

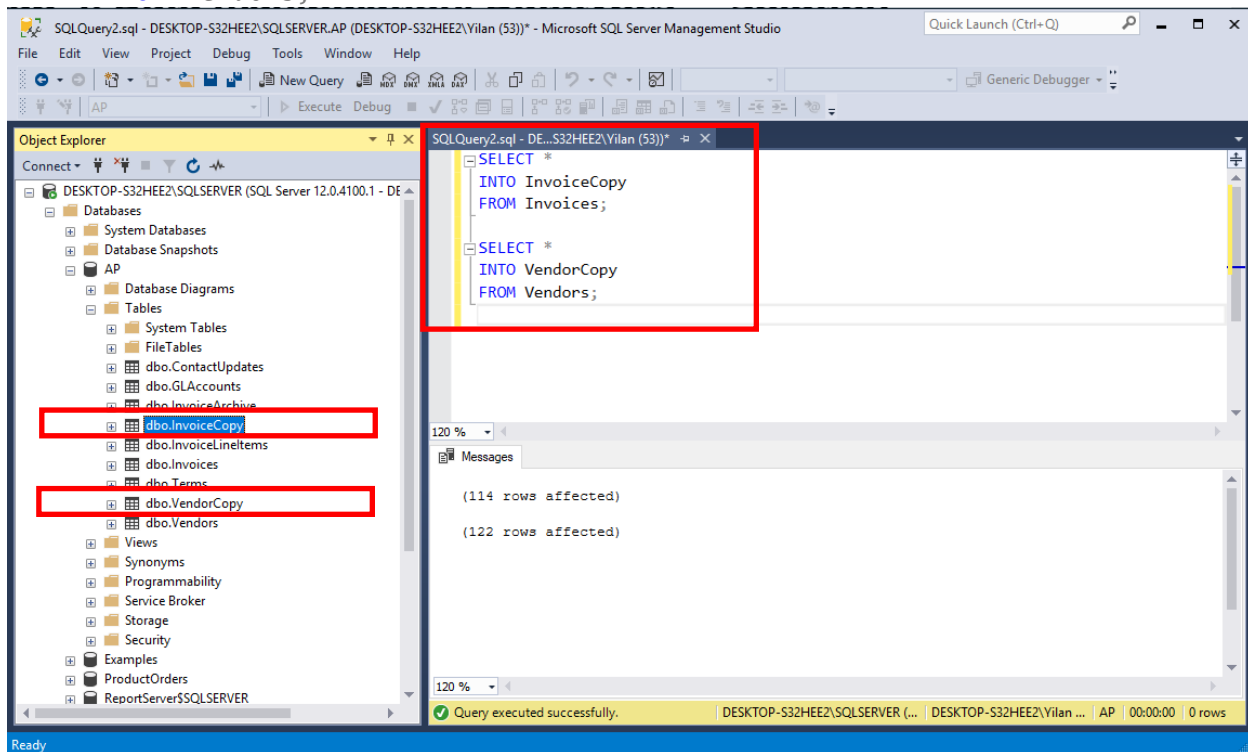


Lab 4: Data manipulation, Datatypes Solution

1. Create VendorCopy table and InvoiceCopy table.

```
SELECT *  
INTO InvoiceCopy  
FROM Invoices;
```

```
SELECT *  
INTO VendorCopy  
FROM Vendors;
```



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: 32 InvoiceTotal: \$ 434.58
 TermsID: 2 InvoiceNumber: AX-014-027
 PaymentTotal: \$0.00 InvoiceDueDate: 07/8/12
 InvoiceDate: 6/21/12 CreditTotal: \$0.00

Before insertion:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window contains the following SQL code:

```
-- before insertion
SELECT *
FROM InvoiceCopy;
```

The results grid displays the following data:

InvoiceID	VendorID	InvoiceNumber	InvoiceDate	InvoiceTotal	PaymentTotal	CreditTotal	TermsID	InvoiceDueDate
1	122	989319-457	2015-12-08 00:00:00	3813.33	3813.33	0.00	3	2016-01-08 00:00:00
2	123	263253241	2015-12-10 00:00:00	40.20	40.20	0.00	3	2016-01-10 00:00:00
3	123	963253234	2015-12-13 00:00:00	138.75	138.75	0.00	3	2016-01-13 00:00:00
4	123	2-000-2993	2015-12-16 00:00:00	144.70	144.70	0.00	3	2016-01-16 00:00:00
5	123	963253251	2015-12-16 00:00:00	15.50	15.50	0.00	3	2016-01-16 00:00:00
6	123	963253261	2015-12-16 00:00:00	42.75	42.75	0.00	3	2016-01-16 00:00:00
7	123	963253237	2015-12-21 00:00:00	172.50	172.50	0.00	3	2016-01-21 00:00:00
8	89	125520-1	2015-12-24 00:00:00	95.00	95.00	0.00	1	2016-01-04 00:00:00
9	121	97/488	2015-12-24 00:00:00	601.95	601.95	0.00	3	2016-01-24 00:00:00

The status bar at the bottom indicates "Query executed successfully." and "114 rows".

Insertion:

```
INSERT InvoiceCopy
VALUES (32, 'AX-014-027', '2012-6-21', 434.58, 0, 0,
2, '2012-07-08', NULL);
```

The screenshot shows the same Microsoft SQL Server Management Studio interface. The query window now contains the following SQL code:

```
-- before insertion
SELECT *
FROM InvoiceCopy;

--insertion
INSERT InvoiceCopy
VALUES (32, 'AX-014-027', '2012-6-21', 434.58, 0, 0,
2, '2012-07-08', NULL);
```

The status bar at the bottom indicates "Query executed successfully." and "1 row affected".

After insertion:

The screenshot displays the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the 'AP' database and its tables. The SQL Query window in the center contains the following code:

```
-- before insertion
SELECT *
FROM InvoiceCopy;

--insertion
INSERT InvoiceCopy
VALUES (32, 'AX-014-027', '2012-6-21', 434.58, 0, 0,
2, '2012-07-08', NULL);

-- after insertion
SELECT *
FROM InvoiceCopy;
```

The Results tab at the bottom shows a table with 115 rows. The columns are: InvoiceID, VendorID, InvoiceNumber, InvoiceDate, InvoiceTotal, PaymentTotal, CreditTotal, TermsID, and InvoiceDueDate. The status bar at the bottom indicates 'Query executed successfully' and '115 rows'.

InvoiceID	VendorID	InvoiceNumber	InvoiceDate	InvoiceTotal	PaymentTotal	CreditTotal	TermsID	InvoiceDueDate
115	32	AX-014-027	2012-06-21 00:00:00	434.58	0.00	0.00	2	2012-07-08 00:00:00
1	122	989319-457	2015-12-08 00:00:00	3813.33	3813.33	0.00	3	2016-01-08 00:00:00
2	123	263253241	2015-12-10 00:00:00	40.20	40.20	0.00	3	2016-01-10 00:00:00
3	123	963253234	2015-12-13 00:00:00	138.75	138.75	0.00	3	2016-01-13 00:00:00
4	123	2-000-2993	2015-12-16 00:00:00	144.70	144.70	0.00	3	2016-01-16 00:00:00
5	123	963253251	2015-12-16 00:00:00	15.50	15.50	0.00	3	2016-01-16 00:00:00
6	123	963253261	2015-12-16 00:00:00	42.75	42.75	0.00	3	2016-01-16 00:00:00
7	123	963253237	2015-12-21 00:00:00	172.50	172.50	0.00	3	2016-01-21 00:00:00
8	89	125520-1	2015-12-24 00:00:00	95.00	95.00	0.00	1	2016-01-04 00:00:00

- Write an UPDATE statement that modifies the VendorCopy table. Change the default account number to 403 for each vendor that has a default account number of 400. (Use SELECT statement to verify data changes in the table before and after the modification)

Before update:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'DESKTOP-S32HEE2\SQLSERVER (SQL Server 12.0.4100.1 - DE...)' under the 'AP' database. The main query window shows a SELECT statement: `-- before update`
`SELECT *`
`FROM VendorCopy;` The Results pane displays a table with 8 columns: VendorState, VendorZipCode, VendorPhone, VendorContactLName, VendorContactFName, DefaultTermsID, and DefaultAccountNo. The data is as follows:

VendorState	VendorZipCode	VendorPhone	VendorContactLName	VendorContactFName	DefaultTermsID	DefaultAccountNo
CA	93888	NULL	Aileen	Joan	1	235
CA	92691	(214) 555-3647	Keeton	Gonzalo	3	540
CA	93718	(559) 555-3005	Chaddick	Derek	3	403
CA	93721	(559) 555-7473	Randrup	Leann	2	400
CA	92621	(714) 555-4541	Lane	Vanesa	3	400
MO	63105	(314) 555-8834	Marques	Malia	3	540
OH	43221	(614) 555-4435	Evan	Emily	2	160
CA	93706	(559) 555-2993	Alexis	Alexandro	3	532
IA	52353	NULL	Halle	Juliana	3	570

The status bar at the bottom indicates 'Query executed successfully.' and '122 rows'.

Update:

```
UPDATE VendorCopy
SET DefaultAccountNo = 403
WHERE DefaultAccountNo = 400;
```

The screenshot shows the Microsoft SQL Server Management Studio interface after executing the UPDATE statement. The Object Explorer on the left is the same as in the previous screenshot. The main query window shows the UPDATE statement: `-- update`
`UPDATE VendorCopy`
`SET DefaultAccountNo = 403`
`WHERE DefaultAccountNo = 400;` The Messages pane displays the message: `(6 rows affected)`. The status bar at the bottom indicates 'Query executed successfully.' and '0 rows'.

After update:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following SQL code:

```
-- before update
SELECT *
FROM VendorCopy;

-- update
UPDATE VendorCopy
SET DefaultAccountNo = 403
WHERE DefaultAccountNo = 400;

-- after update
SELECT *
FROM VendorCopy;
```

The results grid shows the data after the update. The columns are: VendorState, VendorZipCode, VendorPhone, VendorContactLName, VendorContactFName, DefaultTermsID, and DefaultAccountNo. The rows are numbered 16 to 24. A red box highlights the rows where DefaultAccountNo is 403.

VendorState	VendorZipCode	VendorPhone	VendorContactLName	VendorContactFName	DefaultTermsID	DefaultAccountNo
CA	93888	NULL	Aleen	Joan	1	235
CA	92691	(214) 555-3647	Keeton	Gonzalo	3	540
CA	93718	(559) 555-3005	Chaddick	Derek	3	403
CA	93721	(559) 555-7473	Randrup	Leann	2	403
CA	92621	(714) 555-4541	Lane	Vanesa	3	403
MO	63105	(314) 555-8834	Marques	Malia	3	540
OH	43221	(614) 555-4435	Evan	Emily	2	160
CA	93706	(559) 555-2993	Alexis	Alexandro	3	532
IA	52353	NULL	Hallie	Juliana	3	570

The status bar at the bottom indicates: Query executed successfully. DESKTOP-S32HEE2\SQLSERVER (... | DESKTOP-S32HEE2\Yilan ... | AP | 00:00:00 | 122 rows

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

Before update:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'DESKTOP-S32HEE2\SQLSERVER (SQL Server 12.0.4100.1 - DE...)' with the 'AP' database selected. The query editor in the center contains the following SQL code:

```
-- before update
SELECT VendorCopy.VendorID, VendorCopy.DefaultTermsID, InvoiceCopy.TermsID
FROM VendorCopy JOIN InvoiceCopy
ON VendorCopy.VendorID = InvoiceCopy.VendorID
and VendorCopy.DefaultTermsID = 2;
```

The Results pane on the right displays the output of the query as a table with 18 rows. The columns are VendorID, DefaultTermsID, and TermsID. The data is as follows:

VendorID	DefaultTermsID	TermsID
4	119	2
5	97	2
6	95	2
7	83	2
8	95	2
9	95	2
10	81	2
11	80	2
12	94	2
13	95	2
14	95	2
15	82	2
16	90	2
17	83	2
18	80	2

The status bar at the bottom indicates 'Query executed successfully' and '18 rows'.

Update:

```
UPDATE InvoiceCopy
SET TermsID = 4
WHERE VendorID IN
(SELECT VendorID
FROM VendorCopy
WHERE DefaultTermsID = 2);
```

The screenshot shows the Microsoft SQL Server Management Studio interface after executing the UPDATE query. The query editor contains the following SQL code:

```
-- update
UPDATE InvoiceCopy
SET TermsID = 4
WHERE VendorID IN
(SELECT VendorID
FROM VendorCopy
WHERE DefaultTermsID = 2);
```

The Messages pane on the right displays the execution result: '(18 rows affected)'. The status bar at the bottom indicates 'Query executed successfully' and '0 rows'.

After update:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL code:

```
WHERE DefaultTermsID = 2);  
  
-- after update  
SELECT VendorCopy_VendorID, VendorCopy_DefaultTermsID, InvoiceCopy_TermsID  
FROM VendorCopy JOIN InvoiceCopy  
ON VendorCopy_VendorID = InvoiceCopy_VendorID  
and VendorCopy_DefaultTermsID = 2;
```

The results pane shows the following data:

VendorID	DefaultTermsID	TermsID
4	119	2
5	97	2
6	95	2
7	83	2
8	95	2
9	95	2
10	81	2
11	80	2
12	94	2
13	95	2
14	95	2
15	82	2
16	90	2
17	83	2
18	80	2

The status bar at the bottom indicates: Query executed successfully. DESKTOP-S32HEE2\SQLSERVER (... | DESKTOP-S32HEE2\Vilan ... | AP | 00:00:00 | 18 rows

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

Before deletion:

SQLQuery3.sql - DESKTOP-S32HEE2\SQLSERVER.AP (DESKTOP-S32HEE2\Yilan (58)) - Microsoft SQL Server Management Studio

```
-- before deletion
SELECT *
FROM VendorCopy;

SELECT COUNT(*) AS CountTotal
FROM VendorCopy;

SELECT COUNT(*) AS CountCA
FROM VendorCopy
WHERE VendorState = 'CA';
```

VendorID	VendorName	VendorAddress1	VendorAddress2	VendorCity	VendorState	VendorZip
1	US Postal Service	Attn: Supt. Window Services	PO Box 7005	Madison	WI	53703
2	National Information Data Ctr	PO Box 96621	NULL	Washington	DC	20540
3	Register of Copyrights	Library Of Congress	NULL	Washington	DC	20540
4	Jobtrak	1990 Westwood Blvd Ste 260	NULL	Los Angeles	CA	90024
5	Newbrige Book Clubs	3000 Cindel Drive	NULL	Washington	NJ	07033
6	California Chamber Of Commerce	3255 Ramos Cir	NULL	Sacramento	CA	95834

	CountTotal
1	122

	CountCA
1	75

Query executed successfully. DESKTOP-S32HEE2\SQLSERVER (...) DESKTOP-S32HEE2\Yilan ... AP 00:00:00 124 rows

Deletion:

```
DELETE VendorCopy
WHERE VendorState = 'CA';
```

SQLQuery3.sql - DESKTOP-S32HEE2\SQLSERVER.AP (DESKTOP-S32HEE2\Yilan (58)) - Microsoft SQL Server Management Studio

```
FROM VendorCopy
WHERE VendorState = 'CA';

-- deletion
DELETE VendorCopy
WHERE VendorState = 'CA';
```

(75 rows affected)

Query executed successfully. DESKTOP-S32HEE2\SQLSERVER (...) DESKTOP-S32HEE2\Yilan ... AP 00:00:00 0 rows

After deletion:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL code:

```
-- after deletion
SELECT *
FROM VendorCopy;

SELECT COUNT(*) AS CountTotal
FROM VendorCopy;

SELECT COUNT(*) AS CountCA
FROM VendorCopy
WHERE VendorState = 'CA';
```

The Results pane shows the output of the query, which includes a list of vendors and their counts. The results are as follows:

VendorID	VendorName	VendorAddress1	VendorAddress2	VendorCity	VendorState	VendorZip
1	US Postal Service	Attn: Supt. Window Services	PO Box 7005	Madison	WI	53707
2	National Information Data Ctr	PO Box 96621	NULL	Washington	DC	20090
3	Register of Copyrights	Library Of Congress	NULL	Washington	DC	20559
4	Newbrige Book Clubs	3000 Cindel Drive	NULL	Washington	NJ	07882
5	The Library Ltd	7700 Forsyth	NULL	St Louis	MO	63105
6	Micro Center	1555 W Lane Ave	NULL	Columbus	OH	43221

Below the vendor list, the summary results are shown:

CountTotal
1 47

CountCA
1 0

The status bar at the bottom indicates that the query was executed successfully, returning 49 rows.

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)

Hint: Use a subquery coded with "SELECT DISTINCT VendorState" introduced with the NOT IN operator.

Before deletion:

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left displays the database structure for 'DESKTOP-S32HEE2\SQLSERVER (SQL Server 12.0.4100.1 - DE...'. The query window on the right contains the following SQL code:

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

SELECT COUNT(*)
FROM VendorCopy;
```

The Results pane shows the output of the query:

VendorState
10 NC
11 IL
12 IL
13 NY
14 NY
15 IL
16 CT
17 KS

The status bar at the bottom indicates 'Query executed successfully.' and '18 rows'.

Deletion:

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window on the right contains the following SQL code:

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

The Messages pane shows the output of the query:

(17 rows affected)

The status bar at the bottom indicates 'Query executed successfully.' and '0 rows'.

After deletion:

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the Object Explorer with the database structure of 'DESKTOP-S32HEE2\SQLSERVER (SQL Server 12.0.4100.1 - DE...)' expanded, showing tables like 'dbo.ContactUpdates', 'dbo.GLAAccounts', 'dbo.InvoiceArchive', 'dbo.InvoiceCopy', 'dbo.InvoiceLineItems', 'dbo.Invoices', 'dbo.Terms', 'dbo.VendorCopy', and 'dbo.Vendors'. The right pane shows a SQL query window with the following code:

```
-- after deletion
SELECT VendorState
FROM VendorCopy
WHERE VendorState NOT IN
(SELECT DISTINCT VendorState
FROM VendorCopy JOIN InvoiceCopy
ON VendorCopy_VendorID = InvoiceCopy_VendorID);

SELECT COUNT(*)
FROM VendorCopy;
```

The query results are displayed in the 'Results' tab, showing a single row with the value 30. The column header is '(No column name)'. The status bar at the bottom indicates 'Query executed successfully.' and '1 rows'.

(No column name)
30

7. Write an 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 2 digits to the right of the decimal point.
- Use CAST to return the second column as a varchar
- Use CONVERT function to return the third column as the same data type as the first column.
- Use CONVERT to return the fourth column as a varchar, using style 1.

```
SELECT CAST(InvoiceTotal AS DECIMAL(17,2)) AS CastAsDecimal,  
       CAST(InvoiceTotal AS VARCHAR) AS CastAsVarchar,  
       CONVERT(DECIMAL(17,2),InvoiceTotal) AS ConvertToDecimal,  
       CONVERT(VARCHAR,InvoiceTotal,1) AS ConvertToVarchar  
FROM Invoices;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays the following SQL statement:

```
SELECT CAST(InvoiceTotal AS DECIMAL(17,2)) AS CastAsDecimal,  
       CAST(InvoiceTotal AS VARCHAR) AS CastAsVarchar,  
       CONVERT(DECIMAL(17,2),InvoiceTotal) AS ConvertToDecimal,  
       CONVERT(VARCHAR,InvoiceTotal,1) AS ConvertToVarchar  
FROM Invoices;
```

The query has been executed successfully, and the results are displayed in the Results pane. The results table has four columns: CastAsDecimal, CastAsVarchar, ConvertToDecimal, and ConvertToVarchar. The data is as follows:

	CastAsDecimal	CastAsVarchar	ConvertToDecimal	ConvertToVarchar
1	3813.33	3813.33	3813.33	3,813.33
2	40.20	40.20	40.20	40.20
3	138.75	138.75	138.75	138.75
4	144.70	144.70	144.70	144.70
5	15.50	15.50	15.50	15.50
6	42.75	42.75	42.75	42.75
7	172.50	172.50	172.50	172.50
8	95.00	95.00	95.00	95.00
9	601.95	601.95	601.95	601.95
10	42.67	42.67	42.67	42.67
11	42.50	42.50	42.50	42.50
12	662.00	662.00	662.00	662.00
13	16.33	16.33	16.33	16.33

The status bar at the bottom indicates that the query was executed successfully, returning 114 rows.

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 1 and style 10, respectively.
- Use the CAST function to return the fourth column as data type real.

```
SELECT CAST(InvoiceDate AS varchar) AS CastAsVarchar,  
       CONVERT(varchar,InvoiceDate,1) AS ConvertToStyle1,  
       CONVERT(varchar,InvoiceDate,10) AS ConvertToStyle10,  
       CAST(InvoiceDate AS real) AS ConvertToReal
```

FROM Invoices;

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL statement:

```
SELECT CAST(InvoiceDate AS varchar) AS CastAsVarchar,  
       CONVERT(varchar,InvoiceDate,1) AS ConvertToStyle1,  
       CONVERT(varchar,InvoiceDate,10) AS ConvertToStyle10,  
       CAST(InvoiceDate AS real) AS ConvertToReal  
FROM Invoices;
```

The Results pane shows the output of the query, which is a table with four columns: CastAsVarchar, ConvertToStyle1, ConvertToStyle10, and ConvertToReal. The table contains 13 rows of data, with the first row highlighted. The data is as follows:

	CastAsVarchar	ConvertToStyle1	ConvertToStyle10	ConvertToReal
1	Apr 2 2016 12:00AM	04/02/16	04-02-16	42460
2	Apr 1 2016 12:00AM	04/01/16	04-01-16	42459
3	Mar 31 2016 12:00AM	03/31/16	03-31-16	42458
4	Mar 30 2016 12:00AM	03/30/16	03-30-16	42457
5	Mar 28 2016 12:00AM	03/28/16	03-28-16	42455
6	Mar 25 2016 12:00AM	03/25/16	03-25-16	42452
7	Mar 24 2016 12:00AM	03/24/16	03-24-16	42451
8	Mar 24 2016 12:00AM	03/24/16	03-24-16	42451
9	Mar 24 2016 12:00AM	03/24/16	03-24-16	42451
10	Mar 24 2016 12:00AM	03/24/16	03-24-16	42451
11	Mar 23 2016 12:00AM	03/23/16	03-23-16	42450
12	Mar 23 2016 12:00AM	03/23/16	03-23-16	42450
13	Mar 23 2016 12:00AM	03/23/16	03-23-16	42450

The status bar at the bottom indicates that the query was executed successfully and returned 114 rows.