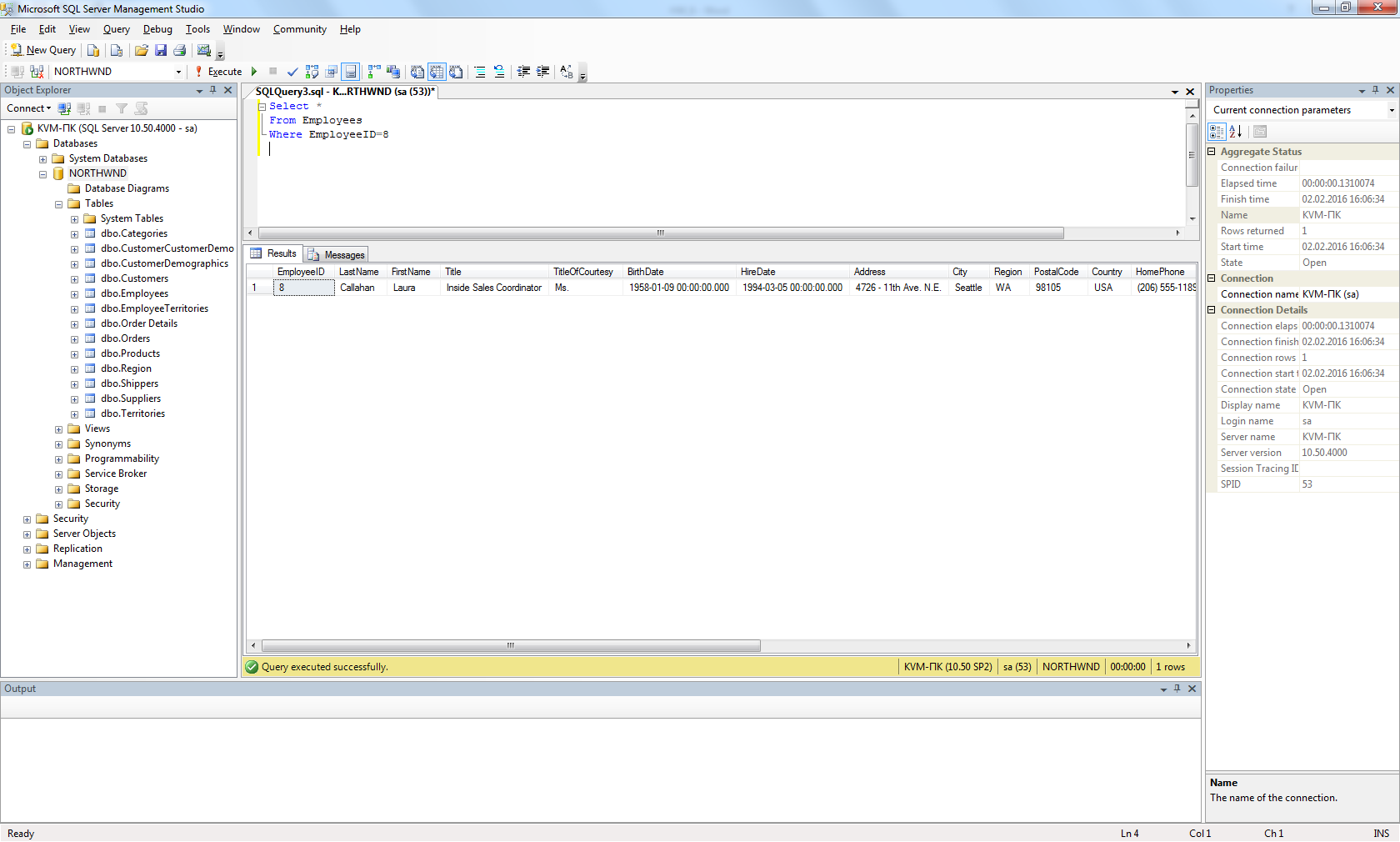
SQL Tasks

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

Select \*

From Employees

Where EmployeeID=8

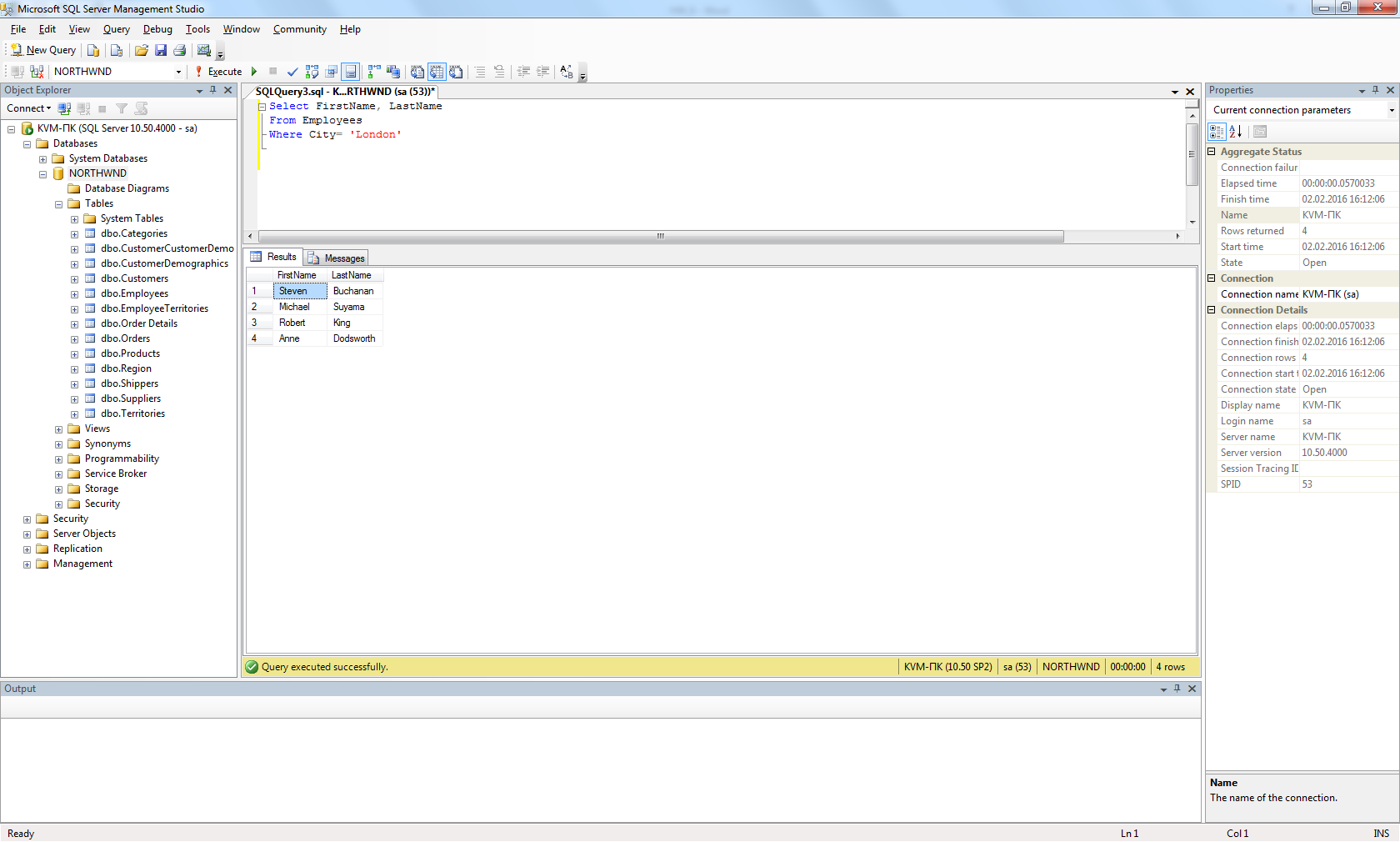


1. Show the list of first and last names of the employees from London.

Select FirstName, LastName

From Employees

Where City= 'London'

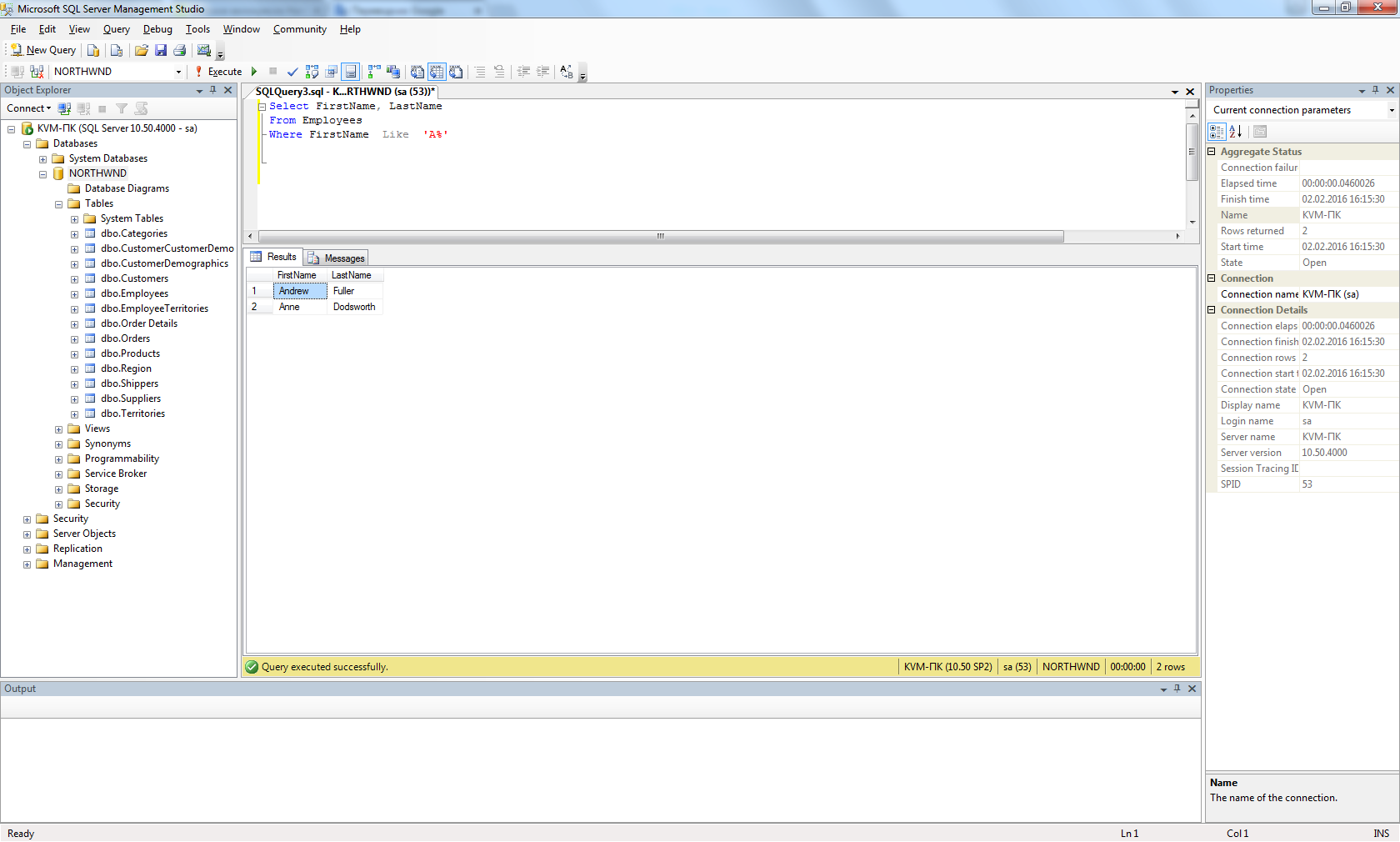


1. Show the list of first and last names of the employees whose first name begins with letter A.

Select FirstName, LastName

From Employees

Where FirstName Like 'A%'



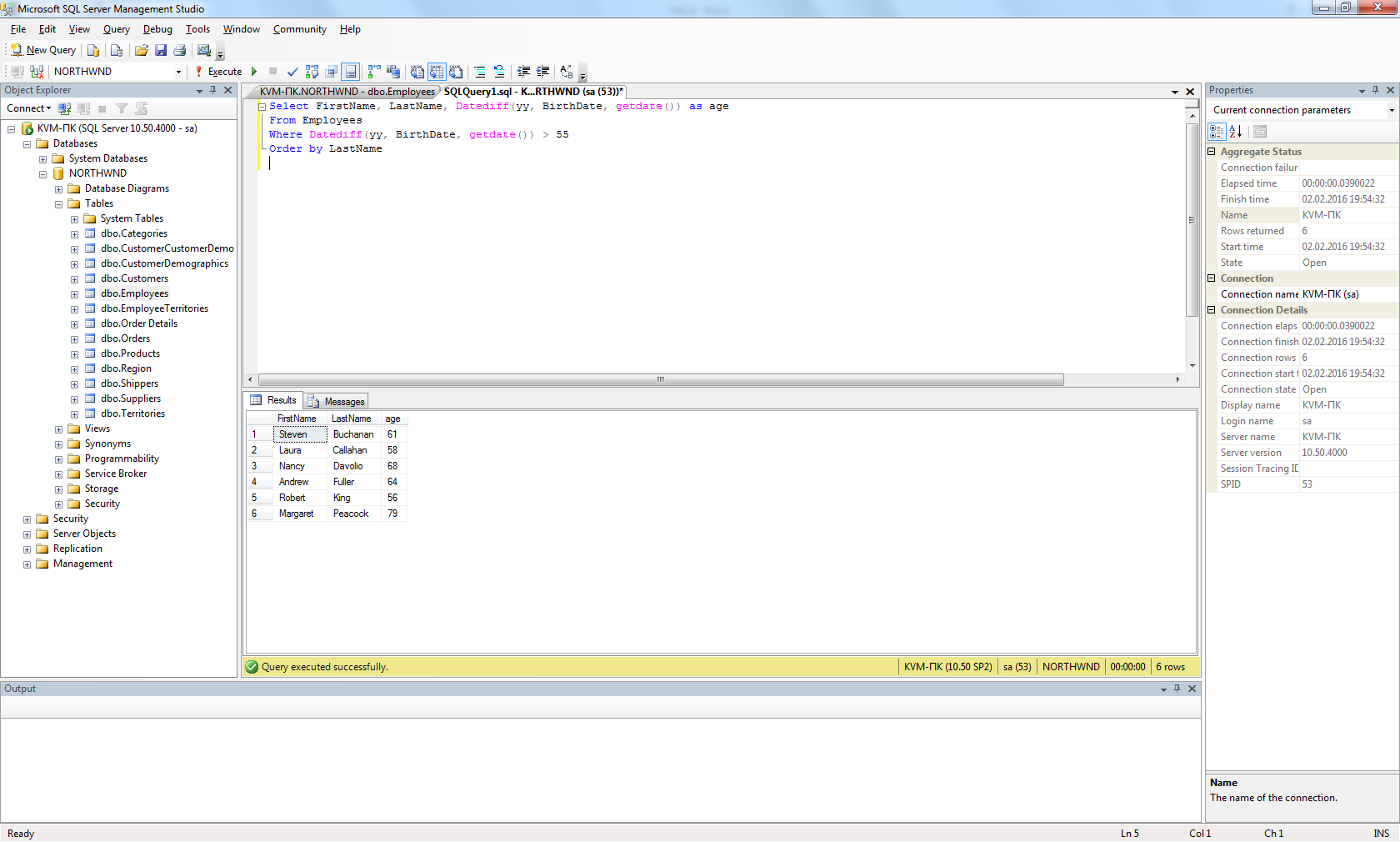
1. Show the list of first, last names and ages of the employees whose age is greater than 55. The result should be sorted by last name.

Select FirstName, LastName, Datediff(yy, BirthDate, getdate()) as age

From Employees

Where Datediff(yy, BirthDate, getdate()) > 55

Order by LastName

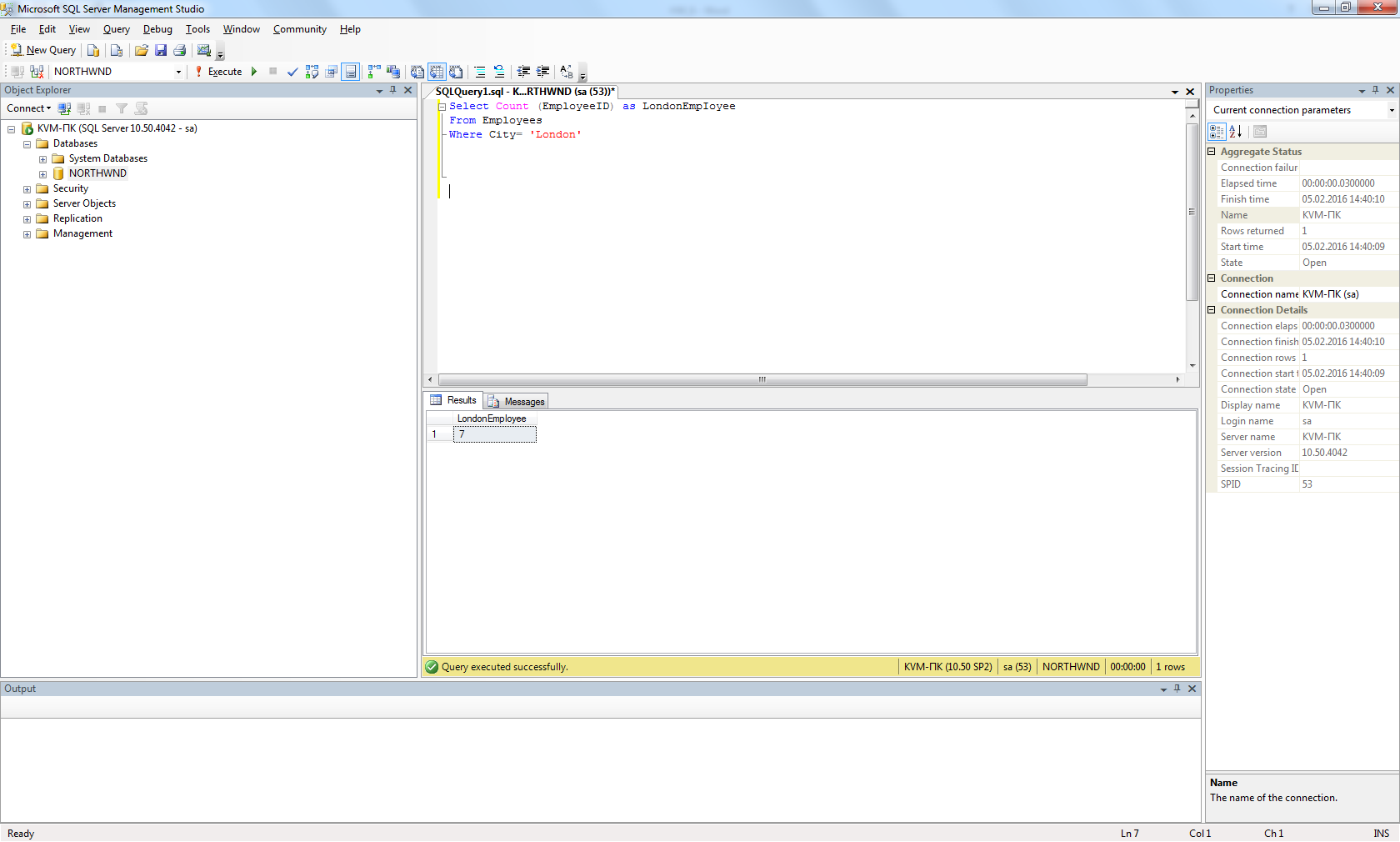


1. Calculate the count of employees from London.

Select Count (EmployeeID) as LondonEmpIoyee

From Employees

Where City= 'London'



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

Select

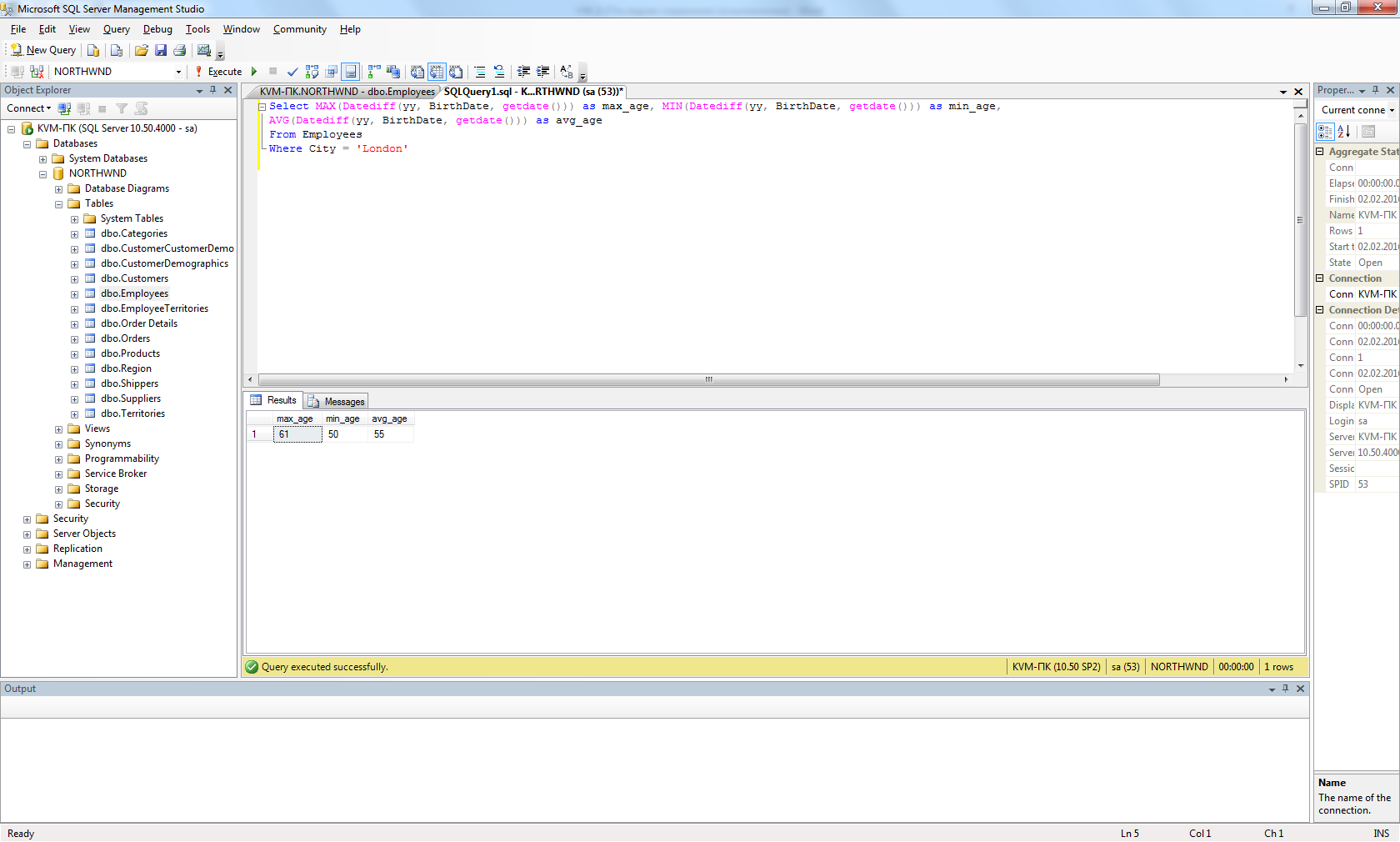
MAX(Datediff(yy, BirthDate, getdate())) as max\_age,

MIN(Datediff(yy, BirthDate, getdate())) as min\_age,

AVG(Datediff(yy, BirthDate, getdate())) as avg\_age

From Employees

Where City = 'London'



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

Select City,

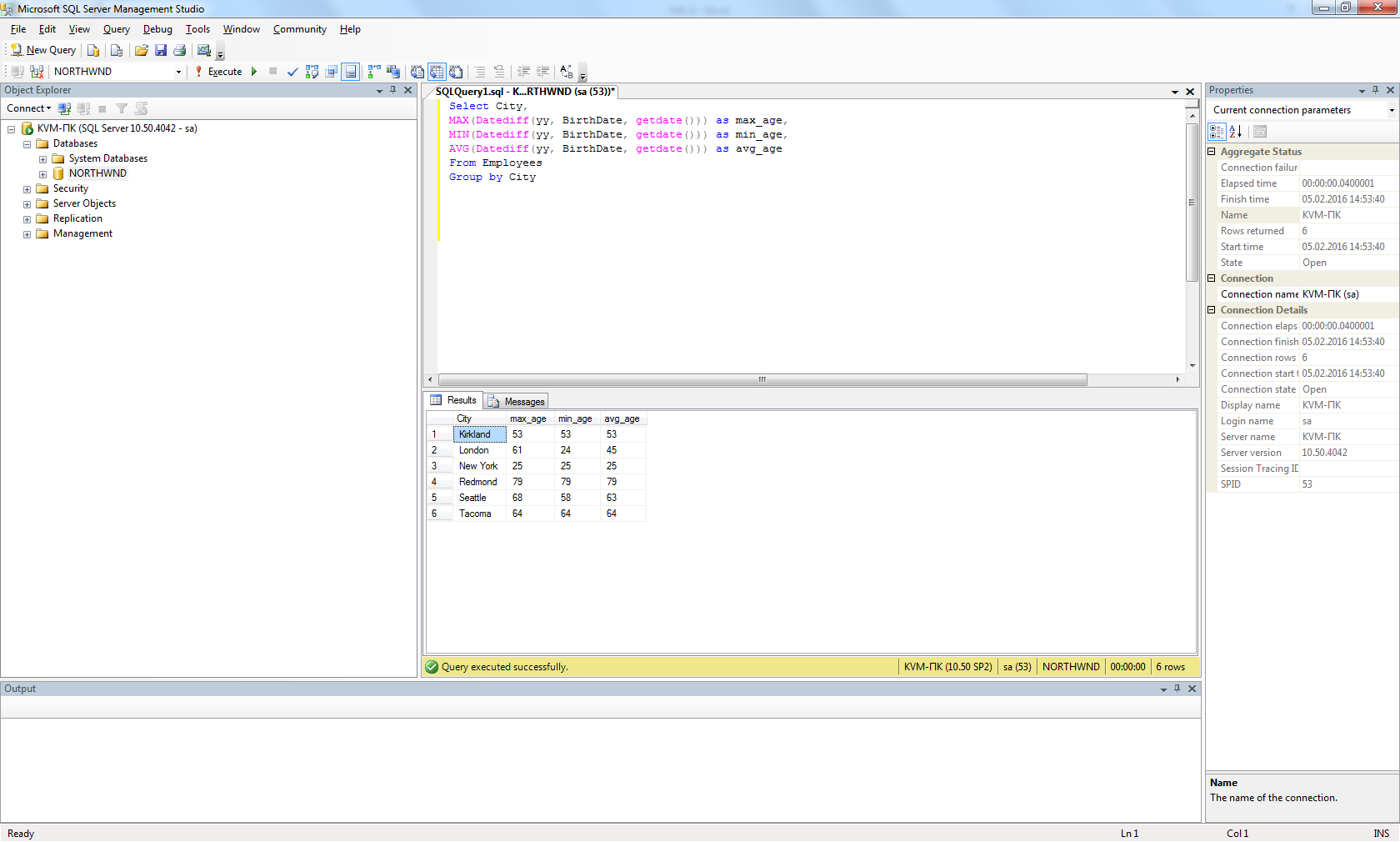
MAX(Datediff(yy, BirthDate, getdate())) as max\_age,

MIN(Datediff(yy, BirthDate, getdate())) as min\_age,

AVG(Datediff(yy, BirthDate, getdate())) as avg\_age

From 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)

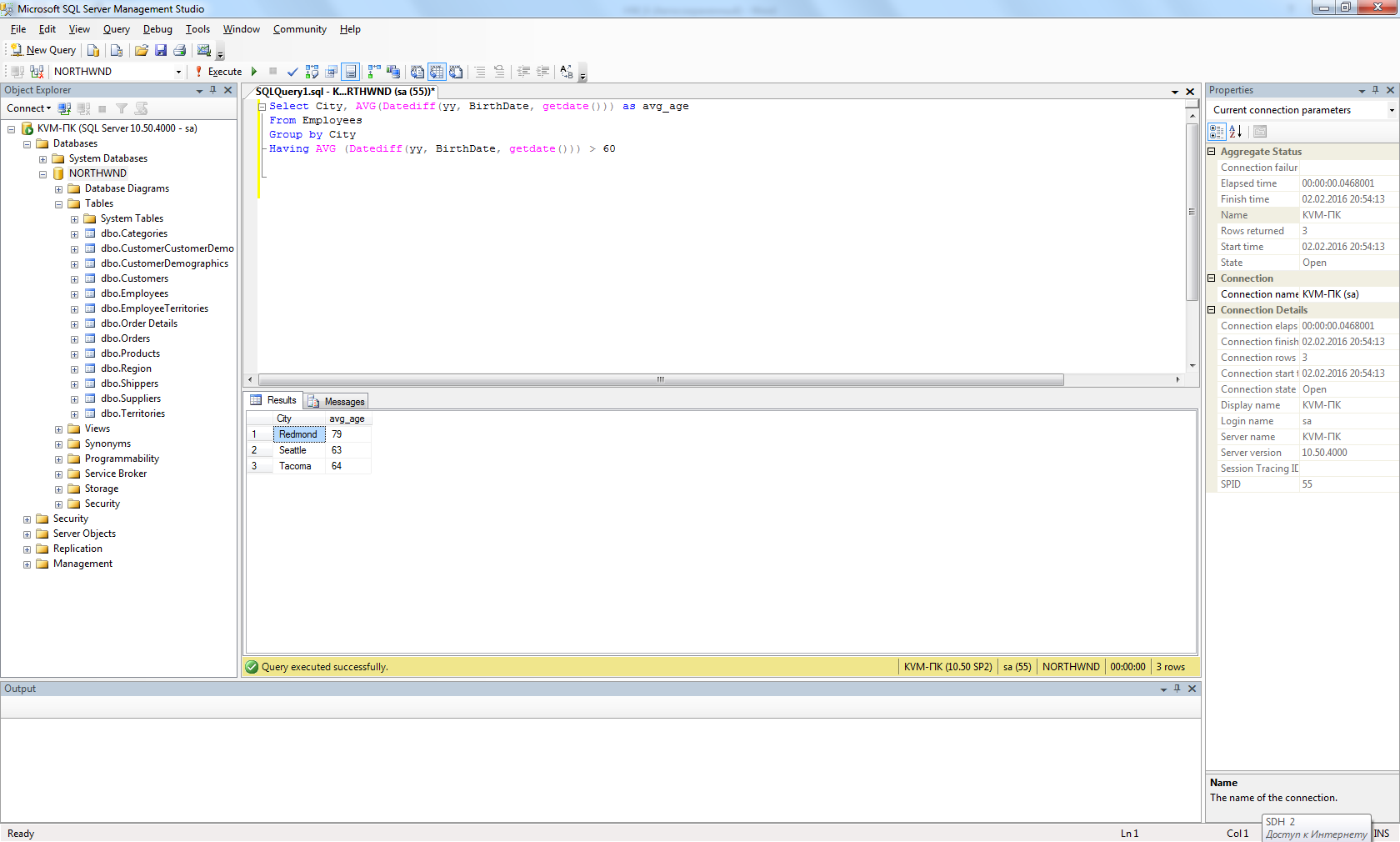
Select City,

AVG(Datediff(yy, BirthDate, getdate())) as avg\_age

From Employees

Group by City

Having AVG (Datediff(yy, BirthDate, getdate())) > 60



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

Select FirstName, LastName

From Employees

Where Datediff(yy, BirthDate, getdate()) =

(Select MAX (Datediff(yy, BirthDate, getdate())) as max\_age

From Employees)

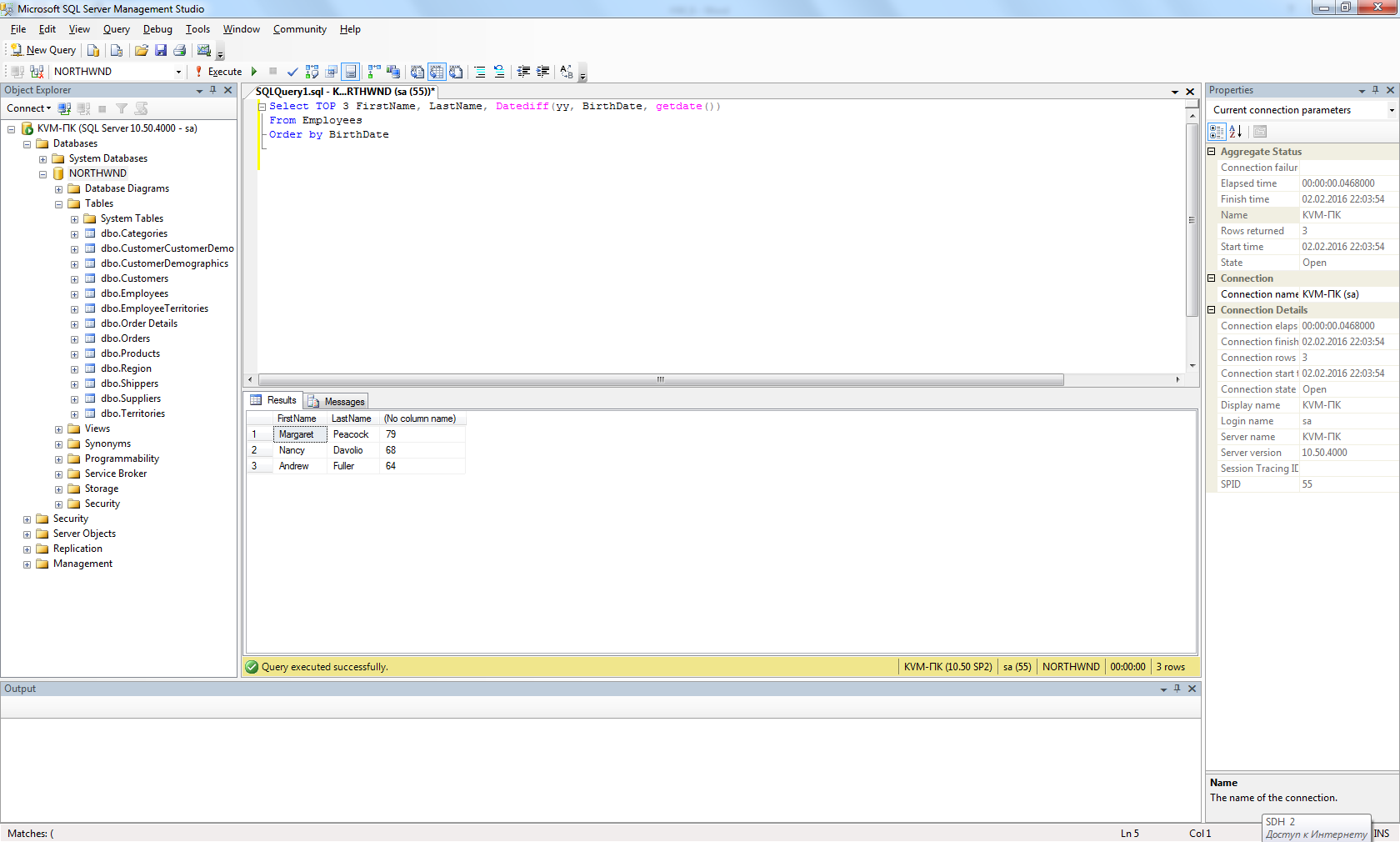


1. Show first, last names and ages of 3 eldest employees.

Select TOP 3 FirstName, LastName, Datediff(yy, BirthDate, getdate())

From Employees

Order by BirthDate

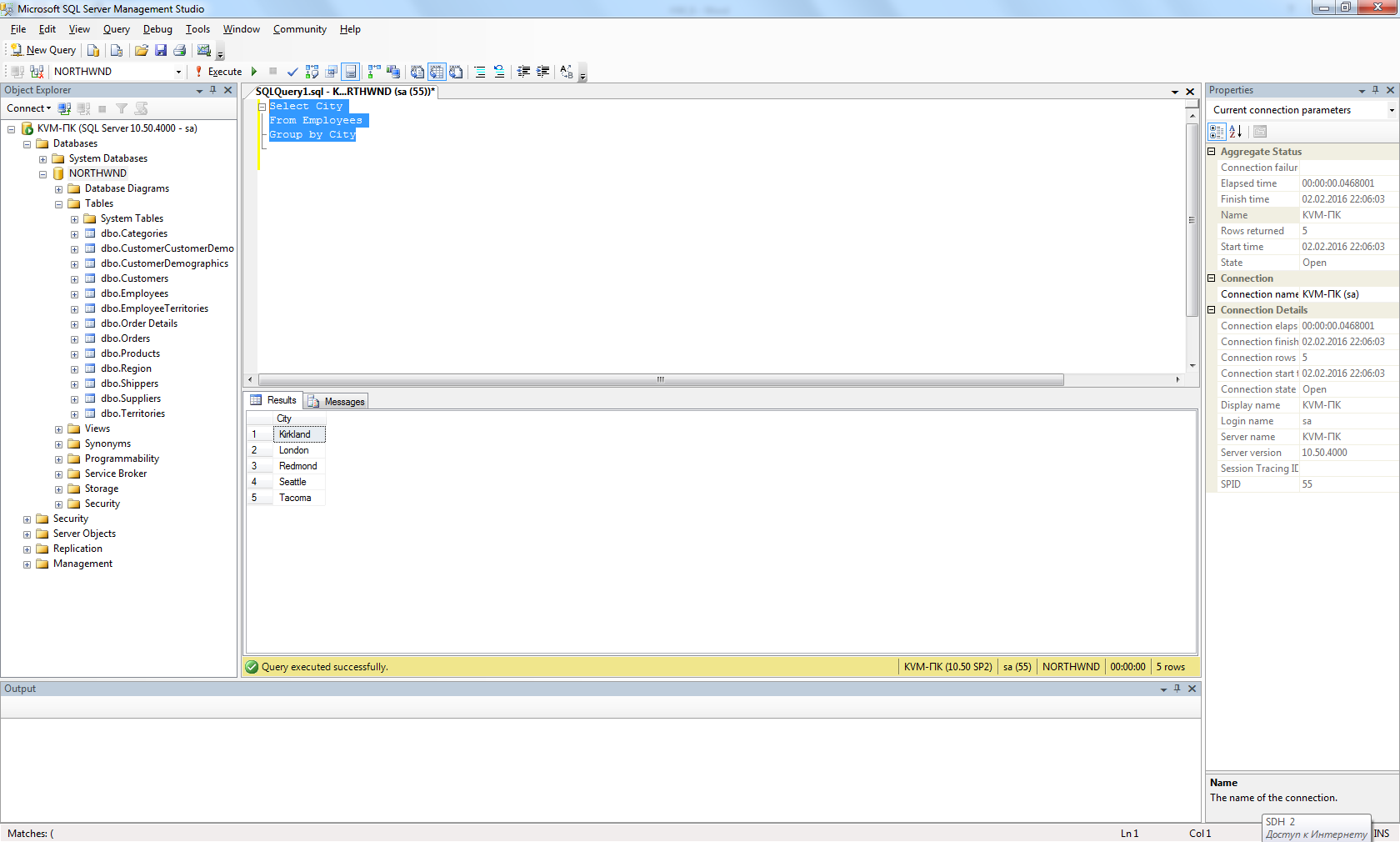


1. Show the list of all cities where the employees are from.

Select City

From Employees

Group by City

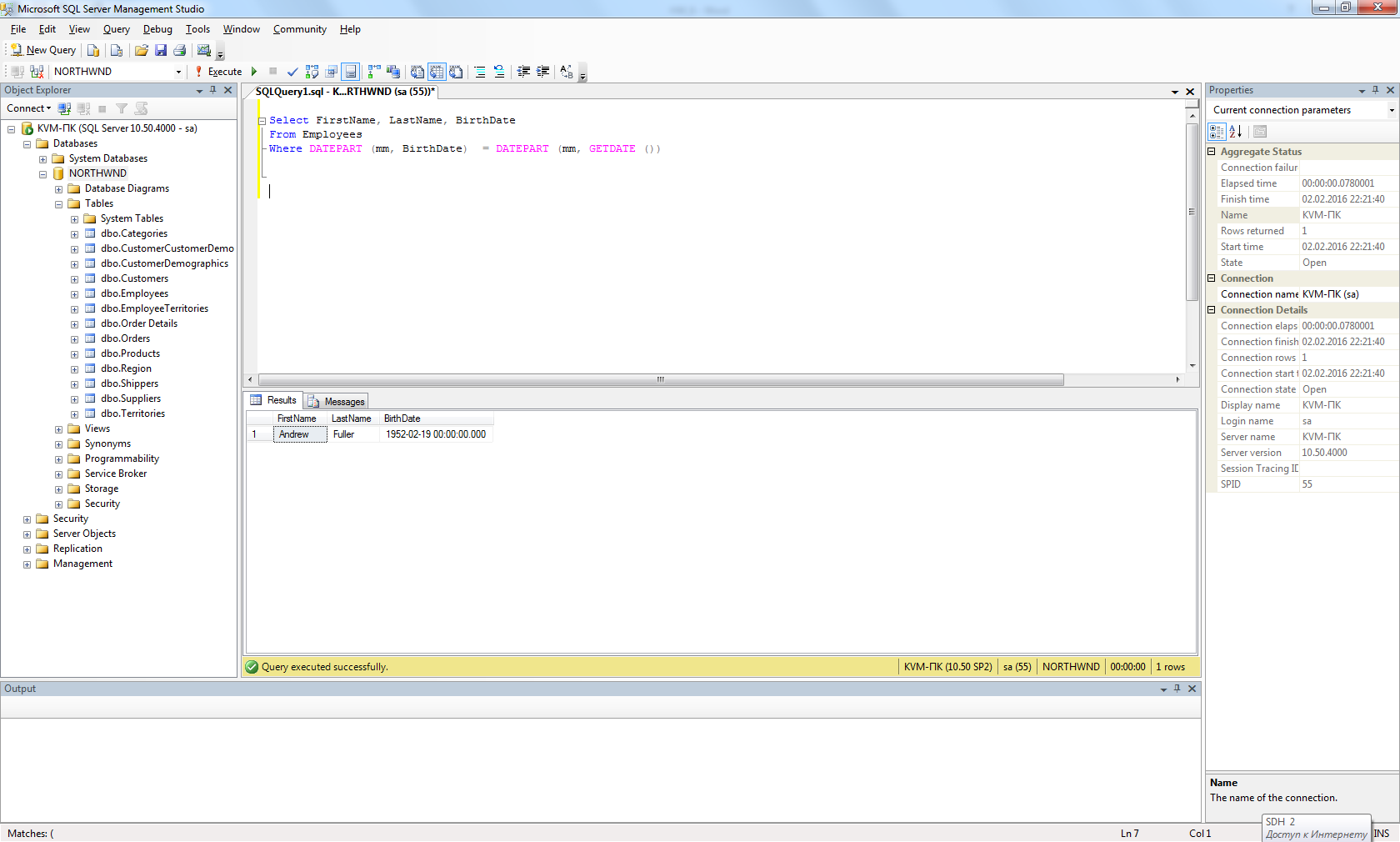


1. Show first, last names and dates of birth of the employees who celebrate their birthdays this month.

Select FirstName, LastName, BirthDate

From Employees

Where DATEPART (mm, BirthDate) = DATEPART (mm, GETDATE ())



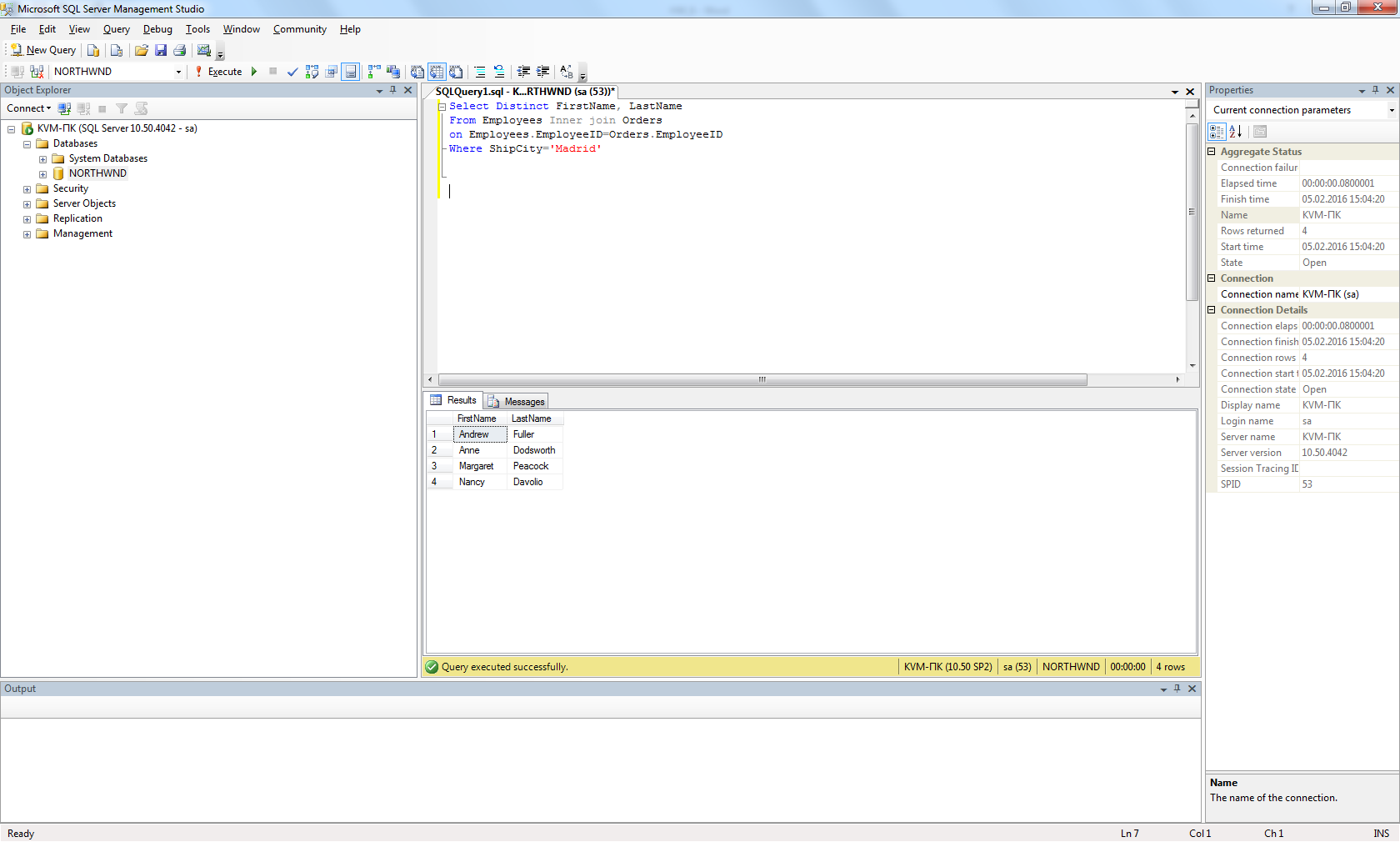
1. Show first and last names of the employees who used to serve orders shipped to Madrid.

Select Distinct FirstName, LastName

From Employees Inner join Orders

on Employees.EmployeeID=Orders.EmployeeID

Where ShipCity='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).

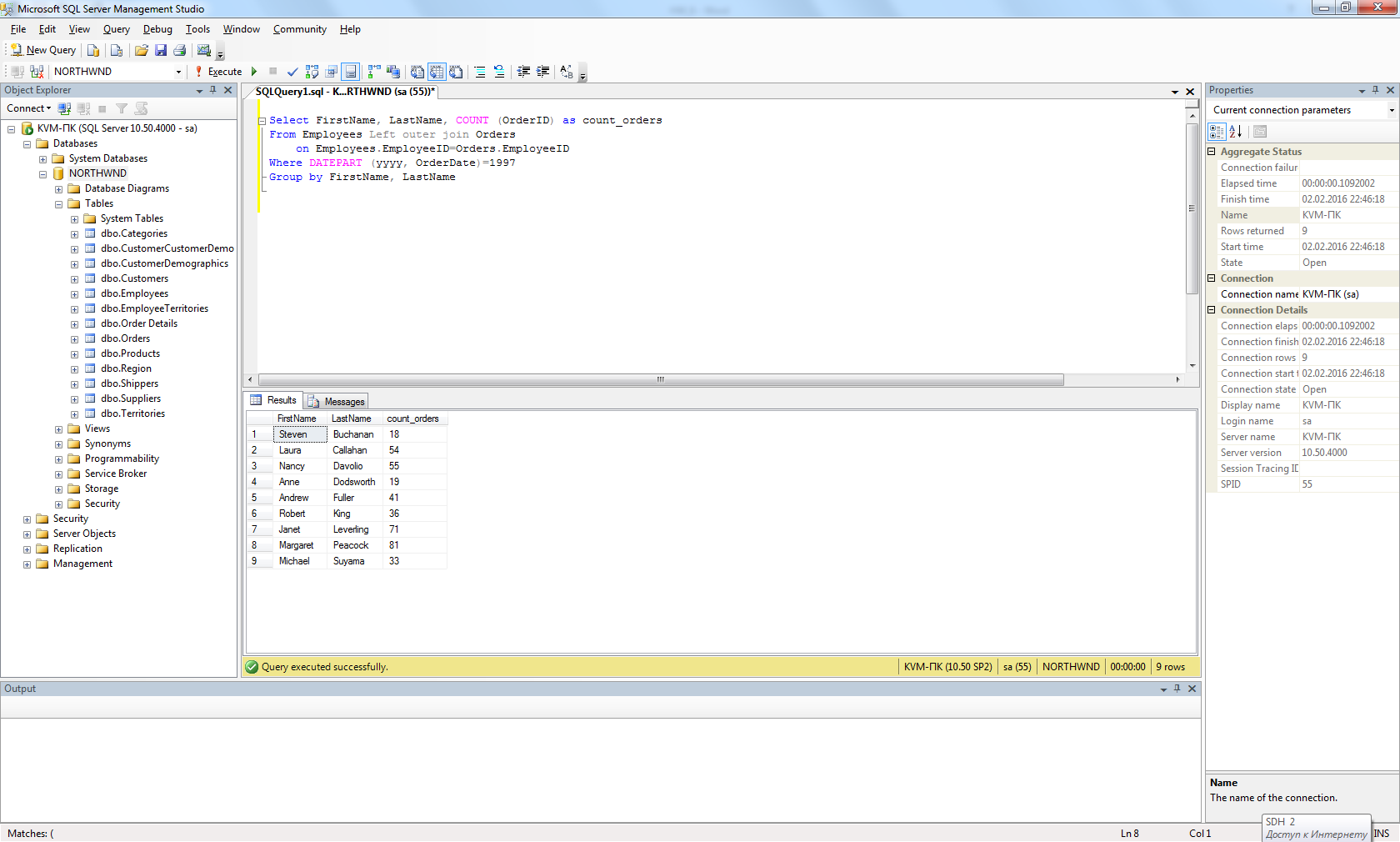
Select FirstName, LastName, COUNT (OrderID) as count\_orders

From Employees Left outer join Orders

on Employees.EmployeeID=Orders.EmployeeID

Where DATEPART (yyyy, 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 Distinct FirstName, LastName, (Select Count(OrderID)

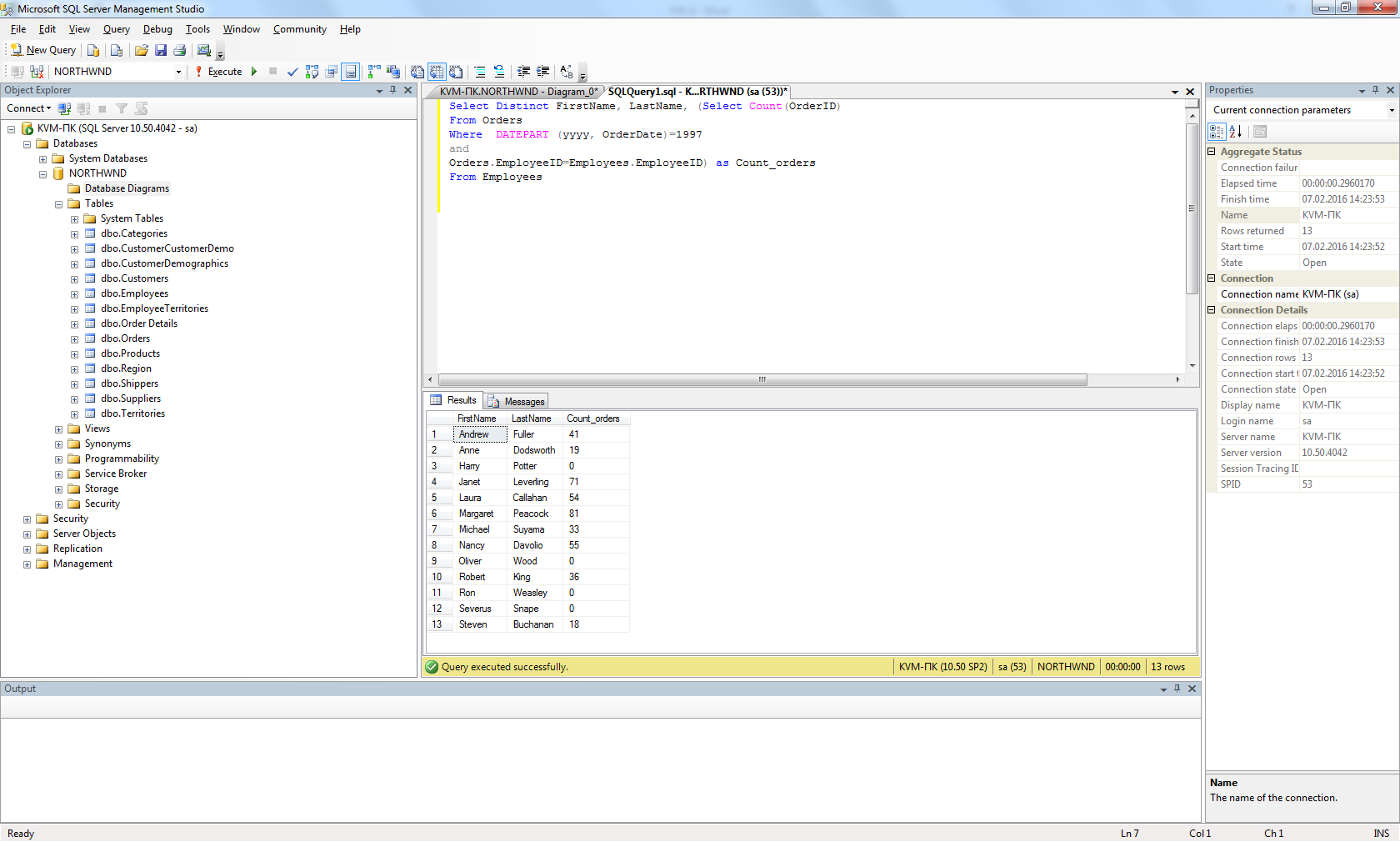
From Orders

Where DATEPART (yyyy, OrderDate)=1997

and

Orders.EmployeeID=Employees.EmployeeID) as Count\_orders

From Employees



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 FirstName, LastName, COUNT (OrderID) as count\_orders

From Orders Left outer join Employees

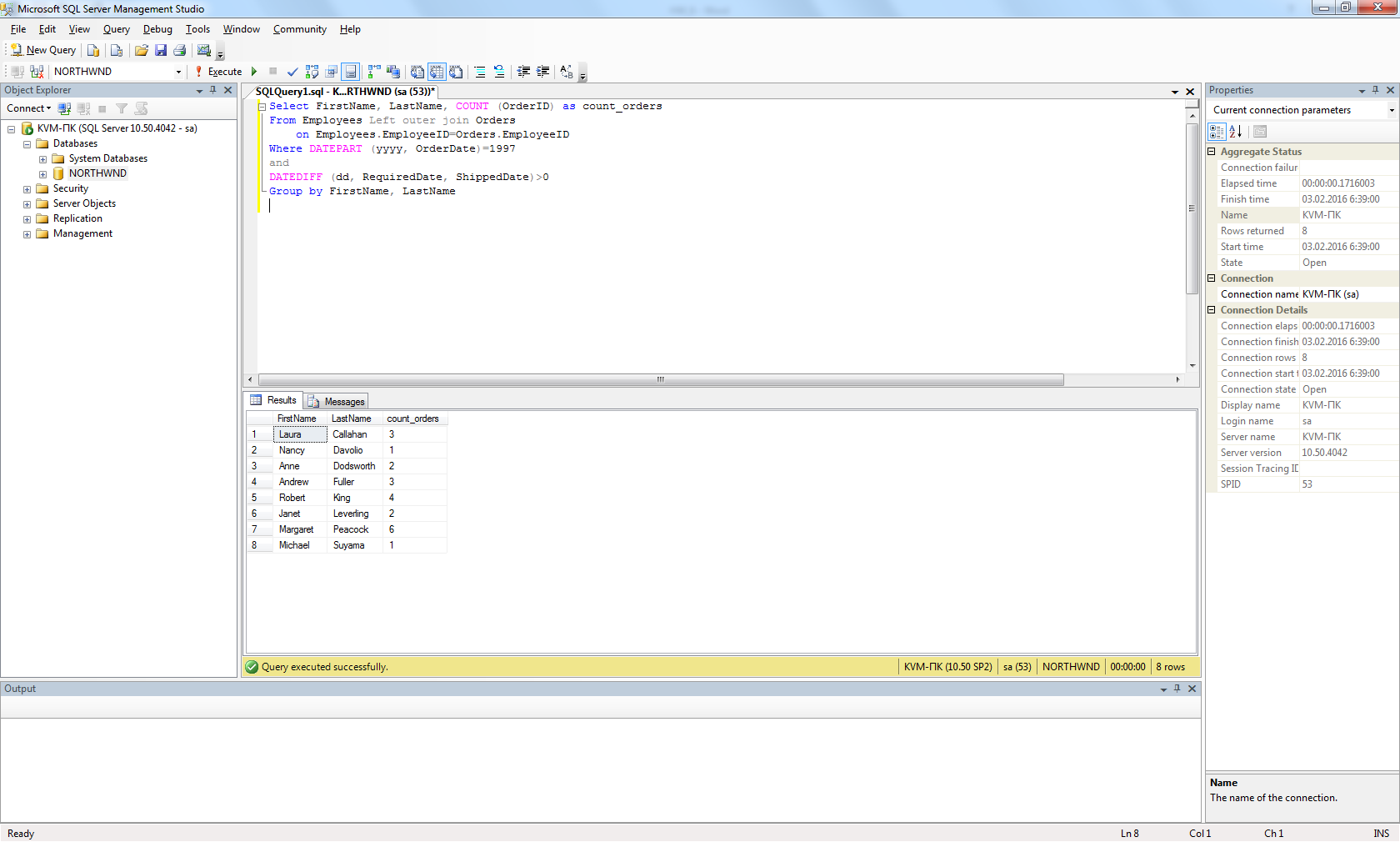
on Employees.EmployeeID=Orders.EmployeeID

Where DATEPART (yyyy, OrderDate)=1997

and

DATEDIFF (dd, RequiredDate, ShippedDate)>0

Group by FirstName, LastName



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

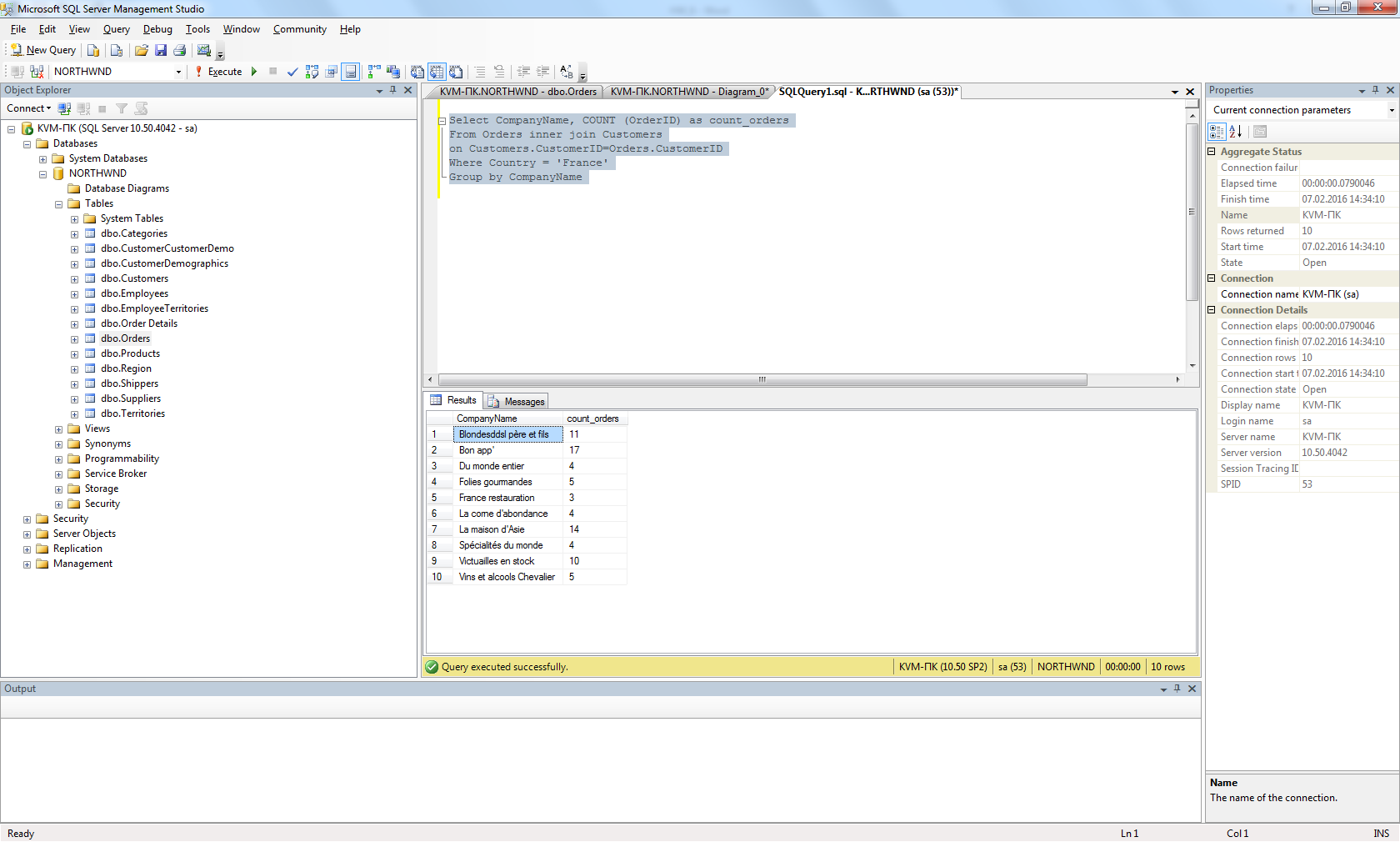
Select CompanyName, COUNT (OrderID) as count\_orders

From Orders inner join Customers

on Customers.CustomerID=Orders.CustomerID

Where Country = 'France'

Group by CompanyName



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

Select CompanyName

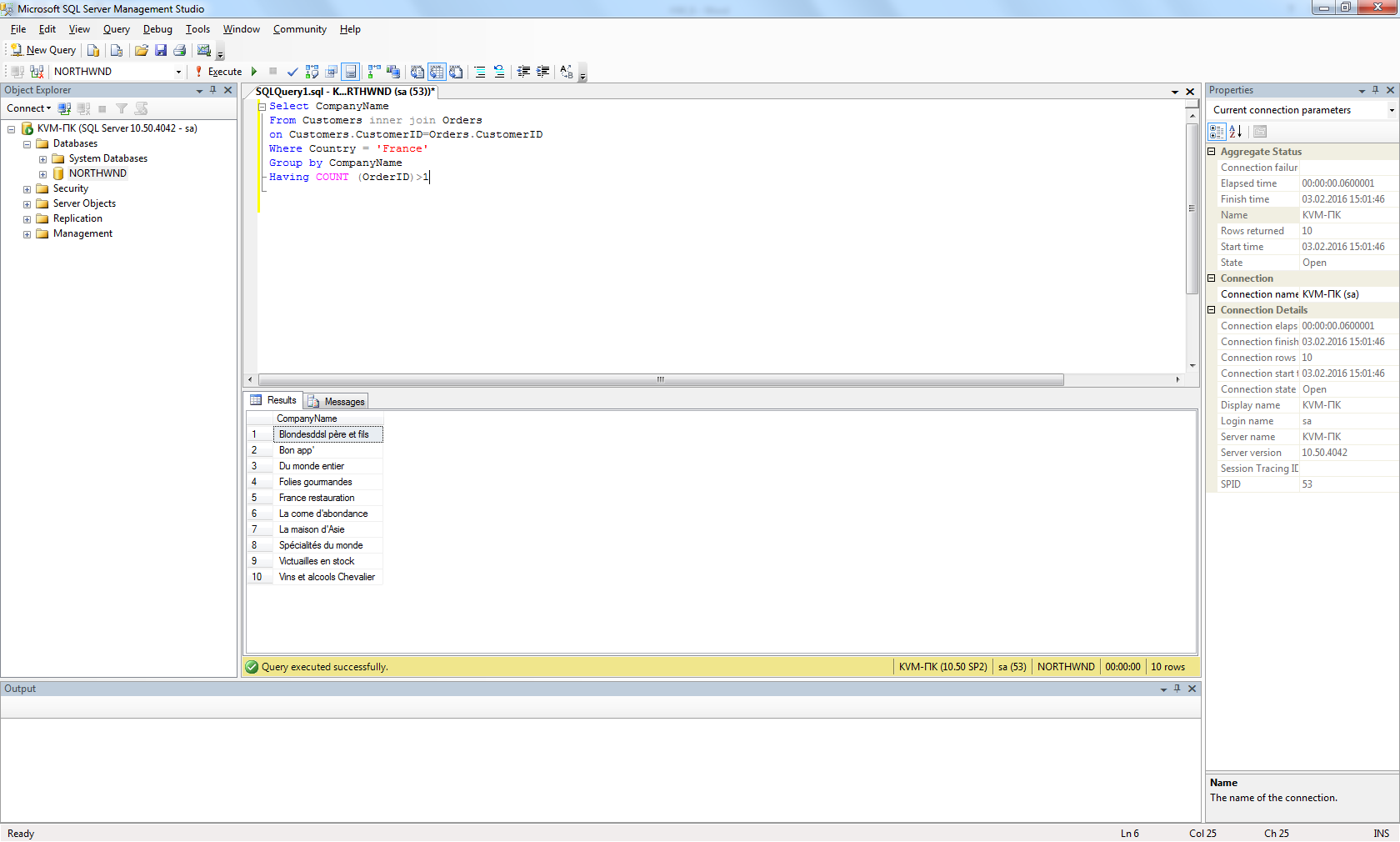
From Customers inner join Orders

on Customers.CustomerID=Orders.CustomerID

Where Country = 'France'

Group by CompanyName

Having COUNT (OrderID)>1



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

Select CompanyName

From Customers

Where CustomerID IN (

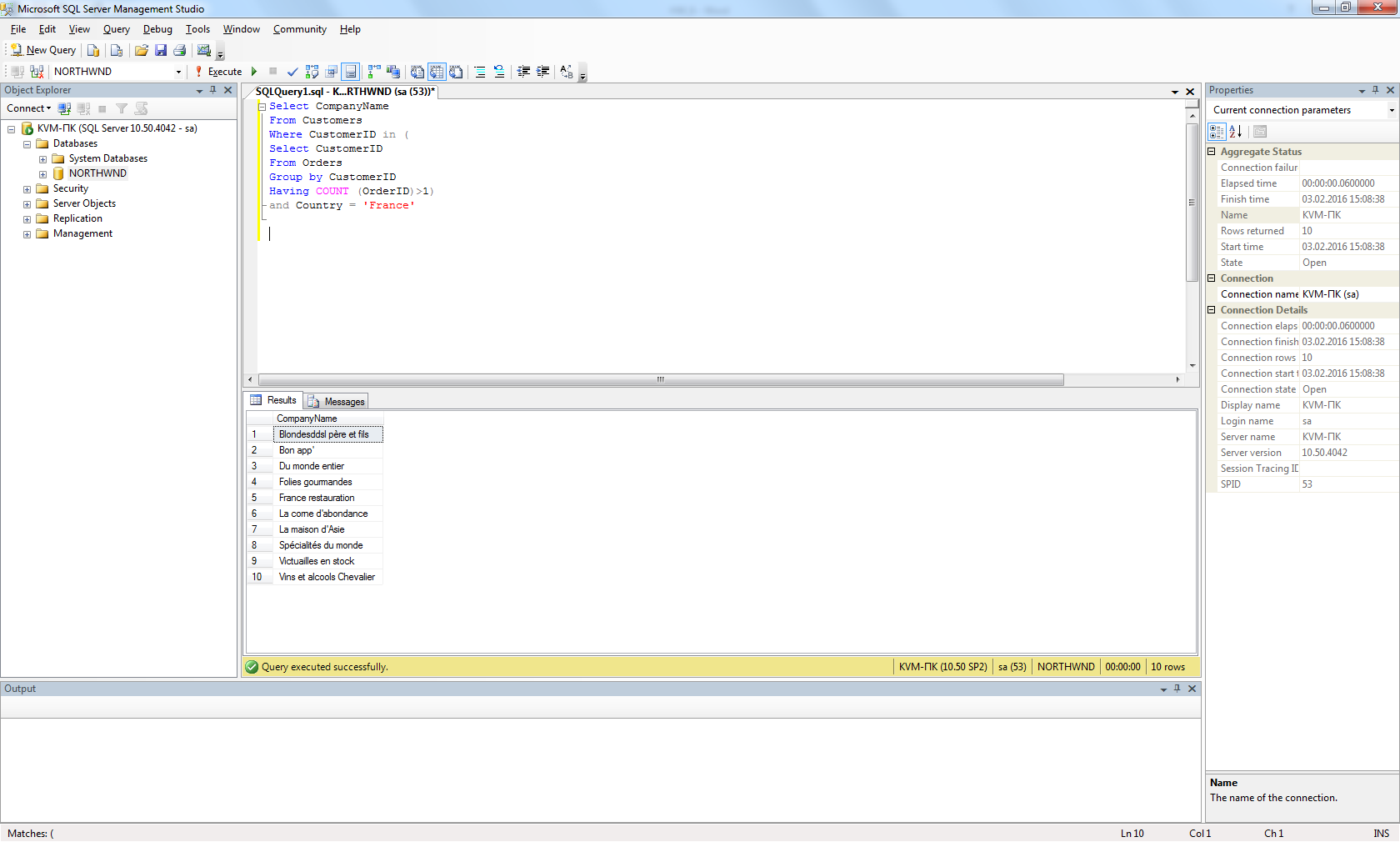
Select CustomerID

From Orders

Group by CustomerID

Having COUNT (OrderID)>1)

and Country = 'France'



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

Select CompanyName

From Customers

Where CustomerID IN(

Select CustomerID

From Orders

Where OrderID IN(

Select OrderID

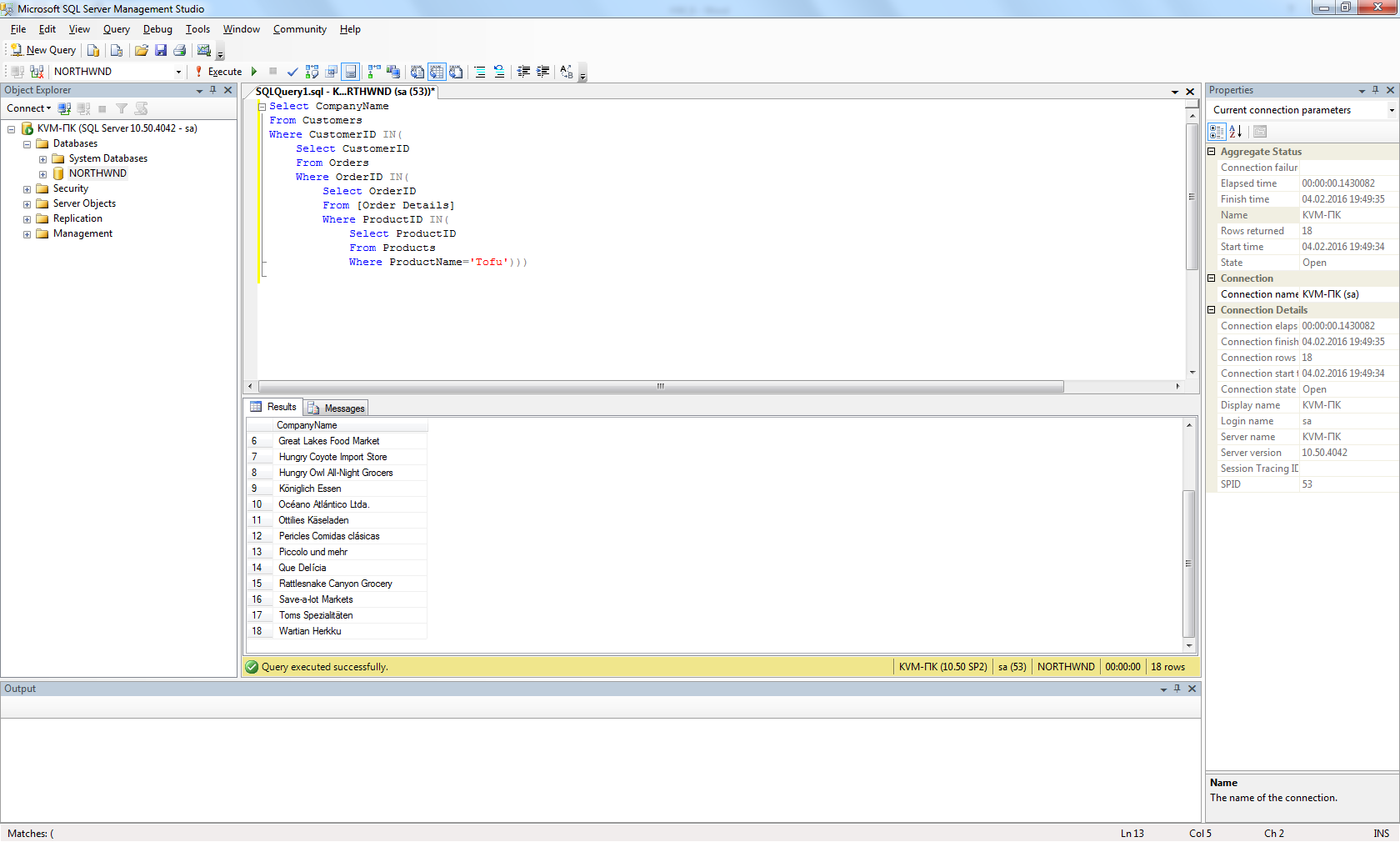
From [Order Details]

Where ProductID IN(

Select ProductID

From Products

Where ProductName='Tofu')))



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 CompanyName,

Sum (Quantity) as Total\_count,

Sum([Order Details].UnitPrice\*Quantity\*(1-Discount)) as Total\_sum

From Products

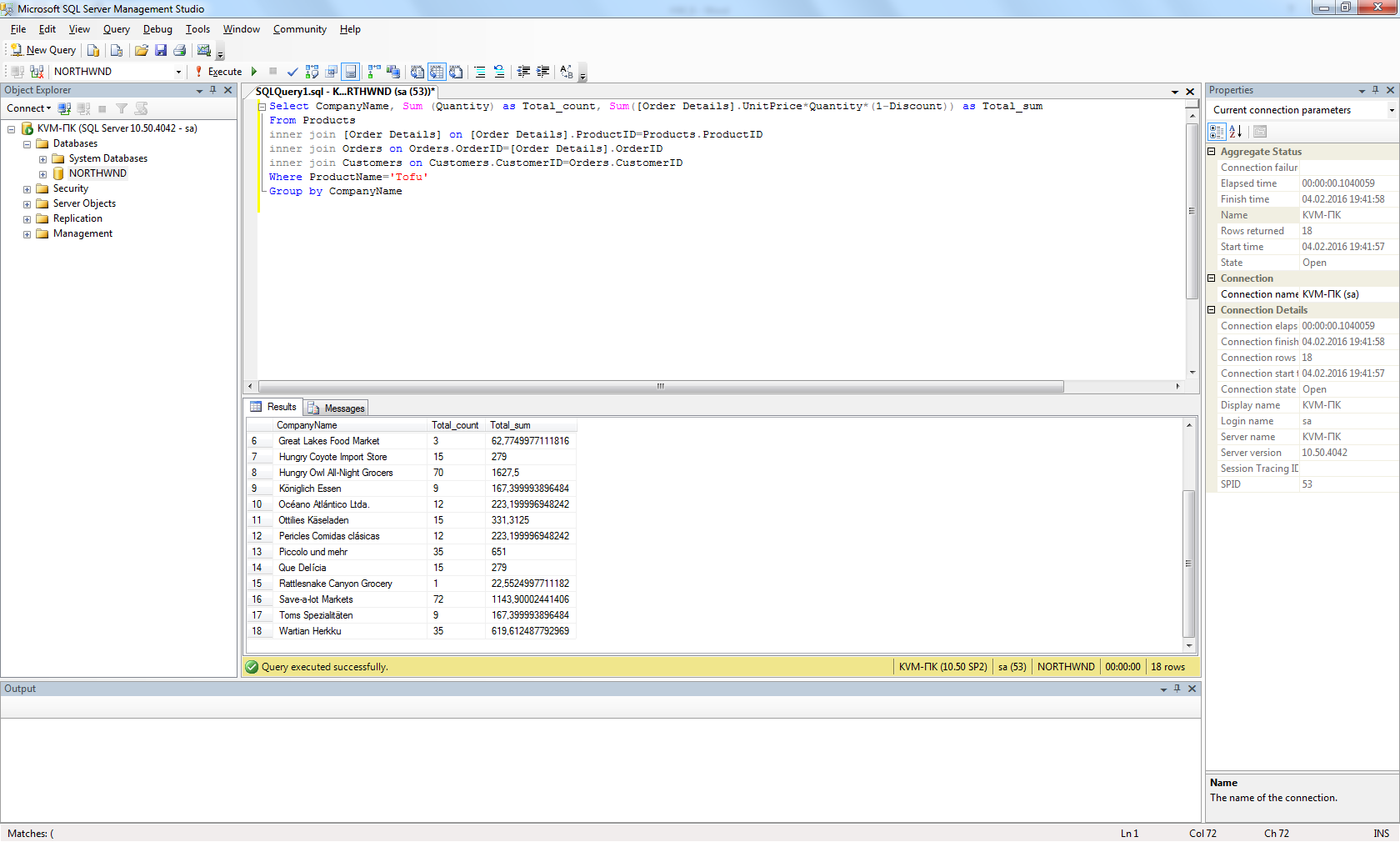
inner join [Order Details] on [Order Details].ProductID=Products.ProductID

inner join Orders on Orders.OrderID=[Order Details].OrderID

inner join Customers on Customers.CustomerID=Orders.CustomerID

Where ProductName='Tofu'

Group by CompanyName



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

Select distinct Customers.CompanyName

From Suppliers

left join Products on Products.SupplierID = Suppliers.SupplierID

left join [Order Details] on [Order Details].ProductID=Products.ProductID

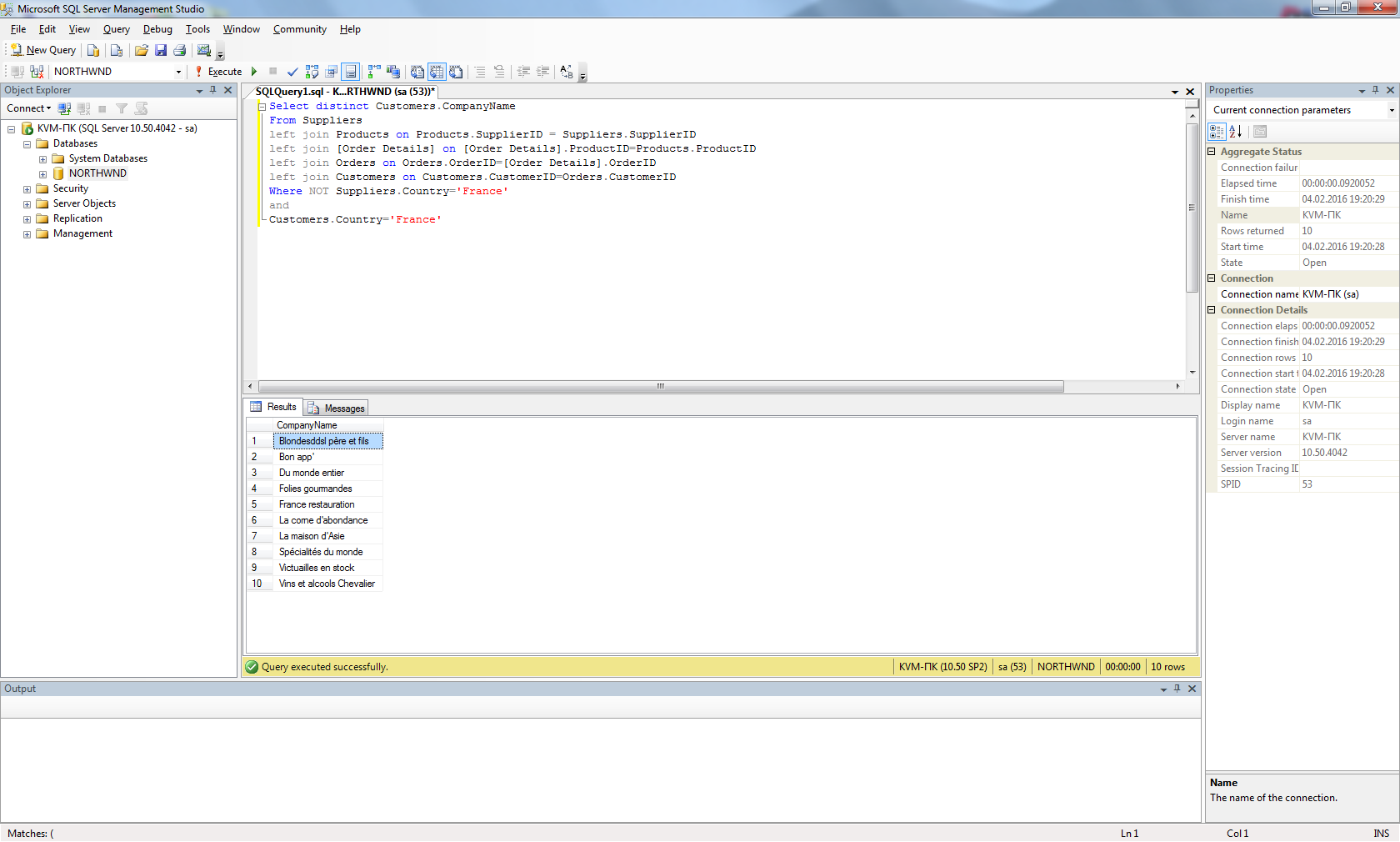
left join Orders on Orders.OrderID=[Order Details].OrderID

left join Customers on Customers.CustomerID=Orders.CustomerID

Where NOT Suppliers.Country='France'

and

Customers.Country='France'



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

Select CompanyName

From Customers

Where CustomerID IN (

Select CustomerID

From Orders

Where OrderID IN(

Select OrderID

From [Order Details]

Where ProductID IN(

Select ProductID

From Products

Where SupplierID IN(

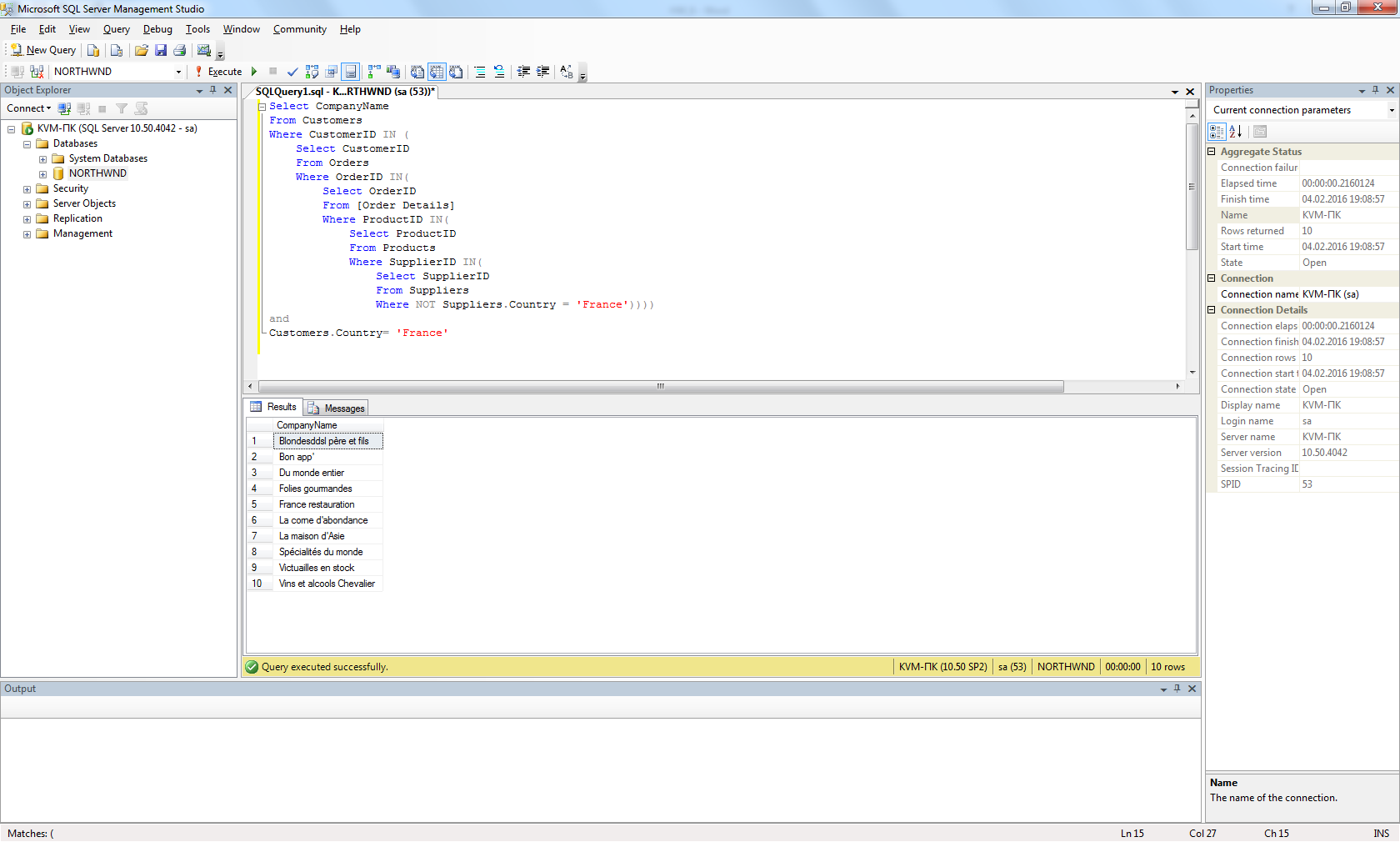
Select SupplierID

From Suppliers

Where NOT Suppliers.Country = 'France'))))

and

Customers.Country= 'France'



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

Select CompanyName

From Customers

Where CustomerID IN (

Select CustomerID

From Orders

Where OrderID IN(

Select OrderID

From [Order Details]

Where ProductID IN(

Select ProductID

From Products

Where SupplierID IN(

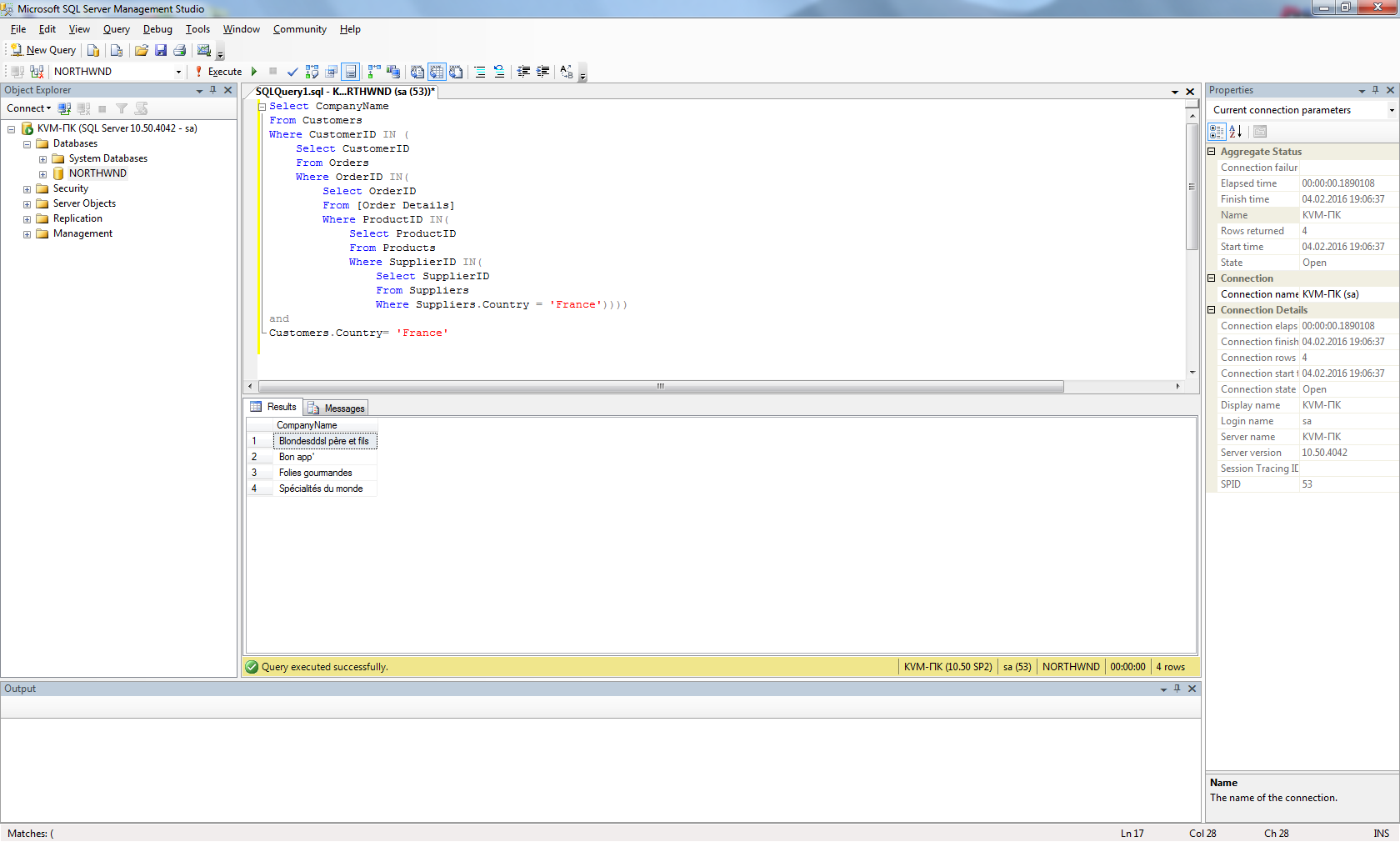
Select SupplierID

From Suppliers

Where Suppliers.Country = 'France'))))

and

Customers.Country= 'France'



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

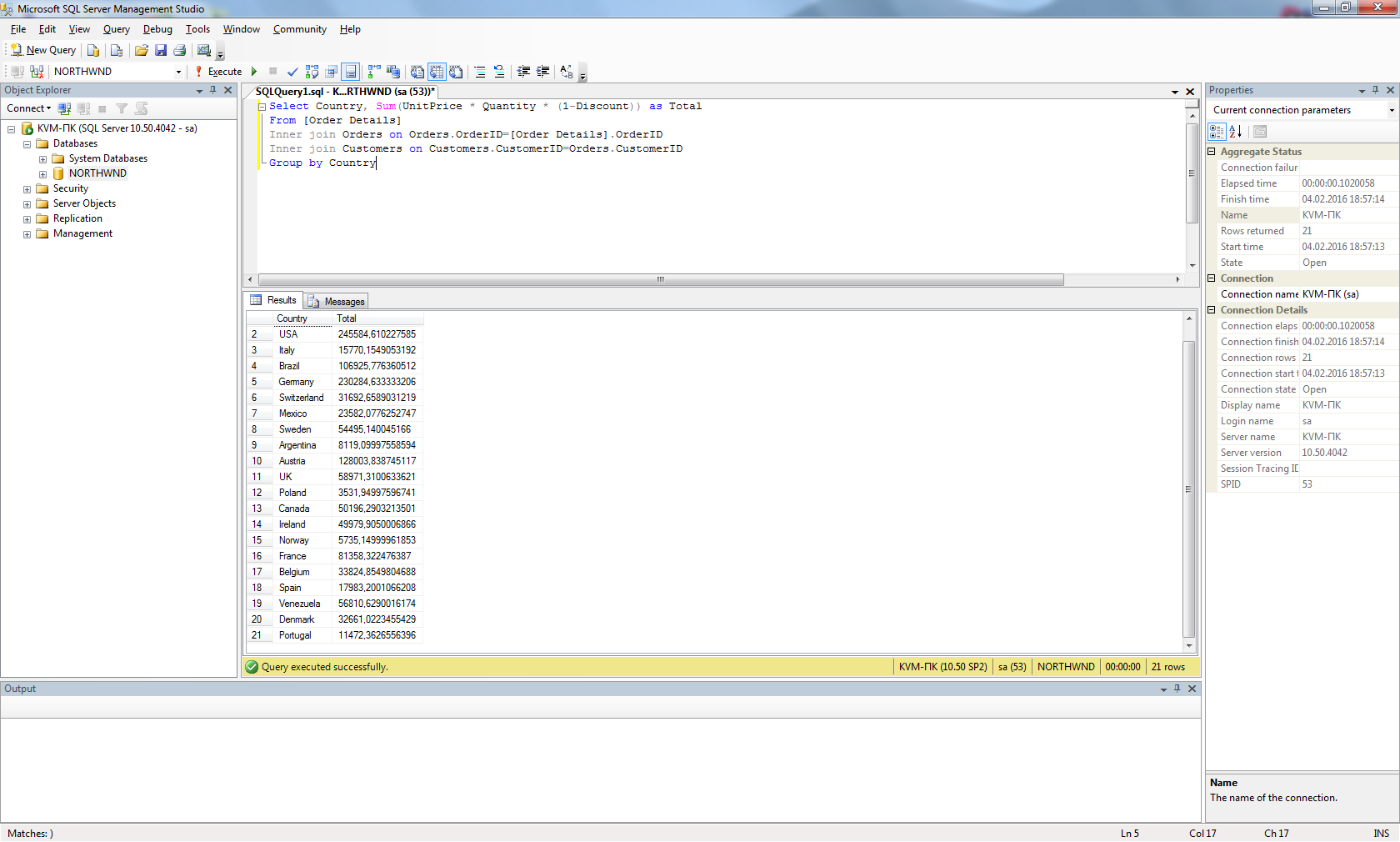
Select Country, Sum(UnitPrice \* Quantity \* (1-Discount)) as Total

From [Order Details]

Inner join Orders on Orders.OrderID=[Order Details].OrderID

Inner join Customers on Customers.CustomerID=Orders.CustomerID

Group by 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 d.Country, d.DomesticSum , nd.NonDomesticSum

From (Select Customers.Country, Sum([Order Details].UnitPrice \* Quantity \* (1-Discount)) as DomesticSum

From Suppliers

Inner join Products on Products.SupplierID = Suppliers.SupplierID

Inner join [Order Details] on [Order Details].ProductID=Products.ProductID

Inner join Orders on Orders.OrderID=[Order Details].OrderID

Inner join Customers on Customers.CustomerID=Orders.CustomerID

Where NOT Customers.Country=Suppliers.Country

Group by Customers.Country) d

Full join

(Select Customers.Country, Sum([Order Details].UnitPrice \* Quantity \* (1-Discount)) as NonDomesticSum

From Suppliers

Inner join Products on Products.SupplierID = Suppliers.SupplierID

Inner join [Order Details] on [Order Details].ProductID=Products.ProductID

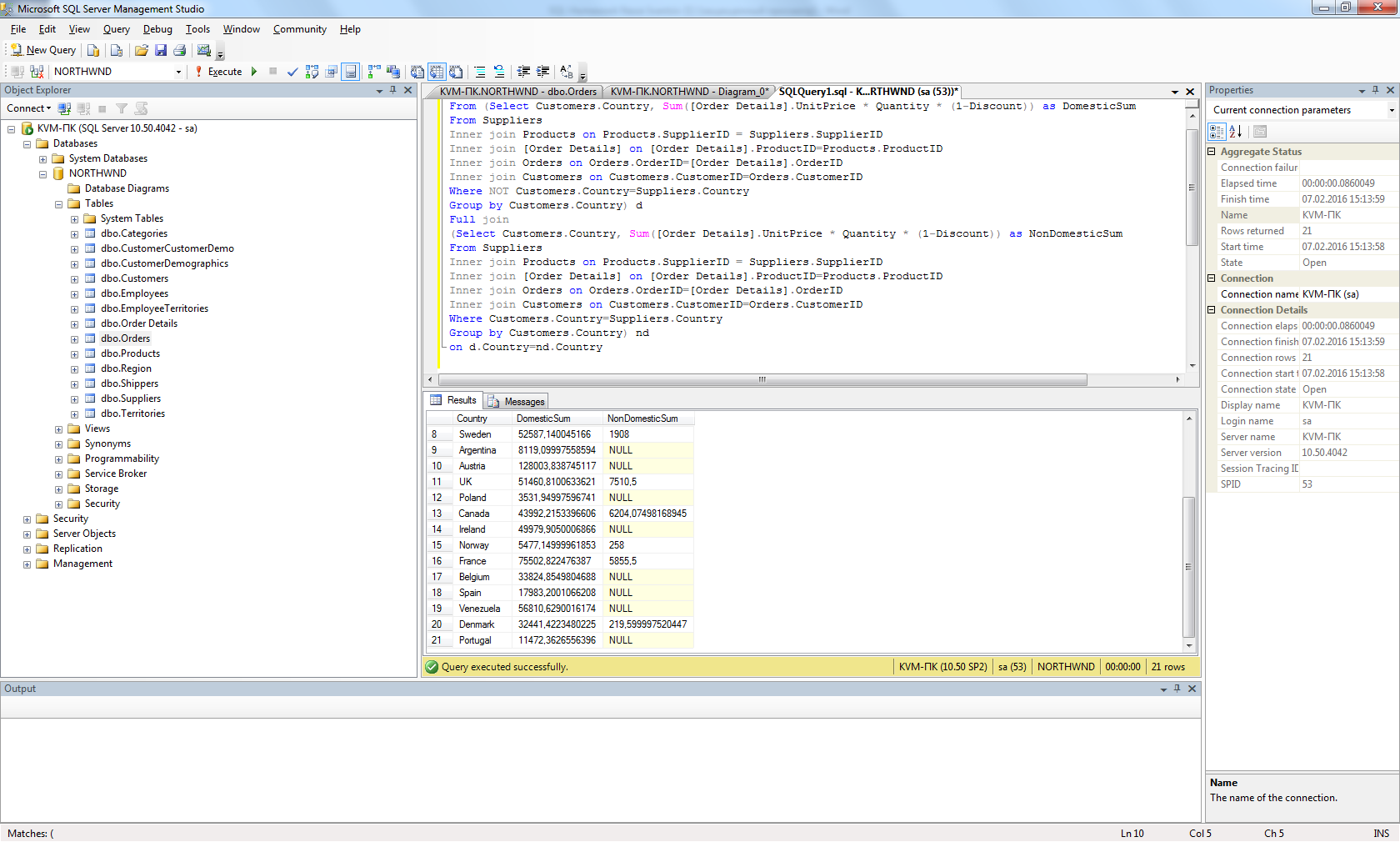
Inner join Orders on Orders.OrderID=[Order Details].OrderID

Inner join Customers on Customers.CustomerID=Orders.CustomerID

Where Customers.Country=Suppliers.Country

Group by Customers.Country) nd

on d.Country=nd.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 CategoryName, Sum ([Order Details].UnitPrice \* Quantity \* (1-Discount)) as Total

From Categories

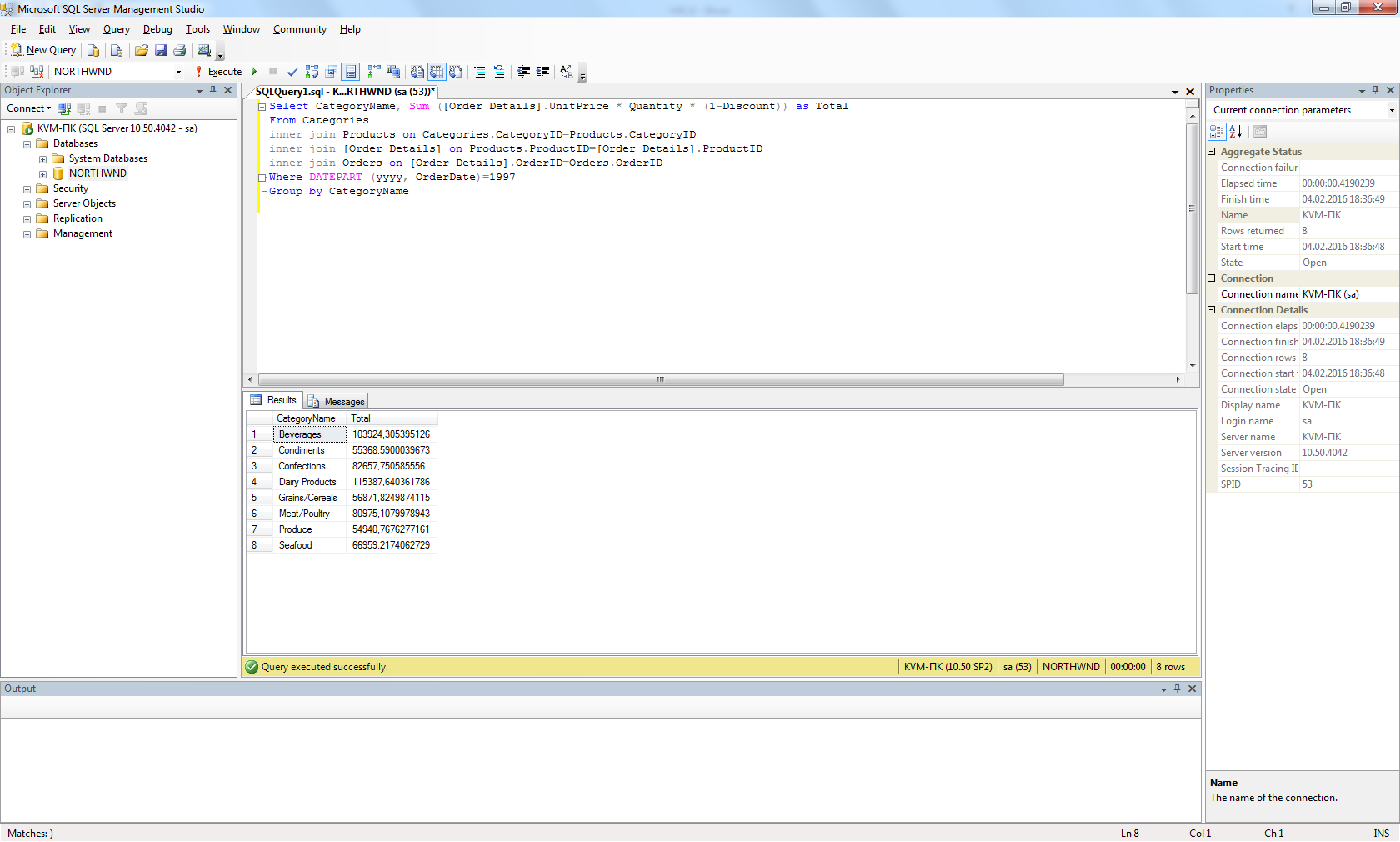
inner join Products on Categories.CategoryID=Products.CategoryID

inner join [Order Details] on Products.ProductID=[Order Details].ProductID

inner join Orders on [Order Details].OrderID=Orders.OrderID

Where DATEPART (yyyy, OrderDate)=1997

Group by CategoryName



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.

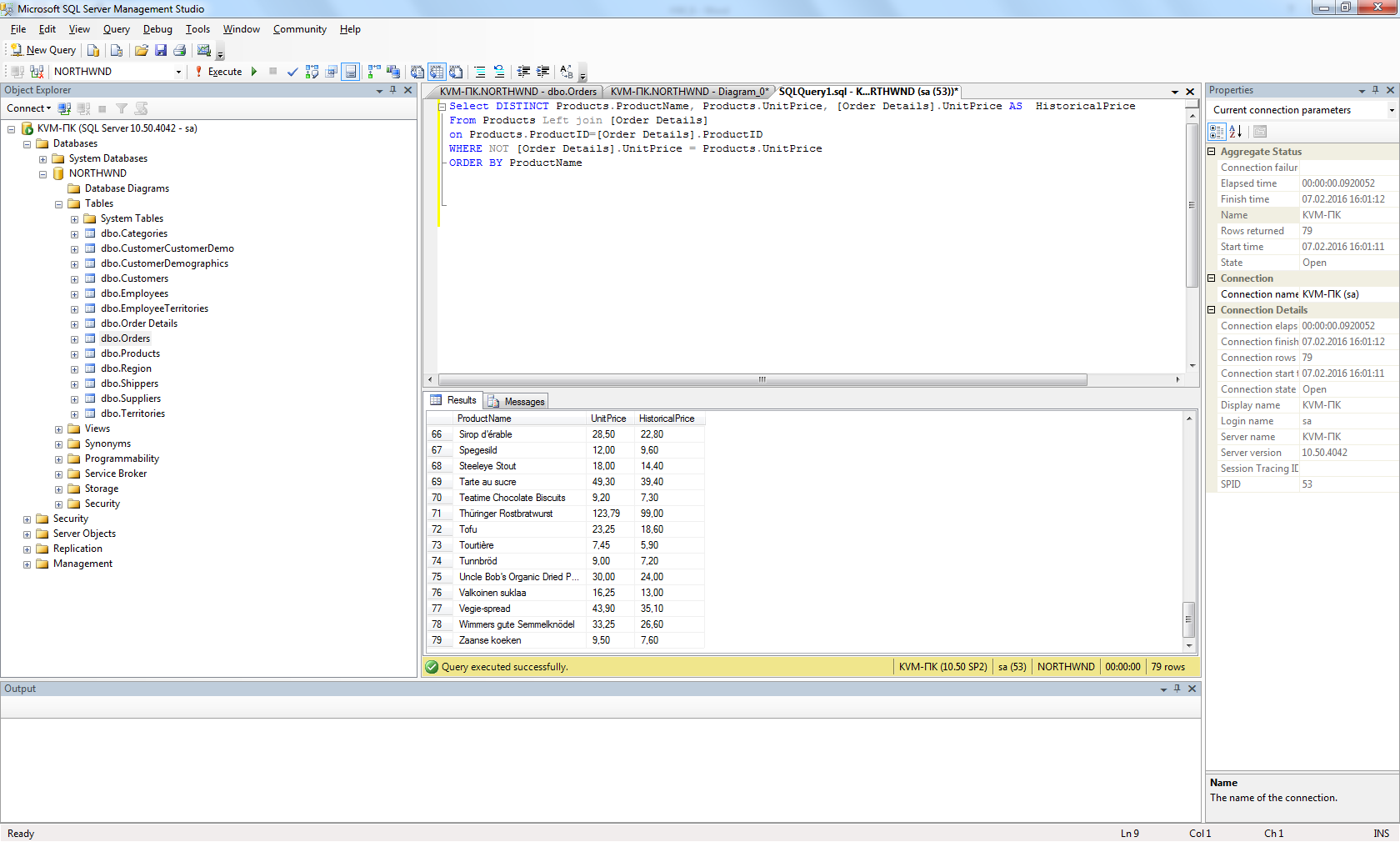
Select DISTINCT Products.ProductName, Products.UnitPrice, [Order Details].UnitPrice AS HistoricalPrice

From Products Left join [Order Details]

on Products.ProductID=[Order Details].ProductID

-- WHERE NOT [Order Details].UnitPrice = Products.UnitPrice

ORDER BY ProductName

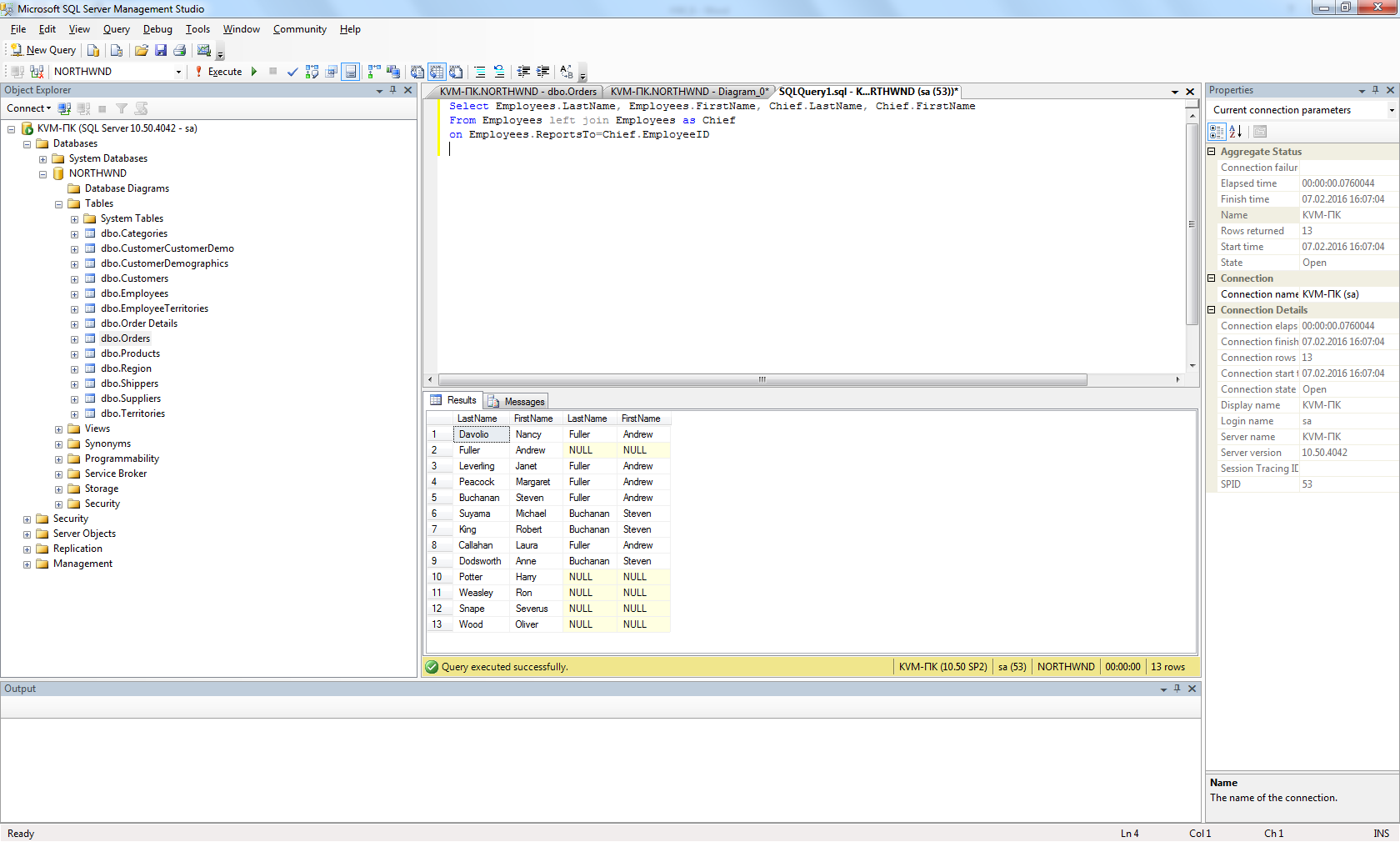


1. \*Show the list of employees’ names along with names of their chiefs (use left join with the same table).

Select Employees.LastName, Employees.FirstName, Chief.LastName, Chief.FirstName

From Employees left join Employees as Chief

on Employees.ReportsTo=Chief.EmployeeID



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

Select Customers.City

From Customers

UNION

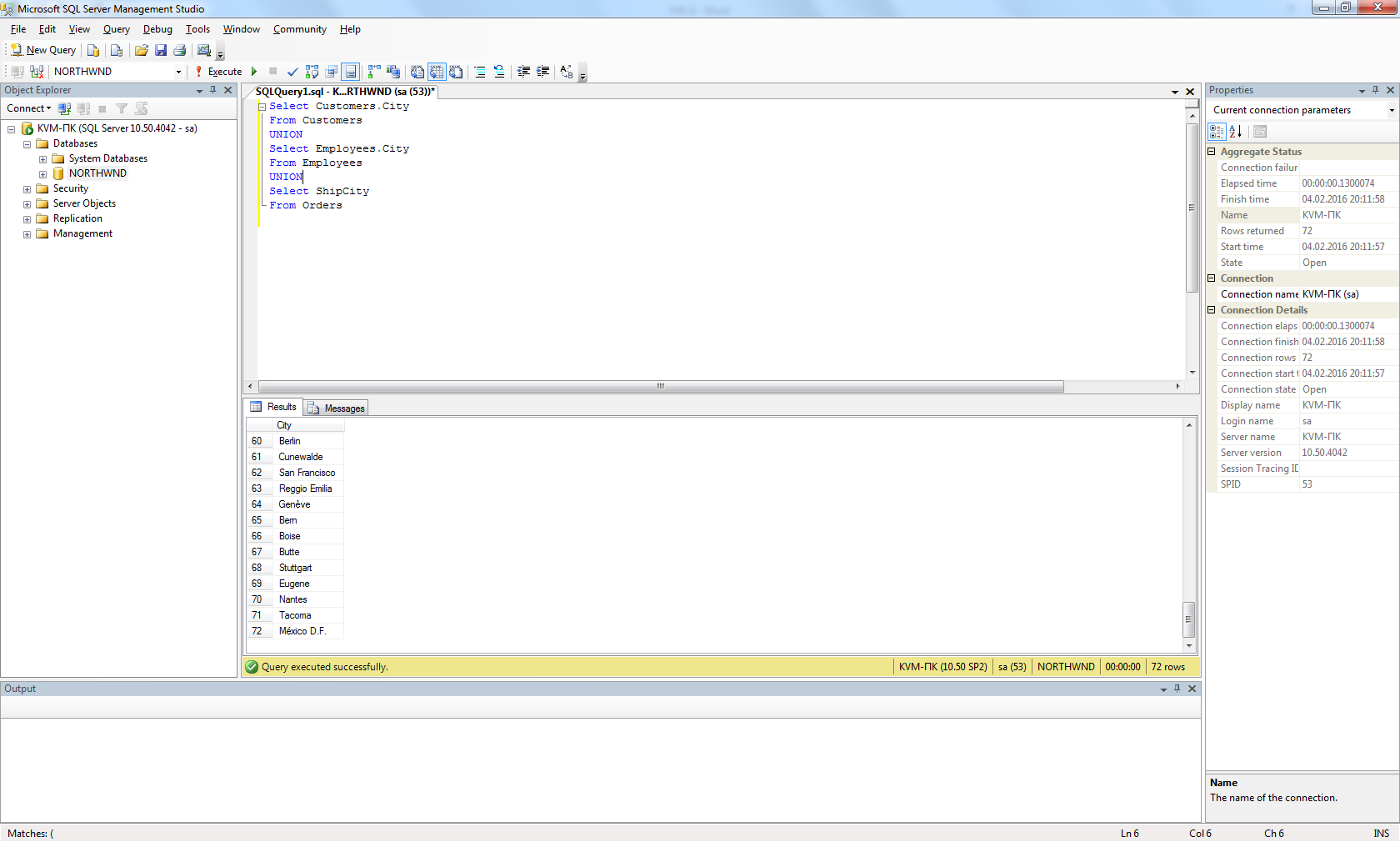
Select Employees.City

From Employees

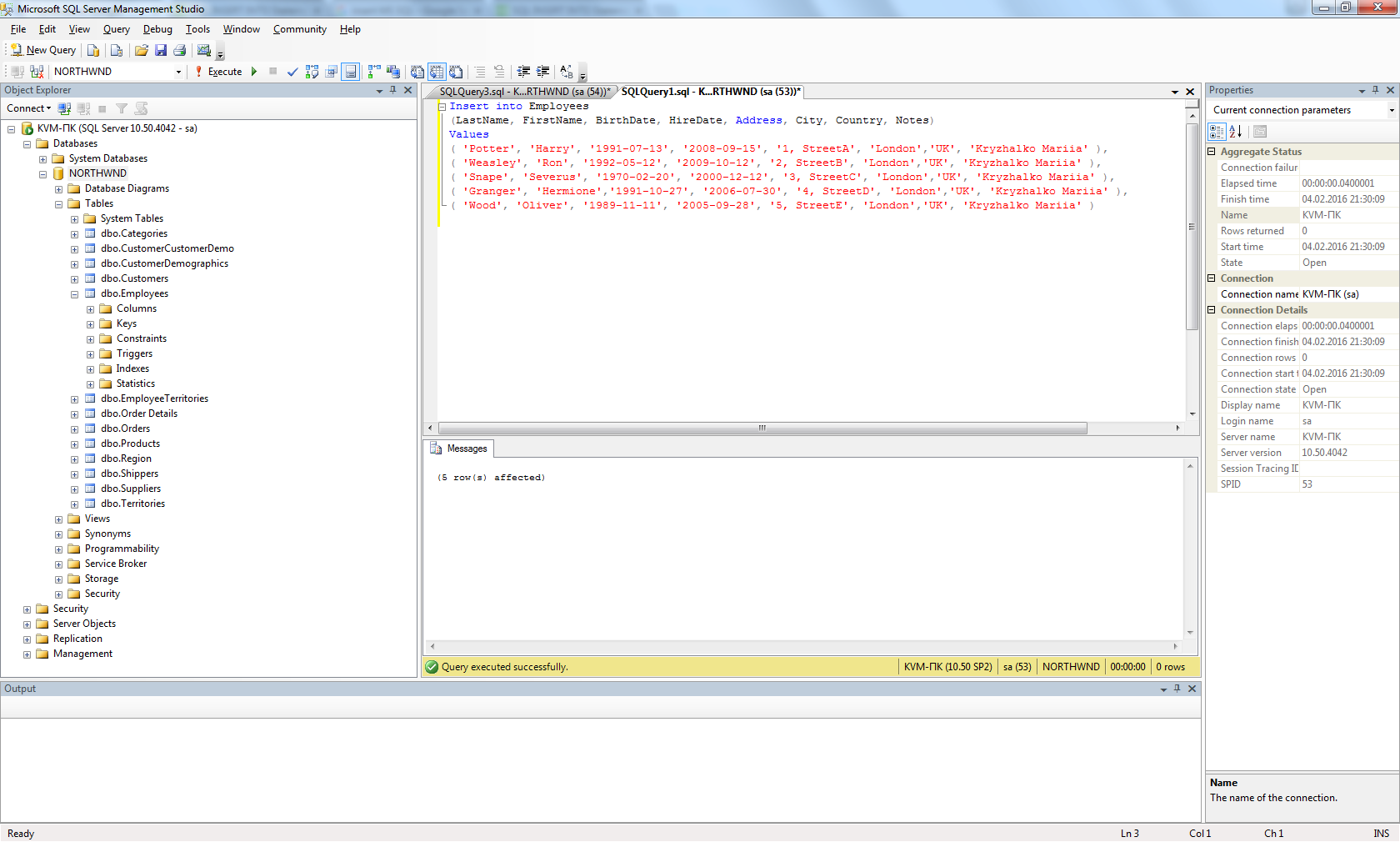
UNION

Select ShipCity

From Orders



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).

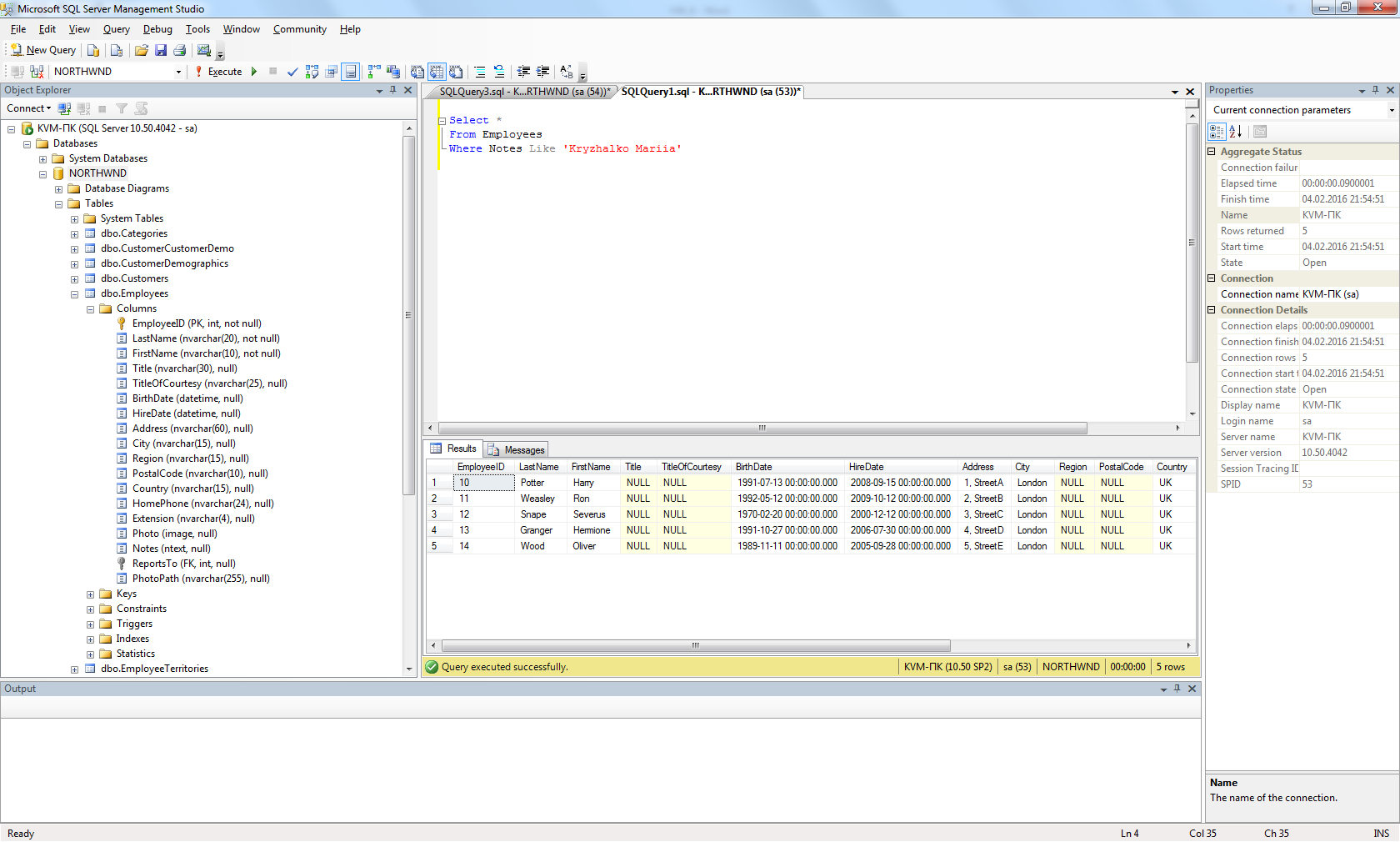


1. \*Fetch the records you have inserted by the SELECT statement

Select \*

From Employees

Where Notes Like 'Kryzhalko Mariia'



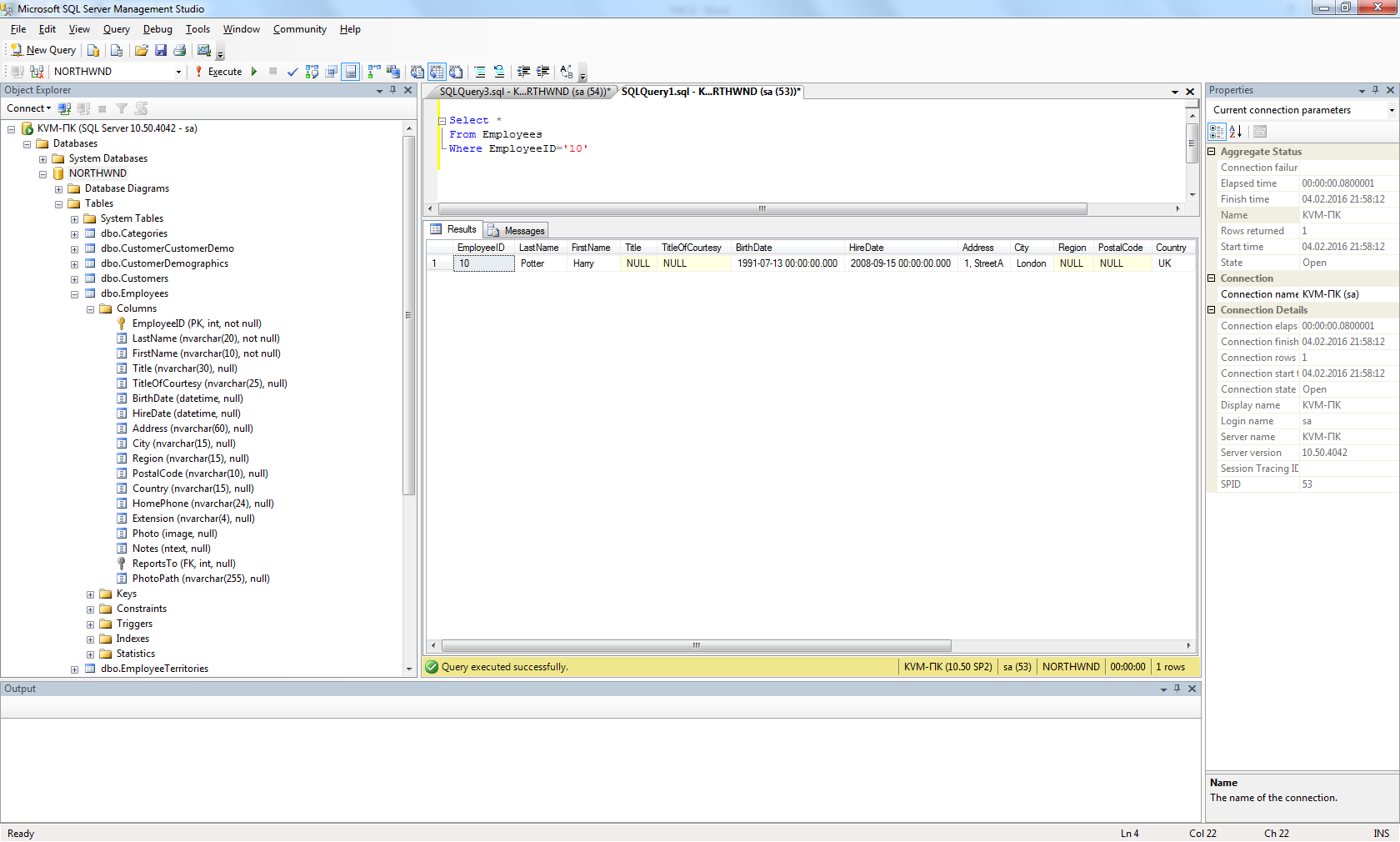
1. \*Change the City field in one of your records using the UPDATE statement (first run the SELECT statement to check whether you are updating the appropriate records!).

1.

Select \*

From Employees

Where EmployeeID='10'



2.

Update Employees

Set City='New York'

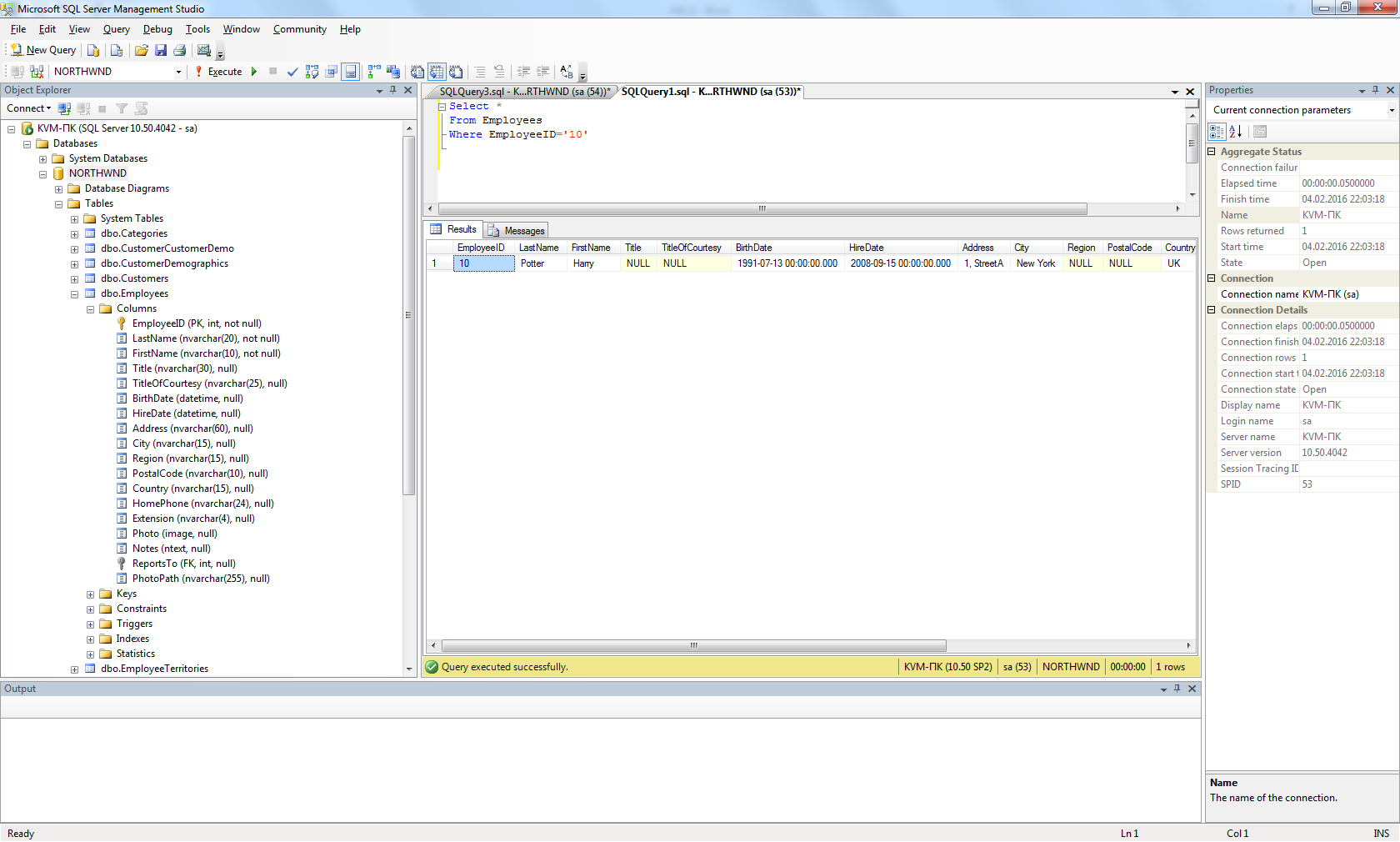
Where EmployeeID='10'

3.

Select \*

From Employees

Where EmployeeID='10'



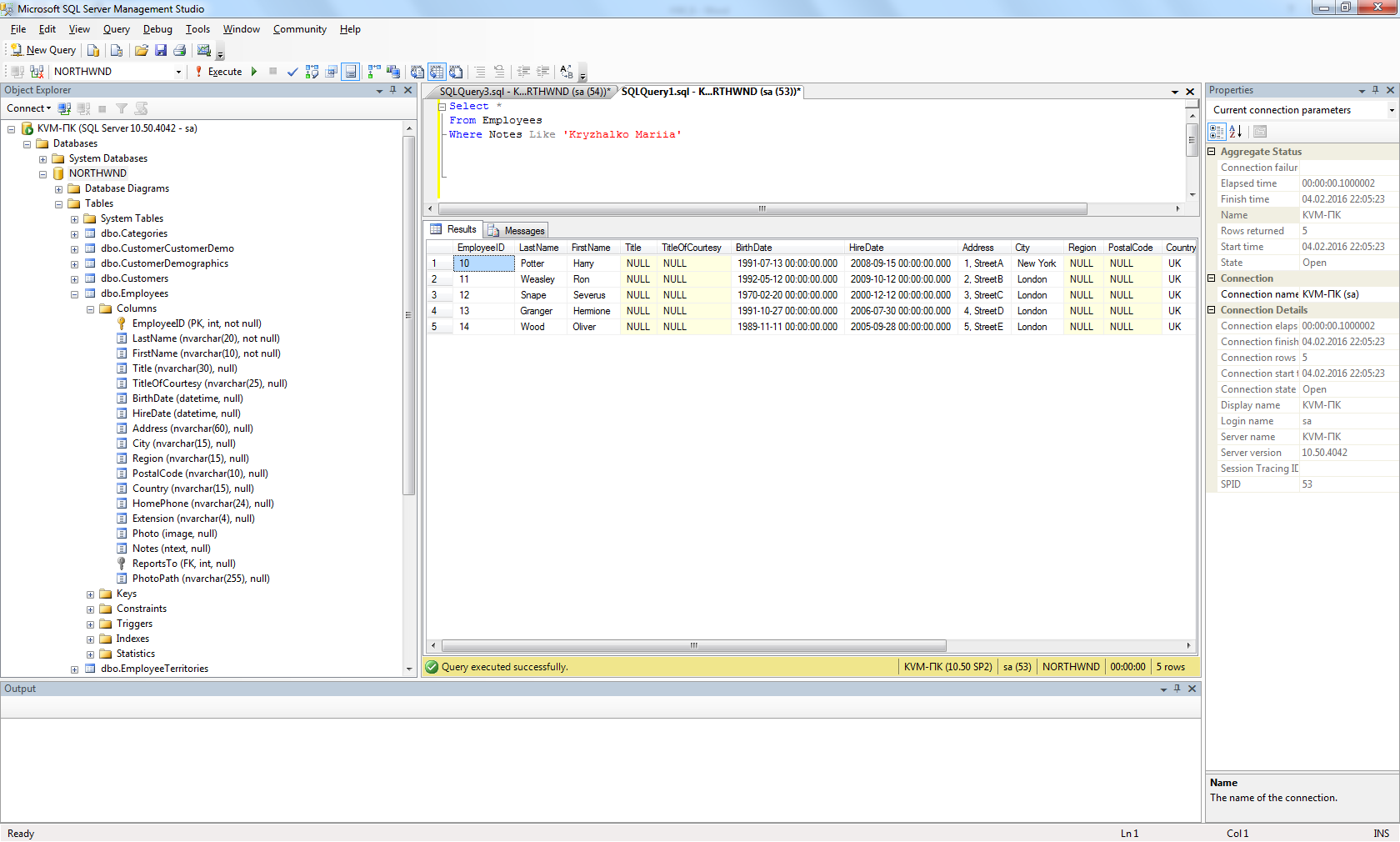
1. \*Change the HireDate field in all your records to current date (first run the SELECT statement to check whether you are updating the appropriate records!).

1.

Select \*

From Employees

Where Notes Like 'Kryzhalko Mariia'



2.

Update Employees

Set HireDate= GETDATE()

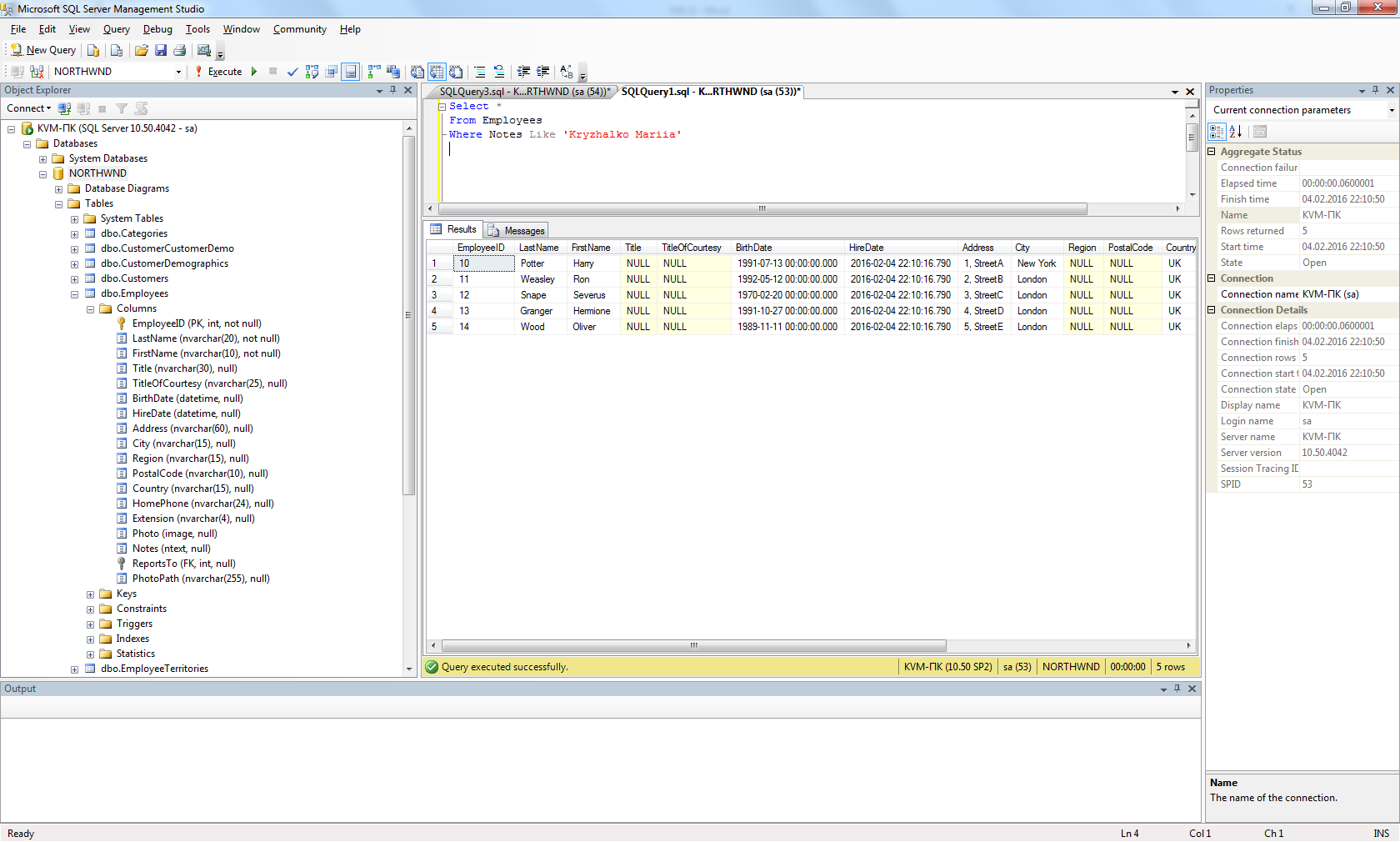
Where Notes like 'Kryzhalko Mariia'

3.

Select \*

From Employees

Where Notes Like 'Kryzhalko Mariia'



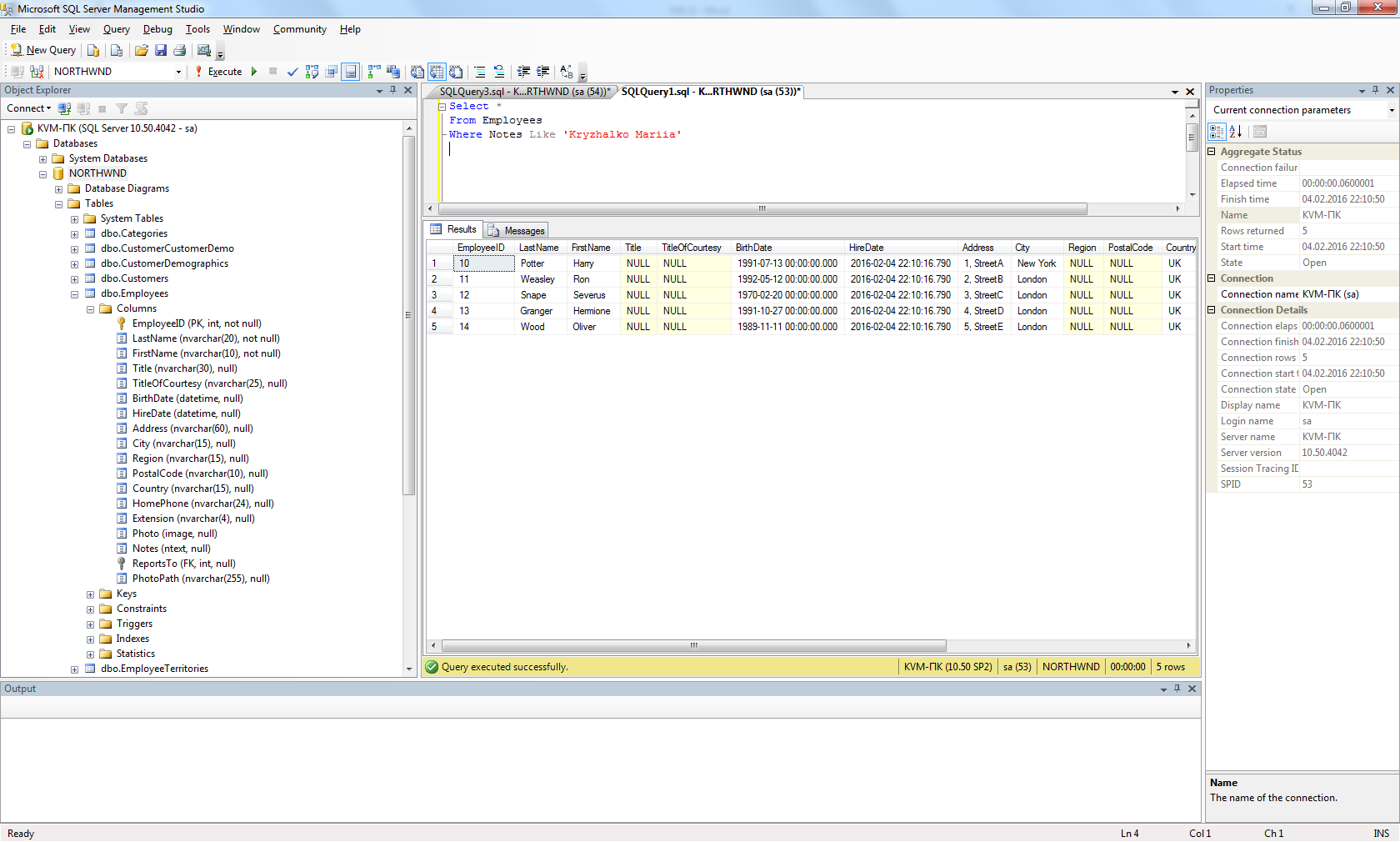
1. \*Delete one of your records (first run the SELECT statement to check whether you are deleting the appropriate record!)

1.

Select \*

From Employees

Where Notes Like 'Kryzhalko Mariia'



2.

Delete from Employees

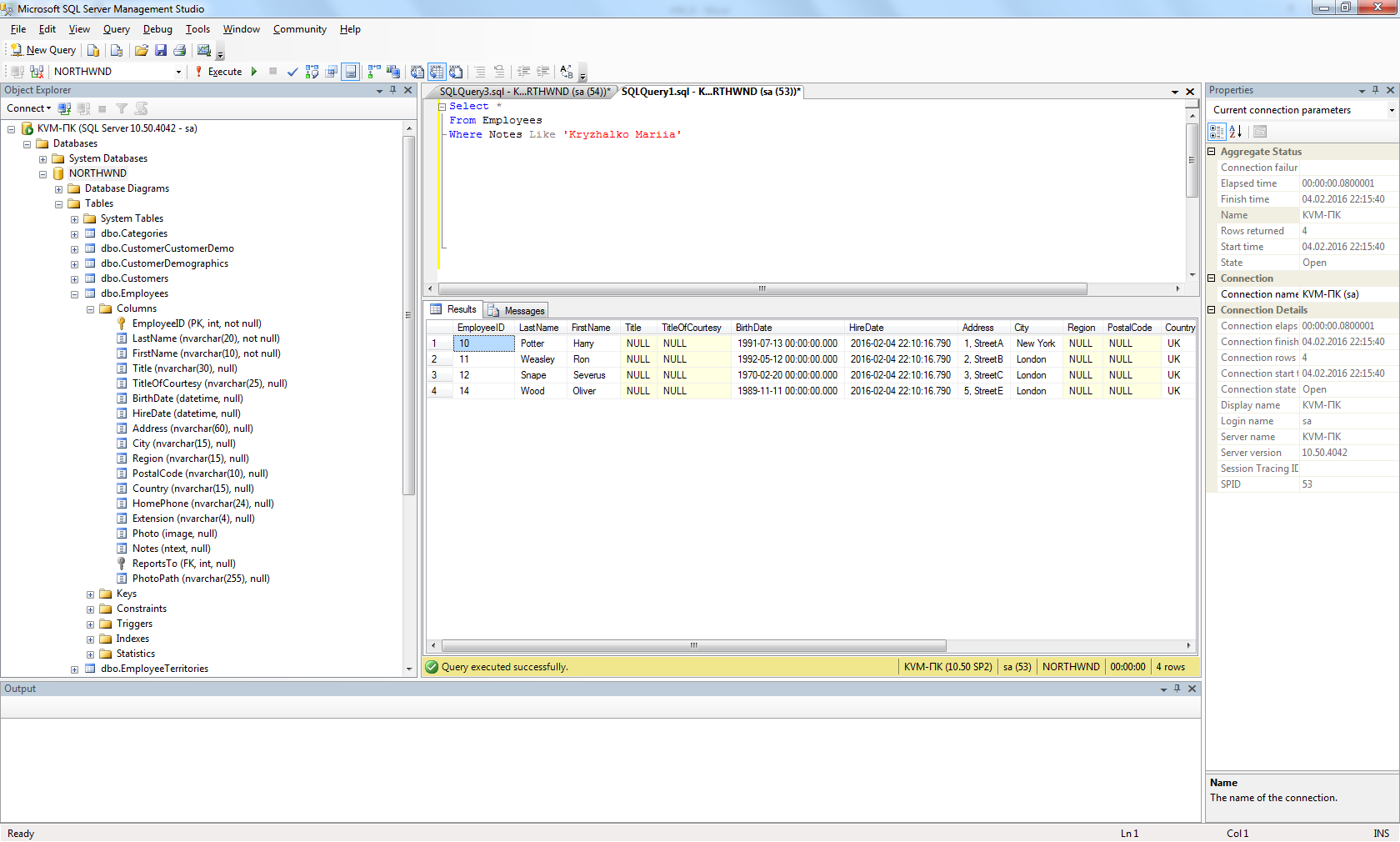
Where EmployeeID=13

3.

Select \*

From Employees

Where Notes Like 'Kryzhalko Mariia'



# 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.