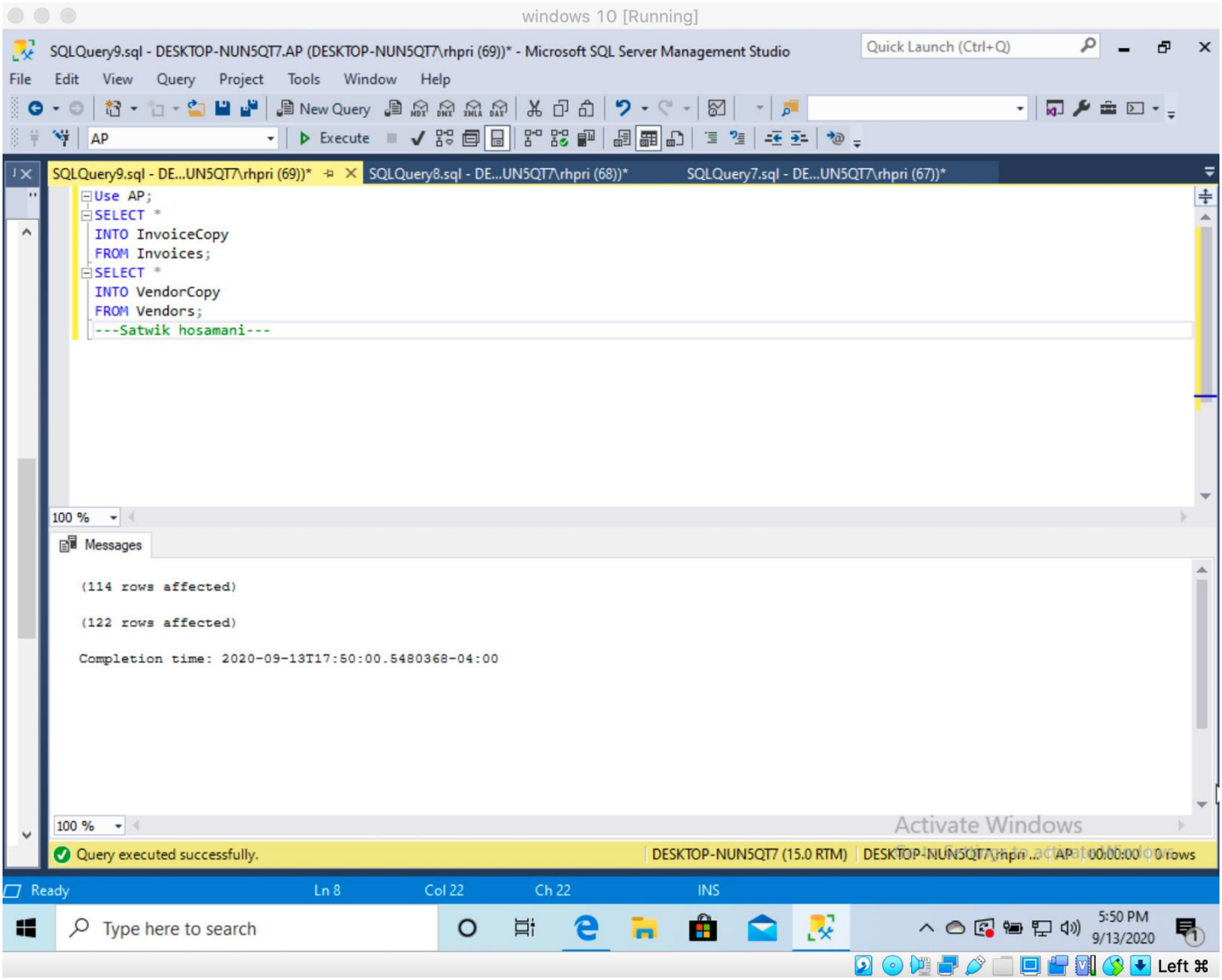
Q1. Create VendorCopy table and InvoiceCopy table

Comments: In this query we had to copy the table rows to another table so I have used the "select into " statement which copies all the rows from the parent table to the new table .The new table is created same as the parent one with all the column names same .We don't have to explicitly create a new table.

Sol:

```
Use AP;
SELECT
INTO InvoiceCopy
From Invoices;
SELECT
INTO VendorCopy
FROM Vendors;
----Satwik Hosamani---
```



Rows in InvoiceCopy :114
Rows in VendorCopy:122

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):

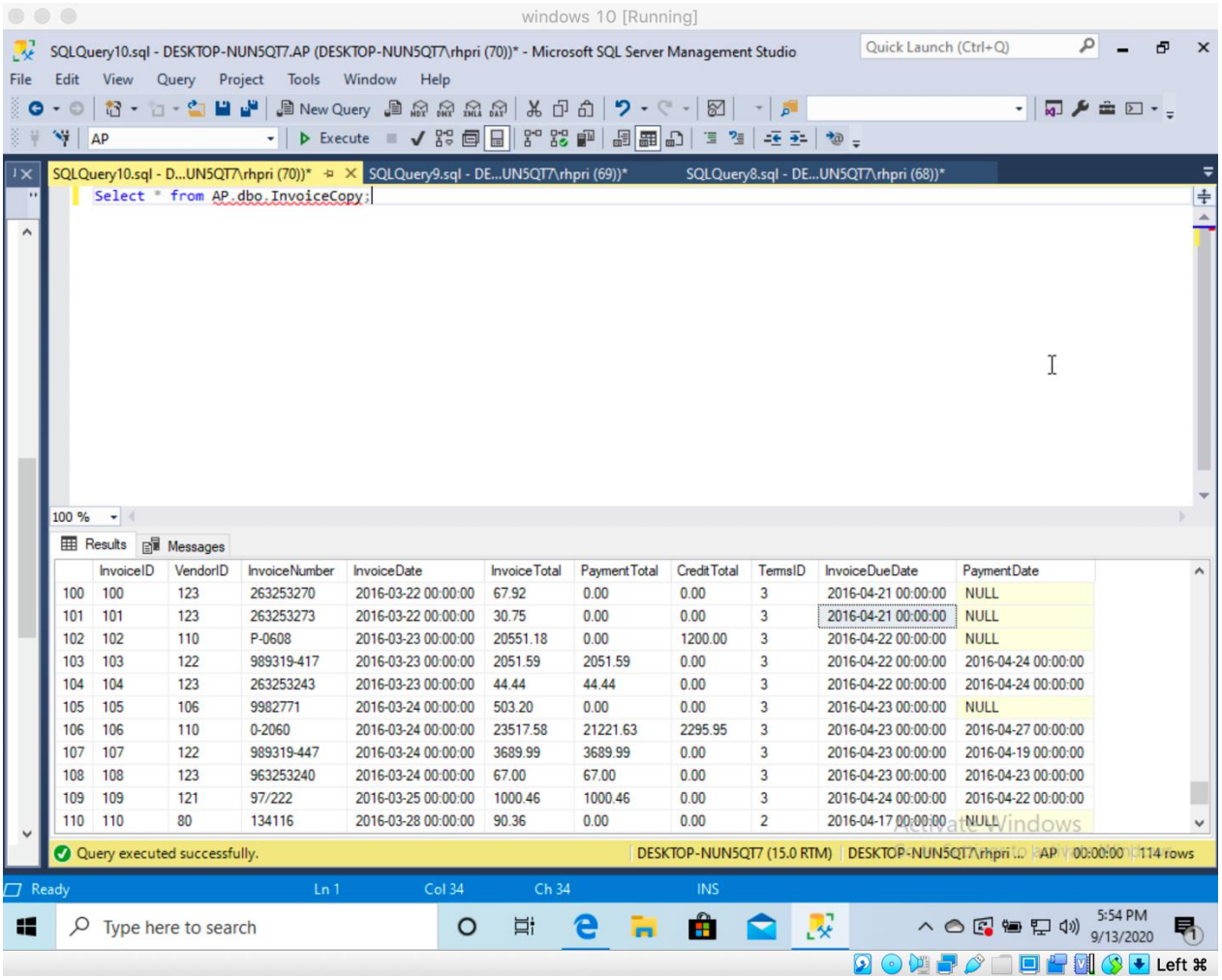
VendorID: 2 InvoiceTotal: $401.40
TermsID: 3 InvoiceNumber: CM-007-700
PaymentTotal: $2.99 InvoiceDueDate: 09/01/15
InvoiceDate: 08/24/15 CreditTotal: $5.99
Do we explicitly need to insert InvoiceID?

Comments:We don't need to explicitly insert into InvoiceID because select into statement copies the properties of the column too not just the name,InvoiceID has identity(1,1) meaning the first 1 means the starting value of ID and the second 1 means the increment value of ID

Before Insertion:

Select * from AP.dbo.InvoiceCopy;

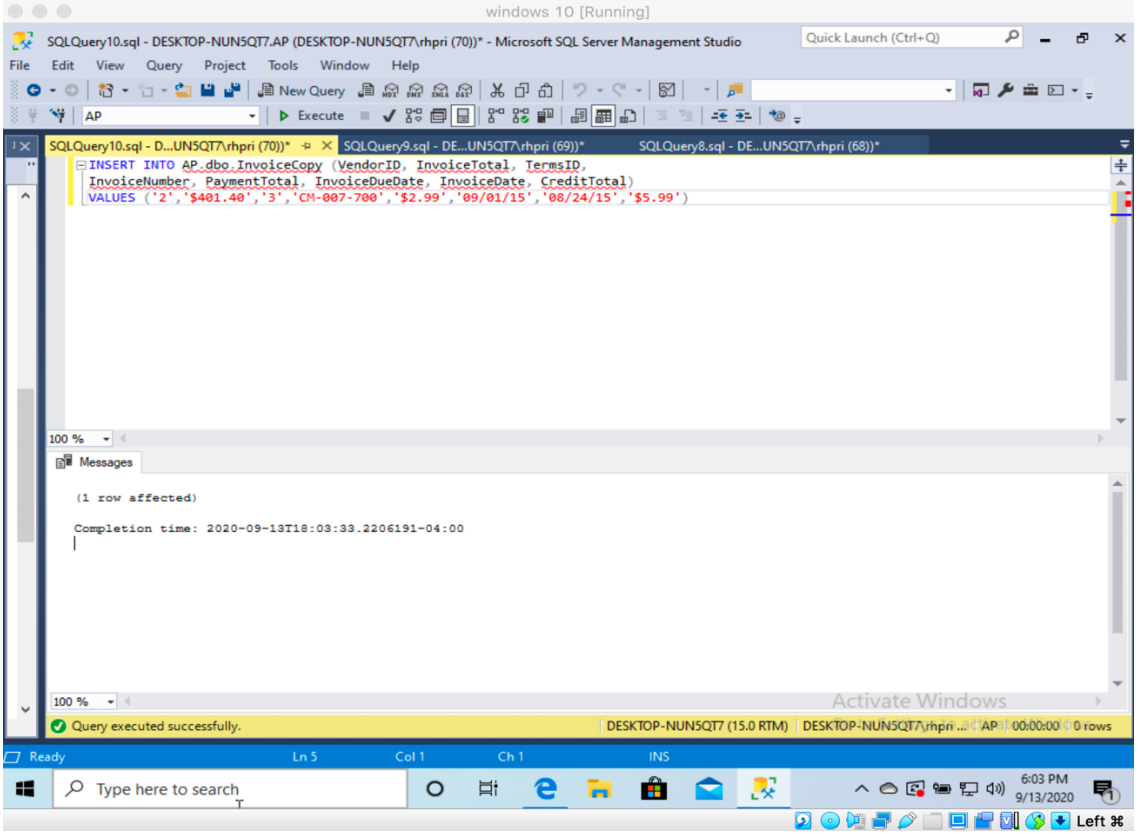
Rows: 114



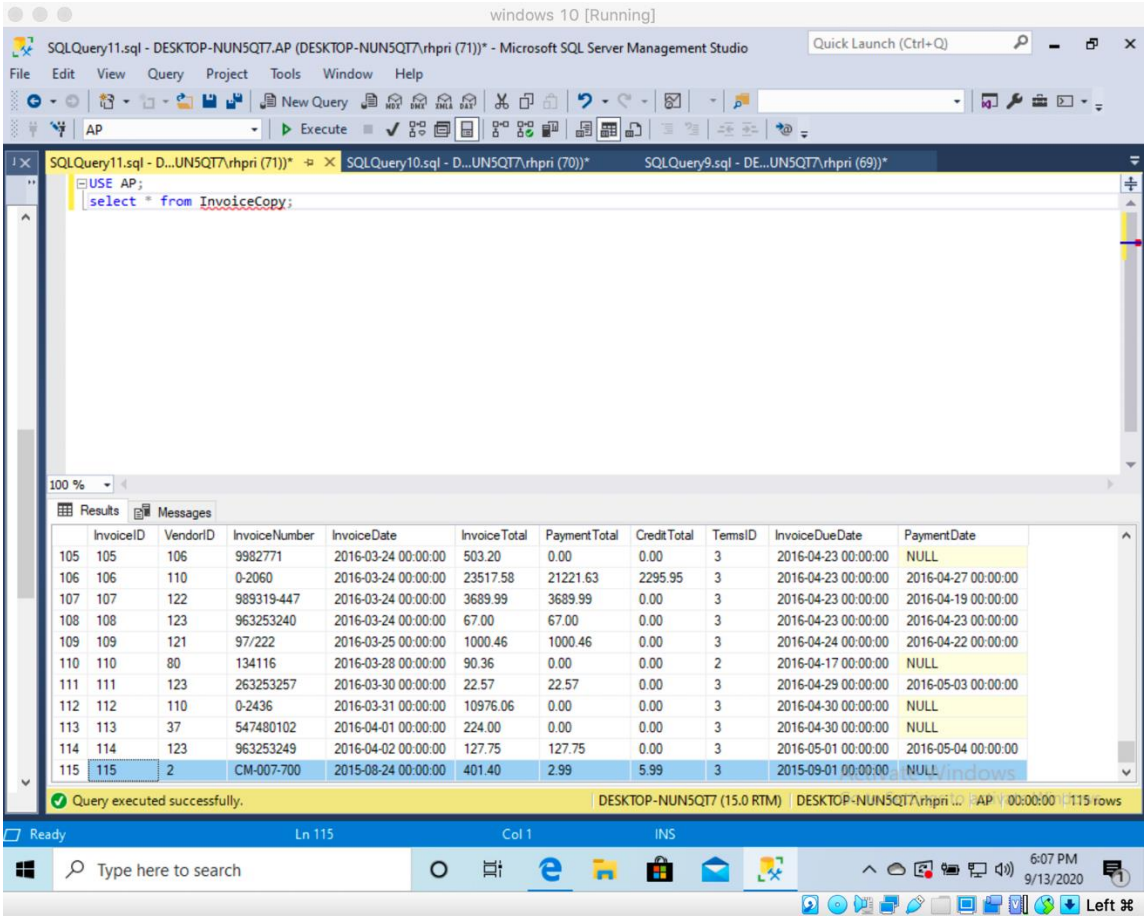
	InvoiceID	VendorID	InvoiceNumber	InvoiceDate	InvoiceTotal	PaymentTotal	CreditTotal	TermsID	InvoiceDueDate	PaymentDate
100	100	123	263253270	2016-03-22 00:00:00	67.92	0.00	0.00	3	2016-04-21 00:00:00	NULL
101	101	123	263253273	2016-03-22 00:00:00	30.75	0.00	0.00	3	2016-04-21 00:00:00	NULL
102	102	110	P-0608	2016-03-23 00:00:00	20551.18	0.00	1200.00	3	2016-04-22 00:00:00	NULL
103	103	122	989319-417	2016-03-23 00:00:00	2051.59	2051.59	0.00	3	2016-04-22 00:00:00	2016-04-24 00:00:00
104	104	123	263253243	2016-03-23 00:00:00	44.44	44.44	0.00	3	2016-04-22 00:00:00	2016-04-24 00:00:00
105	105	106	9982771	2016-03-24 00:00:00	503.20	0.00	0.00	3	2016-04-23 00:00:00	NULL
106	106	110	0-2060	2016-03-24 00:00:00	23517.58	21221.63	2295.95	3	2016-04-23 00:00:00	2016-04-27 00:00:00
107	107	122	989319-447	2016-03-24 00:00:00	3689.99	3689.99	0.00	3	2016-04-23 00:00:00	2016-04-19 00:00:00
108	108	123	963253240	2016-03-24 00:00:00	67.00	67.00	0.00	3	2016-04-23 00:00:00	2016-04-23 00:00:00
109	109	121	97/222	2016-03-25 00:00:00	1000.46	1000.46	0.00	3	2016-04-24 00:00:00	2016-04-22 00:00:00
110	110	80	134116	2016-03-28 00:00:00	90.36	0.00	0.00	2	2016-04-17 00:00:00	NULL

SOL:

```
INSERT INTO AP.dbo.InvoiceCopy (VendorID, InvoiceTotal, TermsID,
InvoiceNumber, PaymentTotal, InvoiceDueDate, InvoiceDate, CreditTotal)
VALUES ('2','$401.40','3','CM-007-700','$2.99','09/01/15','08/24/15','$5.99')
```



After Insertion: Rows-115



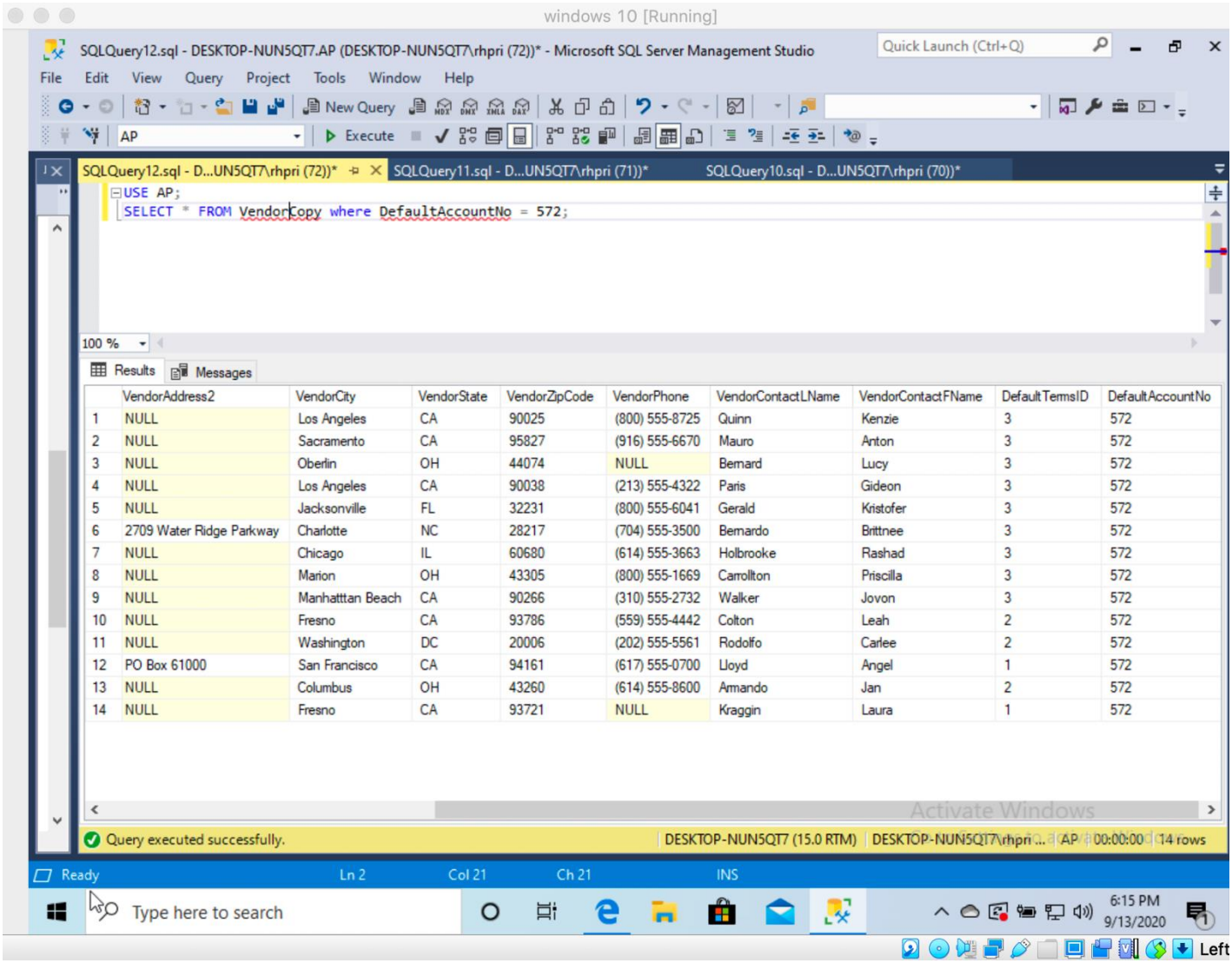
	InvoiceID	VendorID	InvoiceNumber	InvoiceDate	InvoiceTotal	PaymentTotal	CreditTotal	TermsID	InvoiceDueDate	PaymentDate
105	105	106	9982771	2016-03-24 00:00:00	503.20	0.00	0.00	3	2016-04-23 00:00:00	NULL
106	106	110	0-2060	2016-03-24 00:00:00	23517.58	21221.63	2295.95	3	2016-04-23 00:00:00	2016-04-27 00:00:00
107	107	122	989319-447	2016-03-24 00:00:00	3689.99	3689.99	0.00	3	2016-04-23 00:00:00	2016-04-19 00:00:00
108	108	123	963253240	2016-03-24 00:00:00	67.00	67.00	0.00	3	2016-04-23 00:00:00	2016-04-23 00:00:00
109	109	121	97/222	2016-03-25 00:00:00	1000.46	1000.46	0.00	3	2016-04-24 00:00:00	2016-04-22 00:00:00
110	110	80	134116	2016-03-28 00:00:00	90.36	0.00	0.00	2	2016-04-17 00:00:00	NULL
111	111	123	263253257	2016-03-30 00:00:00	22.57	22.57	0.00	3	2016-04-29 00:00:00	2016-05-03 00:00:00
112	112	110	0-2436	2016-03-31 00:00:00	10976.06	0.00	0.00	3	2016-04-30 00:00:00	NULL
113	113	37	547480102	2016-04-01 00:00:00	224.00	0.00	0.00	3	2016-04-30 00:00:00	NULL
114	114	123	963253249	2016-04-02 00:00:00	127.75	127.75	0.00	3	2016-05-01 00:00:00	2016-05-04 00:00:00
115	115	2	CM-007-700	2015-08-24 00:00:00	401.40	2.99	5.99	3	2015-09-01 00:00:00	NULL

3. Write an UPDATE statement that modifies the VendorCopy table. Change the default account number to 542 for each vendor that has a default account number of 572. (USE SELECT statement to verify data changes in the table before and after the modification).

Comments: Update statement is used to modify the values of the rows which satisfy the condition.here the condition is that the account number of vendor should be 572 changed to 542 .The condition is mentioned in the where clause
ROWS:14

Before:

```
USE AP;
SELECT * FROM VendorCopy where DefaultAccountNo = 572;
```

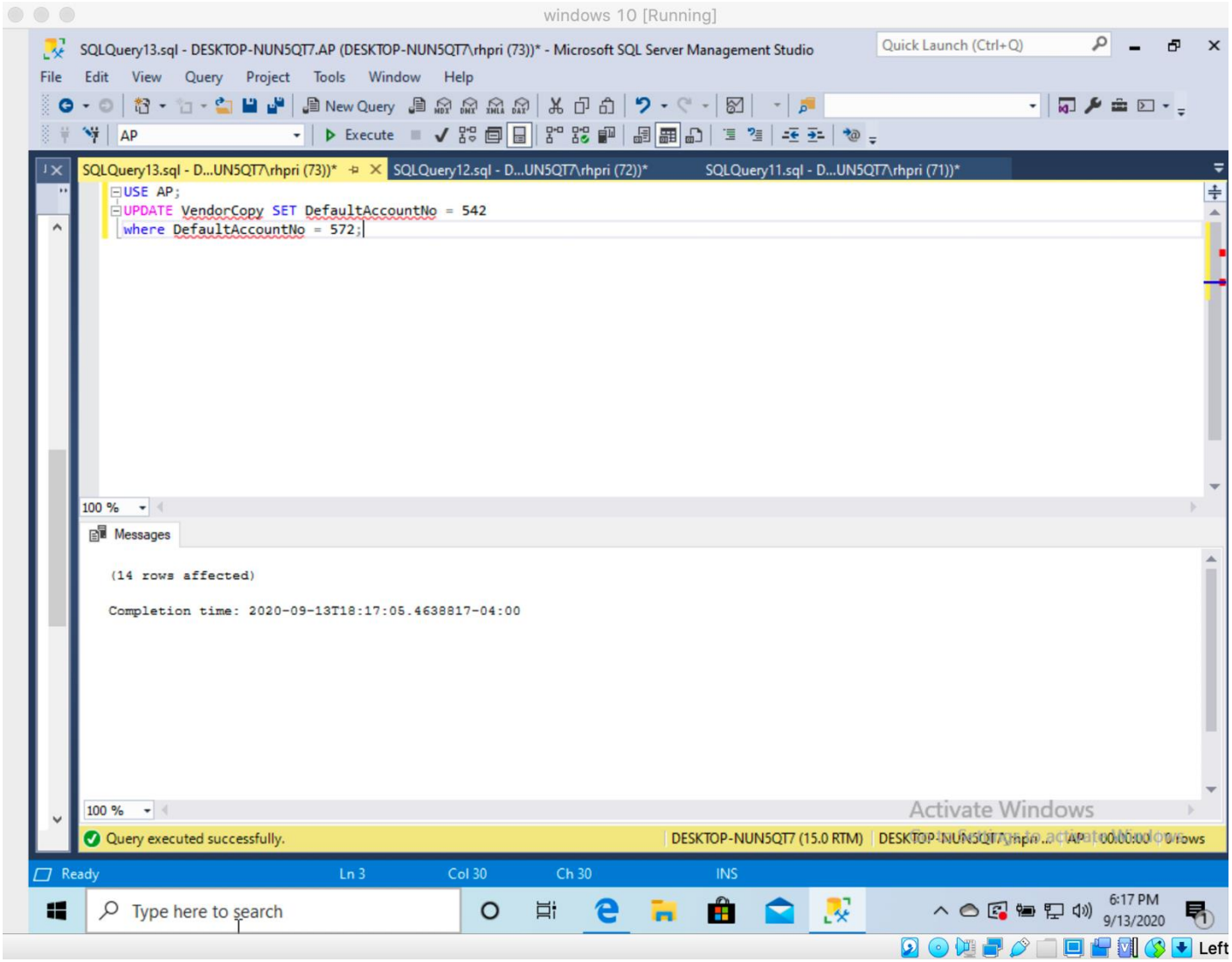


	VendorAddress2	VendorCity	VendorState	VendorZipCode	VendorPhone	VendorContactLName	VendorContactFName	DefaultTermsID	DefaultAccountNo
1	NULL	Los Angeles	CA	90025	(800) 555-8725	Quinn	Kenzie	3	572
2	NULL	Sacramento	CA	95827	(916) 555-6670	Mauro	Anton	3	572
3	NULL	Oberlin	OH	44074	NULL	Bernard	Lucy	3	572
4	NULL	Los Angeles	CA	90038	(213) 555-4322	Paris	Gideon	3	572
5	NULL	Jacksonville	FL	32231	(800) 555-6041	Gerald	Kristofer	3	572
6	2709 Water Ridge Parkway	Charlotte	NC	28217	(704) 555-3500	Bernardo	Brittnee	3	572
7	NULL	Chicago	IL	60680	(614) 555-3663	Holbrooke	Rashad	3	572
8	NULL	Marion	OH	43305	(800) 555-1669	Carrollton	Priscilla	3	572
9	NULL	Manhatttan Beach	CA	90266	(310) 555-2732	Walker	Jovon	3	572
10	NULL	Fresno	CA	93786	(559) 555-4442	Colton	Leah	2	572
11	NULL	Washington	DC	20006	(202) 555-5561	Rodolfo	Carlee	2	572
12	PO Box 61000	San Francisco	CA	94161	(617) 555-0700	Lloyd	Angel	1	572
13	NULL	Columbus	OH	43260	(614) 555-8600	Armando	Jan	2	572
14	NULL	Fresno	CA	93721	NULL	Kraggin	Laura	1	572

TO UPDATE:

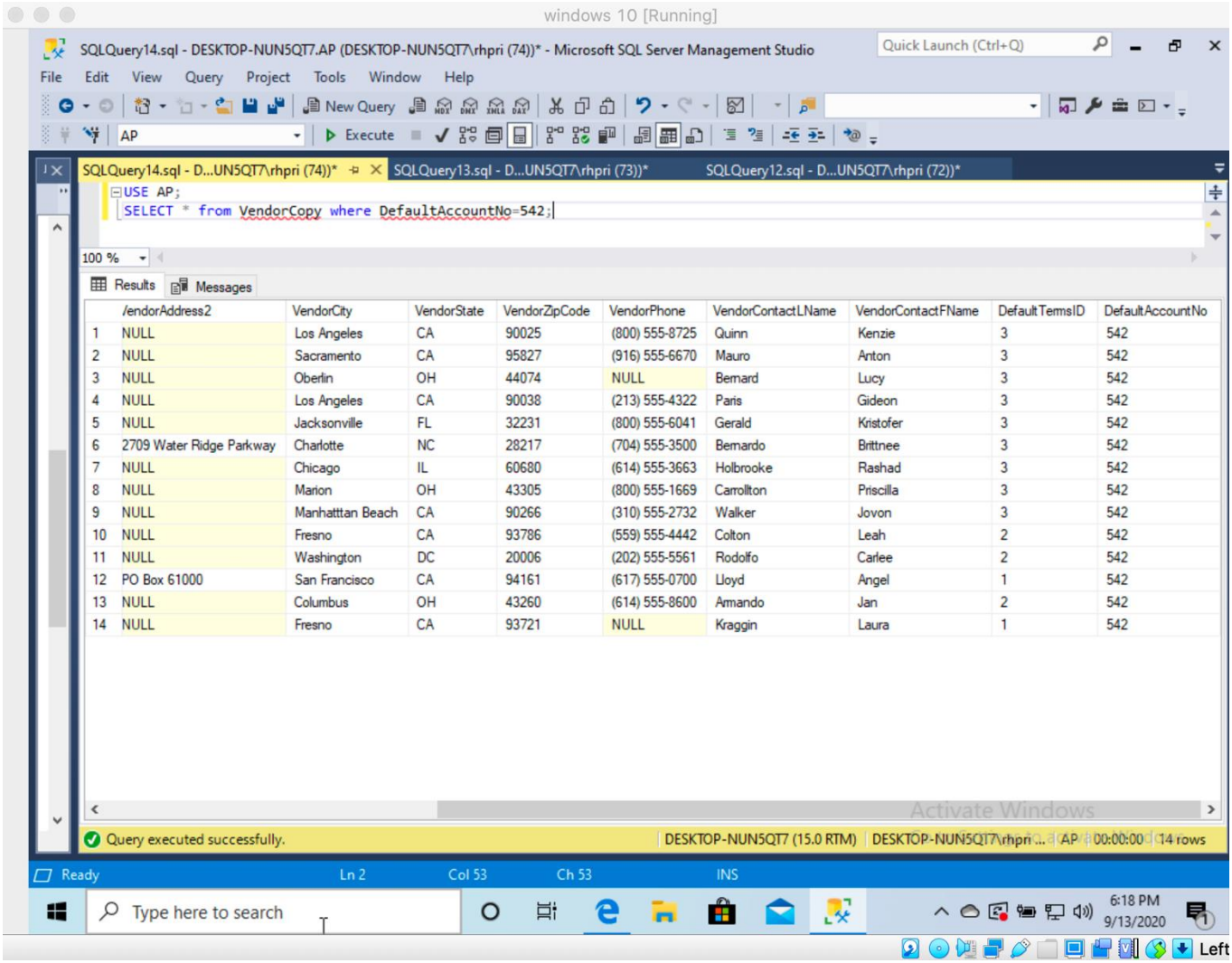
```
USE AP;
UPDATE VendorCopy SET DefaultAccountNo= 542
WHERE DefaultAccountNo = 572;
```

ROWS Affected: 14



CHECKING THE UPDATE:

USE AP;
Select *VendorCopy wherer DefaultAccountNo = 542;



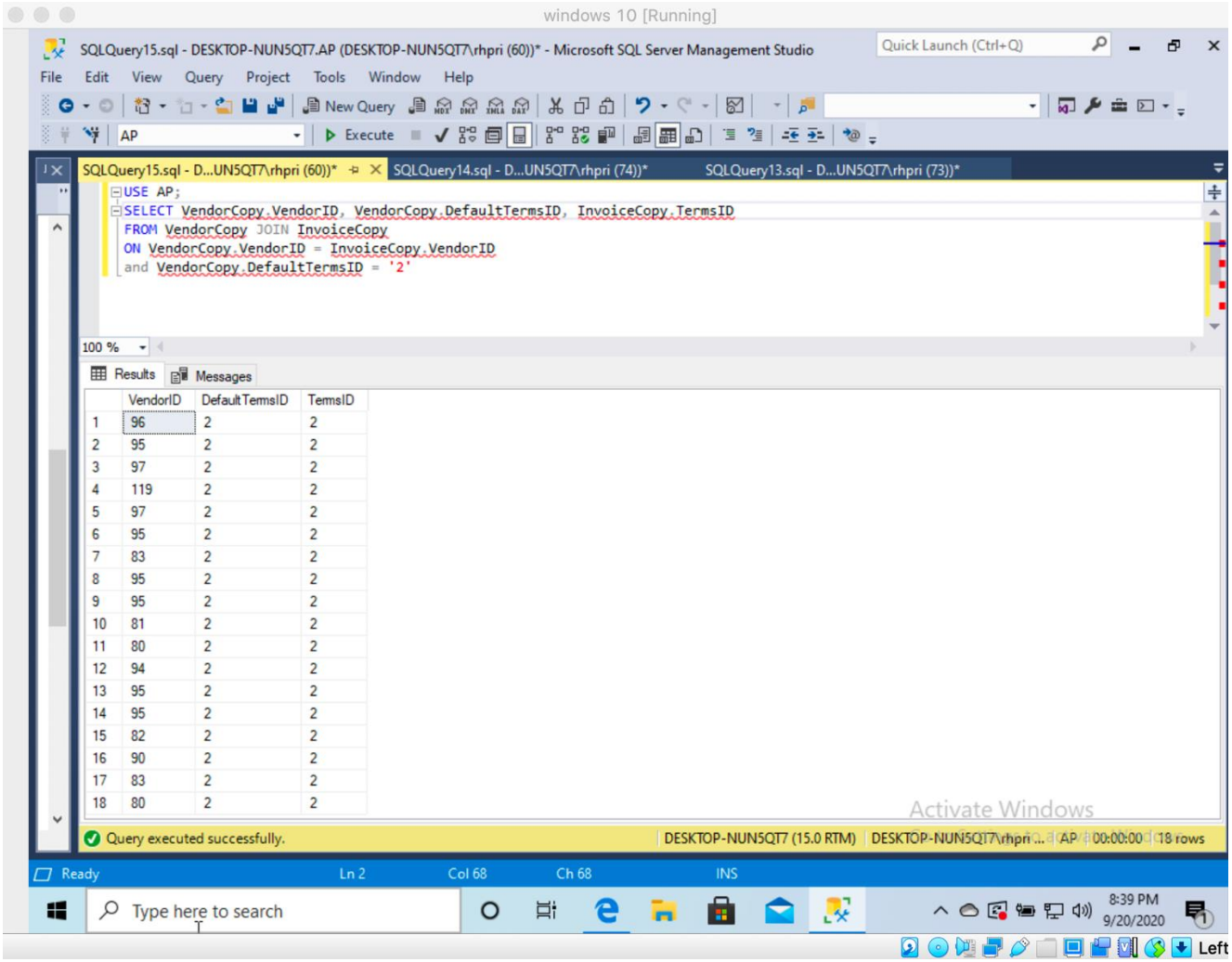
	/endorAddress2	VendorCity	VendorState	VendorZipCode	VendorPhone	VendorContactLName	VendorContactFName	DefaultTermsID	DefaultAccountNo
1	NULL	Los Angeles	CA	90025	(800) 555-8725	Quinn	Kenzie	3	542
2	NULL	Sacramento	CA	95827	(916) 555-6670	Mauro	Anton	3	542
3	NULL	Oberlin	OH	44074	NULL	Bernard	Lucy	3	542
4	NULL	Los Angeles	CA	90038	(213) 555-4322	Paris	Gideon	3	542
5	NULL	Jacksonville	FL	32231	(800) 555-6041	Gerald	Kristofer	3	542
6	2709 Water Ridge Parkway	Charlotte	NC	28217	(704) 555-3500	Bernardo	Brittnee	3	542
7	NULL	Chicago	IL	60680	(614) 555-3663	Holbrooke	Rashad	3	542
8	NULL	Marion	OH	43305	(800) 555-1669	Carrollton	Priscilla	3	542
9	NULL	Manhatttan Beach	CA	90266	(310) 555-2732	Walker	Jovon	3	542
10	NULL	Fresno	CA	93786	(559) 555-4442	Colton	Leah	2	542
11	NULL	Washington	DC	20006	(202) 555-5561	Rodolfo	Carlee	2	542
12	PO Box 61000	San Francisco	CA	94161	(617) 555-0700	Lloyd	Angel	1	542
13	NULL	Columbus	OH	43260	(614) 555-8600	Armando	Jan	2	542
14	NULL	Fresno	CA	93721	NULL	Kraggin	Laura	1	542

4). Write an UPDATE statement that modifies the InvoiceCopy table. Change the TermsID to 5 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).

Comments: Update statement is used to modify the values of the rows which satisfy the condition.here the condition is that the TermsID should be 2 which should be changed to 5.The condition is mentioned in the where clause for the update statement

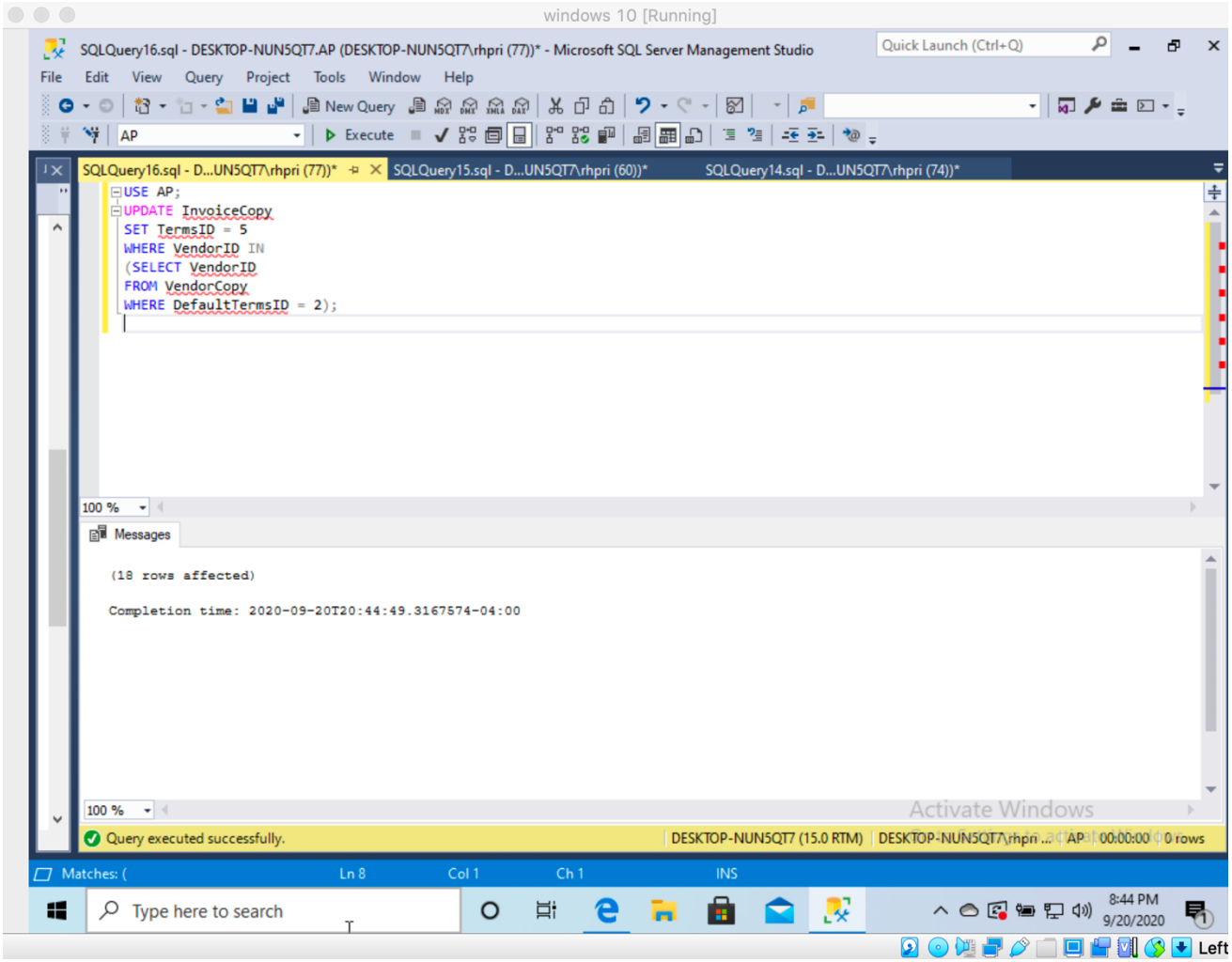
Before Updating:

```
USE AP;
SELECT VendorCopy.VendorID, VendorCopy.DefaultTermsID, InvoiceCopy.TermsID
FROM VendorCopy JOIN InvoiceCopy
ON VendorCopy.VendorID = InvoiceCopy.VendorID
and VendorCopy.DefaultTermsID = '2'
```



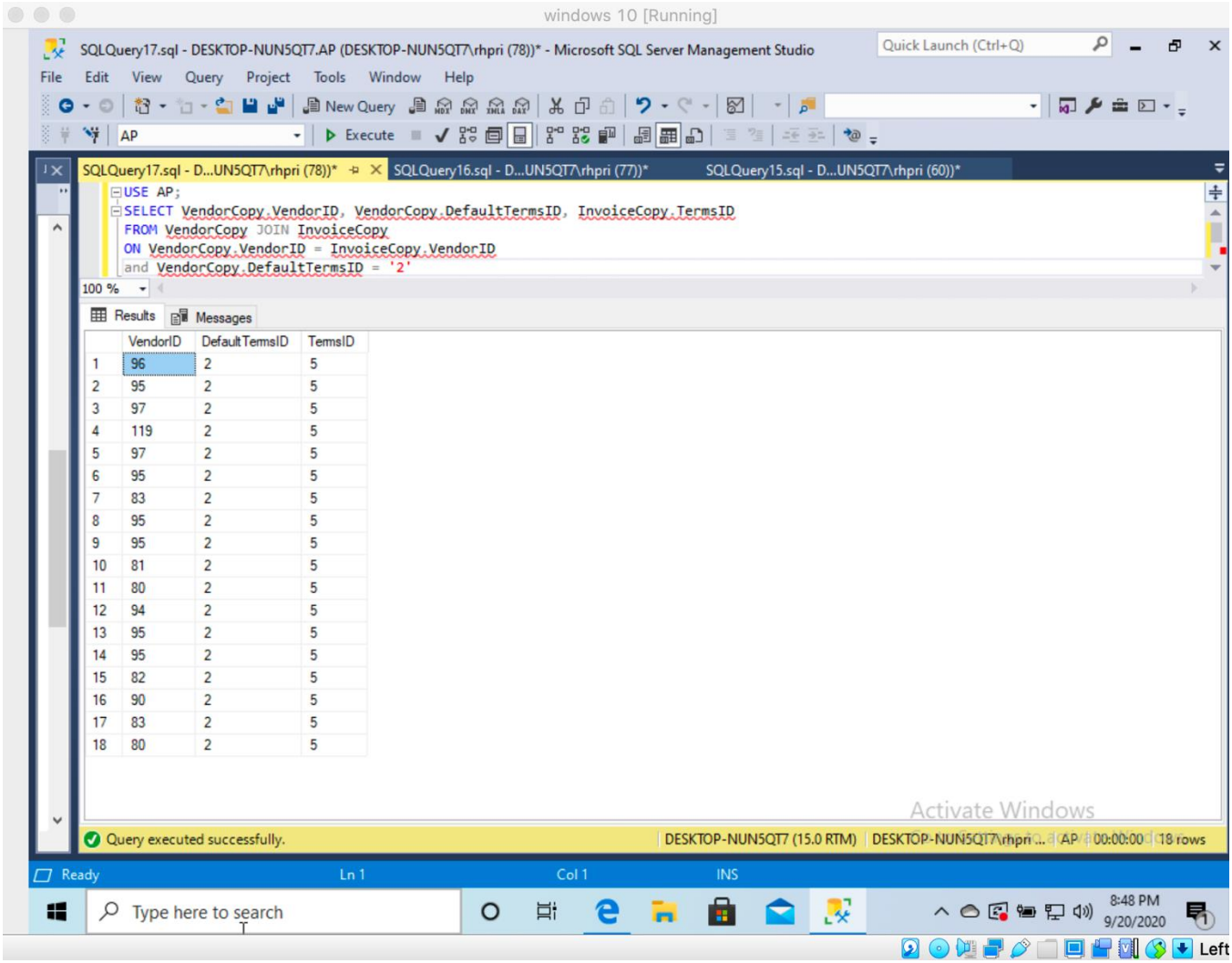
UPDATE:

```
USE AP;
UPDATE InvoiceCopy
SET TermsID = 5
WHERE VendorID IN
(SELECT VendorID
FROM VendorCopy
WHERE DefaultTermsID = 2);
```



After Updating:

```
USE AP;
SELECT VendorCopy.VendorID, VendorCopy.DefaultTermsID, InvoiceCopy.TermsID
FROM VendorCopy JOIN InvoiceCopy
ON VendorCopy.VendorID = InvoiceCopy.VendorID
and VendorCopy.DefaultTermsID = '2'
```

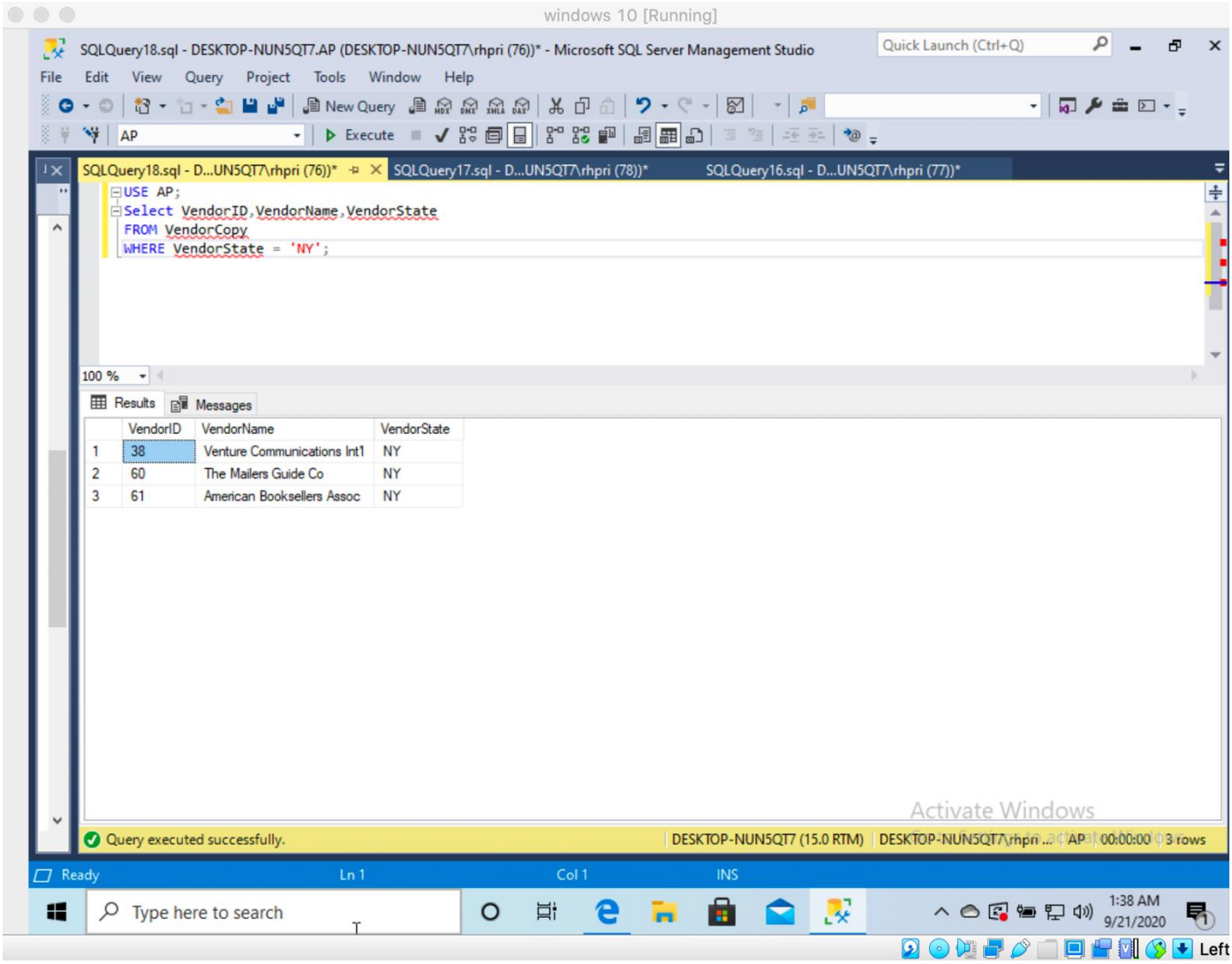


5. Write a DELETE statement that deletes all vendors in the state of ‘New York’ from the VendorCopy table. (USE SELECT statement to verify data changes in the table before and after the modification).

Comments: Delete statement is used to delete the existing rows in the table. Depending on the condition a single or multiple row is deleted, the condition is specified in the where clause. Here we are told to delete the vendors with New York state.

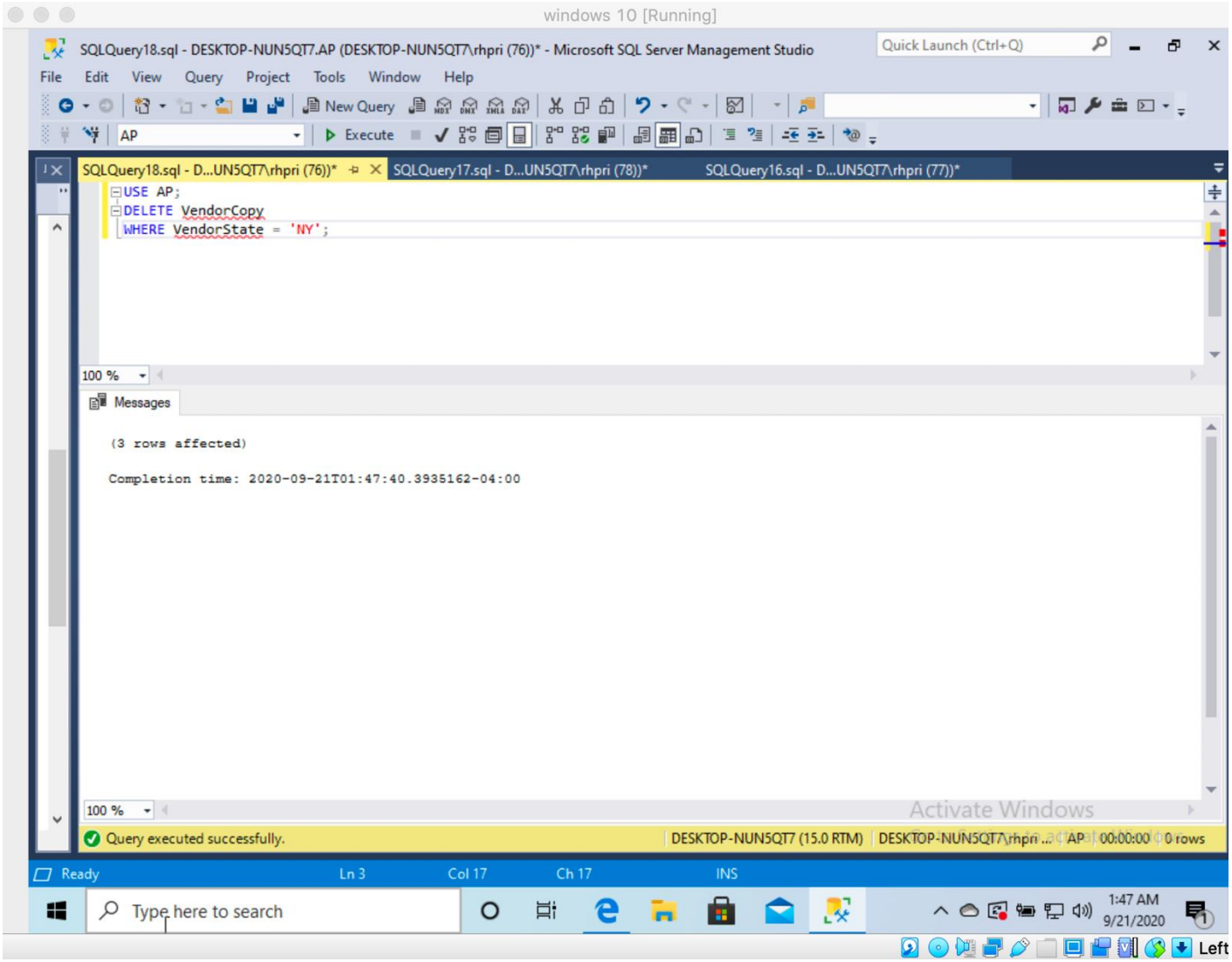
Sol:

```
USE AP;
SELECT VendorID,VendorName,VendorState
FROM VendorCopy
WHERE VendorState='NY';
```



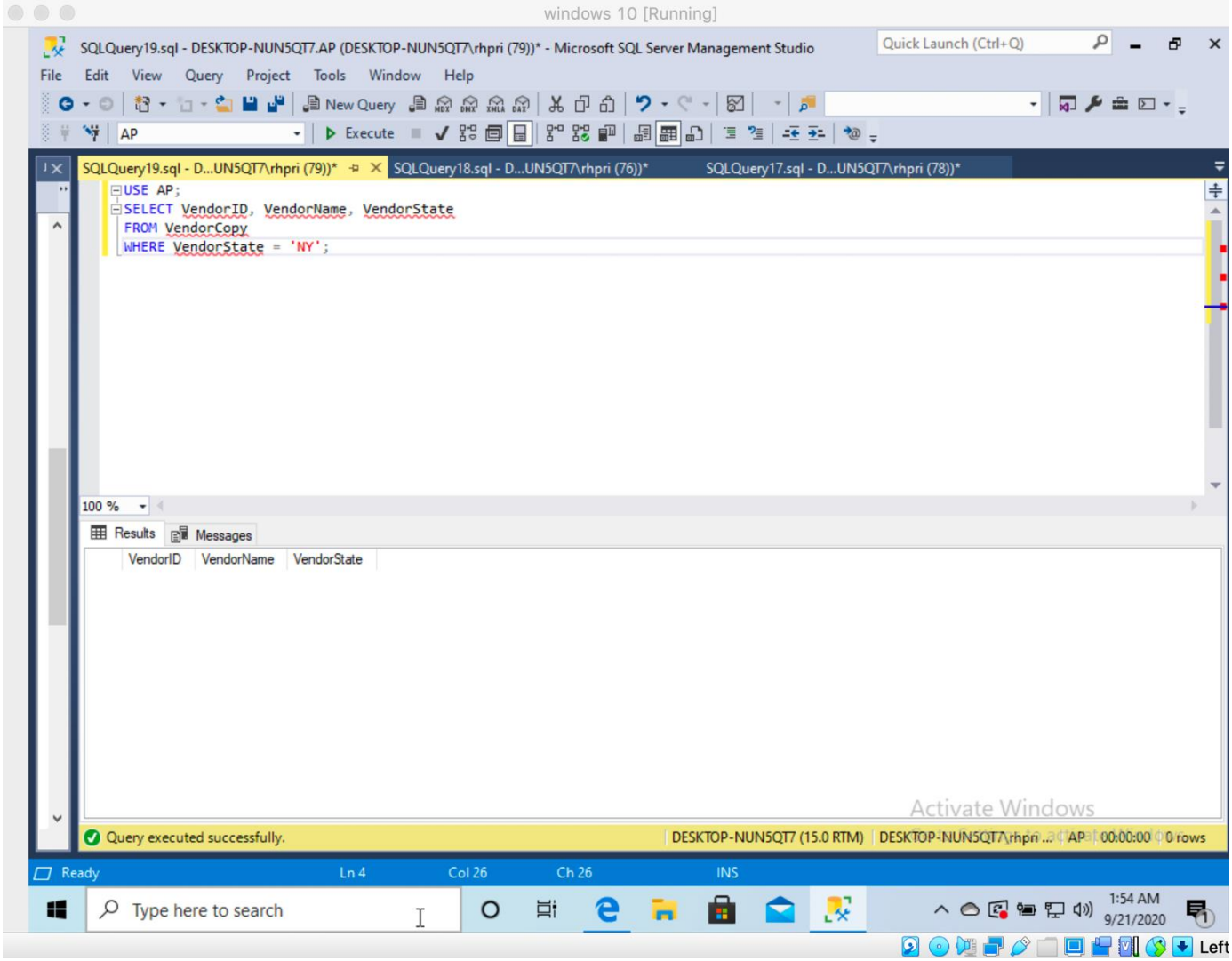
Sol:

```
USE AP;
DELETE VendorCopy
WHERE VendorState='NY';
```



After Deleting:

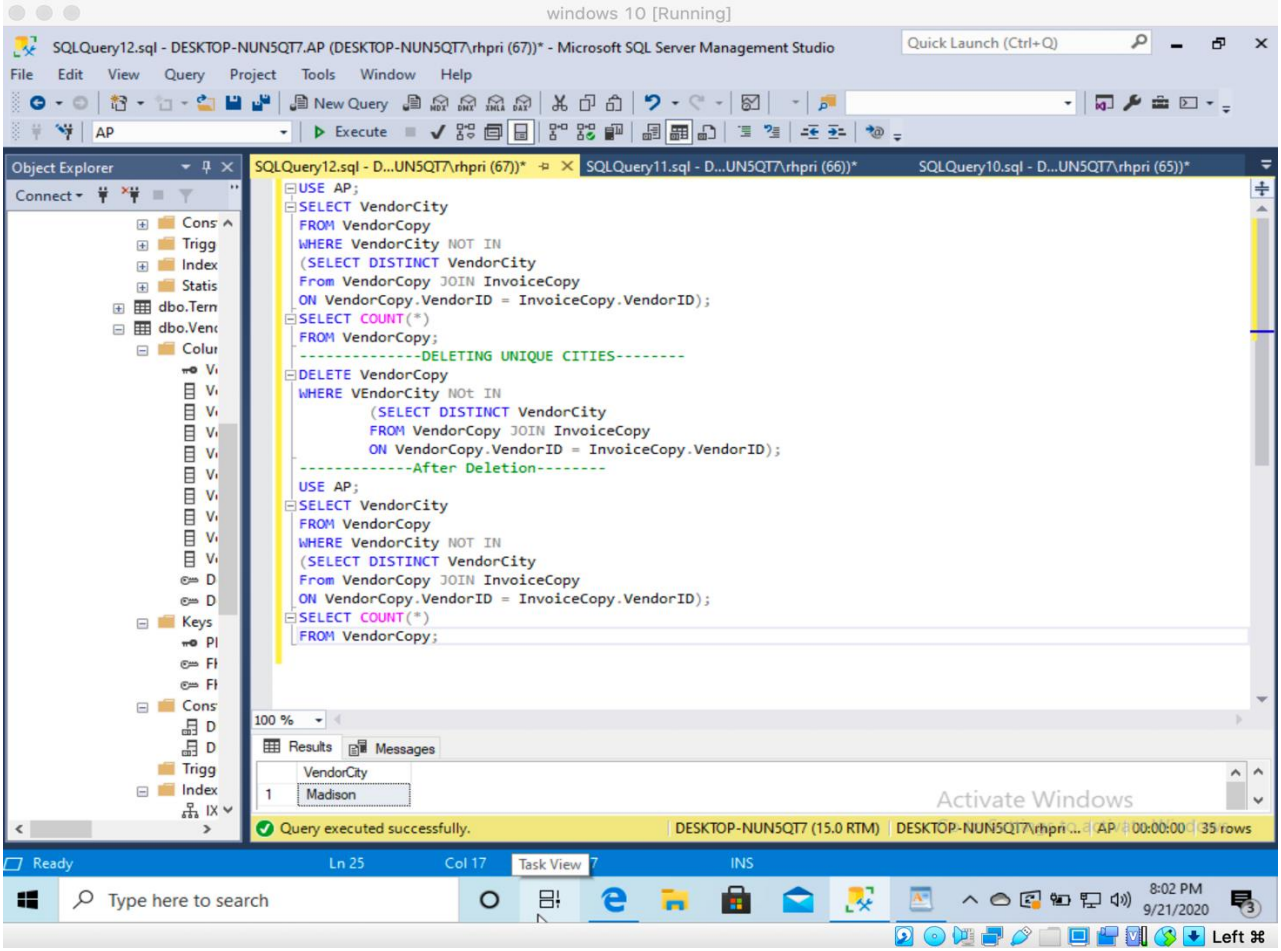
```
USE AP;
SELECT VendorID,VendorName,VendorState
FROM VendorCopy
WHERE VendorState='NY';
```

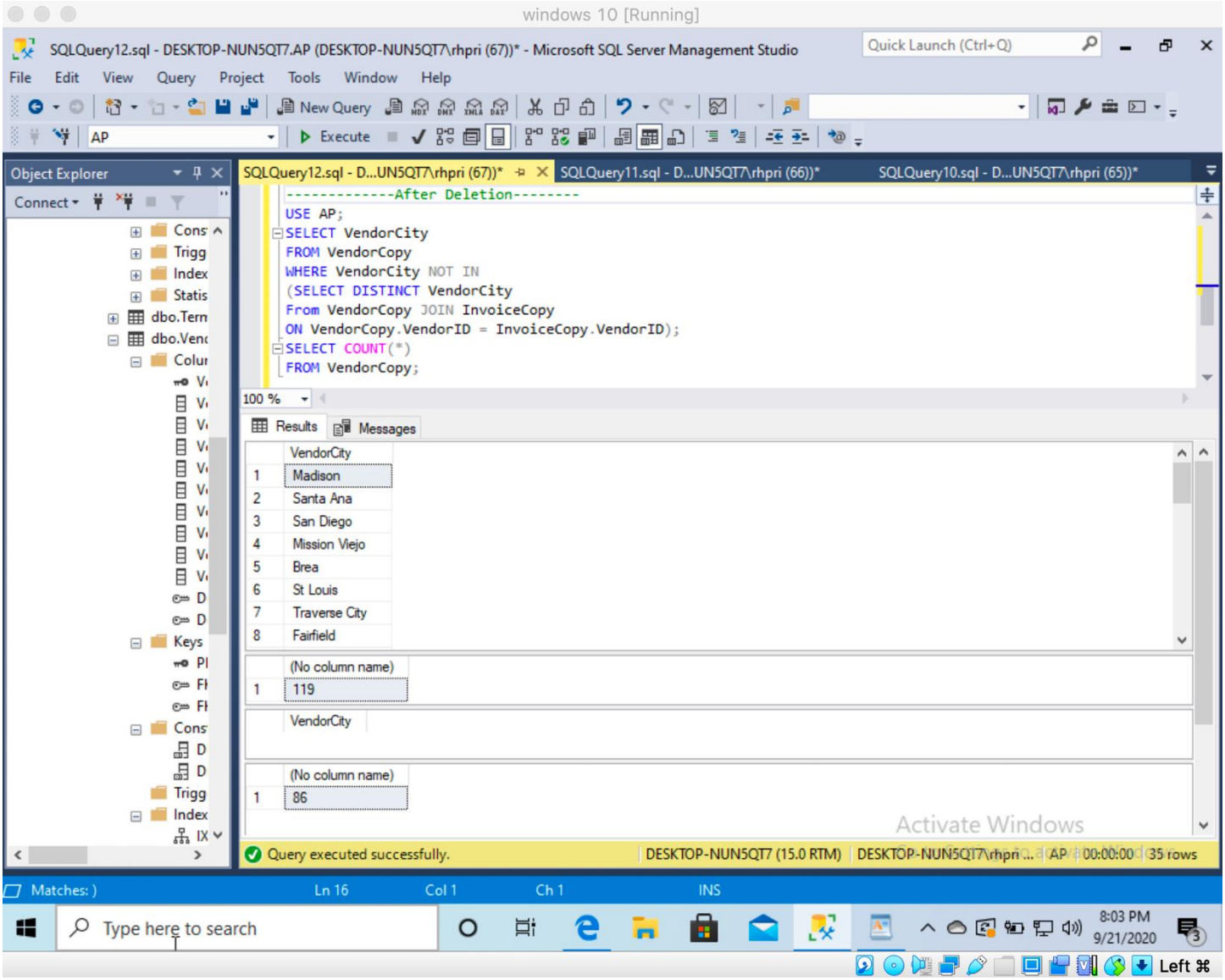


6.Write a DELETE statement for the VendorCopy table. Delete the vendors that are located in cities from which no vendor has ever sent an invoice. (USE SELECT statement to verify data changes in the table before and after the modification)

Comments: Here the question is delete the rows which have no vendor invoice to do that I have joined vendorCopy and Invoice copy table and then not selecting those vendors which are there in the invoice table
BEFORE DELETION :119
AFTER DELETION:86

```
USE AP;
SELECT VendorCity
FROM VendorCopy
WHERE VendorCity NOT IN
(SELECT DISTINCT VendorCity
From VendorCopy JOIN InvoiceCopy
ON VendorCopy.VendorID = InvoiceCopy.VendorID);
SELECT COUNT (*)
FROM VendorCopy;
--------------DELETING UNIQUE CITIES--------
DELETE VendorCopy
WHERE VEndorCity NOt IN
                (SELECT DISTINCT VendorCity
                FROM VendorCopy JOIN InvoiceCopy
                ON VendorCopy.VendorID = InvoiceCopy.VendorID);
-------------After Deletion--------
USE AP;
SELECT VendorCity
FROM VendorCopy
WHERE VendorCity NOT IN
(SELECT DISTINCT VendorCity
From VendorCopy JOIN InvoiceCopy
ON VendorCopy.VendorID = InvoiceCopy.VendorID);
SELECT COUNT(*)
FROM VendorCopy;
```





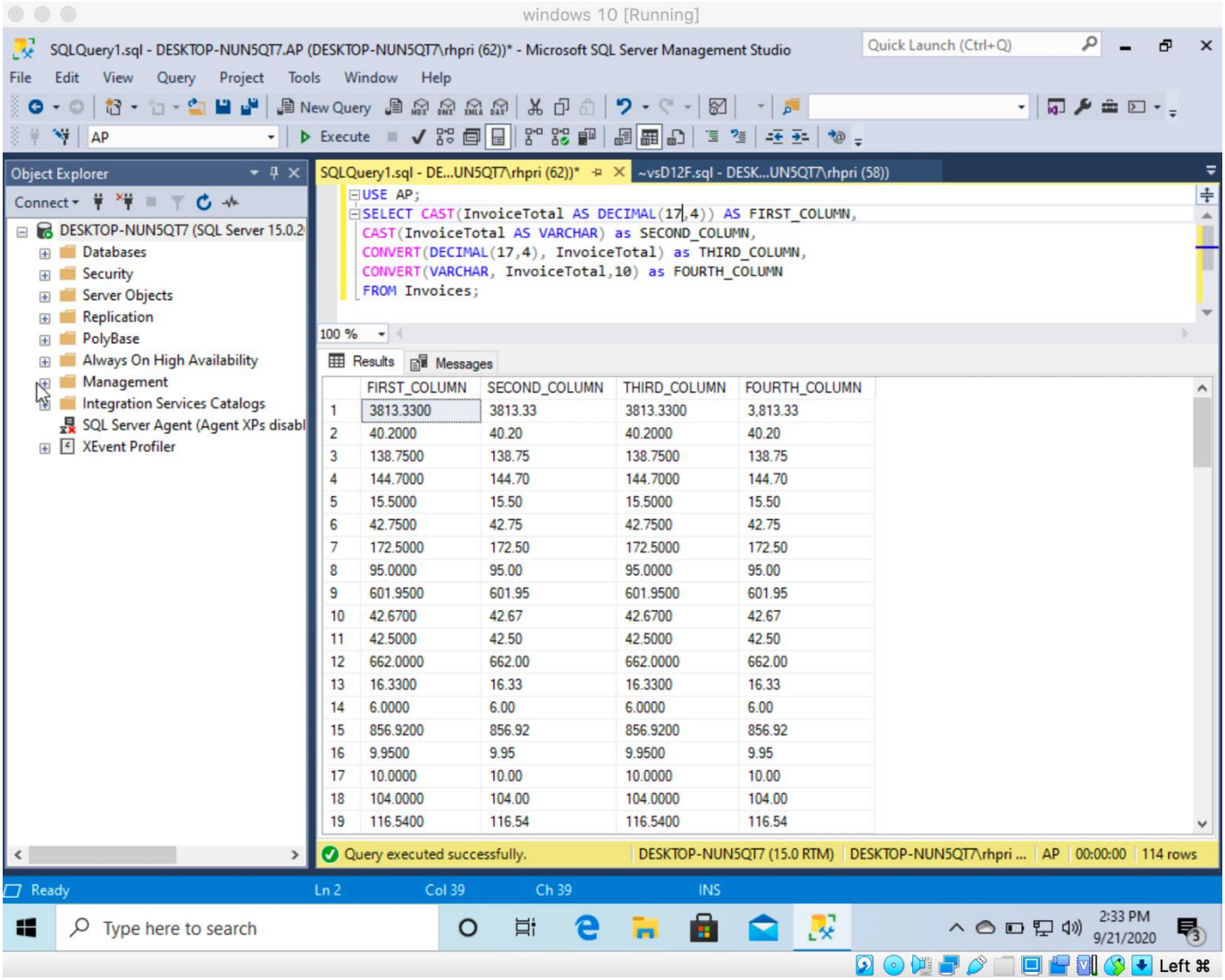
7. Write a SELECT statement that returns four columns based on the InvoiceTotal column of the Invoices table:

- Use CAST function to return the first column as data type decimal with 4 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 10.

Comments: CAST function converts an expression from one datatype to another datatype. If the conversion fails, the function will return an error CAST function converts an expression from one datatype to another datatype. If the conversion fails, the function will return an error

SOL:

```
USE AP;
SELECT CAST (InvoiceTotal AS DECIMAL(17,4)) AS FIRST_COLUMN,
CAST (InvoiceTotal AS VARCHAR) AS SECOND_COLUMN,
CONVERT(DECIMAL (17,4),InvoiceTotal) AS THIRD_COLUMN,
CONVERT (VARCHAR, InvoiceTotal,2) AS FOURTH_COLUMN
FROM Invoices;
```



8. Write a SELECT statement that returns four columns based on the InvoiceDate 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 5 and style 9, respectively.
- Use the CAST function to return the fourth column as a data type real.

Sol:

```
USE AP;
SELECT CAST(InvoiceDate AS VARCHAR) AS FIRST_COLUMN,
CONVERT(VARCHAR,InvoiceDate,2) AS SECOND_COLUMN,
CONVERT(VARCHAR,InvoiceDate,10) AS THIRD_COLUMN,
CAST(InvoiceDate AS REAL) AS FOURTH_COLUMN
FROM Invoices;
```

Comments: Here we have casted first column as Varchar directly ,CAST (expression AS [data type]) this is the format of the cast statement ,style is an integer expression which determines how will the function traslate.