**char and varchar (Transact-SQL)**

* [SQL Server 2012](https://msdn.microsoft.com/en-us/library/ms176089(v=sql.110).aspx)
* [SQL Server 2008 R2](https://msdn.microsoft.com/en-us/library/ms176089(v=sql.105).aspx)

**THIS TOPIC APPLIES TO:** yesSQL Server (starting with 2008)yesAzure SQL DatabaseyesAzure SQL Data Warehouse yesParallel Data Warehouse

Are string data types of either fixed length or variable length.

[**Arguments**](javascript:void(0))

**char** [ ( *n* ) ]  
Fixed-length, non-Unicode string data. *n* defines the string length and must be a value from 1 through 8,000. The storage size is *n* bytes. The ISO synonym for **char** is **character**.

**varchar** [ ( *n* | **max** ) ]  
Variable-length, non-Unicode string data. *n* defines the string length and can be a value from 1 through 8,000. **max** indicates that the maximum storage size is 2^31-1 bytes (2 GB). The storage size is the actual length of the data entered + 2 bytes. The ISO synonyms for **varchar** are **charvarying** or **charactervarying**.

[**Remarks**](javascript:void(0))

When *n* is not specified in a data definition or variable declaration statement, the default length is 1. When *n* is not specified when using the CAST and CONVERT functions, the default length is 30.

Objects that use **char** or **varchar** are assigned the default collation of the database, unless a specific collation is assigned using the COLLATE clause. The collation controls the code page that is used to store the character data.

If you have sites that support multiple languages, consider using the Unicode **nchar** or **nvarchar** data types to minimize character conversion issues. If you use **char** or **varchar**, we recommend the following:

* Use **char** when the sizes of the column data entries are consistent.
* Use **varchar** when the sizes of the column data entries vary considerably.
* Use **varchar(max)** when the sizes of the column data entries vary considerably, and the size might exceed 8,000 bytes.

If SET ANSI\_PADDING is OFF when either CREATE TABLE or ALTER TABLE is executed, a **char** column that is defined as NULL is handled as **varchar**.

When the collation code page uses double-byte characters, the storage size is still *n* bytes. Depending on the character string, the storage size of *n* bytes can be less than *n* characters.

[**Converting Character Data**](javascript:void(0))

When character expressions are converted to a character data type of a different size, values that are too long for the new data type are truncated. The **uniqueidentifier** type is considered a character type for the purposes of conversion from a character expression, and therefore is subject to the truncation rules for converting to a character type. See the Examples section that follows.

When a character expression is converted to a character expression of a different data type or size, such as from **char(5)** to **varchar(5)**, or **char(20)** to **char(15)**, the collation of the input value is assigned to the converted value. If a noncharacter expression is converted to a character data type, the default collation of the current database is assigned to the converted value. In either case, you can assign a specific collation by using the [COLLATE](https://msdn.microsoft.com/en-us/library/ms184391.aspx) clause.

|  |
| --- |
| **System_CAPS_ICON_note.jpg Note** |
| Code page translations are supported for **char** and **varchar** data types, but not for **text** data type. As with earlier versions of SQL Server, data loss during code page translations is not reported. |

Character expressions that are being converted to an approximate **numeric** data type can include optional exponential notation (a lowercase e or uppercase E followed by an optional plus (+) or minus (-) sign and then a number).

Character expressions that are being converted to an exact **numeric** data type must consist of digits, a decimal point, and an optional plus (+) or minus (-). Leading blanks are ignored. Comma separators, such as the thousands separator in 123,456.00, are not allowed in the string.

Character expressions being converted to **money** or **smallmoney** data types can also include an optional decimal point and dollar sign ($). Comma separators, as in $123,456.00, are allowed.

[**Examples**](javascript:void(0))

**A. Showing the default value of n when used in variable declaration.**

The following example shows the default value of *n* is 1 for the char and varchar data types when they are used in variable declaration.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_1244a73a-35d8-4ae3-904c-46e68ae053f3');" \o "Copy to clipboard.)

DECLARE @myVariable AS varchar = 'abc';

DECLARE @myNextVariable AS char = 'abc';

--The following returns 1

SELECT DATALENGTH(@myVariable), DATALENGTH(@myNextVariable);

GO

**B. Showing the default value of n when varchar is used with CAST and CONVERT.**

The following example shows that the default value of *n* is 30 when the char or varchar data types are used with the CAST and CONVERT functions.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_a509d766-d5f5-44ea-93be-c4b0c4c2984f');)

DECLARE @myVariable AS varchar(40);

SET @myVariable = 'This string is longer than thirty characters';

SELECT CAST(@myVariable AS varchar);

SELECT DATALENGTH(CAST(@myVariable AS varchar)) AS 'VarcharDefaultLength';

SELECT CONVERT(char, @myVariable);

SELECT DATALENGTH(CONVERT(char, @myVariable)) AS 'VarcharDefaultLength';

**C. Converting Data for Display Purposes**

The following example converts two columns to character types and applies a style that applies a specific format to the displayed data. A **money** type is converted to character data and style 1 is applied, which displays the values with commas every three digits to the left of the decimal point, and two digits to the right of the decimal point. A **datetime** type is converted to character data and style 3 is applied, which displays the data in the format dd/mm/yy. In the WHERE clause, a **money** type is cast to a character type to perform a string comparison operation.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_5b4fc77e-1a23-48ad-854c-f50256219ff4');)

USE AdventureWorks2012;

GO

SELECT BusinessEntityID,

SalesYTD,

CONVERT (varchar(12),SalesYTD,1) AS MoneyDisplayStyle1,

GETDATE() AS CurrentDate,

CONVERT(varchar(12), GETDATE(), 3) AS DateDisplayStyle3

FROM Sales.SalesPerson

WHERE CAST(SalesYTD AS varchar(20) ) LIKE '1%';

Here is the result set.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_c1e3feda-1e28-4767-9da9-306278caaa29');)

BusinessEntityID SalesYTD DisplayFormat CurrentDate DisplayDateFormat

---------------- --------------------- ------------- ----------------------- -----------------

278 1453719.4653 1,453,719.47 2011-05-07 14:29:01.193 07/05/11

280 1352577.1325 1,352,577.13 2011-05-07 14:29:01.193 07/05/11

283 1573012.9383 1,573,012.94 2011-05-07 14:29:01.193 07/05/11

284 1576562.1966 1,576,562.20 2011-05-07 14:29:01.193 07/05/11

285 172524.4512 172,524.45 2011-05-07 14:29:01.193 07/05/11

286 1421810.9242 1,421,810.92 2011-05-07 14:29:01.193 07/05/11

288 1827066.7118 1,827,066.71 2011-05-07 14:29:01.193 07/05/11

**D. Converting Uniqueidentifer Data**

The following example converts a uniqueidentifier value to a char data type.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_d7f1d533-51f4-4c0f-bd08-30c4602c0fb8');)

DECLARE @myid uniqueidentifier = NEWID();

SELECT CONVERT(char(255), @myid) AS 'char';

The following example demonstrates the truncation of data when the value is too long for the data type being converted to. Because the **uniqueidentifier** type is limited to 36 characters, the characters that exceed that length are truncated.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_65ccd628-13df-4ccf-943b-cff629605b9e');)

DECLARE @ID nvarchar(max) = N'0E984725-C51C-4BF4-9960-E1C80E27ABA0wrong';

SELECT @ID, CONVERT(uniqueidentifier, @ID) AS TruncatedValue;

Here is the result set.

[Copy](javascript:if%20(window.epx.codeSnippet)window.epx.codeSnippet.copyCode('CodeSnippetContainerCode_ec9daf3e-62da-4e88-8c35-5ecc1ab184c8');)

String TruncatedValue

-------------------------------------------- ------------------------------------

0E984725-C51C-4BF4-9960-E1C80E27ABA0wrong 0E984725-C51C-4BF4-9960-E1C80E27ABA0

(1 row(s) affected)