

# Lab 4: Data manipulation, Data types

## 1. Create VendorCopy table and InvoiceCopy table.

As you can see as below, the two table are created.

The image displays two screenshots of the SQL Server Enterprise Manager interface, specifically the SQL Query window, showing the execution of two SQL queries to create new tables.

**Top Screenshot:** The query window shows the following SQL statement:

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

The Messages pane below the query window indicates the execution results:

```
(122 rows affected)  
Completion time: 2020-09-23T19:06:16.5324101+08:00
```

**Bottom Screenshot:** The query window shows the following SQL statement:

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

The Messages pane below the query window indicates the execution results:

```
(114 rows affected)  
Completion time: 2020-09-23T19:08:10.7779865+08:00
```

At the bottom of the interface, a status bar shows a green checkmark and the message: "Query executed successfully."

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):

The screenshot is as below:

And we actually do not need insert InvoiceID because it would display automatically.

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the following SQL script:

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

SELECT *
FROM InvoiceCopy
WHERE InvoiceNumber = 'CM-007-700';
```

The bottom pane shows the results of the query, displaying a single row in the 'Results' tab:

	InvoiceID	VendorID	InvoiceNumber	InvoiceDate	InvoiceTotal	PaymentTotal	CreditTotal	TermsID	InvoiceDueDate	PaymentDate
1	120	2	CM-007-700	2015-08-24 00:00:00	401.40	2.99	5.99	3	2015-09-01 00:00:00	NULL

The status bar at the bottom indicates: "Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (51) AP 00:00:00 1 rows".

**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).**

The screenshot is as below and the default account number have changed from 572 to 542.

Query3.sql - DES...PTHKBA0\lyq (52) X

```

SELECT *
FROM VendorCopy
WHERE DefaultAccountNo = 572;
UPDATE VendorCopy
SET DefaultAccountNo = 542
WHERE DefaultAccountNo = 572;
SELECT *
FROM VendorCopy
WHERE DefaultAccountNo = 542;

```

120 %

Results Messages

	VendorCity	VendorState	VendorZipCode	VendorPhone	VendorContactLName	VendorContactFName	DefaultTermsID	DefaultAccountNo
1	Los Angeles	CA	90025	(800) 555-8725	Quinn	Kenzie	3	572
2	Sacramento	CA	95827	(916) 555-6670	Mauro	Anton	3	572
3	Oberlin	OH	44074	NULL	Bernard	Lucy	3	572
4	Los Angeles	CA	90038	(213) 555-4322	Paris	Gideon	3	572
5	Jacksonville	FL	32231	(800) 555-6041	Gerald	Kristofer	3	572
6	Charlotte	NC	28217	(704) 555-3500	Bernardo	Brittnee	3	572
7	Chicago	IL	60680	(614) 555-3663	Holbrooke	Rashad	3	572
8	Marion	OH	43305	(800) 555-1669	Carrollton	Priscilla	3	572

  

	Address2	VendorCity	VendorState	VendorZipCode	VendorPhone	VendorContactLName	VendorContactFName	DefaultTermsID	DefaultAccountNo
1		Los Angeles	CA	90025	(800) 555-8725	Quinn	Kenzie	3	542
2		Sacramento	CA	95827	(916) 555-6670	Mauro	Anton	3	542
3		Oberlin	OH	44074	NULL	Bernard	Lucy	3	542
4		Los Angeles	CA	90038	(213) 555-4322	Paris	Gideon	3	542

Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (52) AP 00:00:00 28 rows

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).

As you can see in the screenshot the 'TermsID' have changed to 5.

Query4.sql - DESKTOP-PTHKBA0\lyq (59)

```

SELECT *
FROM InvoiceCopy
WHERE VendorID IN (SELECT VendorID FROM VendorCopy
WHERE DefaultTermsID = 2);

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

SELECT *
FROM InvoiceCopy
WHERE VendorID IN (SELECT VendorID FROM VendorCopy
WHERE DefaultTermsID = 2);

```

120 %

Results Messages

	InvoiceID	VendorID	InvoiceNumber	InvoiceDate	InvoiceTotal	PaymentTotal	CreditTotal	TermsID	InvoiceDueDate	PaymentDate
1	12	96	I77271-001	2015-12-26 00:00:00	662.00	662.00	0.00	2	2016-01-16 00:00:00	2016-01-13 0
2	13	95	111-92R-10096	2015-12-30 00:00:00	16.33	16.33	0.00	2	2016-01-20 00:00:00	2016-01-23 0
3	16	97	21-4748363	2016-01-03 00:00:00	9.95	9.95	0.00	2	2016-01-23 00:00:00	2016-01-22 0
4	21	119	10843	2016-01-11 00:00:00	4901.26	4901.26	0.00	2	2016-01-31 00:00:00	2016-01-29 0
5	23	97	21-4923721	2016-01-13 00:00:00	9.95	9.95	0.00	2	2016-02-02 00:00:00	2016-01-28 0
6	44	95	111-92R-10094	2016-02-01 00:00:00	19.67	19.67	0.00	2	2016-02-21 00:00:00	2016-02-24 0
7	47	83	31359783	2016-02-03 00:00:00	1575.00	1575.00	0.00	2	2016-02-23 00:00:00	2016-02-21 0
8	49	95	111-92R-10097	2016-02-04 00:00:00	16.33	16.33	0.00	2	2016-02-24 00:00:00	2016-02-26 0

  

	InvoiceID	VendorID	InvoiceNumber	InvoiceDate	InvoiceTotal	PaymentTotal	CreditTotal	TermsID	InvoiceDueDate	PaymentDate
1	12	96	I77271-001	2015-12-26 00:00:00	662.00	662.00	0.00	5	2016-01-16 00:00:00	2016-01-13 0
2	13	95	111-92R-10096	2015-12-30 00:00:00	16.33	16.33	0.00	5	2016-01-20 00:00:00	2016-01-23 0
3	16	97	21-4748363	2016-01-03 00:00:00	9.95	9.95	0.00	5	2016-01-23 00:00:00	2016-01-22 0
4	21	119	10843	2016-01-11 00:00:00	4901.26	4901.26	0.00	5	2016-01-31 00:00:00	2016-01-29 0

Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (59) AP 00:00:00 36 rows

**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).**

As you can see, the vendors in the state of New York have been delete and could not be searched any more.

The screenshot shows a SQL Server Enterprise Manager interface. At the top, there are two tabs: 'Query5.sql - DES...PTHKBA0\lyq (70))' and 'Query4.sql - DES...PTHKBA0\lyq (59))'. The active query window contains the following SQL code:

```
SELECT *  
FROM VendorCopy  
WHERE VendorState = 'NY';  
  
DELETE FROM VendorCopy  
WHERE VendorState = 'NY';  
  
SELECT *  
FROM VendorCopy  
WHERE VendorState = 'NY';
```

Below the query window, the 'Results' tab is selected, displaying a grid with 3 rows and 10 columns. The columns are: VendorID, VendorName, VendorAddress1, VendorAddress2, VendorCity, VendorState, VendorZipCode, VendorPhone, VendorContactLName, and VendorContactFName. The data rows are:

	VendorID	VendorName	VendorAddress1	VendorAddress2	VendorCity	VendorState	VendorZipCode	VendorPhone	VendorContactLName	VendorContactFName
1	38	Venture Communications Int'l	60 Madison Ave	NULL	New York	NY	10010	(212) 555-4800	Neftaly	
2	60	The Mailers Guide Co	PO Box 1550	NULL	New Rochelle	NY	10802	NULL	Lacy	
3	61	American Booksellers Assoc	828 S Broadway	NULL	Tarrytown	NY	10591	(800) 555-0037	Angelica	

At the bottom of the window, a status bar indicates: 'Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (70) AP 00:00:00 3 rows'.

**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).**

As you can see in the screenshot, the vendors that located in the cities from which no vendor has ever sent an invoice have been deleted.

The screenshot shows a SQL query editor with three tabs: Query6.sql, Query5.sql, and Query4.sql. The active tab, Query6.sql, contains the following SQL code:

```
SELECT *  
FROM VendorCopy  
WHERE VendorID NOT IN (SELECT DISTINCT VendorID FROM InvoiceCopy);  
  
DELETE VendorCopy  
WHERE VendorID NOT IN (SELECT DISTINCT VendorID FROM InvoiceCopy);  
  
SELECT *  
FROM VendorCopy  
WHERE VendorID NOT IN (SELECT DISTINCT VendorID FROM InvoiceCopy);
```

Below the query editor, the 'Results' pane displays a table with 10 rows and 8 columns. The columns are VendorID, VendorName, VendorAddress1, VendorAddress2, VendorCity, VendorState, and VendorZipCode. The data is as follows:

	VendorID	VendorName	VendorAddress1	VendorAddress2	VendorCity	VendorState	VendorZipCode
1	1	US Postal Service	Attn: Supt. Window Services	PO Box 7005	Madison	WI	53707
2	3	Register of Copyrights	Library Of Congress	NULL	Washington	DC	20559
3	4	Jobtrak	1990 Westwood Blvd Ste 260	NULL	Los Angeles	CA	90025
4	5	Newbrige Book Clubs	3000 Cindel Drive	NULL	Washington	NJ	07882
5	6	California Chamber Of Commerce	3255 Ramos Cir	NULL	Sacramento	CA	95827
6	7	Towne Advertiser's Mailing Svcs	Kevin Minder	3441 W Macarthur Blvd	Santa Ana	CA	92704
7	8	BFI Industries	PO Box 9369	NULL	Fresno	CA	93792
8	9	Pacific Gas & Electric	Box 52001	NULL	San Francisco	CA	94152
9	10	Robbins Mobile Lock And Key	4669 N Fresno	NULL	Fresno	CA	93726

At the bottom of the screenshot, a status bar indicates: 'Query executed successfully. DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (67) AP 00:00:00 84 rows'.

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

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

The screenshot is as below.

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a SQL query in the 'Query7.sql' window. The query is as follows:

```
SELECT  
CAST ( InvoiceTotal AS decimal(10,4)) AS DecimalTotal1,  
CAST ( InvoiceTotal AS varchar) AS VarcharTotal1,  
CONVERT ( decimal(10,4), InvoiceTotal ) AS DecimalTotal2,  
CONVERT ( varchar, InvoiceTotal,10) AS VarcharTotal2  
FROM Invoices;
```

The bottom pane shows the 'Results' tab with a table containing 14 rows of data. The columns are DecimalTotal1, VarcharTotal1, DecimalTotal2, and VarcharTotal2. The first row is highlighted.

	DecimalTotal1	VarcharTotal1	DecimalTotal2	VarcharTotal2
1	3813.3300	3813.33	3813.3300	3,813.33
2	40.2000	40.20	40.2000	40.20
3	138.7500	138.75	138.7500	138.75
4	144.7000	144.70	144.7000	144.70
5	15.5000	15.50	15.5000	15.50
6	42.7500	42.75	42.7500	42.75
7	172.5000	172.50	172.5000	172.50
8	95.0000	95.00	95.0000	95.00
9	601.9500	601.95	601.9500	601.95
10	42.6700	42.67	42.6700	42.67
11	42.5000	42.50	42.5000	42.50
12	662.0000	662.00	662.0000	662.00
13	16.3300	16.33	16.3300	16.33
14	6.0000	6.00	6.0000	6.00

The status bar at the bottom indicates 'Query executed successfully.' and 'DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (74) AP 00:00:00 114 rows'.

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

The screenshot is as below:

The screenshot displays the SQL Server Enterprise Manager interface. At the top, three query windows are open: 'Query8.sql - DES...THKBA0\lyq (52)', 'Query7.sql - DES...PTHKBA0\lyq (74)', and 'Query2.sql - DES...PTHKBA0\lyq (51)'. The active window shows a SQL query:

```
SELECT  
CAST ( InvoiceDate AS varchar) AS VarcharDate,  
CONVERT ( varchar, InvoiceDate, 5) AS VarcharDate_5,  
CONVERT ( varchar, InvoiceDate, 9) AS VarchDate_9,  
CAST ( InvoiceDate AS REAL) AS RealCast  
FROM Invoices;
```

Below the query editor, the 'Results' tab is selected, showing a table with 14 rows and 4 columns: 'VarcharDate', 'VarcharDate\_5', 'VarchDate\_9', and 'RealCast'. The data is as follows:

	VarcharDate	VarcharDate_5	VarchDate_9	RealCast
1	Apr 2 2016 12:00AM	02-04-16	Apr 2 2016 12:00:00:000AM	42460
2	Apr 1 2016 12:00AM	01-04-16	Apr 1 2016 12:00:00:000AM	42459
3	Mar 31 2016 12:00AM	31-03-16	Mar 31 2016 12:00:00:000AM	42458
4	Mar 30 2016 12:00AM	30-03-16	Mar 30 2016 12:00:00:000AM	42457
5	Mar 28 2016 12:00AM	28-03-16	Mar 28 2016 12:00:00:000AM	42455
6	Mar 25 2016 12:00AM	25-03-16	Mar 25 2016 12:00:00:000AM	42452
7	Mar 24 2016 12:00AM	24-03-16	Mar 24 2016 12:00:00:000AM	42451
8	Mar 24 2016 12:00AM	24-03-16	Mar 24 2016 12:00:00:000AM	42451
9	Mar 24 2016 12:00AM	24-03-16	Mar 24 2016 12:00:00:000AM	42451
10	Mar 24 2016 12:00AM	24-03-16	Mar 24 2016 12:00:00:000AM	42451
11	Mar 23 2016 12:00AM	23-03-16	Mar 23 2016 12:00:00:000AM	42450
12	Mar 23 2016 12:00AM	23-03-16	Mar 23 2016 12:00:00:000AM	42450
13	Mar 23 2016 12:00AM	23-03-16	Mar 23 2016 12:00:00:000AM	42450
14	Mar 22 2016 12:00AM	22-03-16	Mar 22 2016 12:00:00:000AM	42449

At the bottom, a status bar indicates 'Query executed successfully.' and 'DESKTOP-PTHKBA0 (15.0 RTM) DESKTOP-PTHKBA0\lyq (52) AP 00:00:00 114 rows'.