

# Database Management Systems

## DATA TYPES IN T-SQL

September 2019

# Contents

## Data types

- Numeric
- String
- Date/time
- Large value

## Data conversion

- Implicit conversion
- Explicit conversion

# Data Types

Dozens of data types divided into 4 categories:

- String: Strings of character data
- Numeric: Integers, floating point numbers, currency, other numeric data
- Temporal (date/time): Dates, times, or both
- Other: Large character and binary values, XML, geometric data, geographic data, hierarchical data

Supports most ANSI-standard data type (synonyms can be used, automatically mapped to corresponding SQL Server data types)

- e.g. rowversion → timestamp

# ANSI-standard Data Types & SQL Server Equivalents

Synonym for ANSI-standard data type	SQL Server data type used
binary varying	varbinary
char varying	varchar
character varying	
character	char
dec	decimal
double precision	float
float	real or float
integer	int
national char	nchar
national character	
national char varying	nvarchar
national character varying	
national text	ntext
rowversion	timestamp

# Numeric Data Types

Exact numeric data types (exact precision)

- Integer: whole numbers
- Decimal: decimal values

precision: total number of digits can be stored

scale: number of digits can be stored to the right of the decimal point

Approximate numeric data types (may not represent a value exactly)

- Real: floating-point numbers
  - limited number of significant digits

# Integer Data Type

Type	Bytes	Description
<b>bigint</b>	8	Large integers from -9,223,372,036,854,775,808 through 9,223,372,036,854,775,807.
<b>int</b>	4	Integers from -2,147,483,648 through 2,147,483,647.
<b>smallint</b>	2	Small integers from -32,768 through 32,767.
<b>tinyint</b>	1	Very small positive integers from 0 through 255.
<b>bit</b>	1	Integers with a value of 1 or 0.

## Decimal data types

Type	Bytes	Description
<b>decimal</b> [(p[,s])]	5-17	Decimal numbers with fixed precision ( <i>p</i> ) and scale ( <i>s</i> ) from $-10^{38}+1$ through $10^{38}-1$ . The precision can be any number between 1 and 38; the default is 18. The scale can be any number between 0 and the precision; the default is 0.
<b>numeric</b> [(p[,s])]	5-17	Synonymous with decimal.
<b>money</b>	8	Monetary values with four decimal places from -922,337,203,685,477.5808 through 922,337,203,685,477.5807. Synonymous with decimal(19,4).
<b>smallmoney</b>	4	Monetary values with four decimal places from -214,748.3648 through 214,748.3647. Synonymous with decimal(10,4).

# Real Data Type

Type	Bytes	Description
<b>float</b> [ (n) ]	4 or 8	Double-precision floating-point numbers from $-1.79 \times 10^{308}$ through $1.79 \times 10^{308}$ . <i>n</i> represents the number of bits used to store the decimal portion of the number (the mantissa): <i>n</i> =24 is single-precision; <i>n</i> =53 is double-precision. The default is 53.
<b>real</b>	4	Single-precision floating point numbers from $-3.4 \times 10^{38}$ through $3.4 \times 10^{38}$ . Synonymous with float(24).



## String Data Type

Store standard characters (single byte storage) or Unicode characters (two bytes storage)

char/nchar for fixed-length strings

- same amount of storage regardless of actual string length

Varchar/nvarchar for variable-length strings

- used only amount of storage needed for a given string

Should use char/varchar unless in multi-language environment

## String Data Type (cont)

Type	Bytes	Description
<b>char</b> ( <i>n</i> )	<i>n</i>	Fixed-length strings of character data. <i>n</i> is the number of characters between 1 and 8000. The default is 1.
<b>varchar</b> ( <i>n</i> )		Variable-length strings of character data. <i>n</i> is the maximum number of characters between 1 and 8000. The default is 1. The number of bytes used to store the string depends on the actual length of the string.

Type	Bytes	Description
<b>nchar</b> ( <i>n</i> )	2× <i>n</i>	Fixed-length strings of Unicode character data. <i>n</i> is the number of characters between 1 and 4000. The default is 1.
<b>nvarchar</b> ( <i>n</i> )		Variable-length strings of Unicode character data. <i>n</i> is the maximum number of characters between 1 and 4000. The default is 1. The number of bytes used to store the string depends on the actual length of the string. Two bytes are needed to store each character.

# Date/Time Data Types

Since SQL Server 2008, 4 date/time types

- Date values without time component (Date)
- Time values without date component (Time), fractional seconds precision can be specified (amount of disk space required varies according to, e.g. 3 bytes for 1 digit, 5 bytes for 7 digits)
- Datetime2 combines date & time
- Datetimeoffset works like datetime2 & number of hours ahead or behind GMT (requires 2 more bytes of storage)

Specifying date/time value by coding literal (value enclosed in single quotes)

- Default time value (for date type, if not specified) is 12:00am
- Default two-digit cutoff is 50, e.g. 00 to 49 interpreted as 2000 to 2049, 50 through 99 interpreted as 1950 through 1999
- Time could be 12-hour (default is am) or 24-hour format.

# Date/Time Types (SQL Server 2008 onward)

Type	Bytes	Description
<b>date</b>	3	Dates only (no time part) from January 1, 0001 through December 31, 9999.
<b>time (n)</b>	3-5	Times only (no date part) from 00:00:00.0000000 through 23:59:59.9999999, with an accuracy of .0000001 seconds. <i>n</i> is the number of digits from 0 to 7 that are used for fractional second precision.
<b>datetime2 (n)</b>	6-8	Dates from January 1, 0001 through December 31, 9999 with time values from 00:00:00.0000000 through 23:59:59.9999999.
<b>datetimeoffset (n)</b>	8-10	An extension of the datetime2 type that also includes a time zone offset from -14 to +14.

## Common date formats

Format	Example
<b>yyyy-mm-dd</b>	2016-04-30
<b>mm/dd/yyyy</b>	4/30/2016
<b>mm-dd-yy</b>	4-30-16
<b>Month dd, yyyy</b>	April 30, 2016
<b>Mon dd, yy</b>	Apr 30, 16
<b>dd Mon yy</b>	30 Apr 16

## Common time formats

Format	Example
<b>hh:mi</b>	16:20
<b>hh:mi am/pm</b>	4:20 pm
<b>hh:mi:ss</b>	4:20:36
<b>hh:mi:ss:mmm</b>	4:20:36:12
<b>hh:mi:ss.nnnnnnn</b>	4:20:36.1234567

# Large Value Data Types

Types: varchar (max), nvarchar (max), varbinary (max)

- text, ntext, image (prior to SQL Server 2005) deprecated
- Used to store images & other type of large character or binary data
- Use 'max' specifier to increase storage capacity of the column up to 2 gigabytes
- Work like their smaller counterparts

## Large Value Data Types (cont.)

Type	Description
<b>varchar(max)</b>	Works the same as the varchar type described in figure 8-3, but the max specifier allows this data type to store up to 2,147,483,648 bytes of data.
<b>nvarchar(max)</b>	Works the same as the nvarchar type described in figure 8-3, but the max specifier allows this data type to store up to 2,147,483,648 bytes of data.
<b>varbinary(max)</b>	Stores variable-length binary data up to a maximum of 2,147,483,648 bytes. The number of bytes used to store the data depends on the actual length of the data.

SQL Server 2005 and later	Prior to 2005
<b>varchar(max)</b>	<b>text</b>
<b>nvarchar(max)</b>	<b>ntext</b>
<b>varbinary(max)</b>	<b>image</b>

# Data Conversion

Implicit conversion

- Lower precedence to higher precedence

Explicit conversion

- CAST
- CONVERT
- TRY\_CONVERT
- Others

# Order of Precedence for Common Data Types

Precedence	Category	Data type
Highest	Date/time	datetime
		smalldatetime
	Numeric	float
		real
		decimal
		money
		smallmoney
		int
		smallint
		tinyint
		bit
	String	nvarchar
		nchar
		varchar
		char
Lowest		



# Implicit Conversion

Example:

```
InvoiceTotal * .0775      -- InvoiceTotal (money) converted to decimal
PaymentTotal - 100        -- Numeric literal converted to money
PaymentDate = '2016-04-05' -- Date literal converted to smalldatetime value
```

Conversions can be done implicitly:

From data type	To data type
char, varchar, nchar, nvarchar	money, smallmoney
datetime, smalldatetime	decimal, numeric, float, real, bigint, int, smallint, tinyint, money, smallmoney, bit
money, smallmoney	char, varchar, nchar, nvarchar

# CAST

CAST(expression AS data\_type)

```
SELECT InvoiceDate, InvoiceTotal,  
       CAST(InvoiceDate AS varchar) AS varcharDate,  
       CAST(InvoiceTotal AS int) AS integerTotal,  
       CAST(InvoiceTotal AS varchar) AS varcharTotal  
FROM Invoices;
```

	InvoiceDate	InvoiceTotal	varcharDate	integerTotal	varcharTotal	
1	2015-12-08 00:00:00	3813.33	Dec 8 2015 12:00AM	3813	3813.33	^
2	2015-12-10 00:00:00	40.20	Dec 10 2015 12:00AM	40	40.20	
3	2015-12-13 00:00:00	138.75	Dec 13 2015 12:00AM	139	138.75	
4	2015-12-16 00:00:00	144.70	Dec 16 2015 12:00AM	145	144.70	▼

# CONVERT

CONVERT(data\_type, expression, [,style])

```
SELECT CONVERT(varchar, InvoiceDate) AS varcharDate,  
       CONVERT(varchar, InvoiceDate, 1) AS varcharDate_1,  
       CONVERT(varchar, InvoiceDate, 107) AS varcharDate_107,  
       CONVERT(varchar, InvoiceTotal) AS varcharTotal,  
       CONVERT(varchar, InvoiceTotal, 1) AS varcharTotal_1  
FROM Invoices;
```

	varcharDate	varcharDate_1	varcharDate_107	varcharTotal	varcharTotal_1	
1	Dec 8 2015 12:00AM	12/08/15	Dec 08, 2015	3813.33	3,813.33	^
2	Dec 10 2015 12:00AM	12/10/15	Dec 10, 2015	40.20	40.20	
3	Dec 13 2015 12:00AM	12/13/15	Dec 13, 2015	138.75	138.75	
4	Dec 16 2015 12:00AM	12/16/15	Dec 16, 2015	144.70	144.70	v

## CONVERT (cont.)

Style codes for converting real data to character data

Code	Output
0 (default)	6 digits maximum
1	8 digits; must use scientific notation
2	16 digits; must use scientific notation

Style codes for converting money data to character data

Code	Output
0 (default)	2 digits to the right of the decimal point; no commas to the left
1	2 digits to the right of the decimal point; commas to the left
2	4 digits to the right of the decimal point; no commas to the left

# TRY\_CONVERT

TRY\_CONVERT (data\_type, expression, [,style])

- Return NULL on error

```
SELECT TRY_CONVERT(varchar, InvoiceDate) AS varcharDate,  
       TRY_CONVERT(varchar, InvoiceDate, 1) AS varcharDate_1,  
       TRY_CONVERT(varchar, InvoiceDate, 107) AS varcharDate_107,  
       TRY_CONVERT(varchar, InvoiceTotal) AS varcharTotal,  
       TRY_CONVERT(varchar, InvoiceTotal, 1) AS varcharTotal_1,  
       TRY_CONVERT(date, 'Feb 29 2015') AS invalidDate  
FROM Invoices;
```

	varcharDate	varcharDate_1	varcharDate_107	varcharTotal	varcharTotal_1	invalidDate	
1	Dec 8 2015 12:00AM	12/08/15	Dec 08, 2015	3813.33	3,813.33	NULL	^
2	Dec 10 2015 12:00AM	12/10/15	Dec 10, 2015	40.20	40.20	NULL	
3	Dec 13 2015 12:00AM	12/13/15	Dec 13, 2015	138.75	138.75	NULL	
4	Dec 16 2015 12:00AM	12/16/15	Dec 16, 2015	144.70	144.70	NULL	
5	Dec 16 2015 12:00AM	12/16/15	Dec 16, 2015	15.50	15.50	NULL	
6	Dec 16 2015 12:00AM	12/16/15	Dec 16, 2015	42.75	42.75	NULL	
7	Dec 21 2015 12:00AM	12/21/15	Dec 21, 2015	172.50	172.50	NULL	
8	Dec 24 2015 12:00AM	12/24/15	Dec 24, 2015	95.00	95.00	NULL	v



## Other Common Data Conversion Functions

Function	Description
<code>STR(float[,length[,decimal]])</code>	Converts a floating-point number to a character string with the given length and number of digits to the right of the decimal point. The length must include one character for the decimal point and one character for the sign. The sign is blank if the number is positive.
<code>CHAR(integer)</code>	Converts the ASCII code represented by an integer between 0 and 255 to its character equivalent.
<code>ASCII(string)</code>	Converts the leftmost character in a string to its equivalent ASCII code.
<code>NCHAR(integer)</code>	Converts the Unicode code represented by an integer between 0 and 65535 to its character equivalent.
<code>UNICODE(string)</code>	Converts the leftmost character in a UNICODE string to its equivalent UNICODE code.