# SQL Server Data Types and Ranges (SQL Server 2005 and SQL Server 2000)

*Applies to: SQL Server 2000, SQL Server 2005*

|  |  |
| --- | --- |
|  |  |

## Exact numerics

|  |  |  |
| --- | --- | --- |
| **Type** | **From** | **To** |
| bigint | -9,223,372,036,854,775,808 | 9,223,372,036,854,775,807 |
| int | -2,147,483,648 | 2,147,483,647 |
| smallint | -32,768 | 32,767 |
| tinyint | 0 | 255 |
| bit | 0 | 1 |
| decimal | -10^38 +1 | 10^38 –1 |
| numeric | -10^38 +1 | 10^38 –1 |
| money | -922,337,203,685,477.5808 | +922,337,203,685,477.5807 |
| smallmoney | -214,748.3648 | +214,748.3647 |

numeric and decimal are Fixed precision and scale data types and are functionally equivalent.

## Approximate numerics

|  |  |  |
| --- | --- | --- |
| **Type** | **From** | **To** |
| float | -1.79E + 308 | 1.79E + 308 |
| real | -3.40E + 38 | 3.40E + 38 |

## datetime and smalldatetime

|  |  |  |
| --- | --- | --- |
| **Type** | **From** | **To** |
| datetime (3.33 milliseconds accuracy) | Jan 1, 1753 | Dec 31, 9999 |
| smalldatetime (1 minute accuracy) | Jan 1, 1900 | Jun 6, 2079 |

## Character Strings

|  |  |
| --- | --- |
| **Type** | **Description** |
| char | Fixed-length non-Unicode character data with a maximum length of 8,000 characters. |
| varchar | Variable-length non-Unicode data with a maximum of 8,000 characters. |
| varchar(max) | Variable-length non-Unicode data with a maximum length of 231 characters (**SQL Server 2005 only**). |
| text | Variable-length non-Unicode data with a maximum length of 2,147,483,647 characters. |

## Unicode Character Strings

|  |  |
| --- | --- |
| **Type** | **Description** |
| nchar | Fixed-length Unicode data with a maximum length of 4,000 characters. |
| nvarchar | Variable-length Unicode data with a maximum length of 4,000 characters. |
| nvarchar(max) | Variable-length Unicode data with a maximum length of 230 characters (**SQL Server 2005 only**). |
| ntext | Variable-length Unicode data with a maximum length of 1,073,741,823 characters. |

## Binary Strings

|  |  |
| --- | --- |
| **Type** | **Description** |
| binary | Fixed-length binary data with a maximum length of 8,000 bytes. |
| varbinary | Variable-length binary data with a maximum length of 8,000 bytes. |
| varbinary(max) | Variable-length binary data with a maximum length of 231 bytes (**SQL Server 2005 only**). |
| image | Variable-length binary data with a maximum length of 2,147,483,647 bytes. |

## **Other Data Types**

* **sql\_variant**: Stores values of various SQL Server-supported data types, except text, ntext, and timestamp.
* **timestamp**: Stores a database-wide unique number that gets updated every time a row gets updated.
* **uniqueidentifier**: Stores a globally unique identifier (GUID).
* **xml**: Stores XML data. You can store xml instances in a column or a variable (**SQL Server 2005 only**).
* **cursor**: A reference to a cursor.
* **table**: Stores a result set for later processing. Table variables are just plain cool. If you're using temporary tables and don't need transactions on those tables and want better performance I'd strongly consider using table variables instead. Table variables were introduced in SQL Server 2000. You can use a table variable for just about anything you'd use a regular table.

The following table lists Microsoft SQL Server data types, their equivalents in the common language runtime (CLR) for SQL Server in the **System.Data.SqlTypes** namespace, and their native CLR equivalents in the Microsoft .NET Framework.

|  |  |  |
| --- | --- | --- |
| **SQL Server data type** | **CLR data type (SQL Server)** | **CLR data type (.NET Framework)** |
| **bigint** | **SqlInt64** | **Int64, Nullable<Int64>** |
| **binary** | **SqlBytes, SqlBinary** | **Byte[]** |
| **bit** | **SqlBoolean** | **Boolean, Nullable<Boolean>** |
| **char** | None | None |
| **cursor** | None | None |
| **date** | **SqlDateTime** | **DateTime, Nullable<DateTime>** |
| **datetime** | **SqlDateTime** | **DateTime, Nullable<DateTime>** |
| **datetime2** | **SqlDateTime** | **DateTime, Nullable<DateTime>** |
| **DATETIMEOFFSET** | **None** | **DateTimeOffset, Nullable<DateTimeOffset>** |
| **decimal** | **SqlDecimal** | **Decimal, Nullable<Decimal>** |
| **float** | **SqlDouble** | **Double, Nullable<Double>** |
| **geography** | **SqlGeography**  **SqlGeography** is defined in Microsoft.SqlServer.Types.dll, which is installed with SQL Server and can be downloaded from the SQL Server 2008 [feature pack](http://go.microsoft.com/fwlink/?LinkId=131220). | None |
| **geometry** | **SqlGeometry**  **SqlGeometry** is defined in Microsoft.SqlServer.Types.dll, which is installed with SQL Server and can be downloaded from the SQL Server 2008 [feature pack](http://go.microsoft.com/fwlink/?LinkId=131220). | None |
| **hierarchyid** | **SqlHierarchyId**  **SqlHierarchyId** is defined in Microsoft.SqlServer.Types.dll, which is installed with SQL Server and can be downloaded from the SQL Server 2008 [feature pack](http://go.microsoft.com/fwlink/?LinkId=131220). | None |
| **image** | None | None |
| **int** | **SqlInt32** | **Int32, Nullable<Int32>** |
| **money** | **SqlMoney** | **Decimal, Nullable<Decimal>** |
| **nchar** | **SqlChars, SqlString** | **String, Char[]** |
| **ntext** | None | None |
| **numeric** | **SqlDecimal** | **Decimal, Nullable<Decimal>** |
| **nvarchar** | **SqlChars, SqlString**  **SQLChars** is a better match for data transfer and access, and **SQLString** is a better match for performing String operations. | **String, Char[]** |
| **nvarchar(1), nchar(1)** | **SqlChars, SqlString** | **Char, String, Char[], Nullable<char>** |
| **real** | **SqlSingle** | **Single, Nullable<Single>** |
| **rowversion** | None | **Byte[]** |
| **smallint** | **SqlInt16** | **Int16, Nullable<Int16>** |
| **smallmoney** | **SqlMoney** | **Decimal, Nullable<Decimal>** |
| **sql\_variant** | None | **Object** |
| **table** | None | None |
| **text** | None | None |
| **time** | **TimeSpan** | **TimeSpan, Nullable<TimeSpan>** |
| **timestamp** | None | None |
| **tinyint** | **SqlByte** | **Byte, Nullable<Byte>** |
| **uniqueidentifier** | **SqlGuid** | **Guid, Nullable<Guid>** |
| **User-defined type(UDT)** | None | The same class that is bound to the user-defined type in the same assembly or a dependent assembly. |
| **varbinary** | **SqlBytes, SqlBinary** | **Byte[]** |
| **varbinary(1), binary(1)** | **SqlBytes, SqlBinary** | **byte, Byte[], Nullable<byte>** |
| **varchar** | None | None |
| **xml** | **SqlXml** | None |

Nullable types are instances of the System..::.Nullable<(Of <(T>)>) struct. A nullable type can represent the correct range of values for its underlying value type, plus an additional null value. For example, a Nullable<Int32>, pronounced "Nullable of Int32," can be assigned any value from -2147483648 to 2147483647, or it can be assigned the null value. A Nullable<bool> can be assigned the values truefalse, or null. The ability to assign null to numeric and Boolean types is especially useful when you are dealing with databases and other data types that contain elements that may not be assigned a value. For example, a Boolean field in a database can store the values true or false, or it may be undefined.

The new data types in SQLServer 2008: http://devlicious.com/blogs/sergio\_pereira/archive/2008/04/06/the-new-data-types-in-sql-server-2008.aspx

* The Spatial Data Types
* The DATE type
* The TIME type
* The DATETIME2 type
* The DATETIMEOFFSET type
* The HIERARCHYID type