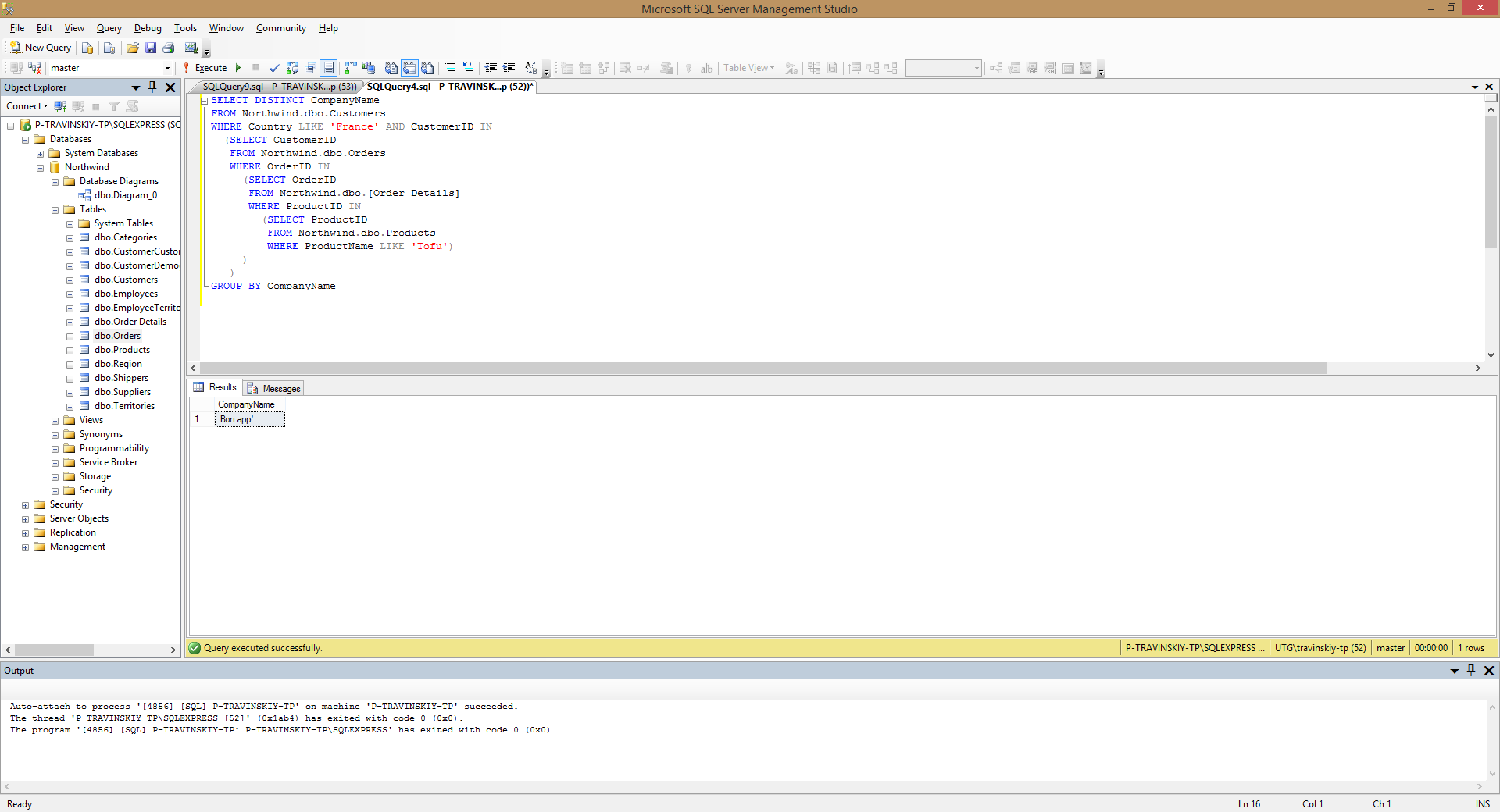
AND a.CustomerID=b.CustomerID

AND b.OrderID = c.OrderID

AND c.ProductID = d.ProductID

AND d.ProductName LIKE 'Tofu'

GROUP BY CompanyName



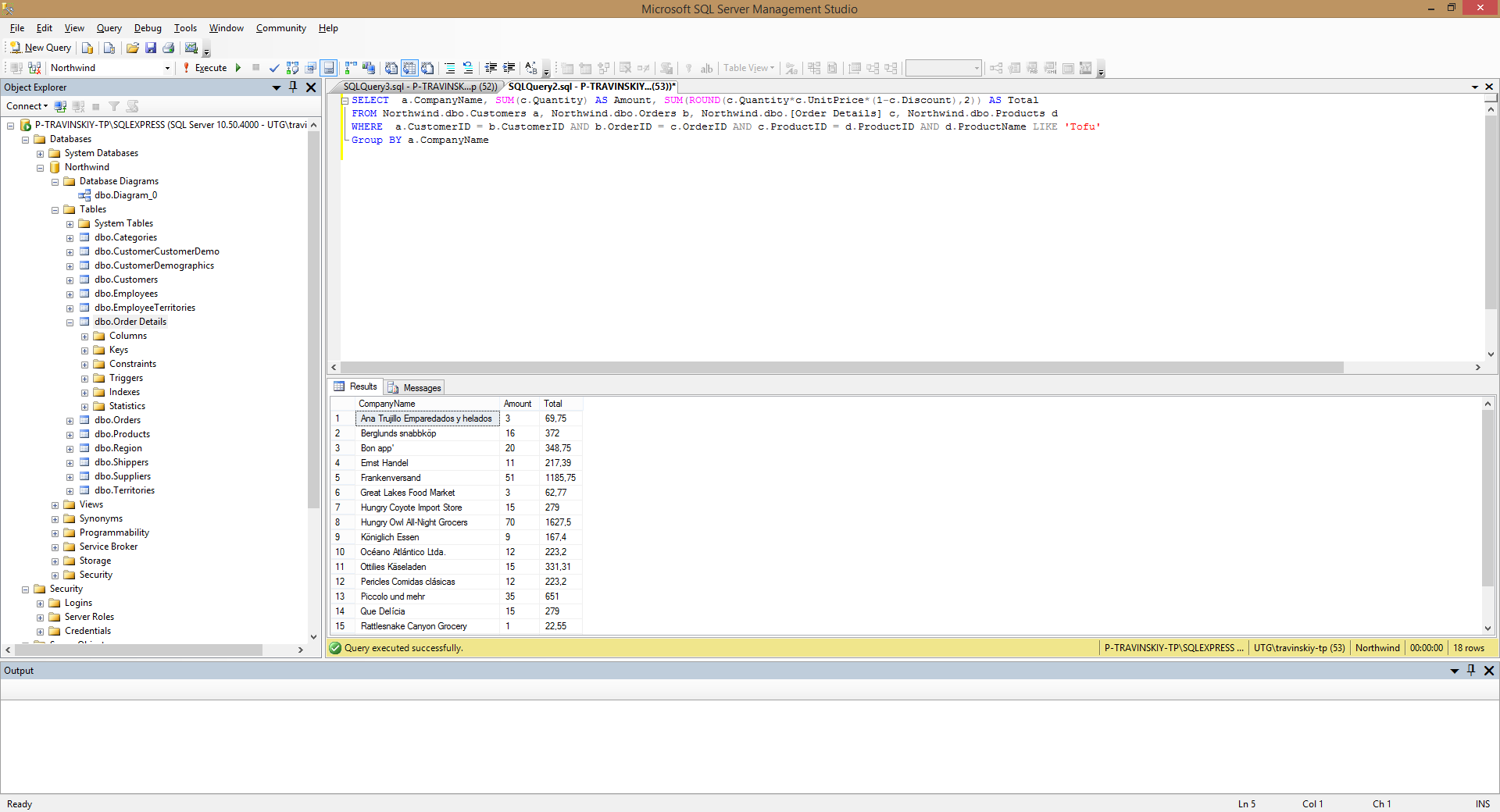
1. \*Show the list of customers’ names who used to order the ‘Tofu’ product, along with the total amount of the product they have ordered and with the total sum for ordered product calculated.

SELECT a.CompanyName, SUM(c.Quantity) AS Amount, SUM(ROUND(c.Quantity\*c.UnitPrice\*(1-c.Discount),2)) AS Total

FROM Northwind.dbo.Customers a, Northwind.dbo.Orders b, Northwind.dbo.[Order Details] c, Northwind.dbo.Products d

WHERE a.CustomerID = b.CustomerID AND b.OrderID = c.OrderID AND c.ProductID = d.ProductID AND d.ProductName LIKE 'Tofu'

Group BY a.CompanyName



1. \*Show the list of french customers’ names who used to order non-french products (use left join).

SELECT a.CompanyName

FROM Northwind.dbo.Customers a

LEFT JOIN Northwind.dbo.Orders b ON a.CustomerID = b.CustomerID

LEFT JOIN Northwind.dbo.[Order Details] c ON b.OrderID = c.OrderID

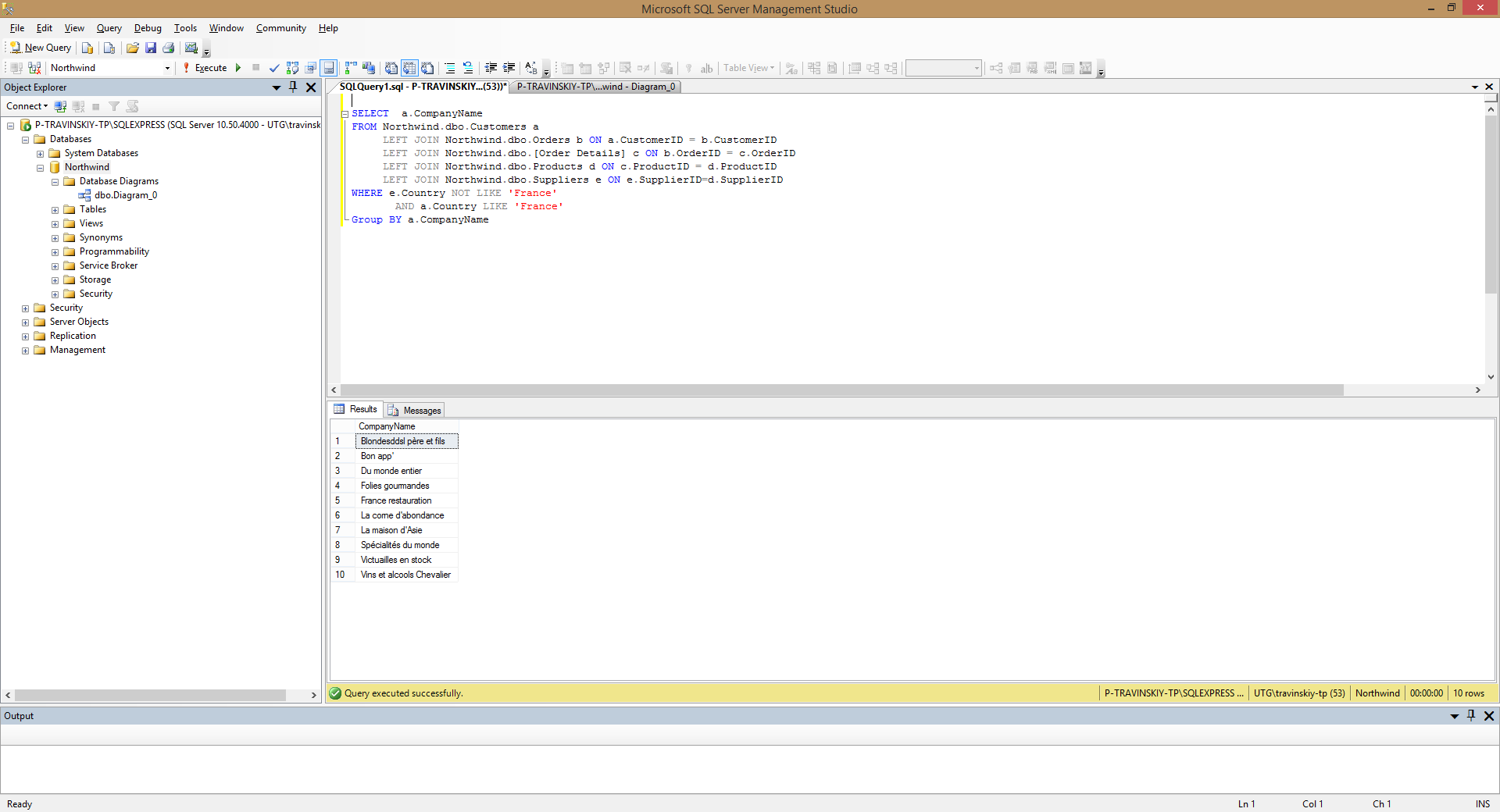
LEFT JOIN Northwind.dbo.Products d ON c.ProductID = d.ProductID

LEFT JOIN Northwind.dbo.Suppliers e ON e.SupplierID=d.SupplierID

WHERE e.Country NOT LIKE 'France'

AND a.Country LIKE 'France'

Group BY a.CompanyName



1. \*Show the list of french customers’ names who used to order non-french products (use a subquery).

SELECT a.CompanyName

FROM Northwind.dbo.Customers a, Northwind.dbo.Orders b, Northwind.dbo.[Order Details] c, Northwind.dbo.Products d

WHERE a.CustomerID = b.CustomerID AND b.OrderID = c.OrderID AND c.ProductID = d.ProductID AND d.SupplierID IN

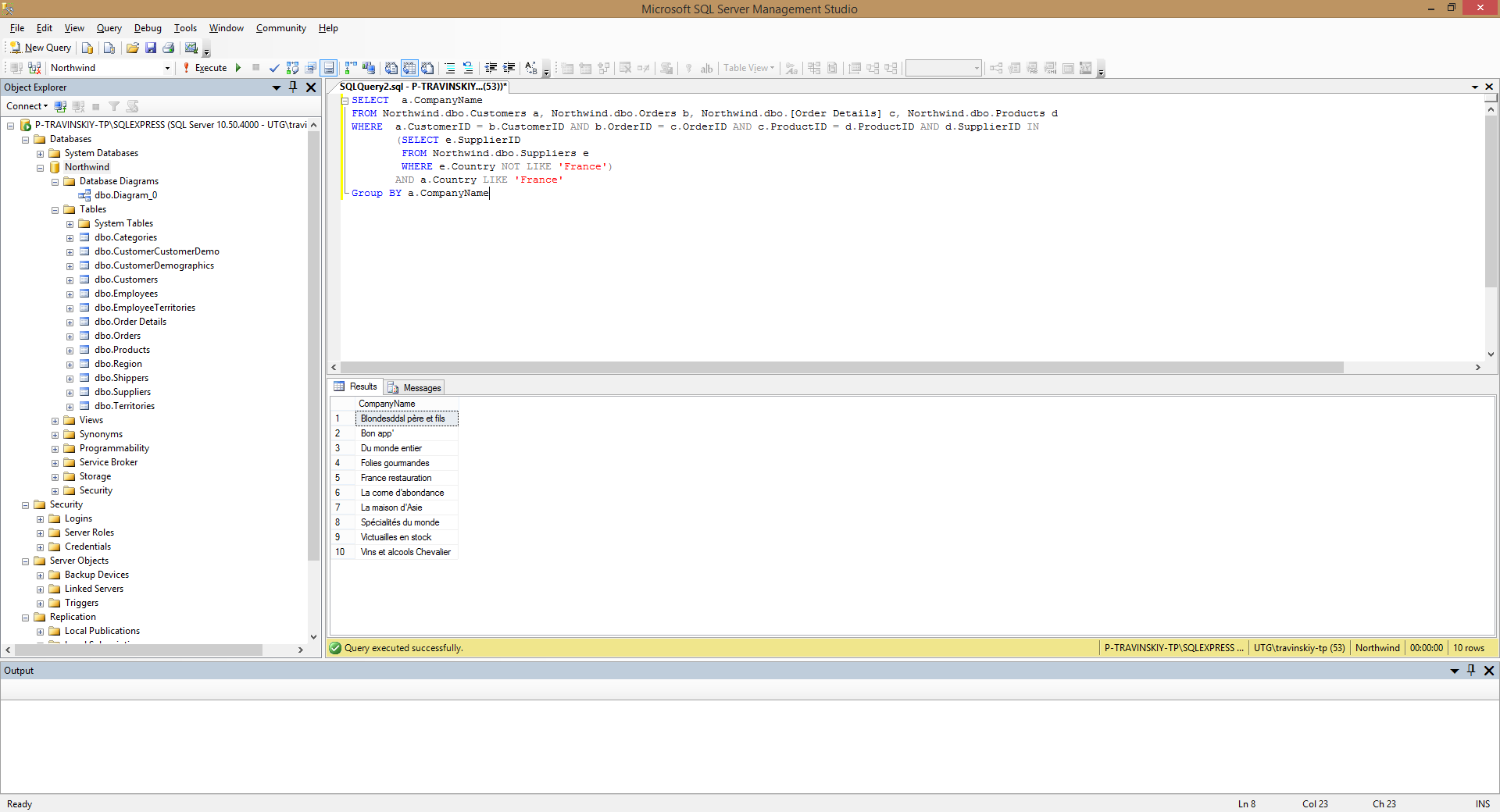
(SELECT e.SupplierID

FROM Northwind.dbo.Suppliers e

WHERE e.Country NOT LIKE 'France')

AND a.Country LIKE 'France'

Group BY a.CompanyName



1. \*Show the list of french customers’ names who used to order french products.

SELECT a.CompanyName

FROM Northwind.dbo.Customers a, Northwind.dbo.Orders b, Northwind.dbo.[Order Details] c, Northwind.dbo.Products d

WHERE a.CustomerID = b.CustomerID AND b.OrderID = c.OrderID AND c.ProductID = d.ProductID AND d.SupplierID IN

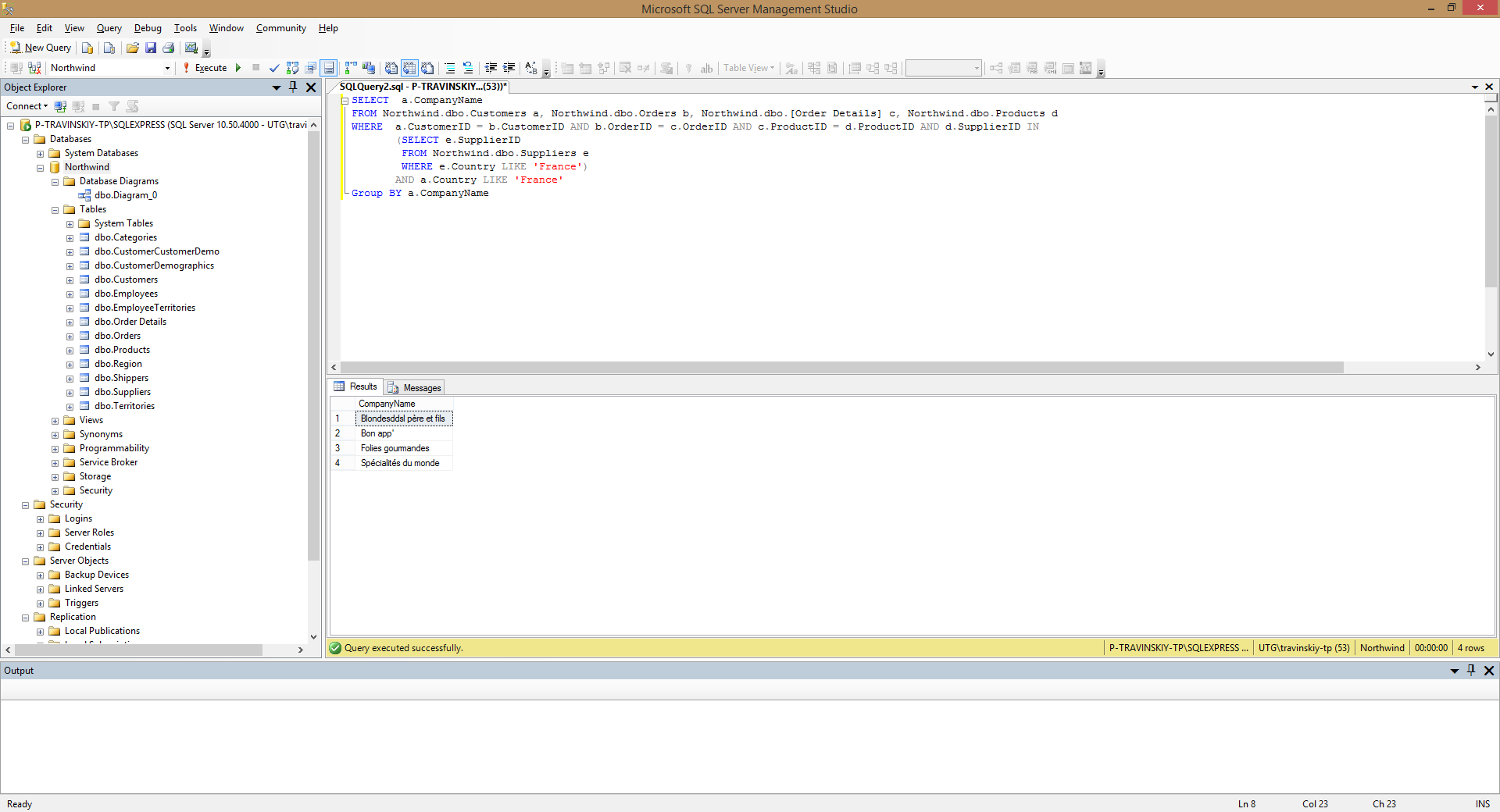
(SELECT e.SupplierID

FROM Northwind.dbo.Suppliers e

WHERE e.Country LIKE 'France')

AND a.Country LIKE 'France'

Group BY a.CompanyName



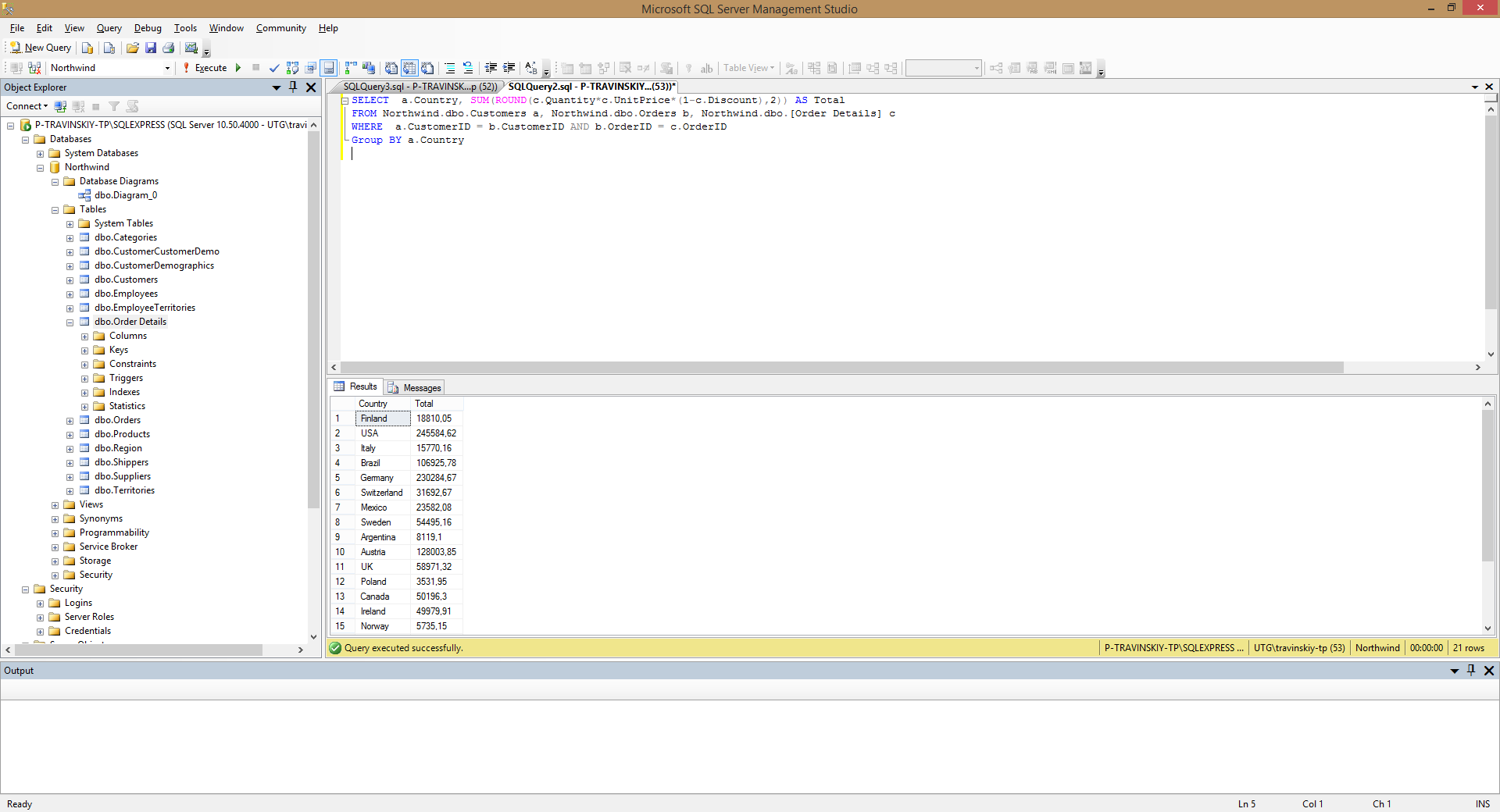
1. \*Show the total ordering sum calculated for each country of customer.

SELECT a.Country, SUM(ROUND(c.Quantity\*c.UnitPrice\*(1-c.Discount),2)) AS Total

FROM Northwind.dbo.Customers a, Northwind.dbo.Orders b, Northwind.dbo.[Order Details] c

WHERE a.CustomerID = b.CustomerID AND b.OrderID = c.OrderID

Group BY a.Country



1. \*Show the total ordering sums calculated for each customer’s country for domestic and non-domestic products separately (e.g.: “France – French products ordered – Non-french products ordered” and so on for each country).

SELECT f.Country, SUM(ROUND(f.DomTotal,2)) AS DomTotal, SUM(ROUND(f.NonDomTotal,2)) AS NonDomTotal

FROM (SELECT a.Country AS Country,

(SELECT SUM(ROUND(c.Quantity\*c.UnitPrice\*(1-c.Discount),2))

FROM Northwind.dbo.Orders b, Northwind.dbo.[Order Details] c, Northwind.dbo.Products d, Northwind.dbo.Suppliers e

WHERE d.SupplierID=e.SupplierID AND a.CustomerID = b.CustomerID AND b.OrderID = c.OrderID AND e.Country LIKE a.Country) AS DomTotal,

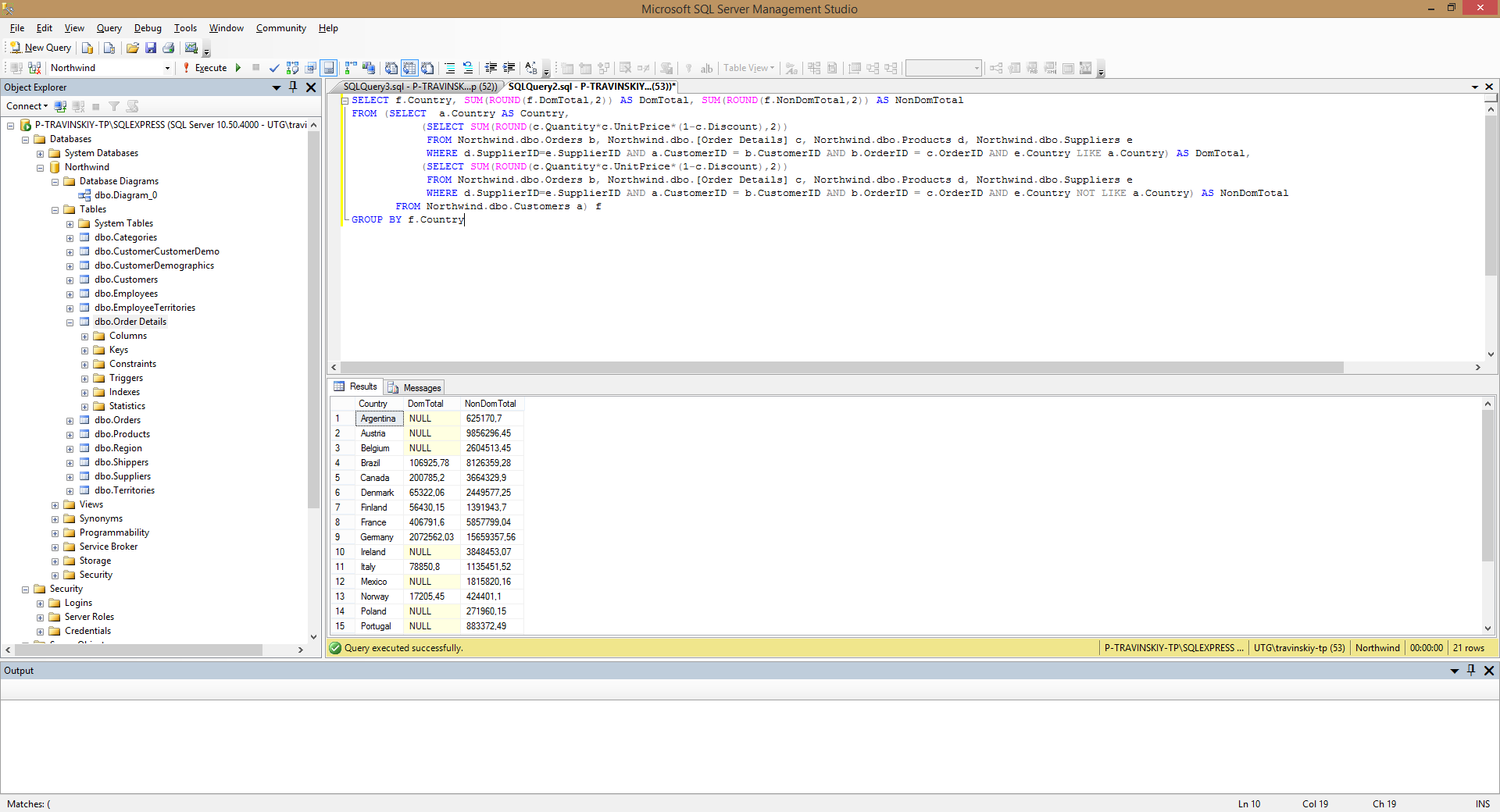
(SELECT SUM(ROUND(c.Quantity\*c.UnitPrice\*(1-c.Discount),2))

FROM Northwind.dbo.Orders b, Northwind.dbo.[Order Details] c, Northwind.dbo.Products d, Northwind.dbo.Suppliers e

WHERE d.SupplierID=e.SupplierID AND a.CustomerID = b.CustomerID AND b.OrderID = c.OrderID AND e.Country NOT LIKE a.Country) AS NonDomTotal

FROM Northwind.dbo.Customers a) f

GROUP BY f.Country



1. \*Show the list of product categories along with total ordering sums calculated for the orders made for the products of each category, during the year 1997.

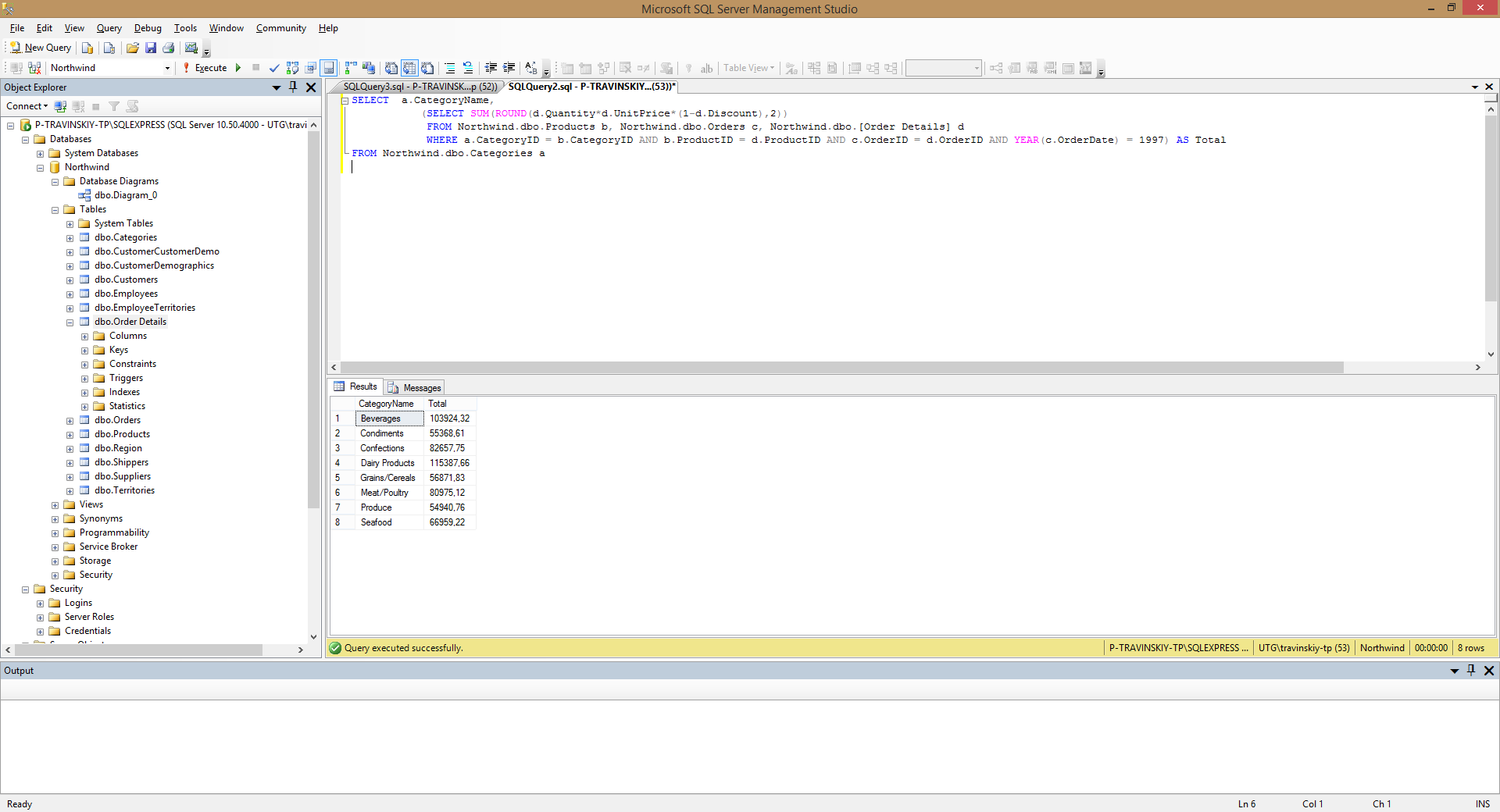
SELECT a.CategoryName,

(SELECT SUM(ROUND(d.Quantity\*d.UnitPrice\*(1-d.Discount),2))

FROM Northwind.dbo.Products b, Northwind.dbo.Orders c, Northwind.dbo.[Order Details] d

WHERE a.CategoryID = b.CategoryID AND b.ProductID = d.ProductID AND c.OrderID = d.OrderID AND YEAR(c.OrderDate) = 1997) AS Total

FROM Northwind.dbo.Categories a



1. \*Show the list of product names along with unit prices and the history of unit prices taken from the orders (show ‘Product name – Unit price – Historical price’). The duplicate records should be eliminated. If no orders were made for a certain product, then the result for this product should look like ‘Product name – Unit price – NULL’. Sort the list by the product name.

&&&&&&&&&&&&&&&&CLARIFY

SELECT a.ProductName,

f.UnitPrice

FROM Northwind.dbo.Products a LEFT JOIN (SELECT d.ProductID,d.UnitPrice

FROM Northwind.dbo.Orders c,

Northwind.dbo.[Order Details] d

WHERE c.OrderID = d.OrderID ) f

ON f.ProductID=a.ProductID

WHERE a.ProductID IN

(SELECT b1.ProductID

FROM Northwind.dbo.Orders a1,

Northwind.dbo.[Order Details] b1

WHERE a1.OrderID = b1.OrderID

)

GROUP BY a.ProductName,

f.UnitPrice

1. \*Show the list of employees’ names along with names of their chiefs (use left join with the same table).

SELECT a.FirstName,

a.LastName,

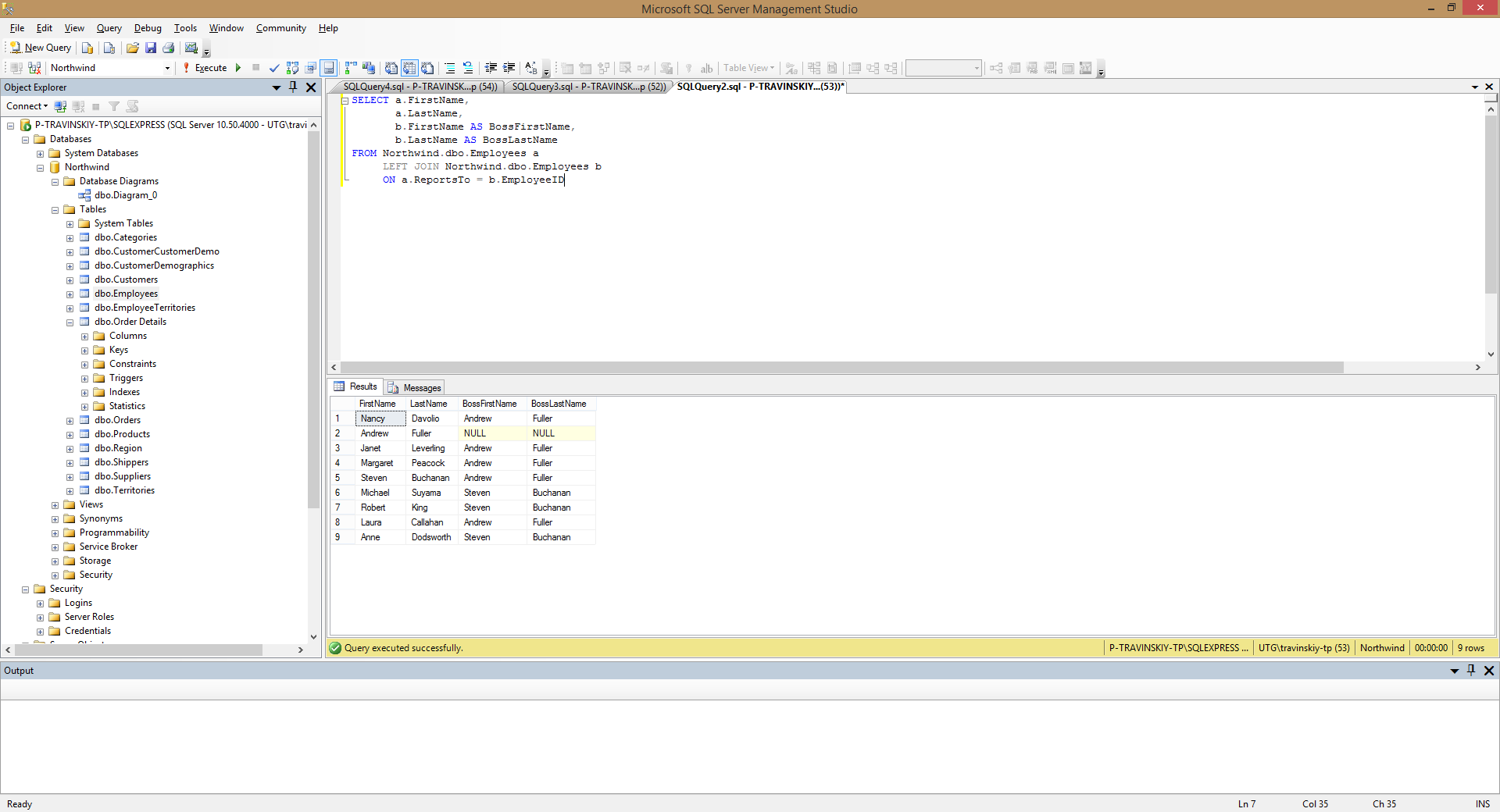
b.FirstName AS BossFirstName,

b.LastName AS BossLastName

FROM Northwind.dbo.Employees a

LEFT JOIN Northwind.dbo.Employees b

ON a.ReportsTo = b.EmployeeID



1. \*Show the list of cities where employees and customers are from and where orders have been made to. Duplicates should be eliminated.

DROP TABLE cities

CREATE TABLE cities

(City varchar(255))

INSERT INTO cities SELECT DISTINCT a.City

FROM Northwind.dbo.Employees a

INSERT INTO cities SELECT DISTINCT a.City

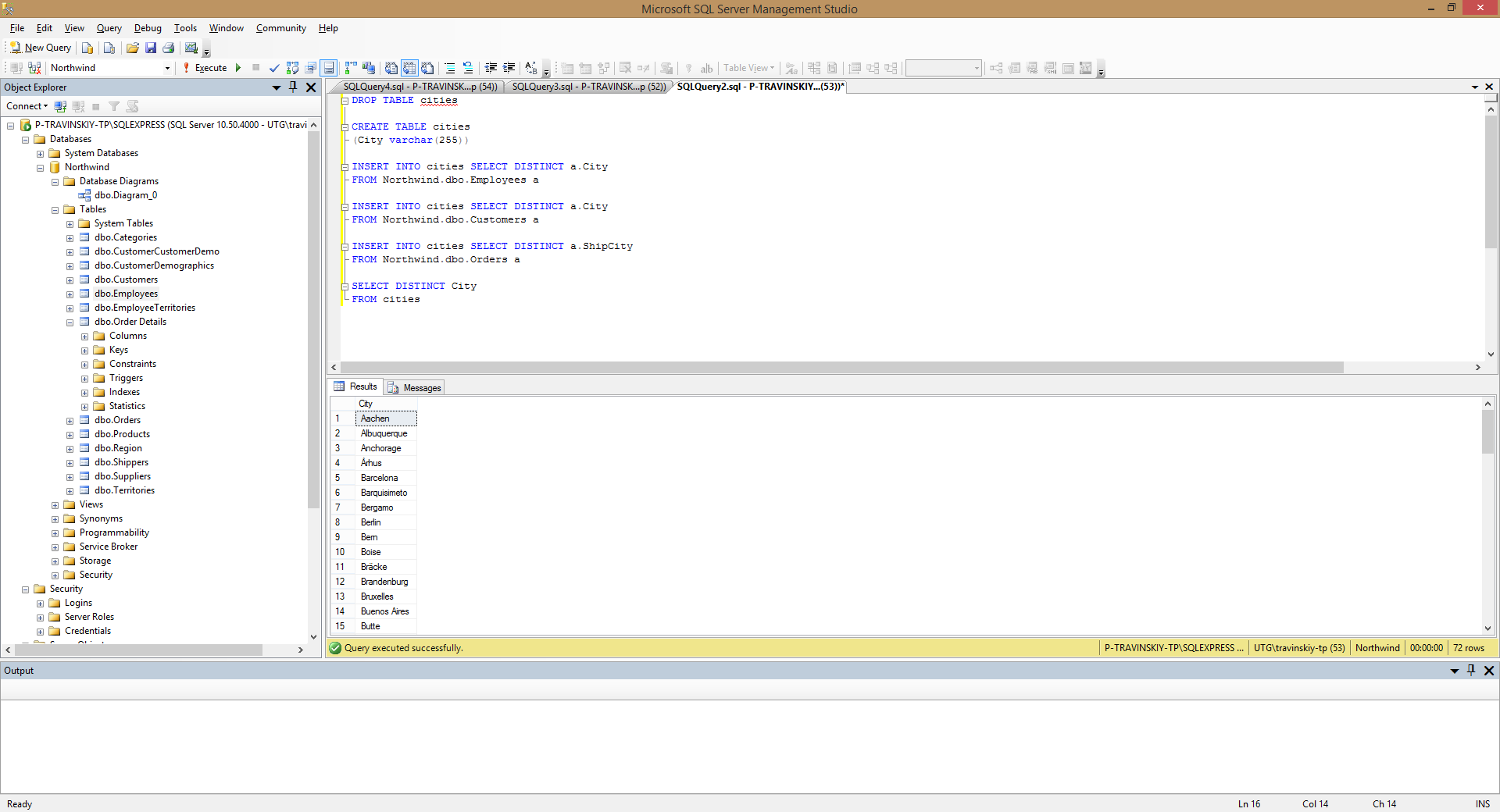
FROM Northwind.dbo.Customers a

INSERT INTO cities SELECT DISTINCT a.ShipCity

FROM Northwind.dbo.Orders a

SELECT DISTINCT City

FROM cities



1. \*Insert 5 new records into Employees table. Fill in the following fields: LastName, FirstName, BirthDate, HireDate, Address, City, Country, Notes. The Notes field should contain your own name (to distinguish your records from the ones inserted by other trainees).

INSERT INTO Northwind.dbo.Employees

(LastName, FirstName, BirthDate, HireDate, Address, City, Country, Notes)

VALUES ('LastName1', 'FirstName1', '1980-01-01', '2001-01-01', 'Address1', 'City1', 'Country1', 'Taras')

INSERT INTO Northwind.dbo.Employees

(LastName, FirstName, BirthDate, HireDate, Address, City, Country, Notes)

VALUES ('LastName2', 'FirstName2', '1980-01-01', '2001-01-01', 'Address2', 'City2', 'Country2', 'Taras')

INSERT INTO Northwind.dbo.Employees

(LastName, FirstName, BirthDate, HireDate, Address, City, Country, Notes)

VALUES ('LastName3', 'FirstName3', '1980-01-01', '2001-01-01', 'Address3', 'City3', 'Country3', 'Taras')

INSERT INTO Northwind.dbo.Employees

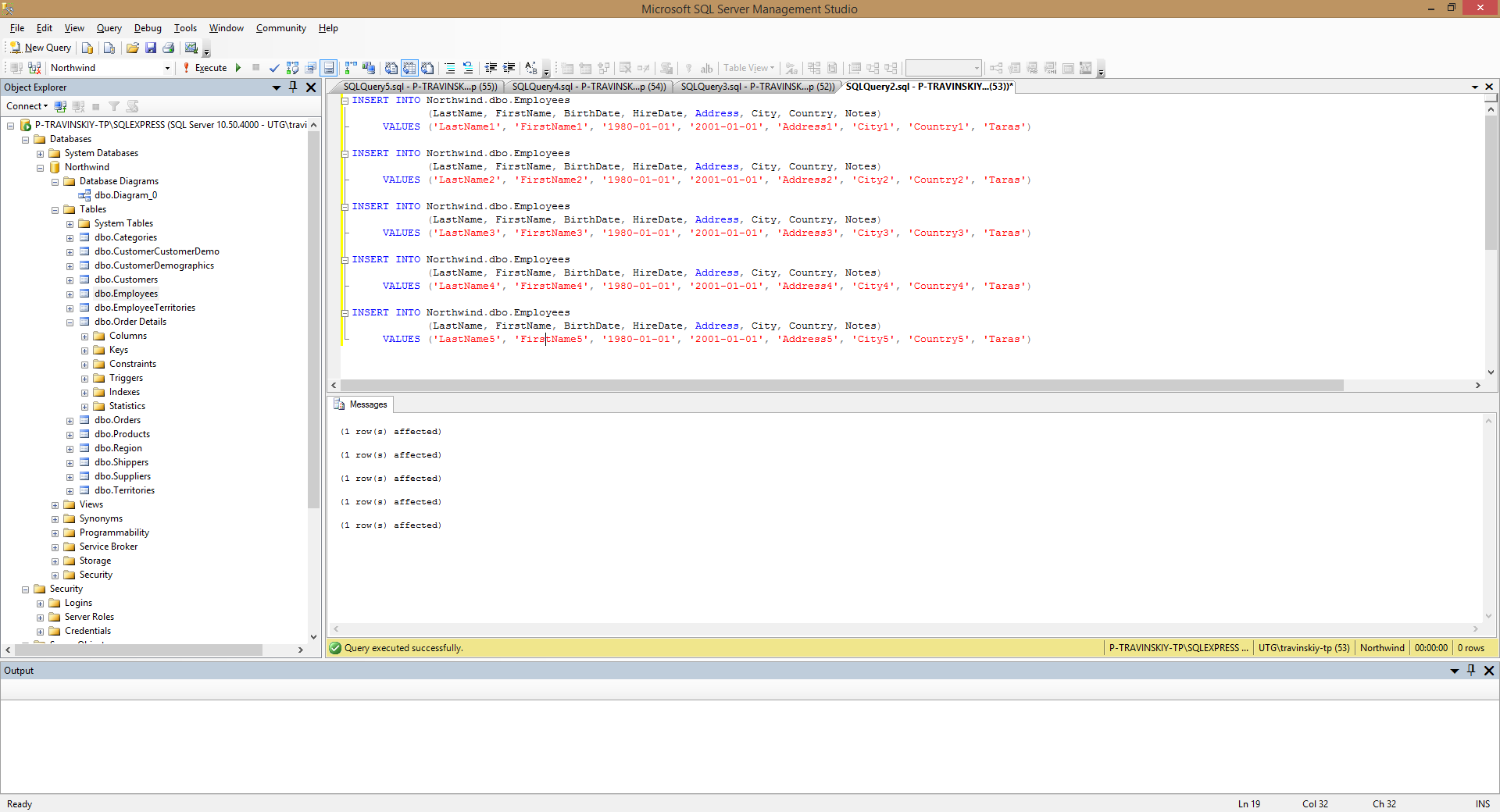
(LastName, FirstName, BirthDate, HireDate, Address, City, Country, Notes)

VALUES ('LastName4', 'FirstName4', '1980-01-01', '2001-01-01', 'Address4', 'City4', 'Country4', 'Taras')

INSERT INTO Northwind.dbo.Employees

(LastName, FirstName, BirthDate, HireDate, Address, City, Country, Notes)

VALUES ('LastName5', 'FirstName5', '1980-01-01', '2001-01-01', 'Address5', 'City5', 'Country5', 'Taras')

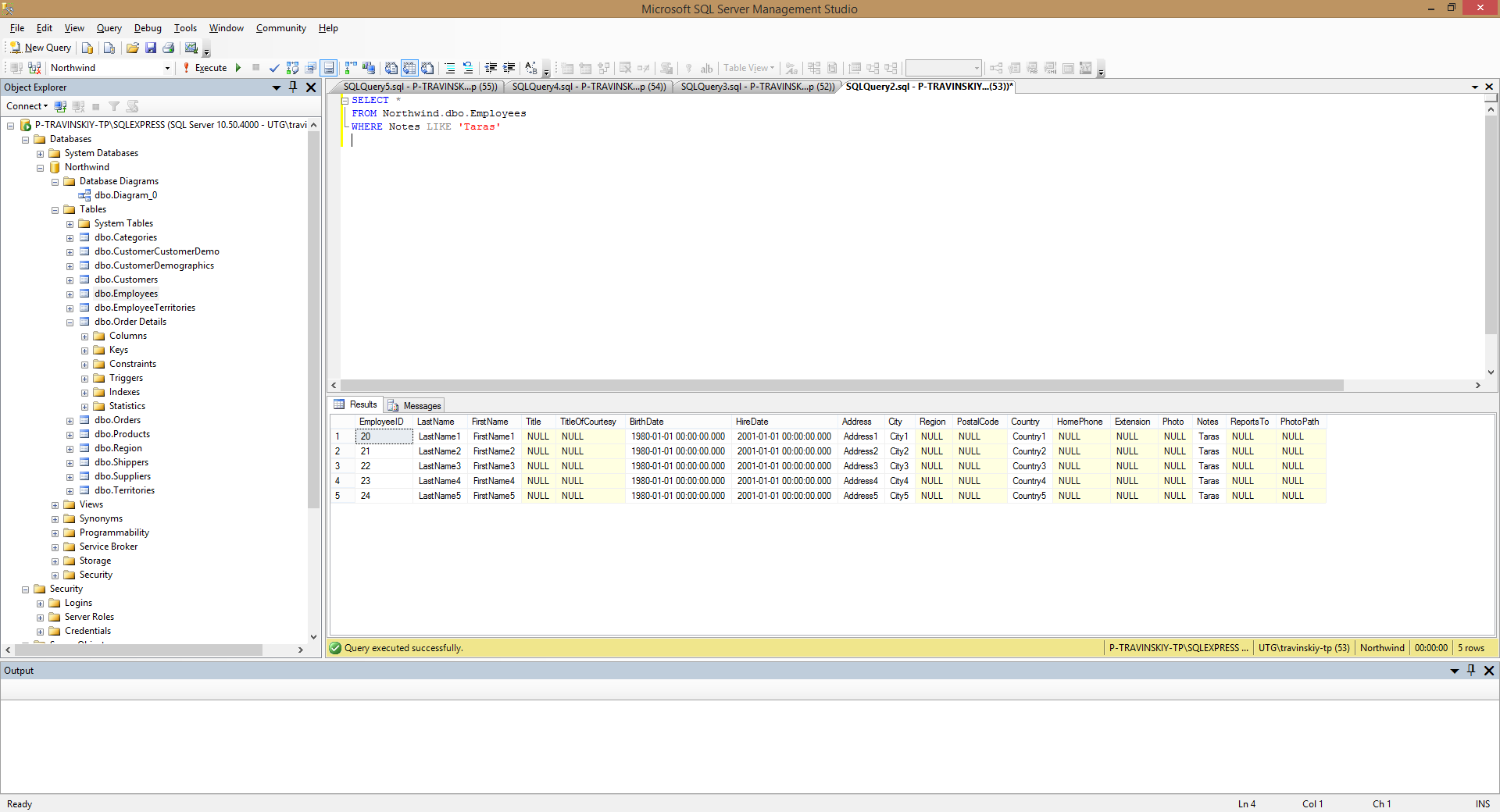


1. \*Fetch the records you have inserted by the SELECT statement

SELECT \*

FROM Northwind.dbo.Employees

WHERE Notes LIKE 'Taras'

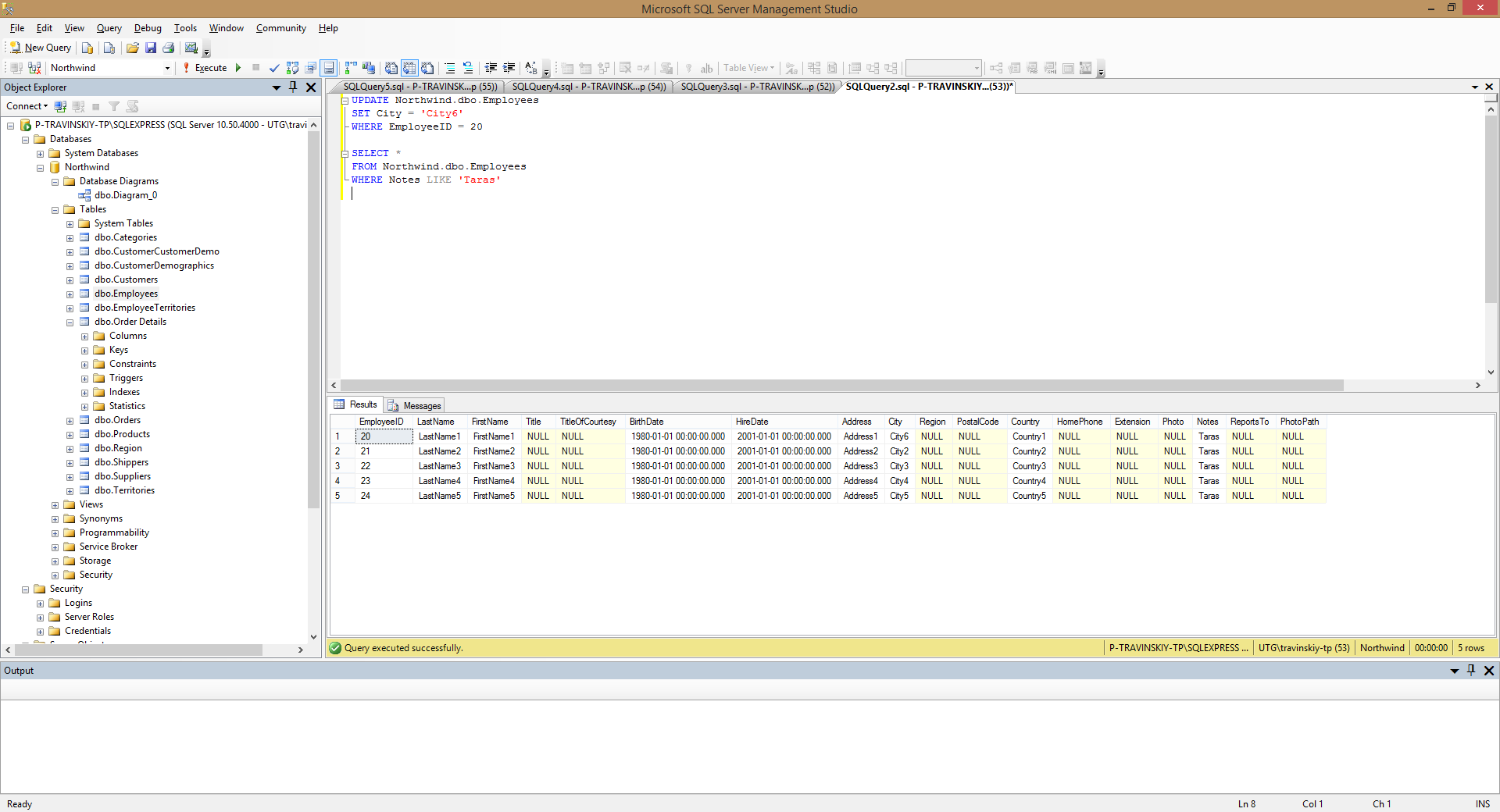


1. \*Change the City field in one of your records using the UPDATE statement (first run the SELECT statement to check whether you are updating the appropriate records!).

UPDATE Northwind.dbo.Employees

SET City = 'City6'

WHERE EmployeeID = 20

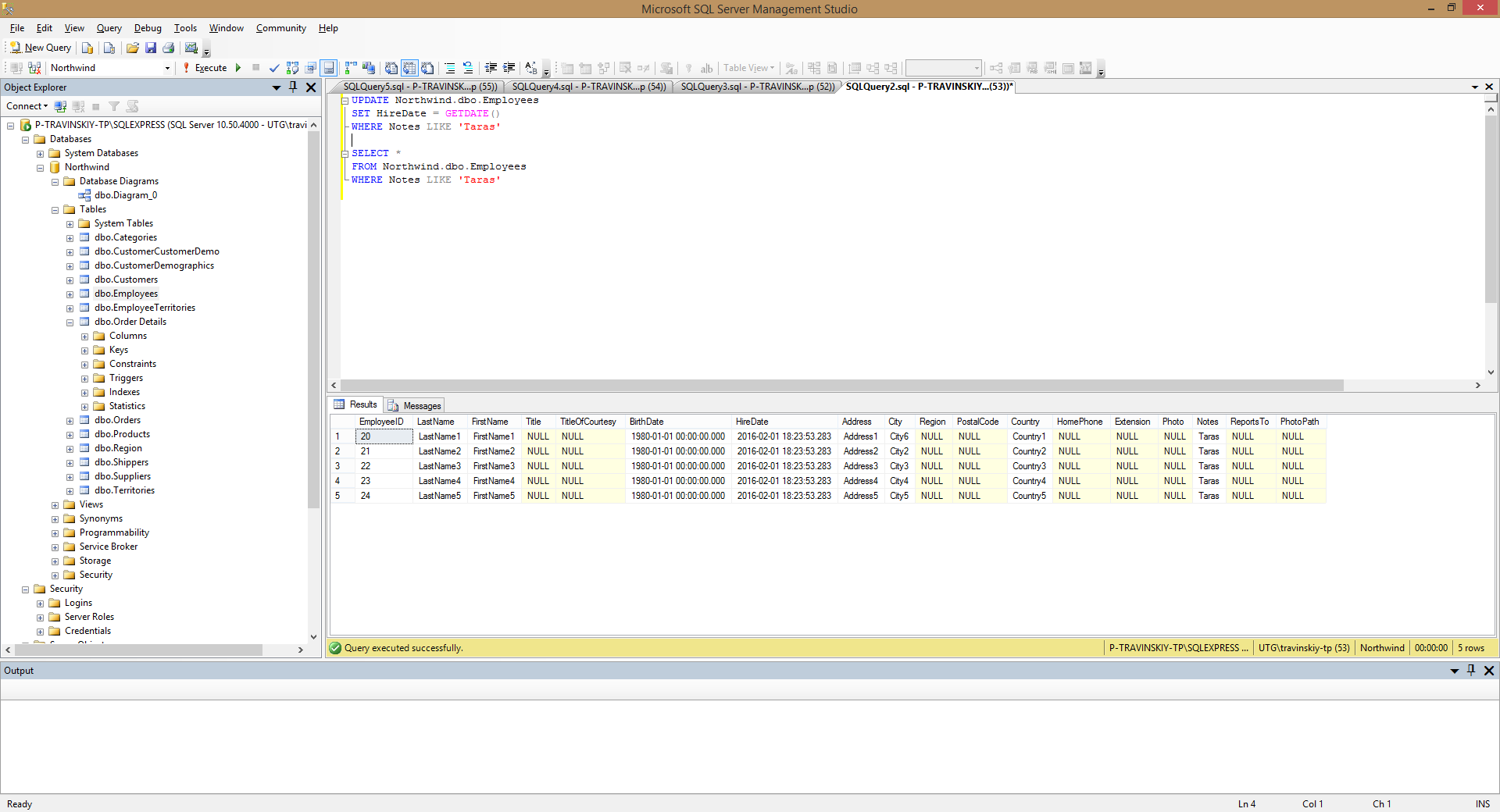


1. \*Change the HireDate field in all your records to current date (first run the SELECT statement to check whether you are updating the appropriate records!).

UPDATE Northwind.dbo.Employees

SET HireDate = GETDATE()

WHERE Notes LIKE 'Taras'

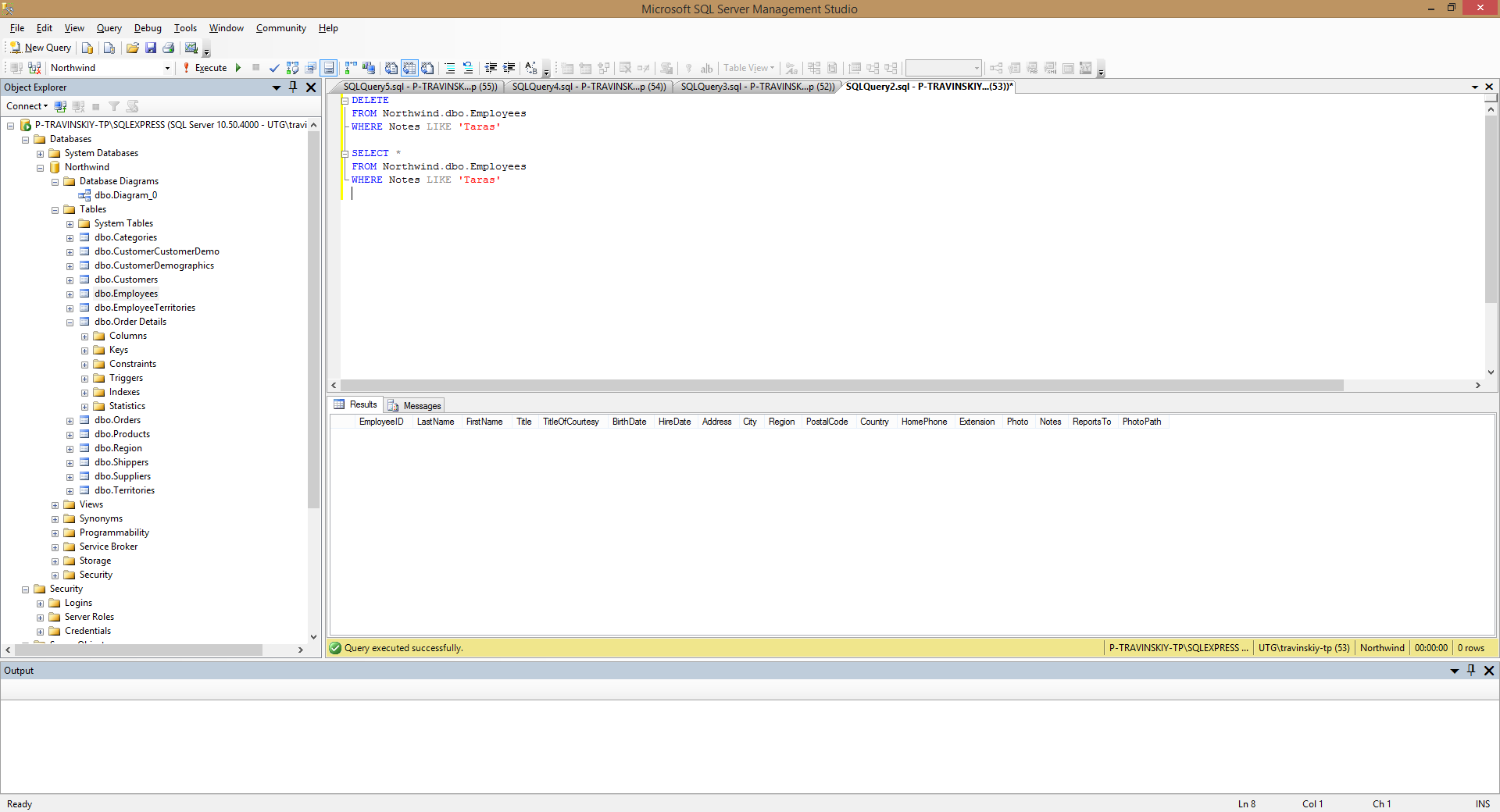


1. \*Delete one of your records (first run the SELECT statement to check whether you are deleting the appropriate record!).

DELETE

FROM Northwind.dbo.Employees

WHERE Notes LIKE 'Taras'



# Home works

1. Write SQL statements for the examples marked with (\*) above.
2. Write you own 20 examples covering all types of queries mentioned in the lecture that are cases: where, like, order by, group by, count(\*), count(<field>), sum, having, order by, inner join, left (right) join, subquery; delete, insert (+results of select), update

**Cautions**:

1. When using INSERT commands, fill some text field with your own name.
2. When using UPDATE or DELETE commands, only your records should be affected.