

Categories of SQL Server data types

SQL Server supports the following data type's categories:

- **Exact numeric:** bit, tinyint, smallint, int, bigint, decimal, numeric, money and smallmoney
- **Approximate numeric:** Real and float
- **Date and time:** date, DateTime, datetime2, datetimeoffset, smalldatetime, time
- **Character strings:** char, varchar, text
- **Unicode character strings:** Nchar, Nvarchar, Ntext
- **Binary strings:** Binary, image and varbinary
- **Other data types:** Cursor, hierarchyid, sql_variant, table, rowversion, uniqueidentifier, XML, Spatial and geography

Exact numeric SQL Server data type

We use exact numeric data types for integer, decimal, and money. Each data type has its own lower, upper limit and memory requirements. We should use the smallest data type to save memory requirements as well. For example, we can use the bit data type for storing true (1) or false (0) values.

Data Type	Lower Range	Upper Range	Storage	Remarks
Bit	0	1	1 byte	We can also store NULL values in this.
tinyint	0	255	1 byte	We can store whole numbers up to 255 in this data type.
Smallint	-2^{15} (-32,768)	$2^{15}-1$ (32,767)	2 bytes	We can store whole numbers between a lower and higher range.
Int	-2^{31} (-2,147,483,648)	$2^{31}-1$ (2,147,483,647)	4 bytes	It also stores the whole number similar to a smallint but its lower and upper limits changes as defined.
Bigint	-2^{63} (-9,223,372,036,854,775,808)	$2^{63}-1$ (9,223,372,036,854,775,807)	8 bytes	We should use bigint data type if we cannot accommodate data in the integer data type.
			It depends upon precision.	
			1 – 9 -> 5 bytes	
			10-19->9 bytes	
			20-28->13 bytes	
Decimal	$-10^{38}+1$	$10^{38}-1$		We use decimal data type for scale and fixed precision numbers.

			29-28->17 bytes	
			It depends upon precision.	
			1 – 9 -> 5 bytes	
			10-19->9 bytes	
			20-28->13 bytes	
Numeric	$-10^{38}+1$	$10^{38}-1$	29-28->17 bytes	Decimal and numeric are synonyms. We can use them interchangeably.
Smallmoney	-214,478.3648	+214,478.3647	4 bytes	We can use this data type for monetary or currency values.
Money	-922,337, 203, 685,477.5808	+922,337, 203, 685,477.5807	8 bytes	

Approximate numeric SQL Server data type

Data Type	Lower Range	Upper Range	Storage	Remarks
Real	$-3.40E+38$	$3.40E+38$	4 bytes	We can use float924) as ISO synonym for real .
			Its storage depends upon value (n)	
			N(1-24) ->4 bytes	It is an Approximate-number data types.
Float(n)	$-1.79E+308$	$1.79E+308$	N(25-53)->8 bytes	The default value of N is 53.

Date and Time SQL Server Data types

We stored date and time data along with time offset using these data types.

Data Type	Lower Range	Upper Range	Storage	Remarks
				1. It stores only dates in SQL Server.
				2. Its default value is 1900-01-01.
				3. It provides default format
Date	0001-01-01	9999-12-31	3 bytes	YYYY-MM-DD.

				<p>1. We can define a date along with time with fractional seconds.</p> <p>2. The default value for this data type is 1900-01-01 00:00:00.</p> <p>3.It provides accuracy in increments of .000, .003, or .007 seconds.</p> <p>4. We should avoid using this data type. We can use Datetime2 instead.</p>
Datetime	1753-01-01	9999-12-31	8 bytes	
			6-8 bytes	1. the default format for this is YYYY-MM-DD hh:mm:ss[.fractional seconds].
			1. Precision<3 -> 6 bytes	2. It provides precision from 0 to 7 digits, with an accuracy of 100ns.
Datetime2	0001-01-01 00:00:00	9999-12-31 23:59:59.9999999	2.Precision 3 or 4-> 7 bytes	2. The default precision for datetime2 is 7 digits.
				1. It is similar to a datetime2 data type but includes time zone offset as well.
Datetimeoffset	0001-01-01 00:00:00	9999-12-31 23:59:59.9999999	10 bytes	2. Timezone offset is -14:00 through +14:00.
				1. It defines a date with the time of the day.
				2. Its default value is 1900-01-01 00:00:00.
smalldatetime	1900-01-01 00:00:00	2079-06-06 23:59:59	4 bytes	3. It provides an accuracy of one minute.
				1. We can use it for storing only time data.
				2. Its default format is hh:mm:ss[.nnnnnnn].
Time	00:00:00.0000000	23:59:59.9999999	5 bytes	3. It provides an accuracy of 100 nanoseconds.

Character Strings SQL Server Data types

Data Type	Lower Range	Upper Range	Storage	Remarks
Char(n)	0 characters	8000 characters	N bytes	1. It provides a fixed-width character data type.
				1.It is a variable length character data type.
Varchar(n)	0 characters	8000 characters	n bytes + 2 bytes	2.N defines the string size.
Varchar(max)	0 characters	2^31 chars	n bytes + 2 bytes ~ 2 GB	We should avoid using this data type unless required due to its huge storage requirement.

				1. It is a variable-length character data type.
Text	0 chars	2,147,483,647 chars	n bytes + 4 bytes	2. We should avoid using this data type as it might get deprecated in future versions of SQL Server.

Unicode character string SQL Server data types

Data Type	Lower Range	Upper Range	Storage	Remarks
Nchar	0 characters	4000 characters	2 times n bytes	It is a Unicode string of fixed width.
Nvarchar	0 chars	4000 Chars	2 times n bytes	Nvarchar is a Unicode string of variable width.

				1. It is a Variable-length Unicode data
Ntext	0 chars	1,073,741,823 char	2 times the string length	2. We should avoid using this data type as it will be deprecated in future SQL releases.

Binary SQL Server data types

Data Type	Lower Range	Upper Range	Storage	Remarks
Binary	0 bytes	8000 bytes	N bytes	This data type is a fixed-width binary string.
varbinary	0 bytes	8000 bytes	Its storage is the actual length of string + 2 bytes.	
Image	0 bytes	2,147,483,647 bytes		Avoid using this data type, as it will be deprecated in future SQL Server releases.

Other data types

There are few data types as well that can be used as per the requirement:

- **Cursor**: It is useful for variables or stored procedure OUTPUT parameter referencing to a cursor
- **Rowversion**: It returns automatically generated, unique binary numbers within a database
- **Hierarchyid**: **it is a system data type with variable length. We use it to represent** a position in a hierarchy
- **Uniqueidentifier**: It provides 16 bytes GUID
- **XML**: It is a special data type for storing the XML data in SQL Server tables
- **Spatial Geometry type**: We can use this for representing data in a flat (Euclidean) coordinate system
- **Spatial Geography Types**: We can use Spatial Geography type for storing ellipsoidal (round-earth) data, such as GPS latitude and longitude coordinates. It represents data in a round-earth coordinate system
- **Table**: It is a special data type useful for storing result set temporarily in a table-valued function. We can use data from this for processing later. It can be used in functions, stored procedures, and batches

