

## APPENDIX A



## SQL Data Types

Name	Type	What Can It Store?	Description
<b>Numeric Data Types</b>			
TINYINT	Numeric	Numbers from 0 to 255	The smallest data type, at 1 byte. Good for small lookup tables
SMALLINT	Numeric	–32,768 to 32,767	2 bytes in size.
INT	Numeric	–2,147,483,648 to 2,147,483,647	4 bytes. The most commonly used integer field.
BIGINT	Numeric	Huge numbers you'll probably never hit!	8 bytes. Only use if you really need BIG numbers!
SMALLMONEY	Numeric	Same range as INT	4 bytes. Stores money values.
MONEY	Numeric	Same range as BIGINT	8 bytes. Stores BIG money values.
DECIMAL	Numeric	Just about any exact decimal number you can think of	Between 5 and 17 bytes, depending upon the number of decimal places.
NUMERIC	Numeric	As DECIMAL	As DECIMAL.
FLOAT	Numeric	–1.79E+308 to 1.79E+308 (very large)	Between 4 and 8 bytes. Approximate floating point numbers. Equivalent to REAL(24).
REAL	Numeric	–340E+38 to 3.40E+38 (very large)	The same as FLOAT, except it can store larger numbers.
<b>Date/Time Data Types</b>			
DATE	Date/Time	0001-01-01 to 9999-12-31	Stores a date in the specified range. 3 bytes.
TIME	Date/Time	00:00:00 to 23:59:59.999	Stores a particular time of day. 5 bytes in size.
DATETIME	Date/Time	01/01/1753 to 31/12/9999	8 bytes. Storage of dates and times.
DATETIME2	Date/Time	01/01/0001 to 31/12/9999	6 to 8 bytes (depending upon precision of time). Stores a date and time with a 24-hour clock time.



SMALLDATETIME	Date/Time	01/01/1900 to 06/06/2079	Use for recent dates/times or dates/time not too far into the future. 4 bytes.
DATETIMEOFFSET	Date/Time	0001-01-01 to 9999-12-31 with fractional second precision	10 bytes. A zone offset from UTC (Coordinated Universal Time) for a time or datetime value. Stores detailed, fractional-second information, accurate to 100 nanoseconds.
<b>String Data Types</b>			
CHAR	String	1 to 8,000 characters	Fixed string information. If you declare CHAR(10), 10 characters will be stored—the string will be padded with spaces if necessary. This cannot be used for multilingual purposes (it is non-Unicode). Requires 1 byte per character.
VARCHAR	String	1 to 8,000 characters, or up to 2GB using MAX	Variable string information. If you declare VARCHAR(10) and declare a string of 5 characters in length, only 5 characters are stored. This cannot be used for multilingual purposes (it is non-Unicode). Also requires 1 byte per character.
NCHAR	String	1 to 4,000 characters	Fixed string information, stored in Unicode format. Works in the same way as CHAR but can store multilingual values. Uses 2 bytes per character.
NVARCHAR	String	1 to 4,000 characters, or up to 2GB using MAX	Variable string information, stored in Unicode format. Works in the same way as VARCHAR but can store multilingual values. Like NCHAR, requires 2 bytes per character.
<b>Binary Data Types</b>			
TEXT	String	Variable-length string up to 2GB in size	This is a non-Unicode data type. It is deprecated and should not be used. If you have a database using it, plan to change it.
NTEXT	String	Variable-length string up to 1GB in size	This is the Unicode version of TEXT. It is deprecated and should not be used. If you have a database using it, plan to change it.
BINARY	Binary Data	1 to 8,000 bytes	Can store any type of binary data. Any data can be stored in this column type and can be read back out and converted.



VARBINARY	Binary Data	Variable-length binary data of 1 to 8,000, or 2 <sup>31</sup> -1 bytes when MAX is used	Can store any type of binary data. Any data can be stored in this column type and can be read back out and converted.
IMAGE	Binary Data	Variable-length binary data up to 2GB in size	This data type is deprecated and should not be used; use VARBINARY(MAX) instead.
<b>Other Data Types</b>			
UNIQUEIDENTIFIER	Other	16 byte GUID	A unique string value, guaranteed to be unique.
TIMESTAMP	Other	An incrementing number, identifying a row version	A synonym for the ROWVERSION data type, 8 bytes in length. Used to determine if the row has changed since it was last read.
CURSOR	Other	A set of rows	There are different types of cursor. All allow you to process data on a row-by-row basis. Cursors are slow and should be avoided where possible.
HIERARCHYID	Other	Positions in a hierarchy	Supports parent/child hierarchy relationships, and assists in making these structures more performant.
SQL_VARIANT	Other	Various types of data	Can contain different data types, such as VARCHAR(MAX), XML, and GEOGRAPHY.
XML	Other	XML data up to 2GB in size	Can store full XML documents or simple XML fragments, which can be individually queried. XML can be forced to meet certain requirements via a schema.
TABLE	Other	Any valid SQL Server table definition, with rows	Use when a temporary result needs to be stored for a certain period, and a temporary table is not suitable. Used in Table-Valued Functions.
GEOGRAPHY	Spatial	Geographic data, such as longitude and latitude	Supports the storage and querying of coordinates, using a round-earth coordinate system.
GEOMETRY	Spatial	Location data	Uses a flat (Euclidean) coordinate system. Complements GEOGRAPHY.
COMPUTED	Calculation	Any valid T-SQL calculation, with some limitations	Use when you want to derive a value from a function or other columns, and have that value returned as a column in a table or view.

