CSE 581 : INTRODUCTION TO DATABASE MANAGEMENT SYSTEMS

LAB : 4

DATE : 9/28/2022

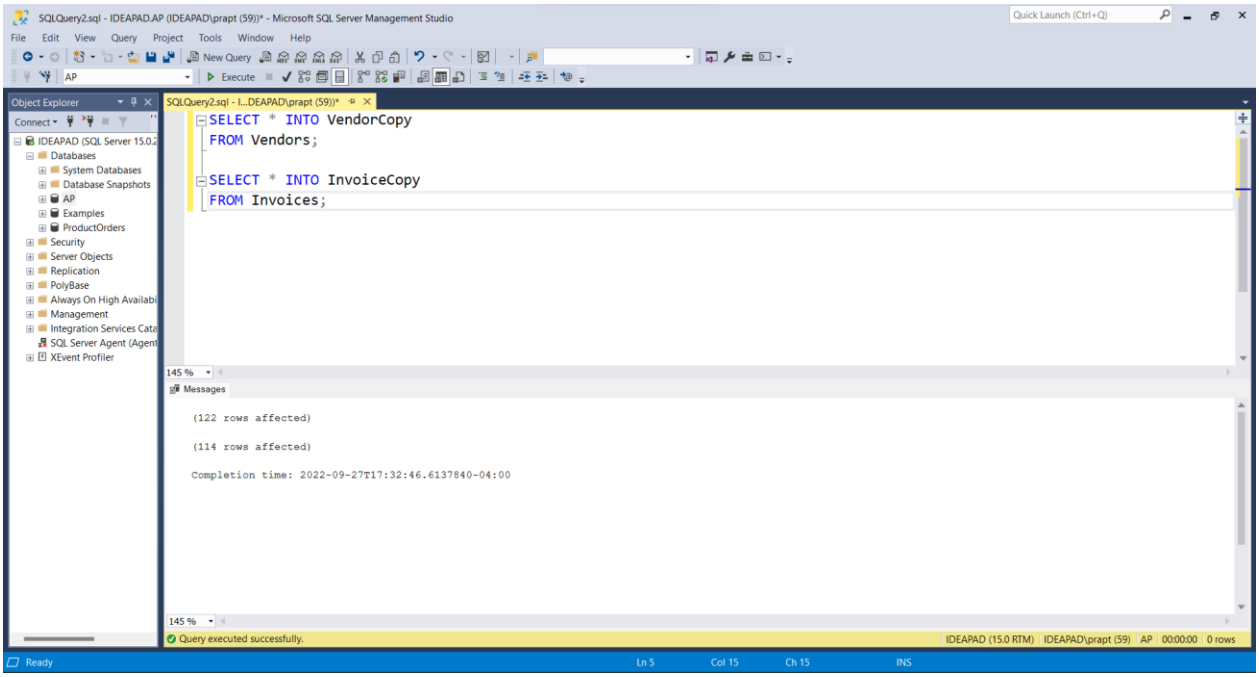
Q1) Create VendorCopy table and InvoiceCopy table.

Ans: SELECT * INTO VendorCopy

FROM Vendors;

SELECT * INTO InvoiceCopy

FROM Invoices;



Comment: Here, SELECT statement is used to select the rows from Vendors and Invoices tables respectively for the two different queries. In the first query, records are selected from the Vendors table and are duplicated/copied into another table named VendorCopy using the SELECT INTO statement. Similarly, in the second query, records are selected from the Invoices table and are duplicated/copied into another table named Invoice Copy using SELECT INTO statement.

Remark: Here, SELECT INTO statement is used to copy the data and the table definition from the specified table.

Q2) 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	InvoiceTotal:	$750.48
TermsID:	2	InvoiceNumber:	UN-004-400
PaymentTotal:	$100.00	InvoiceDueDate:	12/01/22
InvoiceDate:	10/01/22	CreditTotal:	$7.50

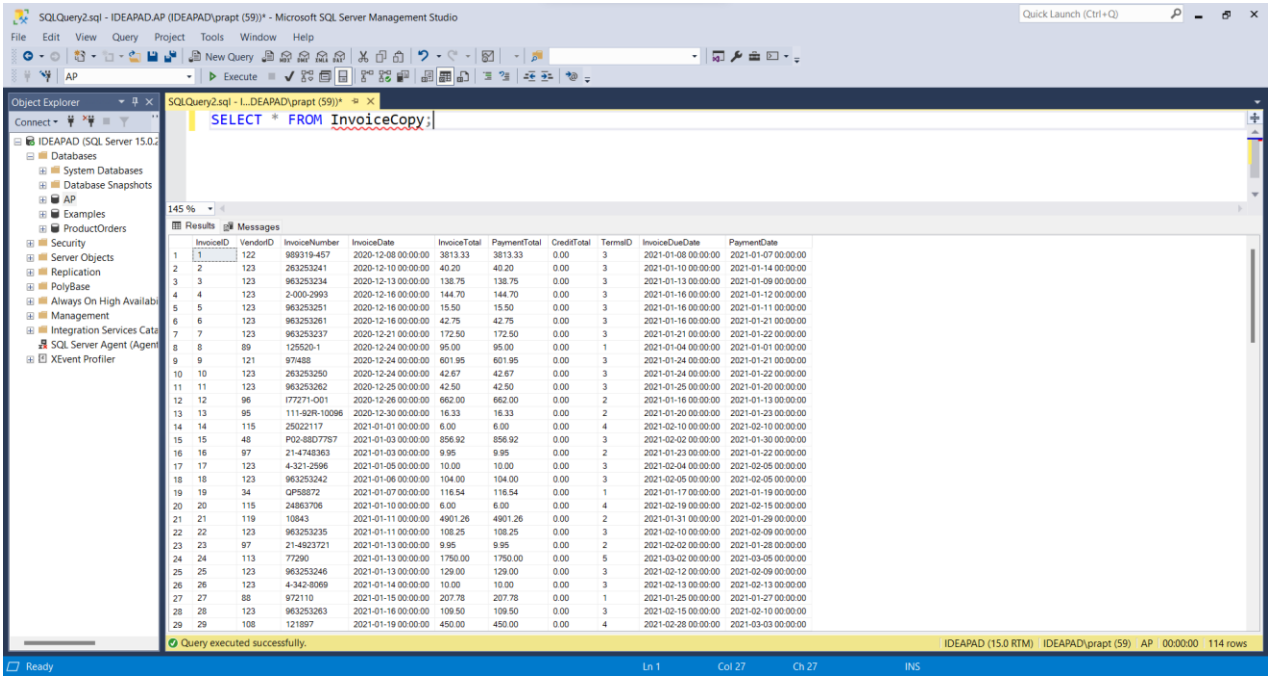
Do we explicitly need to have an InvoiceID to insert?

Ans: INSERT INTO InvoiceCopy

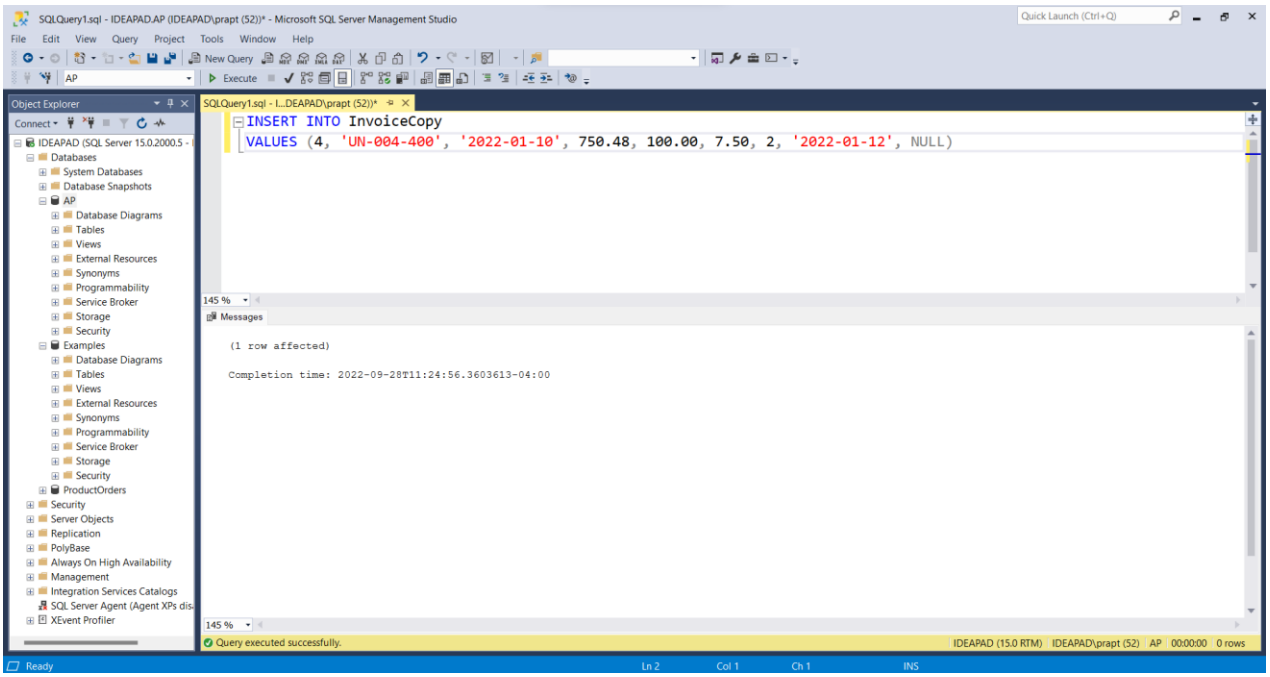
VALUES (4, 'UN-004-400', '2022-01-10', 750.48, 100.00, 7.50, 2, '2022-01-12', NULL)

Before adding the new row:

SELECT * FROM InvoiveCopy

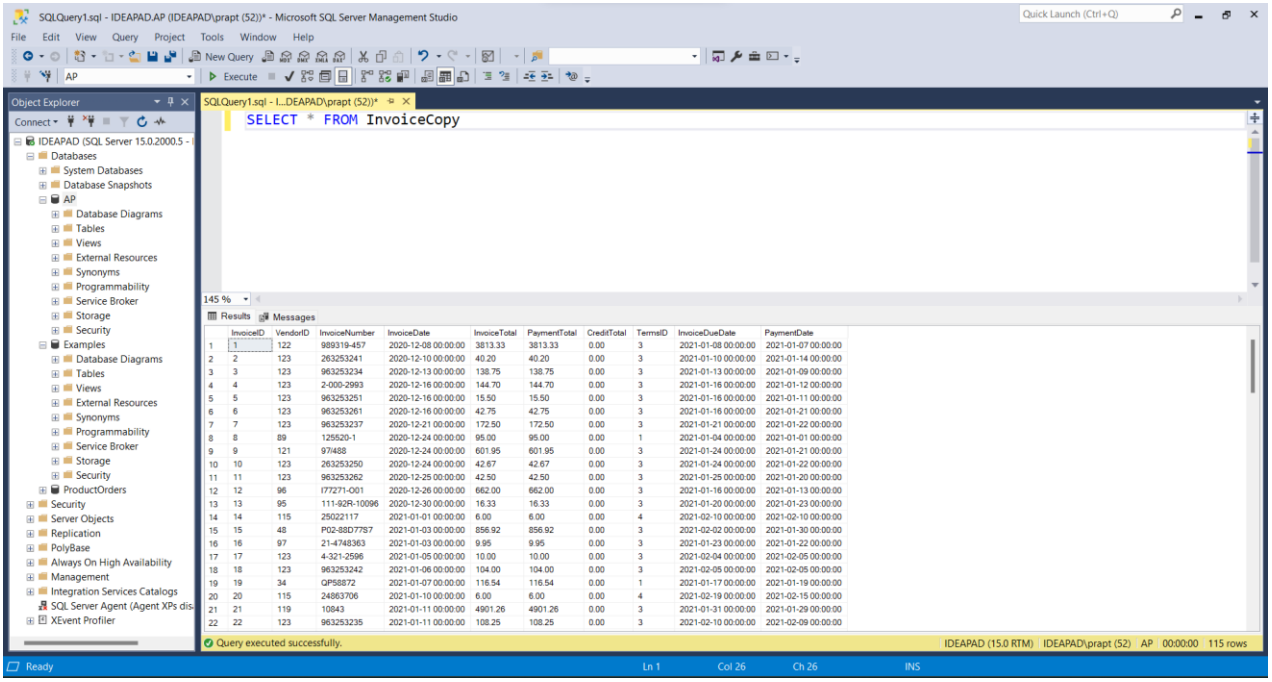


Inserting the new row:



After adding the new row:

SELECT * FROM InvoiveCopy



Comment: Here, INSERT INTO statement is used to insert the given row with the given values into the InvoiceCopy table. Here as we are adding values for all the columns of the table, we do not need to write the column names of the table. The values that are to be

added in the respective columns are written using VALUES keyword and are written in the same order as the columns of the table.

Remark: INSERT INTO is used to insert rows in the table by: 1) specifying the specific column names where you want to enter values, if you do not want to add values to all the columns 2) if you are adding values to all columns then without specifying the column names and writing values in the same order as the columns.

VALUES keyword is used to specify the values that are to be added in a single specific cell.

Q3) 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).

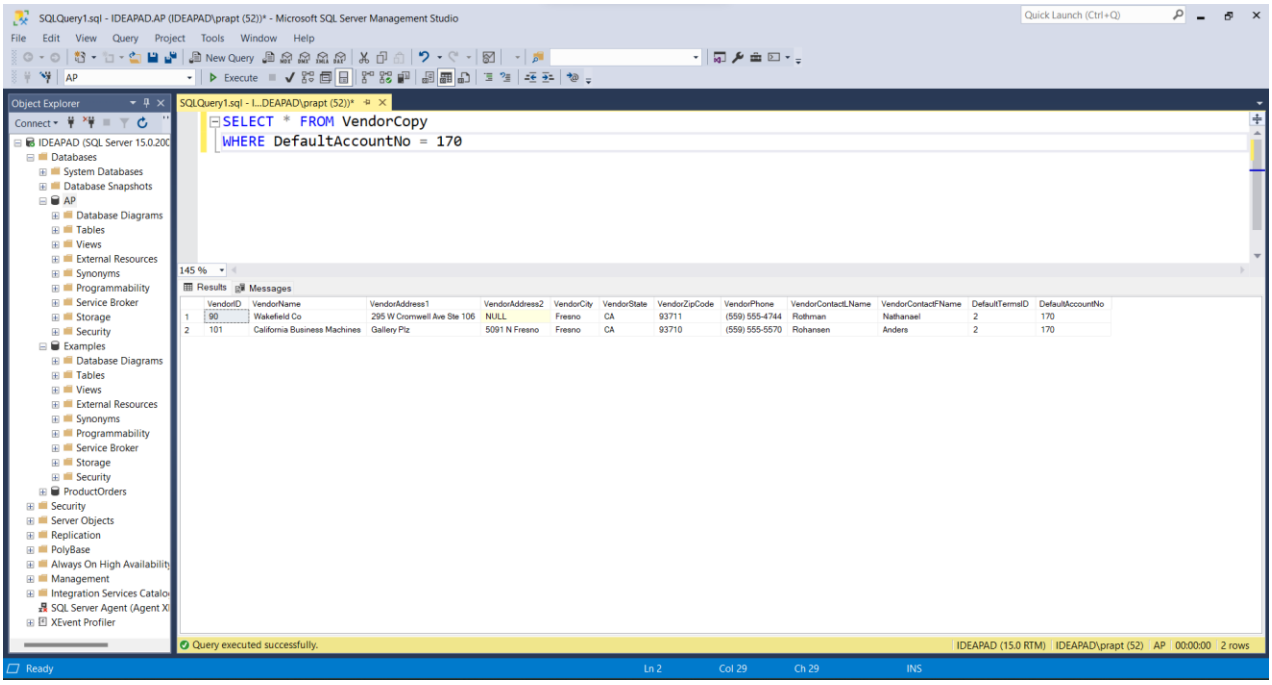
Ans: UPDATE VendorCopy SET DefaultAccountNo = 970

WHERE DefaultAccountNo = 170

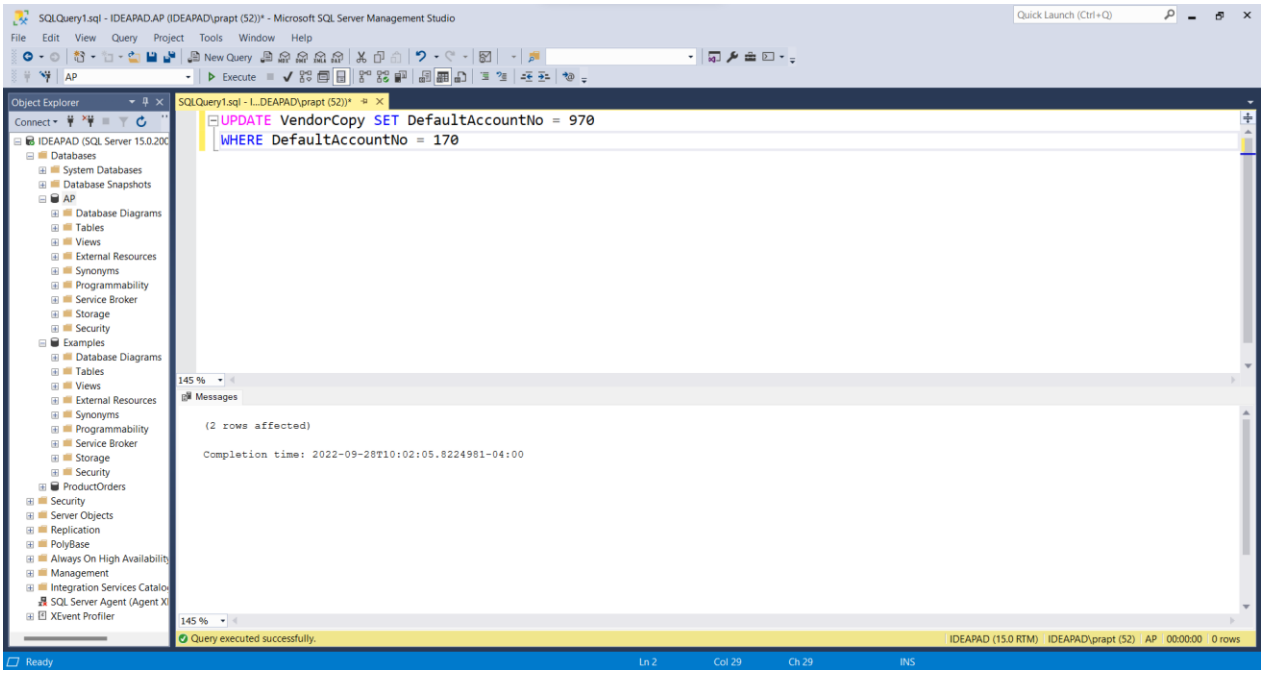
Before updating the records:

SELECT * FROM VendorCopy

WHERE DefaultAccountNo = 170



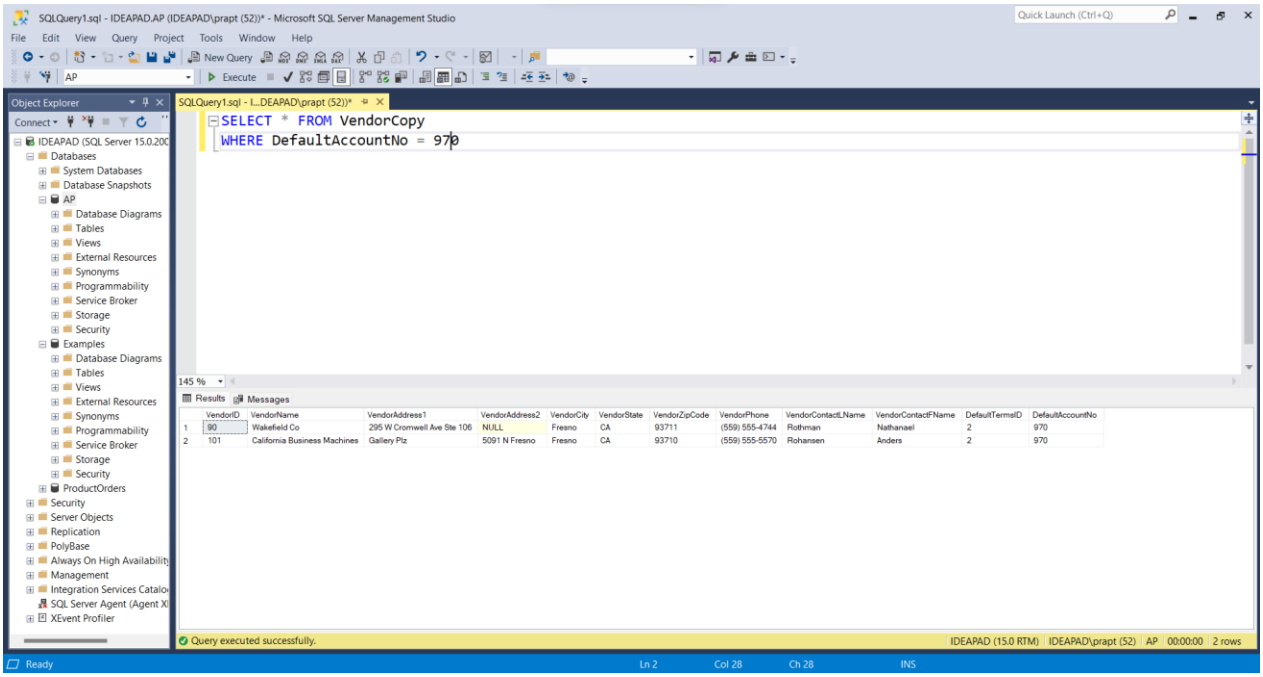
Updating the records:



After updating the records:

SELECT * FROM VendorCopy

WHERE DefaultAccountNo = 170



Comment: Here, UPDATE statement is used for updating/modification of the account number. Using UPDATE statement, the records with DefaultAccountNo 170 is changed to 970 using SET keyword. The condition is put using WHERE keyword.

Remark: To change/modify the existing data of the table, UPDATE statement is used.

SET is used to specify the value which is to be set instead of the previous value.

Q4) 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).

Ans: UPDATE InvoiceCopy SET TermsID = '3'

WHERE VendorID IN(SELECT VendorID

FROM VendorCopy

WHERE DefaultTermsID = '2')

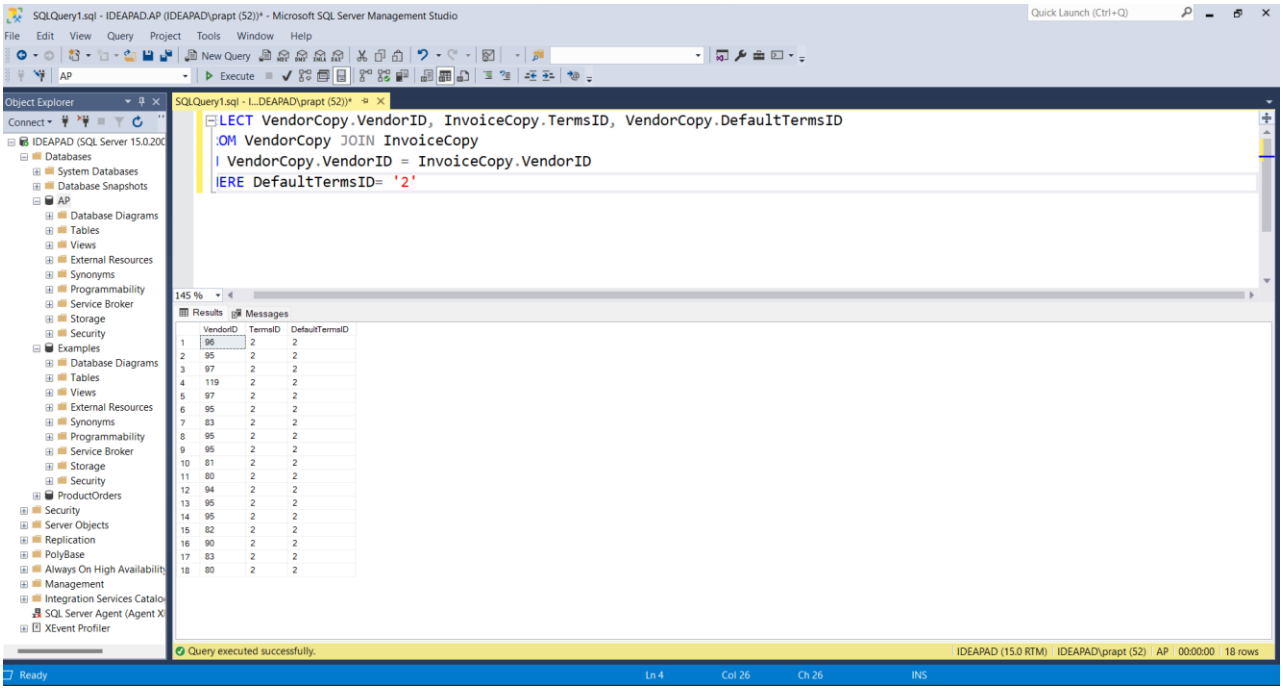
Before Updating the records:

SELECT VendorCopy.VendorID, InvoiceCopy.TermsID, VendorCopy.DefaultTermsID

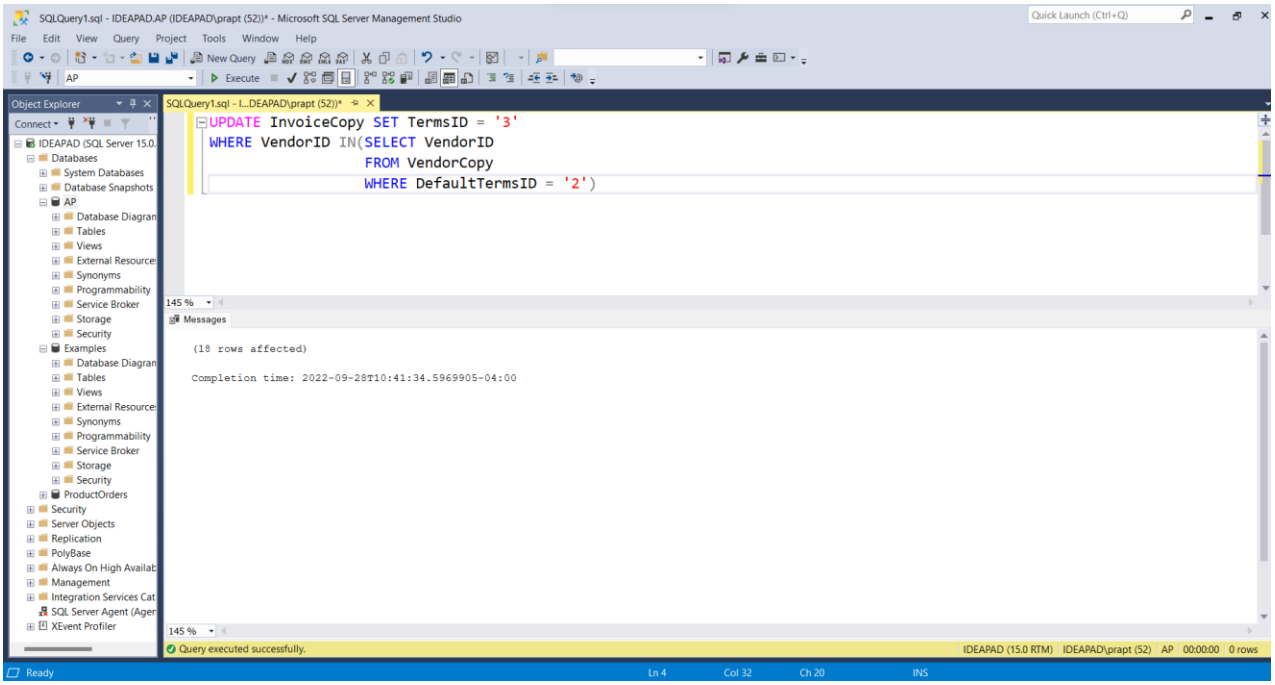
FROM VendorCopy JOIN InvoiceCopy

ON VendorCopy.VendorID = InvoiceCopy.VendorID

WHERE DefaultTermsID = '2'



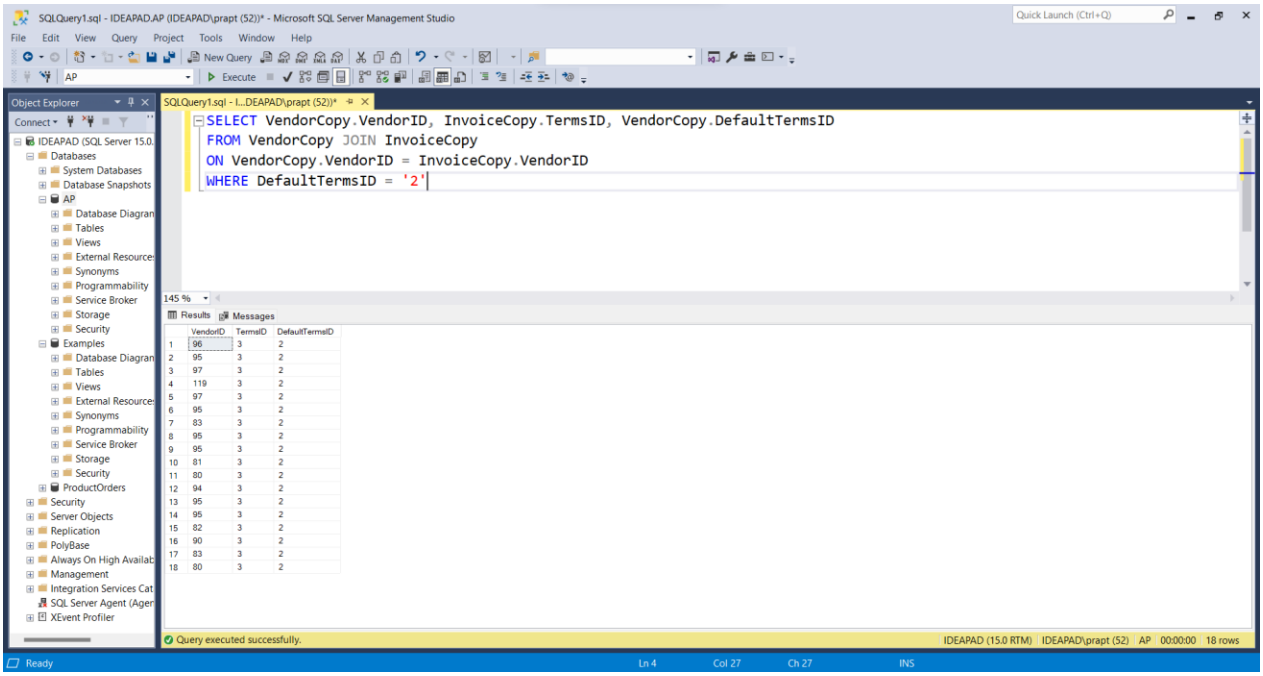
Updating the records:



After updating the records:

```
SELECT VendorCopy.VendorID, InvoiceCopy.TermsID, VendorCopy.DefaultTermsID
    FROM VendorCopy JOIN InvoiceCopy
    ON VendorCopy.VendorID = InvoiceCopy.VendorID
```

WHERE DefaultTermsID = '2'



Comment: Here, an UPDATE statement is used to update/modify the records of the InvoiceCopy table. The records in the VendorCopy table with DefaultTermsID as 2 are selected and the VendorID of the records of the InvoiceCopy table that are from the pool of the subquery's result set are to be changed. This is done using the IN keyword on the subquery. The TermsID of these records are then set to 3 using SET keyword.

Remark: To change/modify the existing data of the table, an UPDATE statement is used. Subqueries can also be used to select certain records that are to be changed.

Q5) 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).

Ans: DELETE FROM VendorCopy

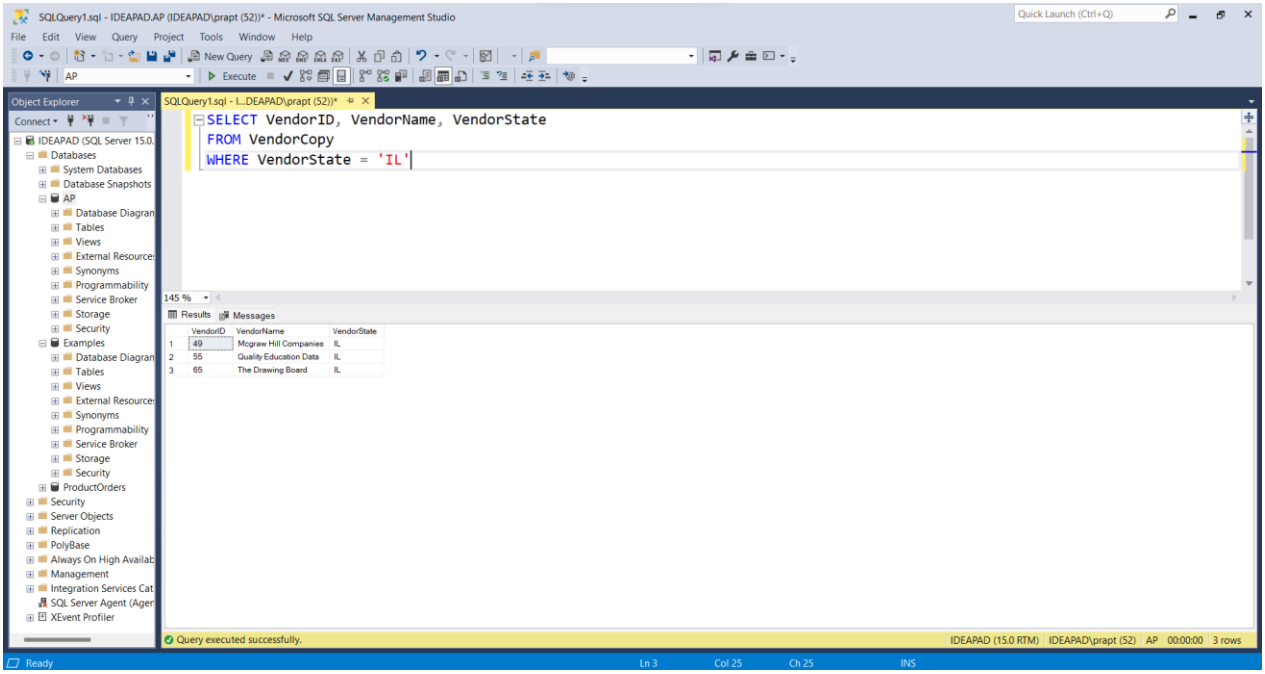
WHERE VendorState = 'IL'

Before deleting records:

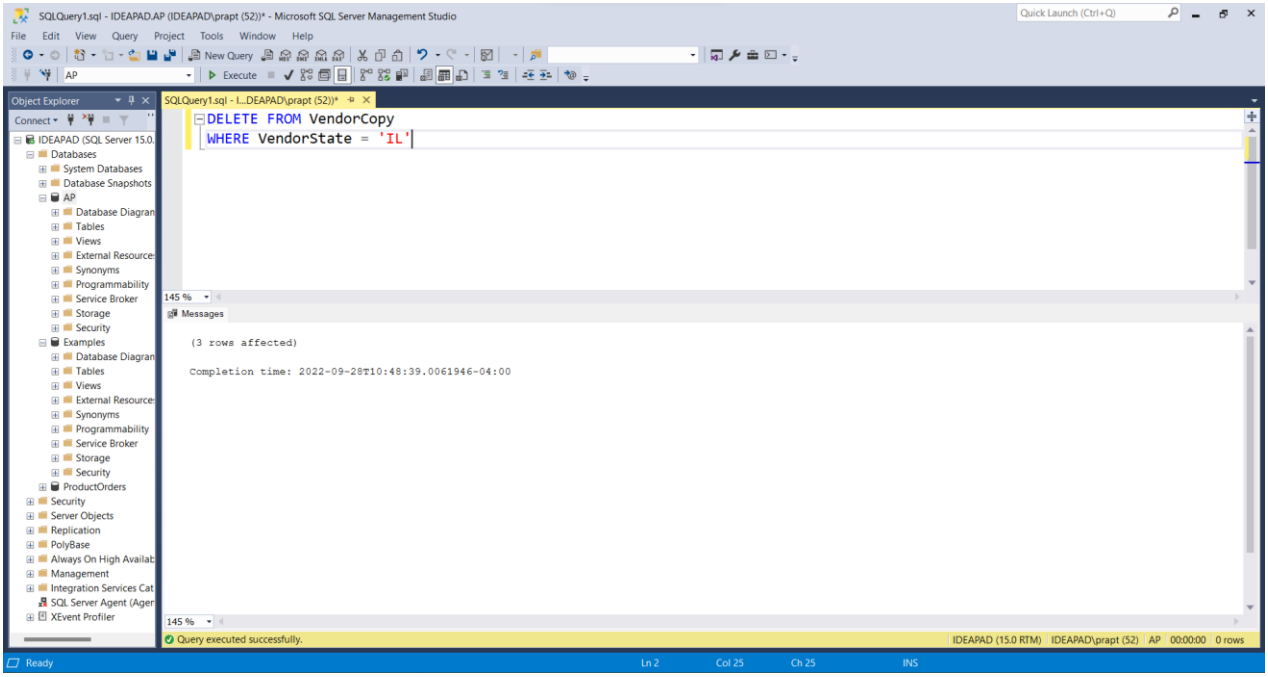
SELECT VendorID, VendorName, VendorState

FROM VendorCopy

WHERE VendorState = 'IL'



Deleting the records:

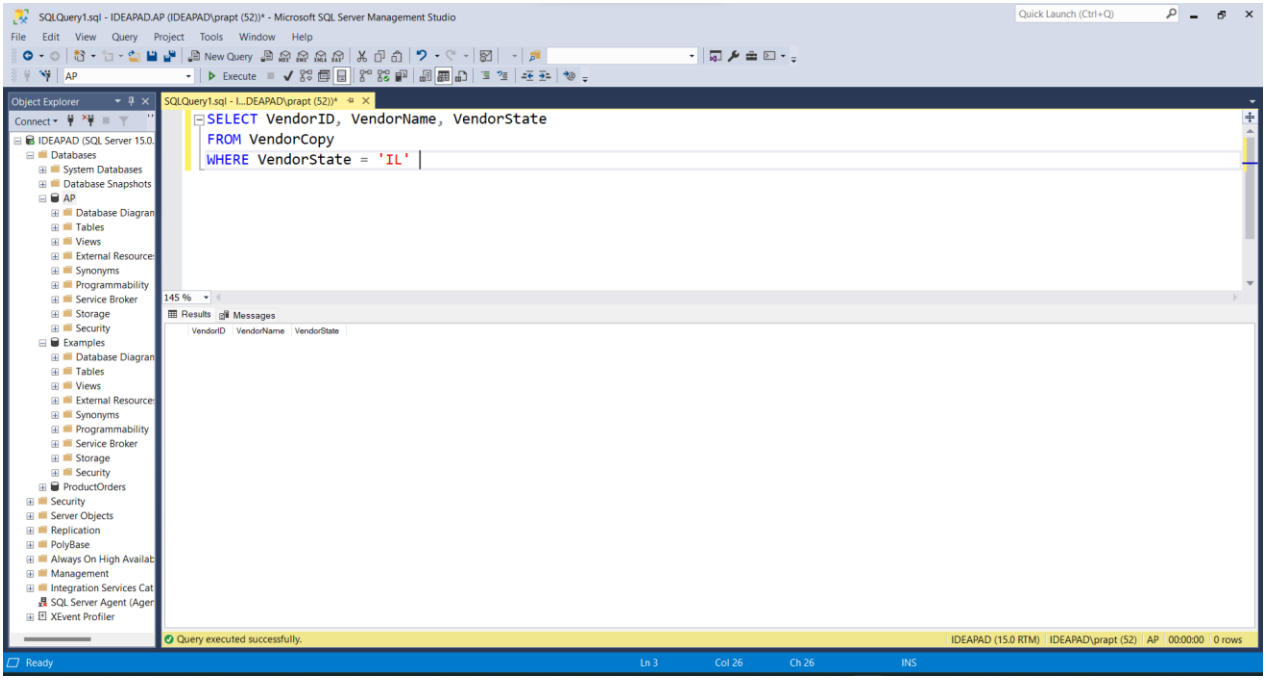


After deleting the records:

SELECT VendorID, VendorName, VendorState

FROM VendorCopy

WHERE VendorState = 'IL'



Comment: Here, DELETE statement is used to delete the records from the VendorCopy table. WHERE clause is used to specify the condition of the records that are to be deleted. So, the records with VendorState as 'IL' (Illinois) are deleted.

Remark: DELETE statement is used to delete all or certain records from a table.

Q6) 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)

Ans: DELETE FROM VendorCopy

WHERE VendorState NOT IN (SELECT VendorState

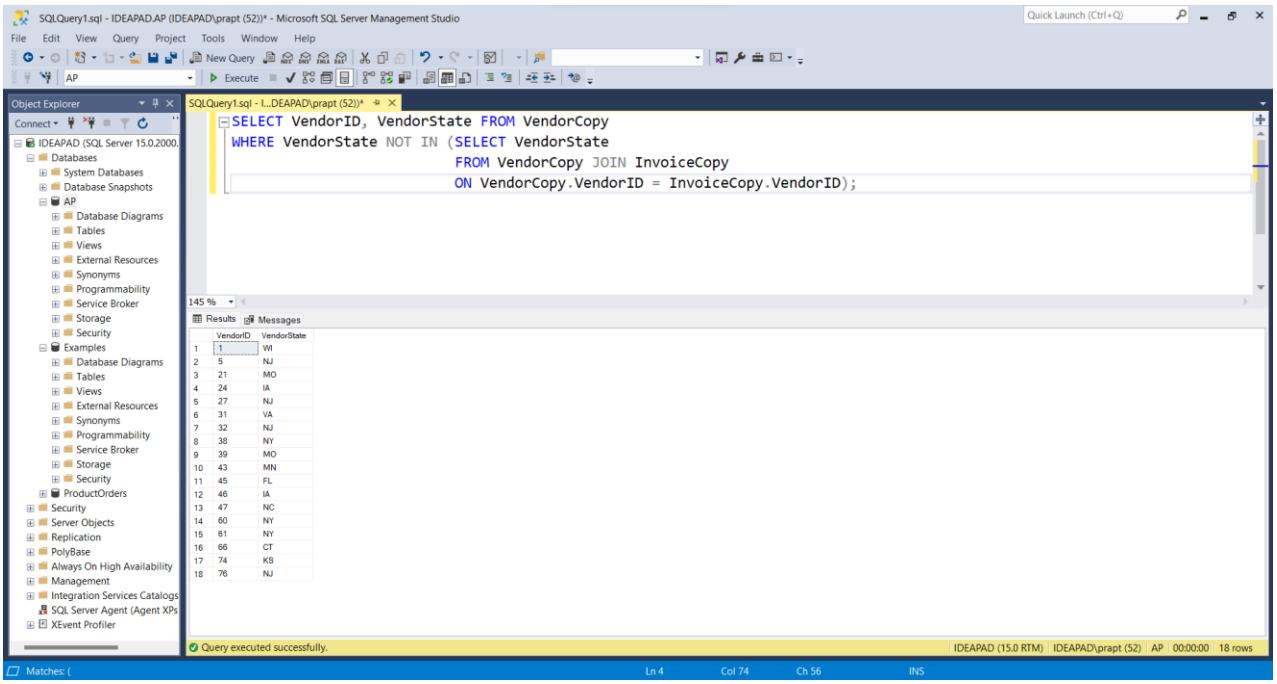
FROM VendorCopy JOIN InvoiceCopy

ON VendorCopy.VendorID = InvoiceCopy.VendorID);

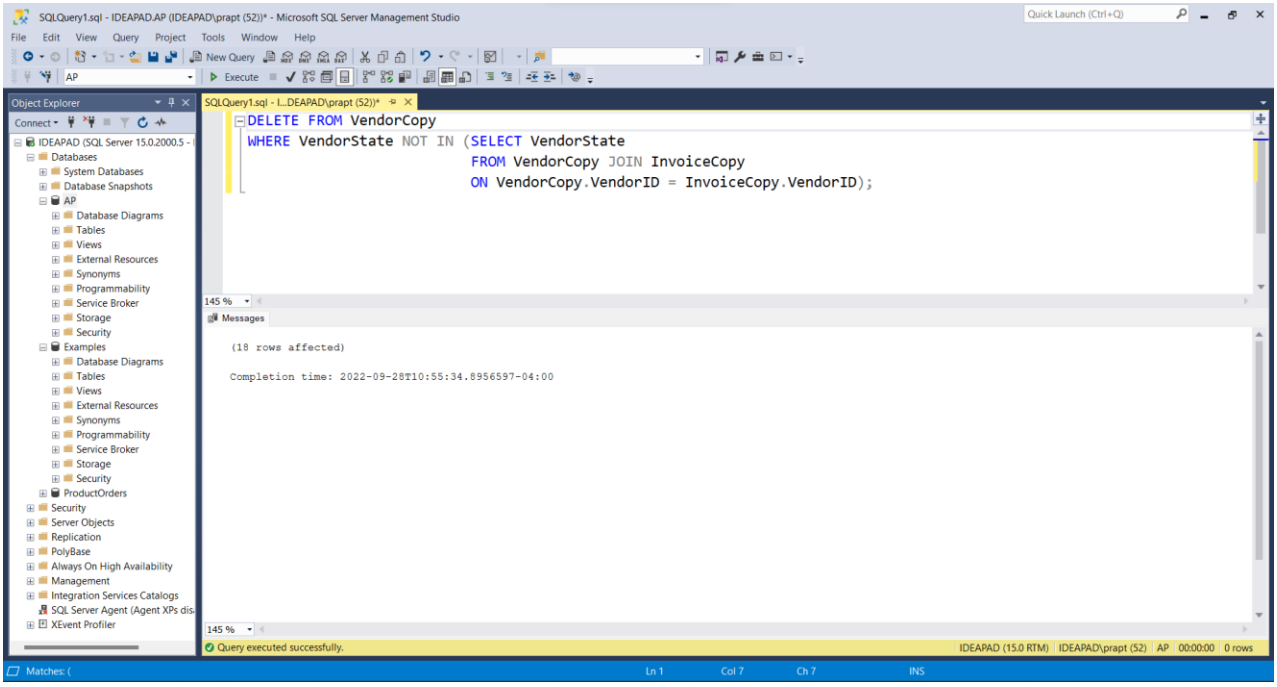
Before deleting records:

SELECT VendorID, VendorState FROM VendorCopy

```
WHERE VendorState NOT IN (SELECT VendorState
                          FROM VendorCopy JOIN InvoiceCopy
                          ON VendorCopy.VendorID = InvoiceCopy.VendorID);
```

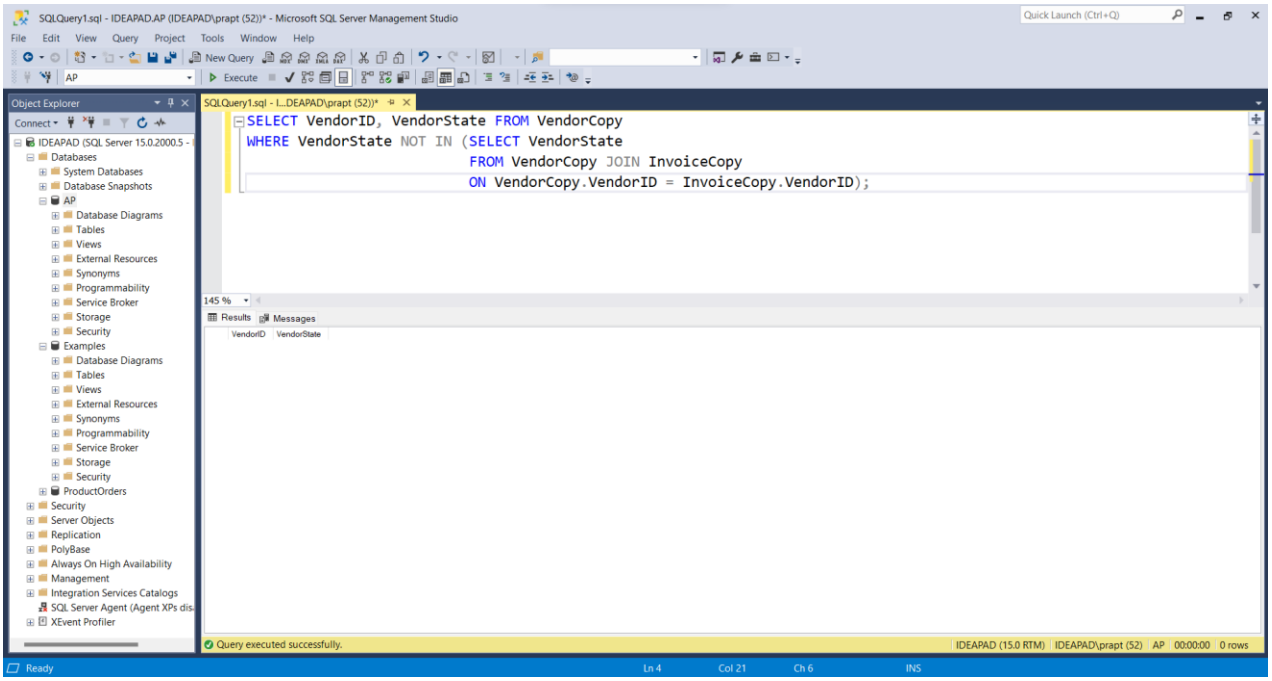


Deleting the records:



After deleting the records:

```
SELECT VendorID, VendorState FROM VendorCopy
WHERE VendorState NOT IN (SELECT VendorState
                          FROM VendorCopy JOIN InvoiceCopy
                          ON VendorCopy.VendorID = InvoiceCopy.VendorID);
```



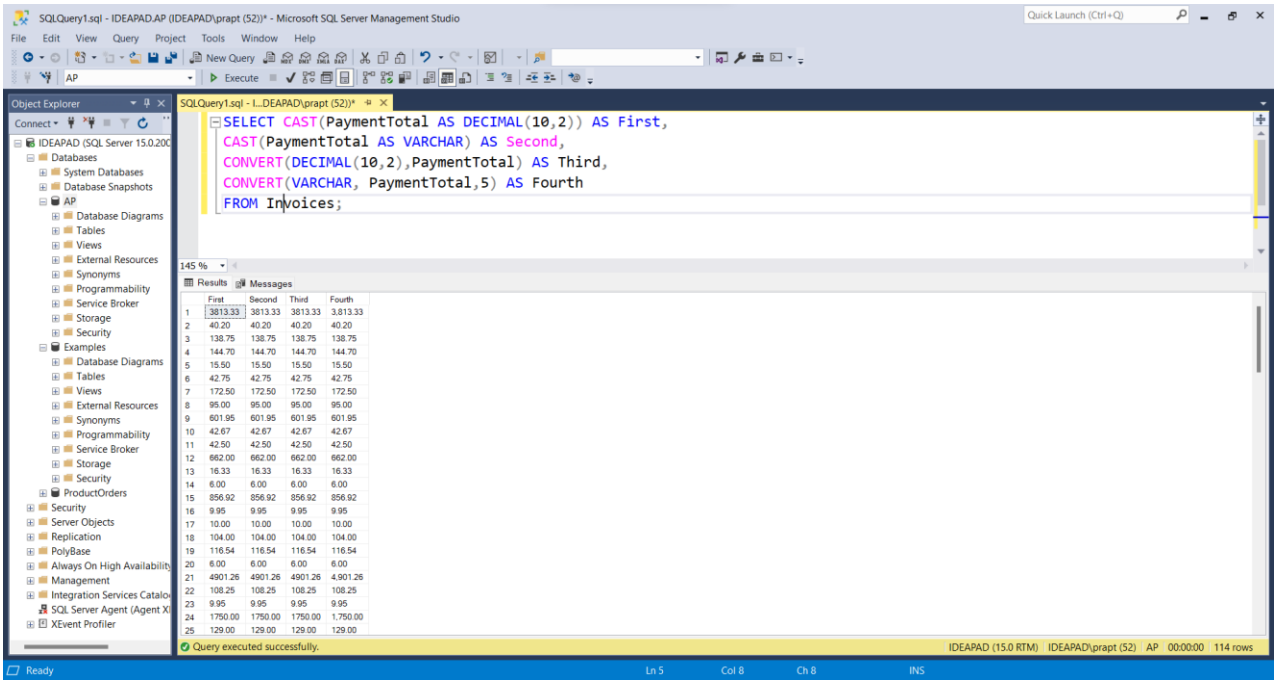
Comment: Here, DELETE statement is used to delete the records from the VendorCopy whose vendors are in such a list of states of vendors who have never sent an invoice. So, a subquery is written which selects the records of such vendors. SELECT statement is used in the subquery to select the VendorState from the JOIN of the VendorCopy and InvoiceCopy table on VendorID I.e., those records that have sent an invoice. NOT IN is used to delete those records that are not in the result set of the subquery.

Remark: NOT IN operator is used to select the values that are not in the specified result set.

Q7) Write a SELECT statement that returns four columns based on the PaymentTotal column of the Invoices table:

1. Use CAST function to return the first column as data type decimal with 2 digits to the right of the decimal point.
2. Use CAST to return the second column as a VARCHAR.
3. Use CONVERT function to return third column as the same type as the first column.
4. Use CONVERT to return the fourth column as a VARCHAR, using style 5.

Ans:
```
SELECT CAST(PaymentTotal AS DECIMAL(10,2)) AS First,
       CAST(PaymentTotal AS VARCHAR) AS Second,
       CONVERT(DECIMAL(10,2),PaymentTotal) AS Third,
       CONVERT(VARCHAR, PaymentTotal,5) AS Fourth
       FROM Invoices;
```



Comment: Here, a SELECT statement is used to display different types of records by applying different functions to the PaymentTotal column of the Invoices table. For the First column, CAST function is used on the PaymentTotal column to return the data of the length of the data type DECIMAL with 2 digits to the right of the decimal point. For the Second column, CAST function is used on the PaymentTotal column to return the data in VARCHAR datatype. For the Third column, CONVERT function is used on the PaymentTotal column to convert the datatype to DECIMAL with 2 digits to the right of

the decimal point. For the Fourth column, CONVERT function is used on the PaymentTotal column to convert the datatype to VARCHAR and format using style 5.

Remark: CAST function is used to convert data. CONVERT function is used to convert and format the data.

Q8) Write a SELECT statement that returns four columns based on the PaymentDate column of the Invoices table:

1. Use the CAST function to return the first column as data type VARCHAR.
2. Use the CONVERT function to return the second and third columns as a VARCHAR, using style 1 and style 9, respectively.
3. Use the CAST function to return the fourth column as a data type real.

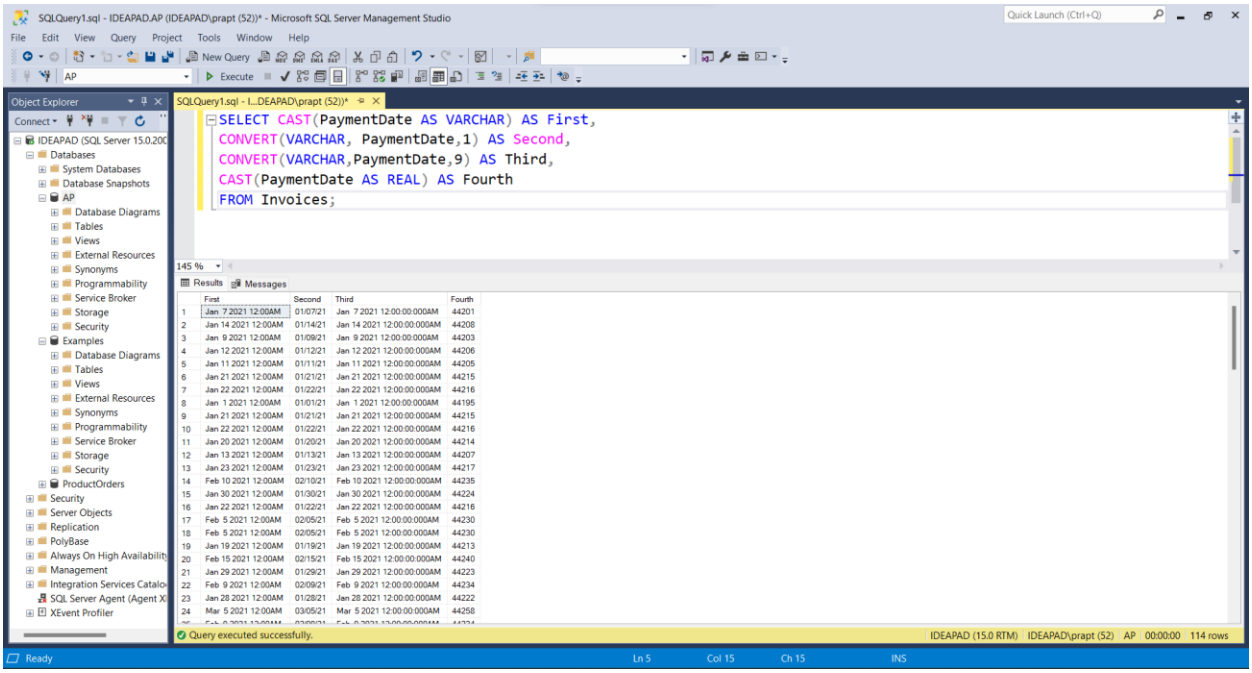
Ans: SELECT CAST(PaymentDate AS VARCHAR) AS First,

CONVERT(VARCHAR, PaymentDate,1) AS Second,

CONVERT(VARCHAR,PaymentDate,9) AS Third,

CAST(PaymentDate AS REAL) AS Fourth

FROM Invoices;



Comment: Here, a SELECT statement is used to display different types of records by applying different functions to the PaymentDate column of the Invoices table. For the First column, CAST function is used on the PaymentDate column to return the data of the data type VARCHAR. For the Second column, CONVERT function is used on the PaymentDate column to return the data in VARCHAR datatype and format using style 1. For the Third column, CONVERT function is used on the PaymentDate column to convert the datatype to VARCHAR and format with style 9. For the Fourth column, CONVERT function is used on the PaymentDate column to convert the datatype to REAL.

Remark for the lab: Concepts related to data manipulation and data types are used including the basic keywords, clauses and statements in this lab.