LAB 4

**1.** Create VendorCopy table and InvoiceCopy table.

**Comment:**

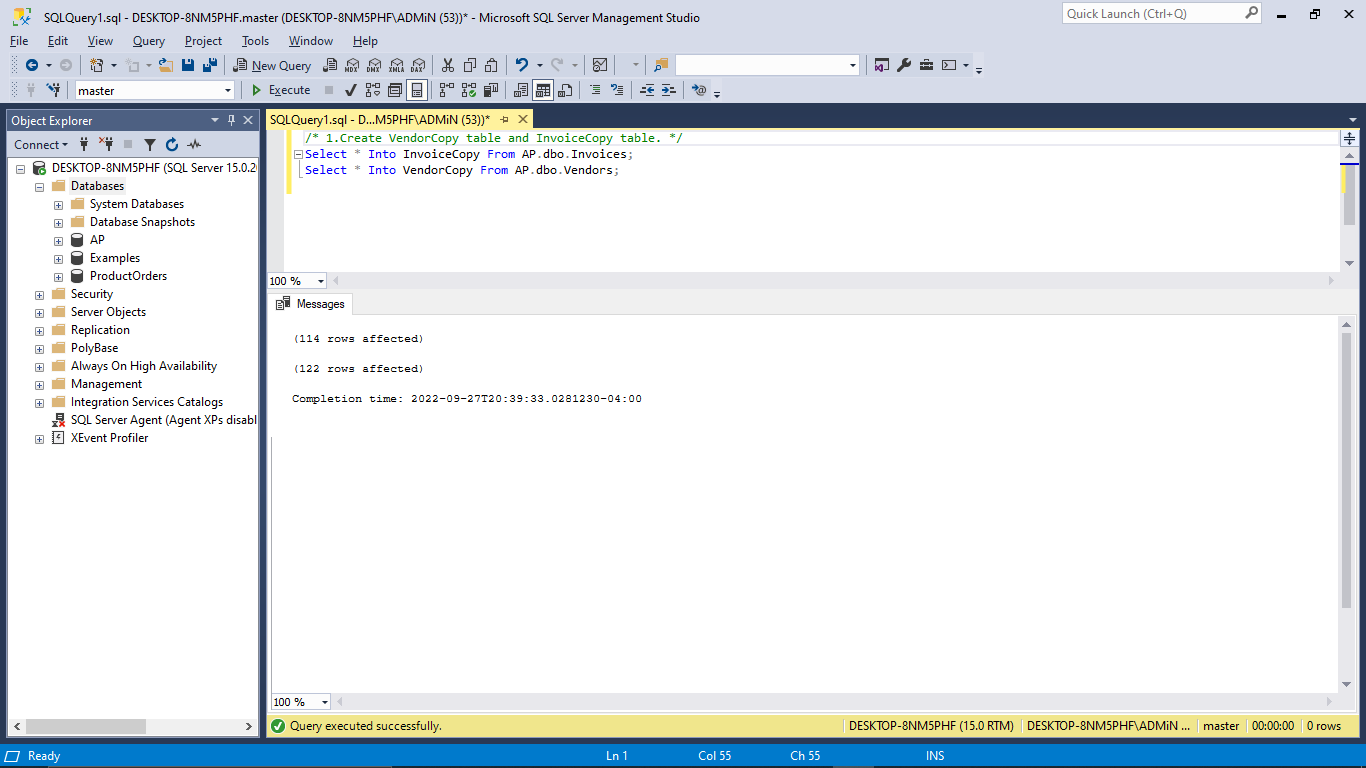
* Using the INTO Clause in the SELECT statement, two tables, VendorCopy and InvoiceCopy, holding the same data of Vendors and Invoice tables, are formed.

**Query:**

Select \* Into VendorCopy From AP.dbo.Vendors;

Select \* Into InvoiceCopy From AP.dbo.Invoices;

**Output:**

****

**2.** Write an INSERT statement that adds a row to the InvoiceCopy table with the following values (USE SELECT statement to verify data changes in the table before and after the modification):

VendorID: 4,

InvoiceNumber: UN-004-400,

InvoiceDate: 10/01/22,

InvoiceTotal: $750.48,

PaymentTotal: $100.00,

CreditTotal: $7.50,

TermsID: 2,

InvoiceDueDate: 12/01/22.

Do we explicitly need to have an InvoiceID to insert?

**Comment:**

* INSERT clause is used in the SELECT query to insert a new record into the InvoiceCopy table.
* There is **no need** to manually insert a row using InvoiceID because the identity specifications from the original Invoice table will be replicated to the InvoiceCopy table.

**Query:**

-- *Changes in the table before modification:*

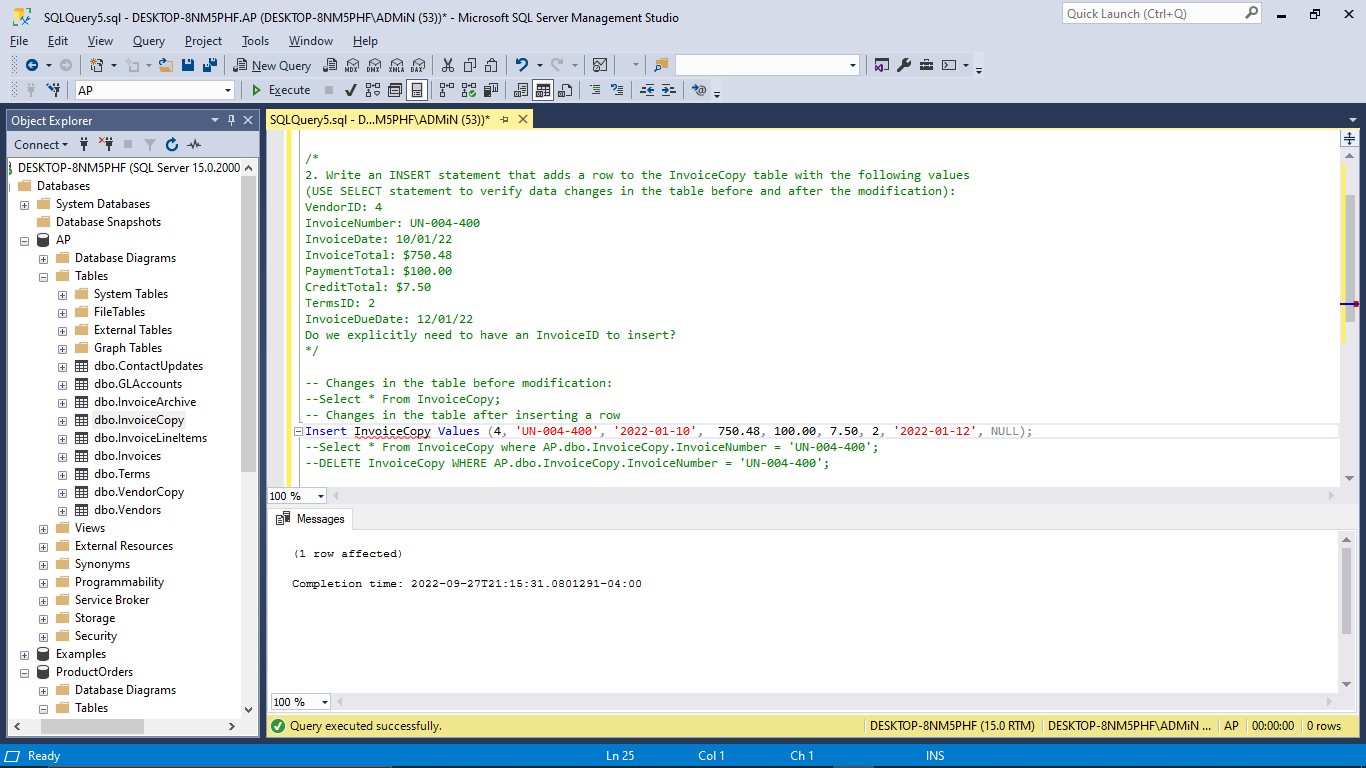
Select \* From InvoiceCopy;

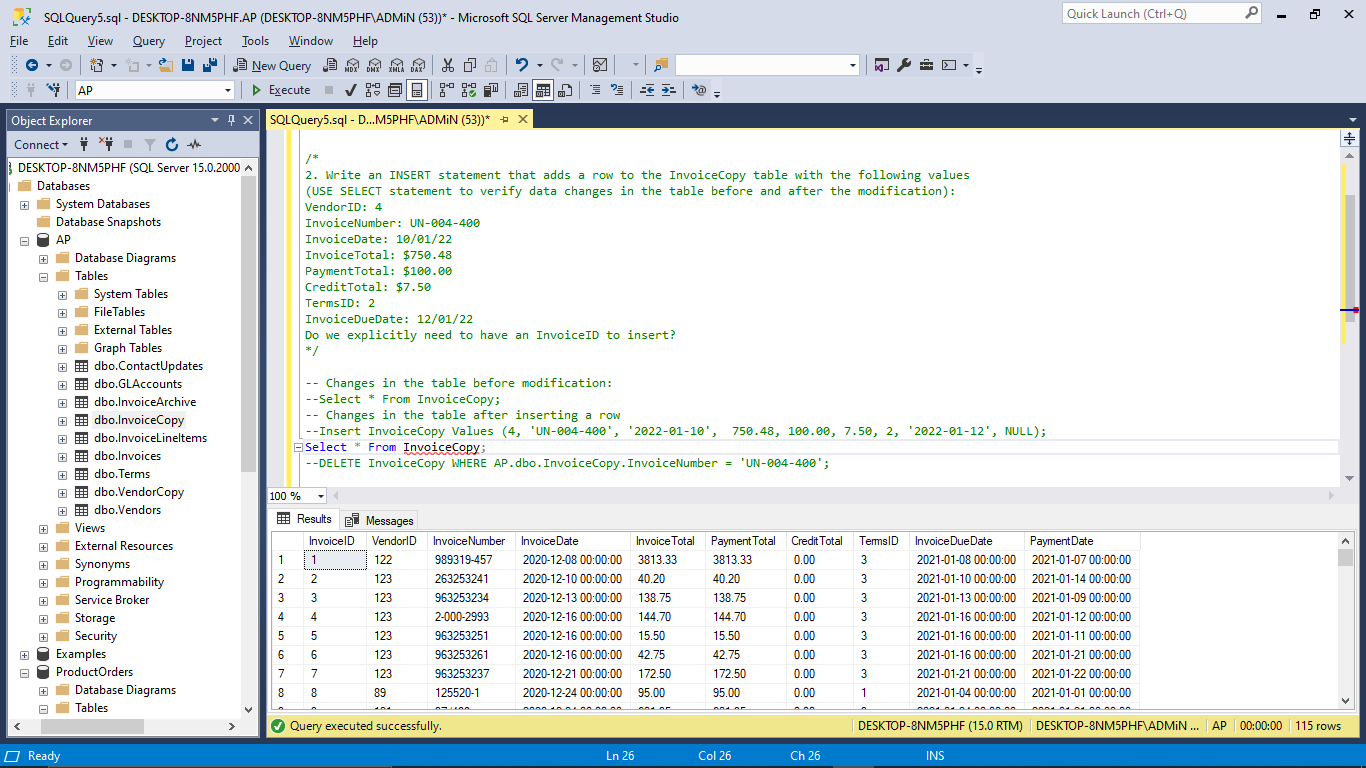
-- *Changes in the table after inserting a row:*

Insert InvoiceCopy Values (4, 'UN-004-400', '2022-01-10', 750.48, 100.00, 7.50, 2, '2022-01-12', NULL);

**Output:**

*After Inserting a row*:

****

****

**3.** Write an UPDATE statement that modifies the VendorCopy table. Change the default account number to 970 for each vendor that has a default account number of 170. (USE SELECT statement to verify data changes in the table before and after the modification).

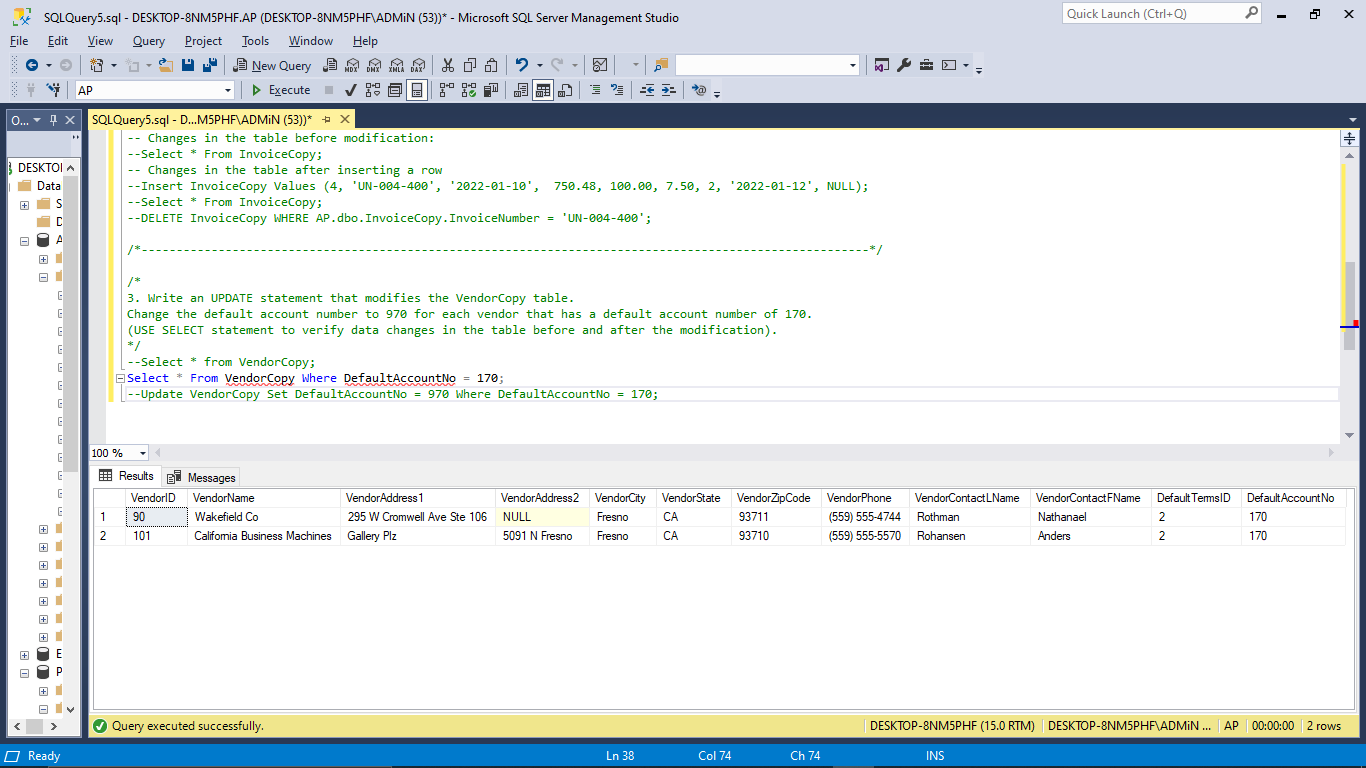
**Comment:**

The UPDATE clause is used in conjunction with the SELECT statement to modify the account number of each vendor in the VendorCopy Table from 170 to 970.

**Output:**

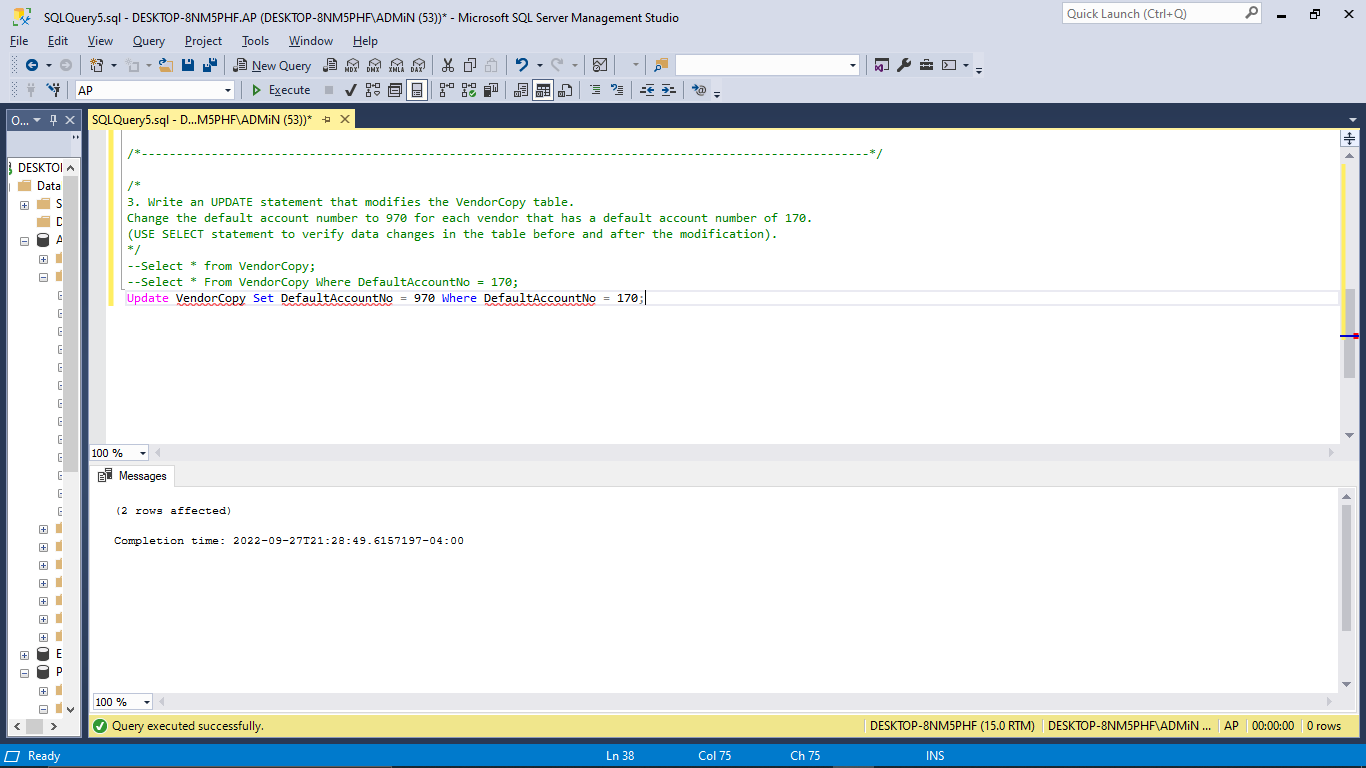
* ***Before Modification:***

**Query:** Select \* From VendorCopy Where DefaultAccountNo = 170;

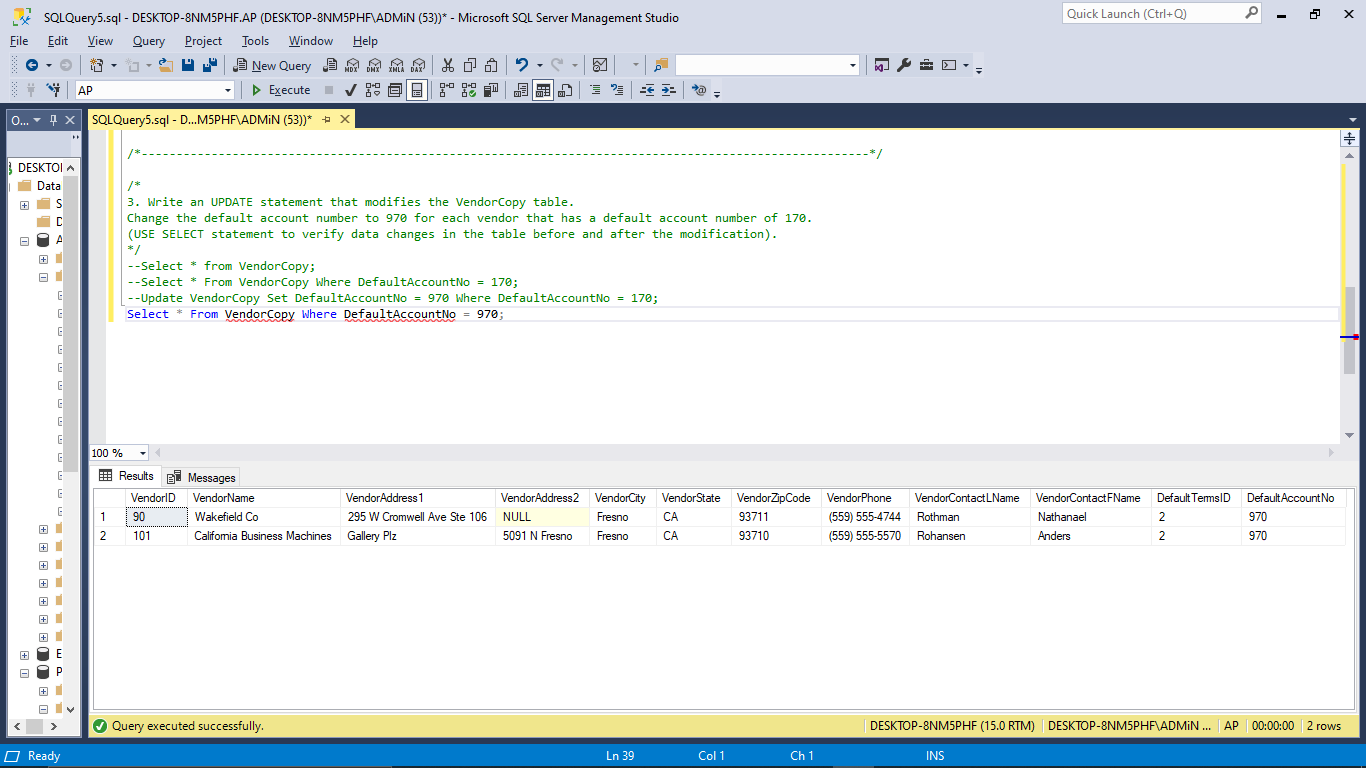
****

* ***After Modification*:**

**Query:** Update VendorCopy Set DefaultAccountNo = 970 Where DefaultAccountNo = 170;

****

**Query:** Select \* From VendorCopy Where DefaultAccountNo = 970;

****

**4.** Write an UPDATE statement that modifies the InvoiceCopy table. Change the TermsID to 3 for each invoice that’s from a vendor with a defaultTermsID of 2. Use a subquery. (USE SELECT statement to verify data changes in the table before and after the modification).

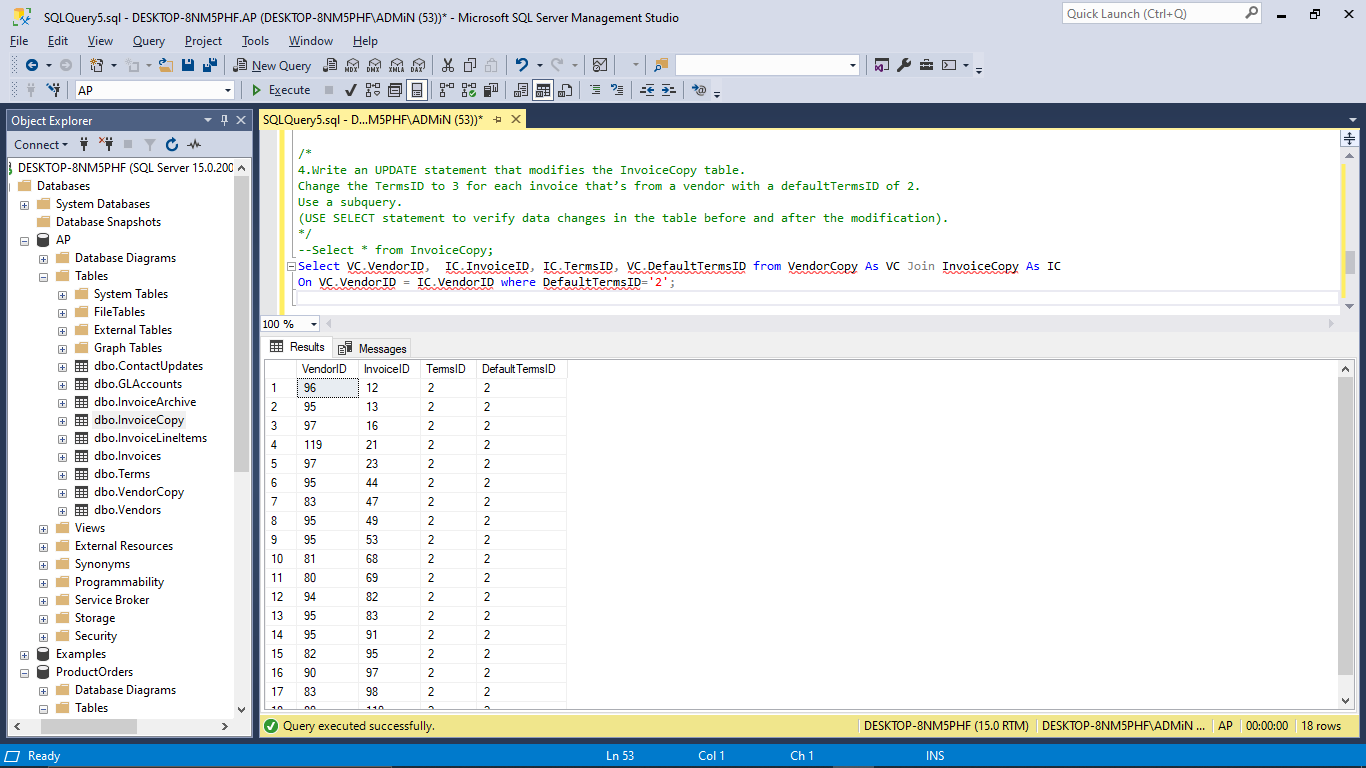
**Comment :**

* The UPDATE statement is used to modify the InvoiceCopy table.
* Using a subquery, changed the TermsID to 3 for each invoice from a vendor with a default TermsID of 2.

**Output:**

* **Before Modification:**

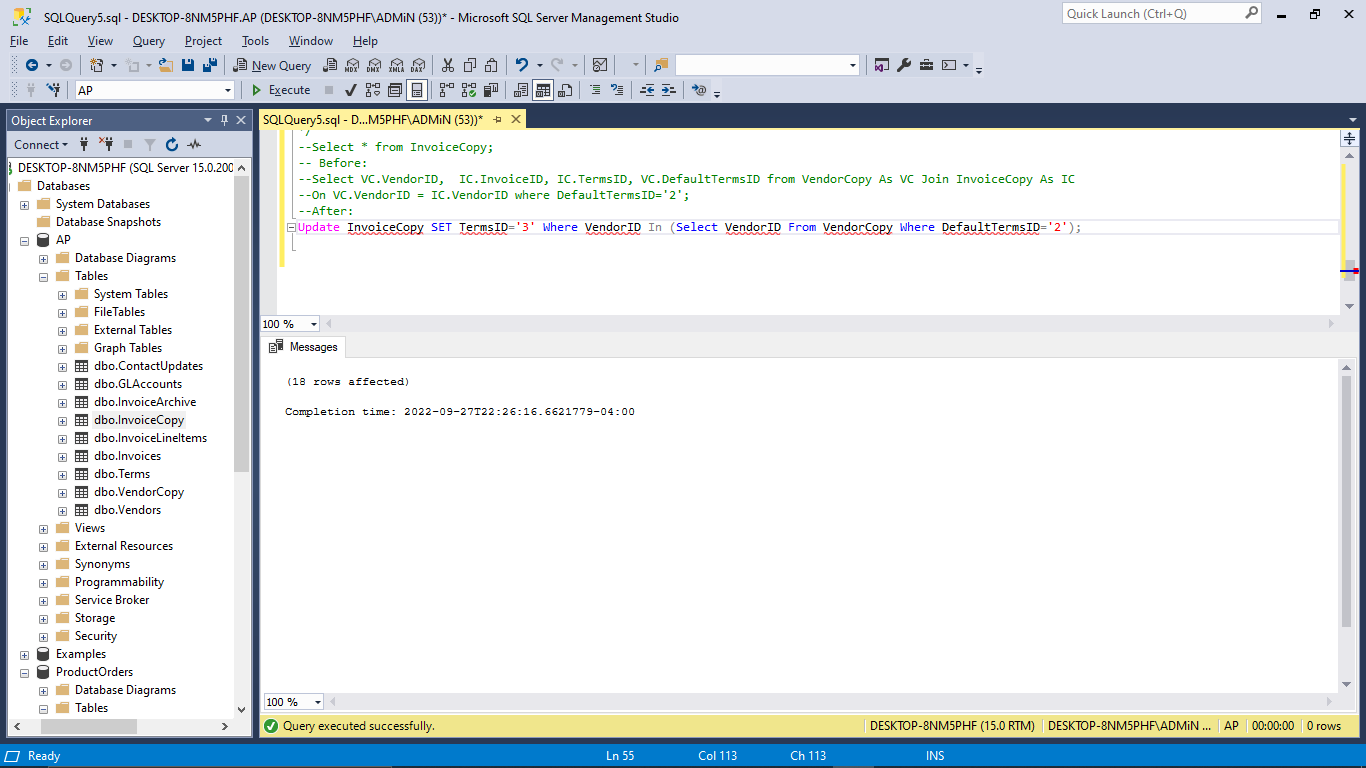
**Query:** Select VC.VendorID, IC.InvoiceID, IC.TermsID, VC.DefaultTermsID from VendorCopy As VC Join InvoiceCopy As IC On VC.VendorID = IC.VendorID where DefaultTermsID='2';

****

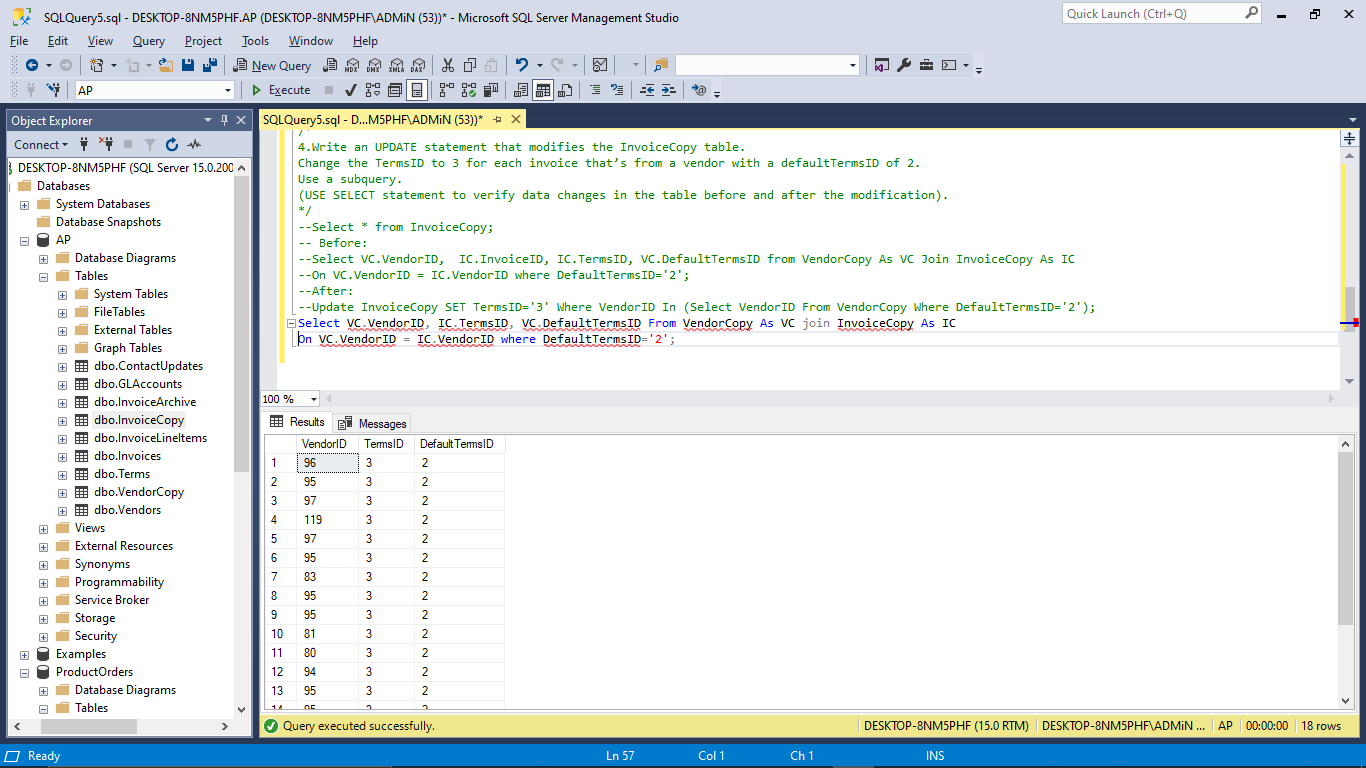
* **After Modification:**

**Query:** Update InvoiceCopy SET TermsID='3' Where VendorID In (Select VendorID From VendorCopy Where DefaultTermsID='2');

**Output:**

****

**Query:** Select VC.VendorID, IC.TermsID, VC.DefaultTermsID From VendorCopy As VC join InvoiceCopy As IC On VC.VendorID = IC.VendorID where DefaultTermsID='2';

****

**5.** Write a DELETE statement that deletes all vendors in the state of Illinois from the VendorCopy table. (USE SELECT statement to verify data changes in the table before and after the modification).

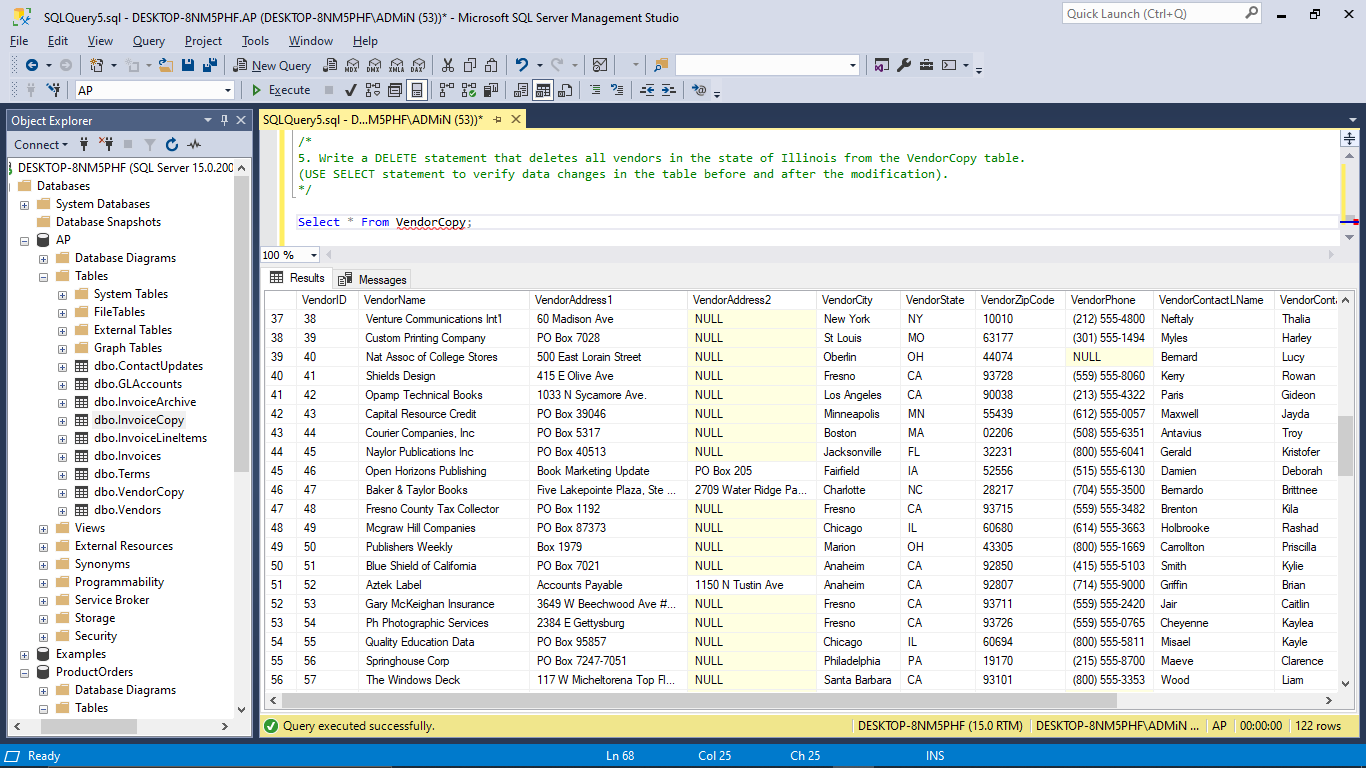
**Comment:**

* The DELETE statement deletes all vendors in the state of Illinois from the VendorCopy database.

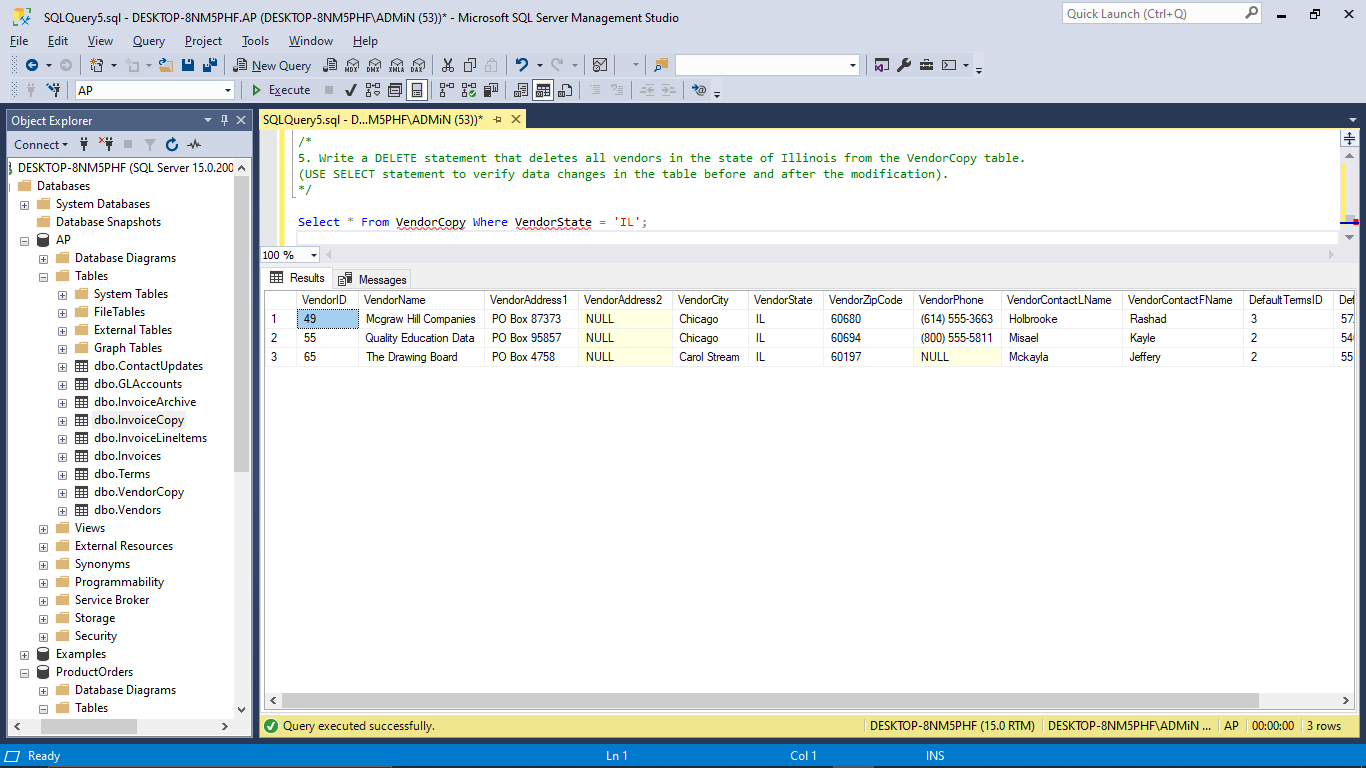
**Output:**

* **Before Modification:**

**Query:** Select \* From VendorCopy;

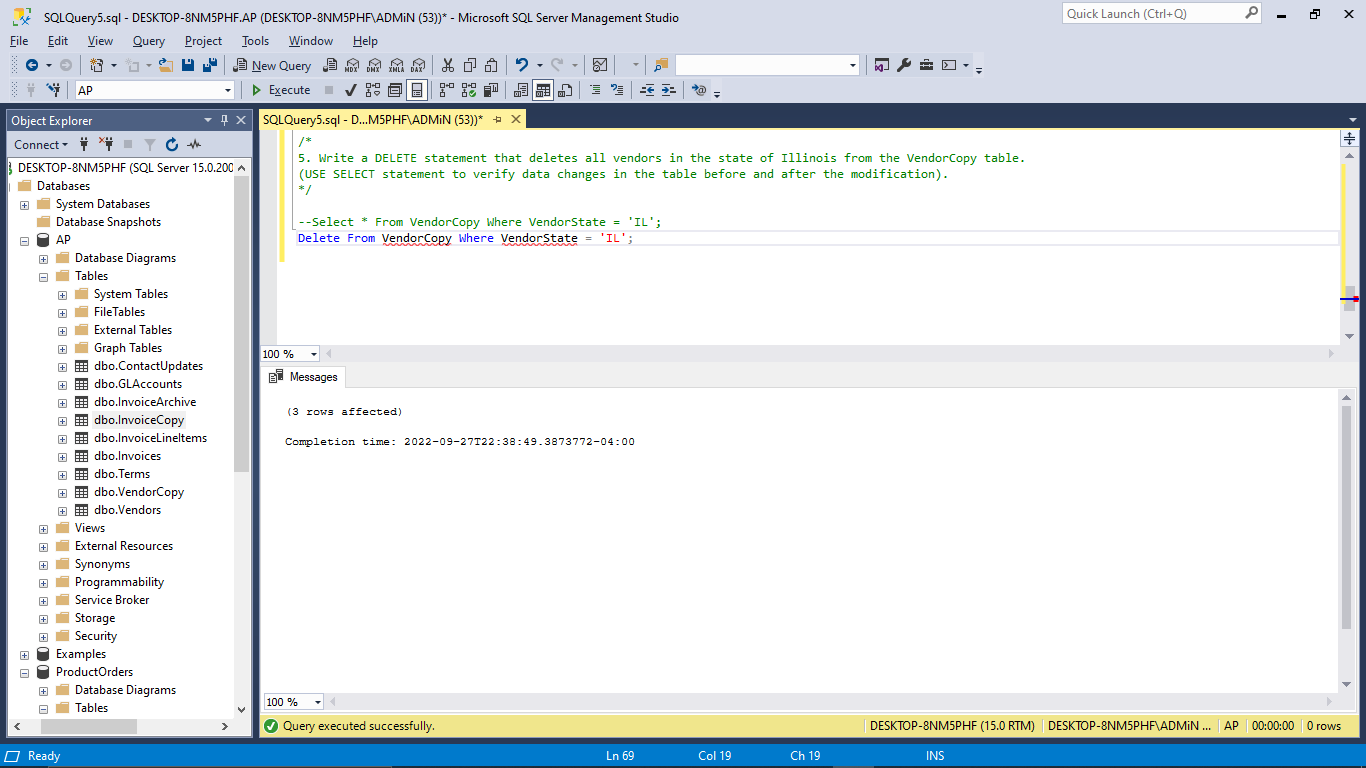
****

**Query:** Select \* From VendorCopy Where VendorState = 'IL';

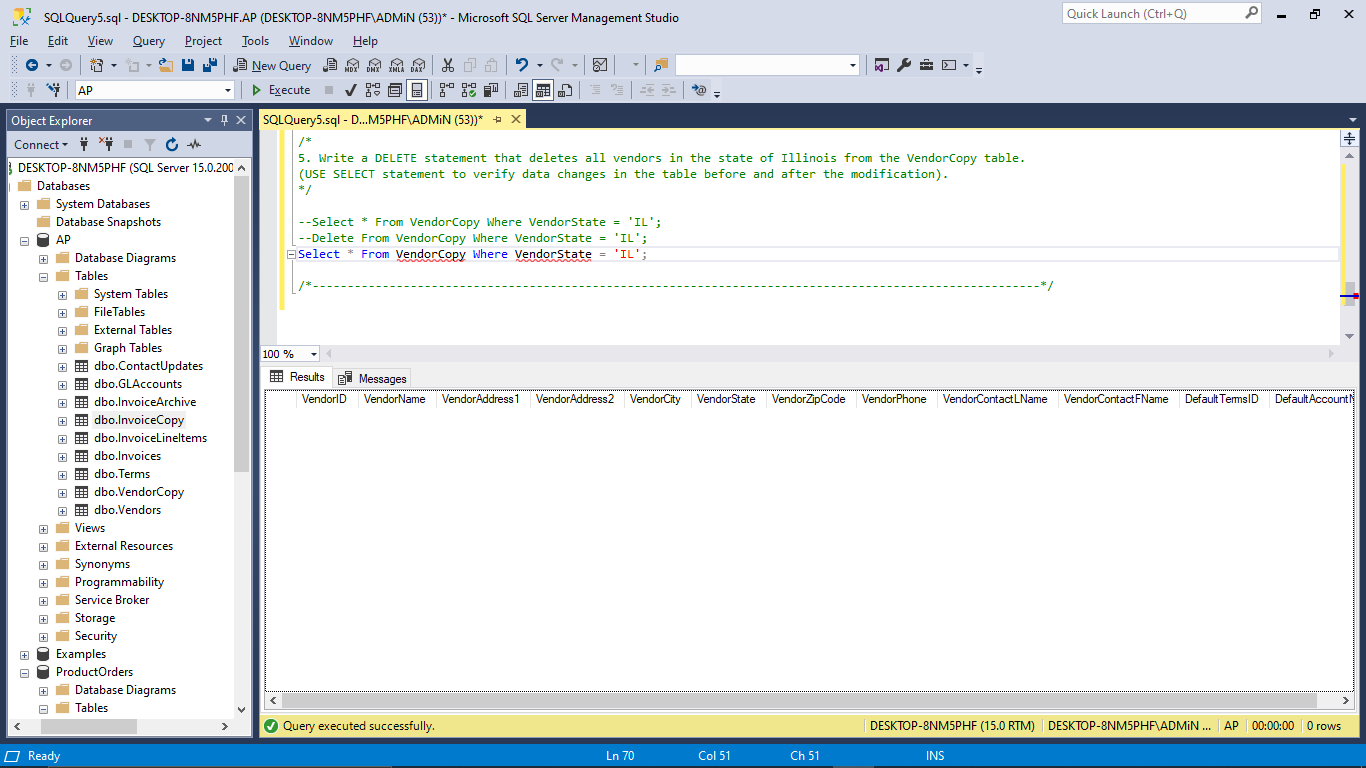


* **After Modification:**

**Query**: Delete From VendorCopy Where VendorState = 'IL';



**Query:** Select \* From VendorCopy Where VendorState = 'IL';



**6.** Write a DELETE statement for the VendorCopy table. Delete the vendors that are located in states from which no vendor has ever sent an invoice. (USE SELECT statement to verify data changes in the table before and after the modification).

**Comment:**

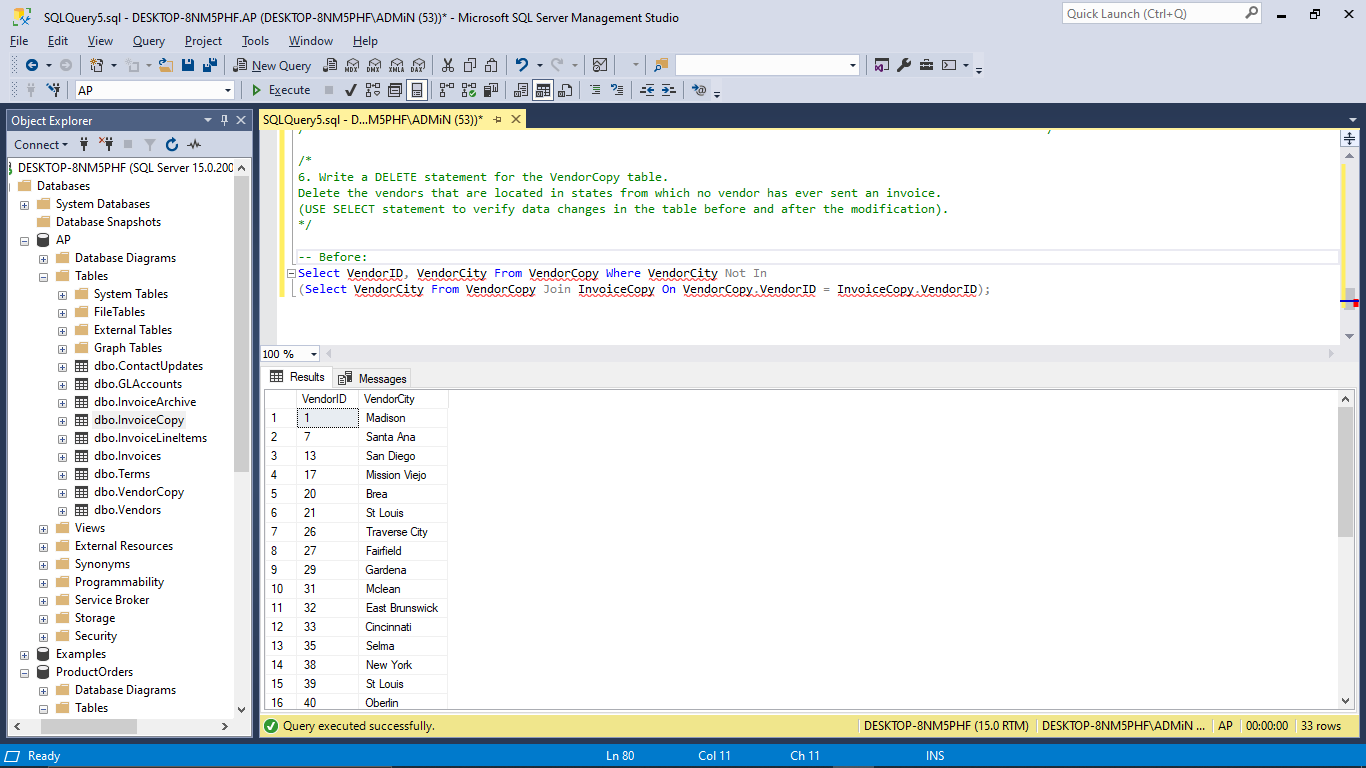
* The DELETE statement is used to delete vendors located in states where no vendor has ever sent an invoice.

**Output:**

* **Before Modification:**

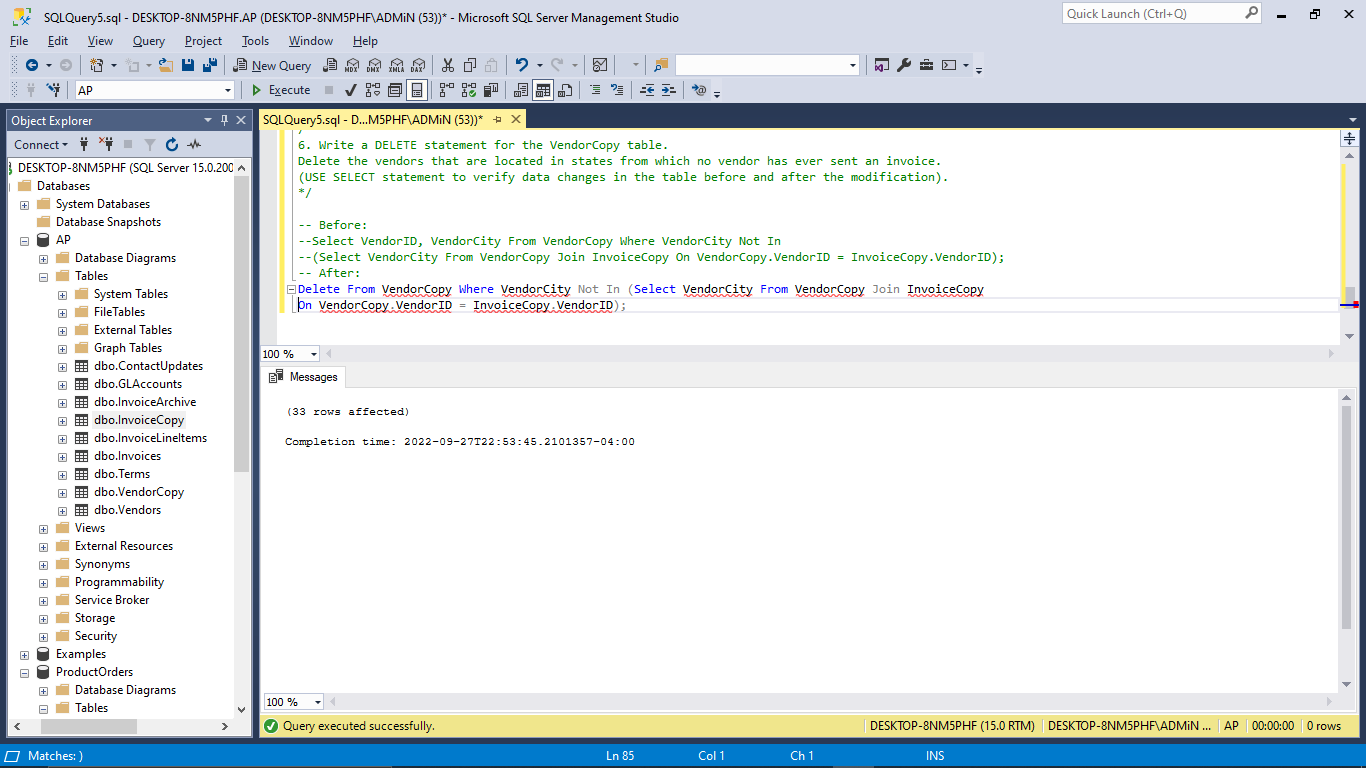
**Query:** Select VendorID, VendorCity From VendorCopy Where VendorCity Not In

(Select VendorCity From VendorCopy Join InvoiceCopy On VendorCopy.VendorID = InvoiceCopy.VendorID);

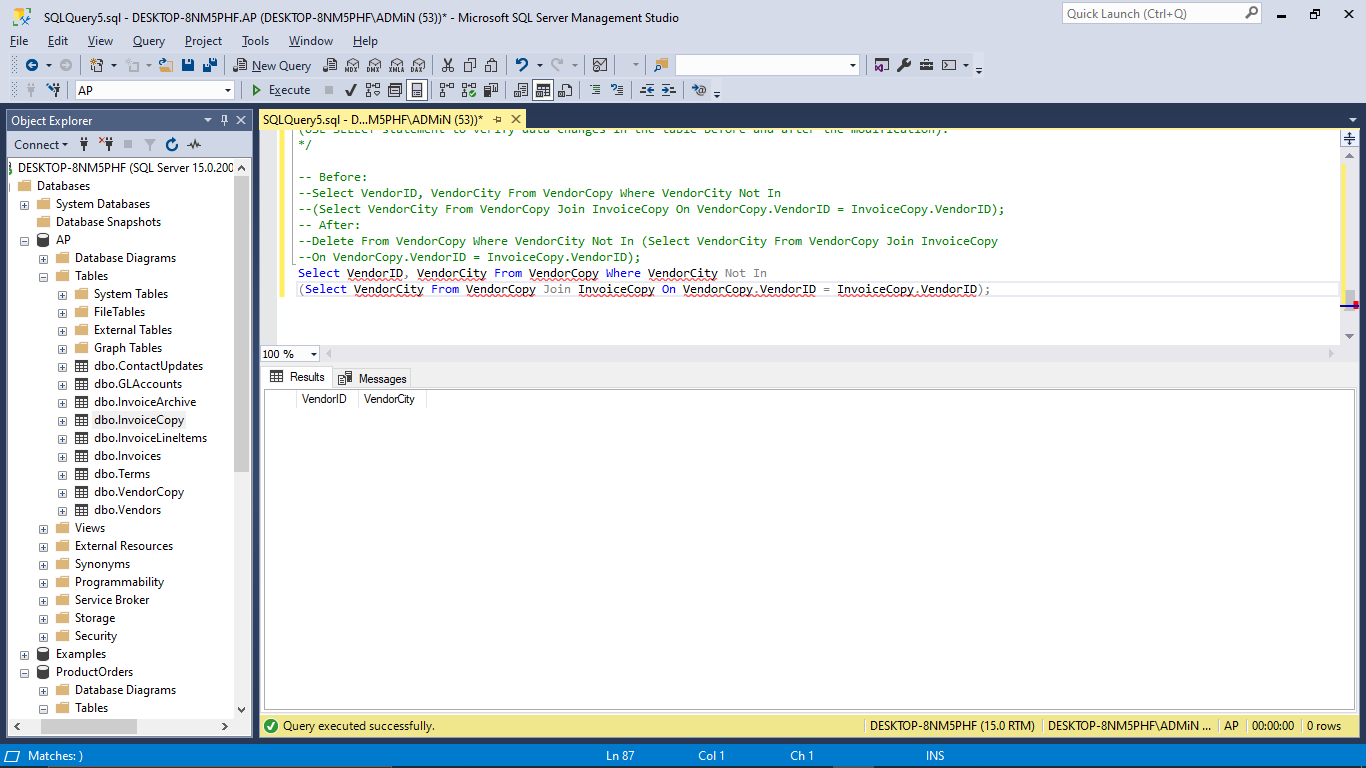
****

* **After Modification:**

**Query:** Delete From VendorCopy Where VendorCity Not In (Select VendorCity From VendorCopy Join InvoiceCopy On VendorCopy.VendorID = InvoiceCopy.VendorID);



**Query:** Select VendorID, VendorCity From VendorCopy Where VendorCity Not In (Select VendorCity From VendorCopy Join InvoiceCopy On VendorCopy.VendorID = InvoiceCopy.VendorID);



**7.** Write a SELECT statement that returns four columns based on the PaymentTotal column of the Invoices table: Use CAST function to return the first column as data type decimal with 2 digits to the right of the decimal point. Use CAST to return the second column as a VARCHAR. Use CONVERT function to return third column as the same type as the first column. Use CONVERT to return the fourth column as a VARCHAR, using style 5.

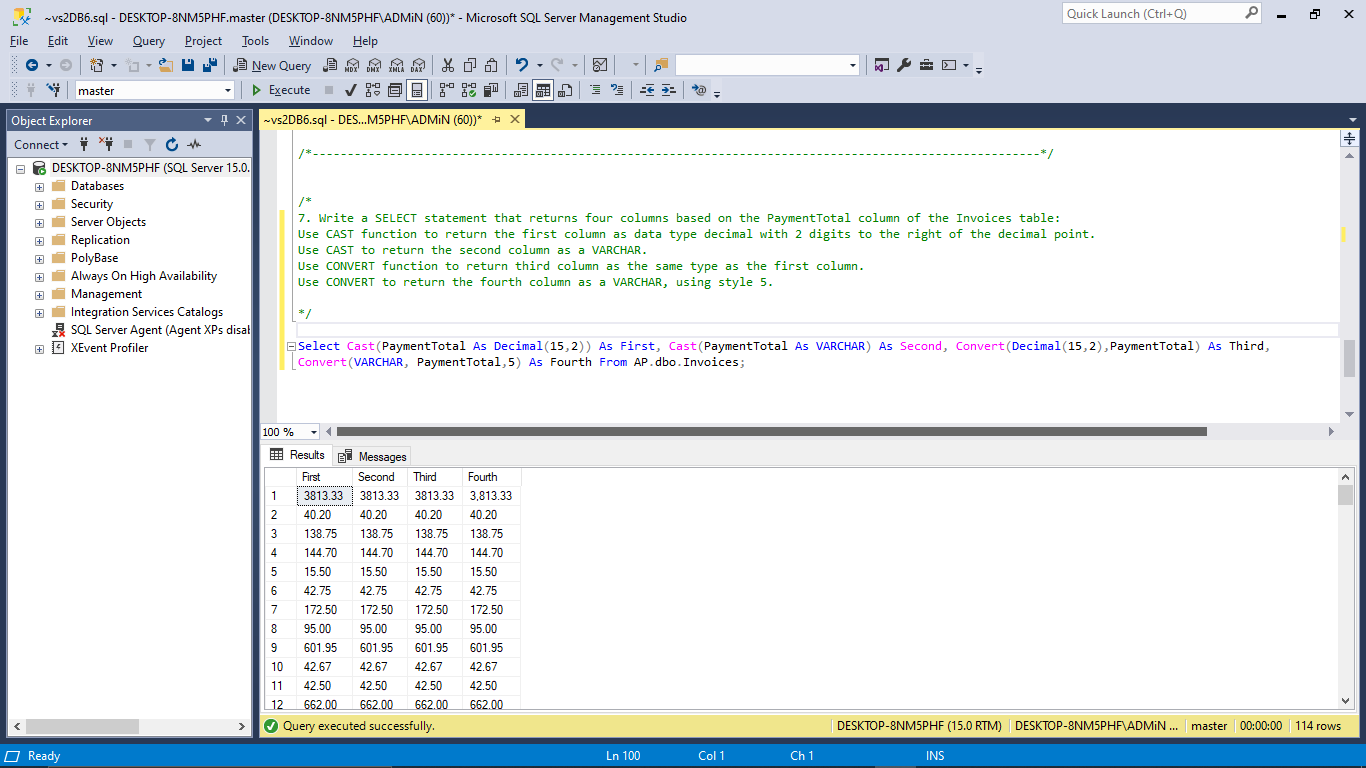
**Comment:**

* Two functions namely, CAST and CONVERT are used seperately on PaymentTotal column of Invoice Table to convert the datatype of it into DECIMAL and VARCHAR such that they create 4 columns with 2 digits to the right of the decimal point.
* *114 rows* are created while executing the query.

**Query:**

Select Cast(PaymentTotal As Decimal(15,2)) As First, Cast(PaymentTotal As VARCHAR) As Second, Convert(Decimal(15,2),PaymentTotal) As Third, Convert(VARCHAR, PaymentTotal,5) As Fourth From AP.dbo.Invoices;

**Output:**

****

**8.** Write a SELECT statement that returns four columns based on the PaymentDate column of the Invoices table: Use the CAST function to return the first column as data type VARCHAR.Use the CONVERT function to return the second and third columns as a VARCHAR, using style 1 and style 9, respectively.Use the CAST function to return the fourth column as a data type real.

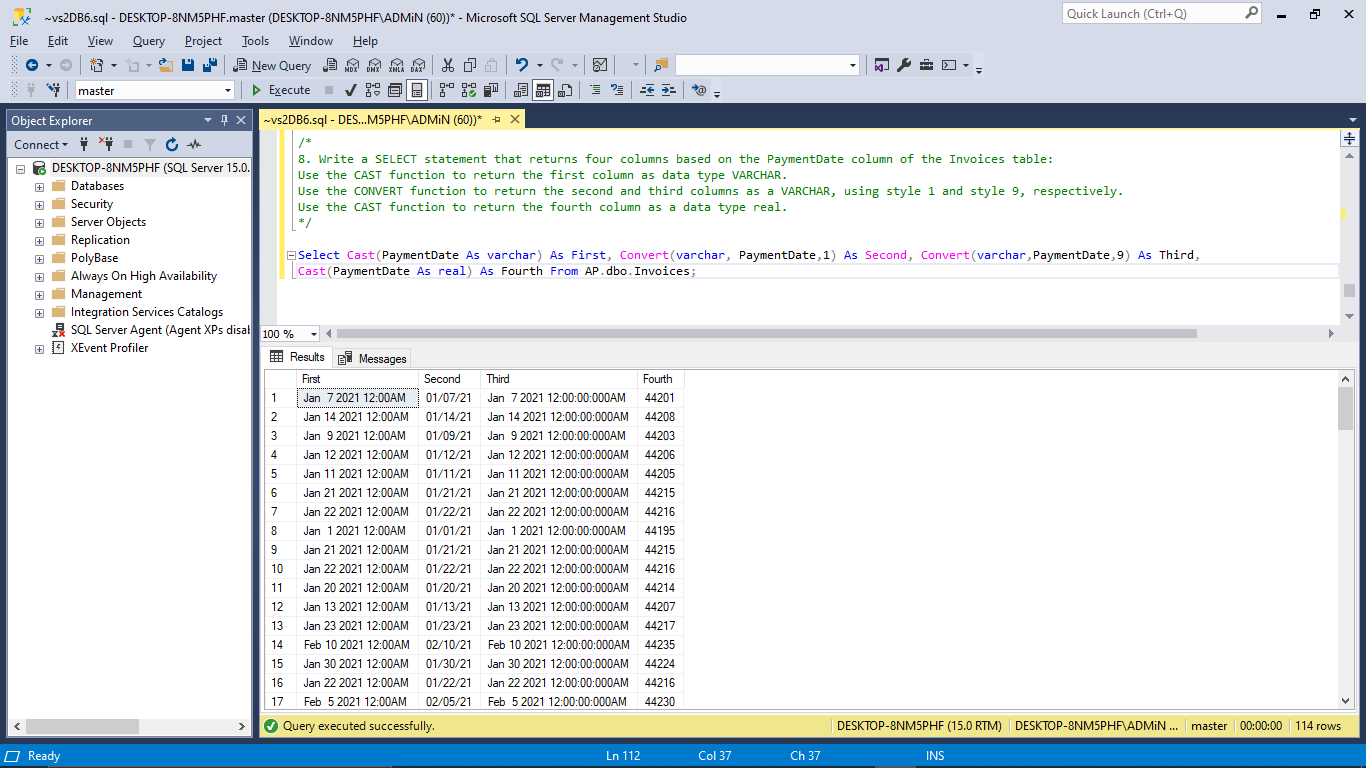
**Comment:**

* Two functions namely, CAST and CONVERT are used seperately on PaymentTotal column of Invoice Table to convert the datatype of it into VARCHAR such that they create 3 columns out of which 2 of them uses style 1 and 9.
* 1 column has the REAL datatype.
* While running this query, 4 columns and *114 rows* are retrieved.

**Query:**

Select Cast(PaymentDate As varchar) As First, Convert(varchar, PaymentDate,1) As Second, Convert(varchar,PaymentDate,9) As Third, Cast(PaymentDate As real) As Fourth From AP.dbo.Invoices;

**Output:**

****

## Remarks for the Lab:

The overall lab consisted the utilization of INSERT, DELETE, INTO, UPDATE etc clauses to create, manipulate and handle the tables.