**What is Data Type?**

A **Data Type** in SQL server is defined as the type of data that any column or variable can store. It is a type of data that an object holds like integer, character, string, etc. While creating any table or variable, in addition to specifying the name, you also set the Type of Data it will store.

**How to use MS SQL datatype**

* You need to define in advance, the type of data a column or variable can store. Determining data type also restricts the user from entering any unexpected or invalid data.
* You can make efficient use of memory by assigning an appropriate data type to variable or column which will allocate only the required amount of system memory for the respective column's data.
* MS SQL offers a broad category of basic data types in SQL as per user's needs like Date, binary images, etc.

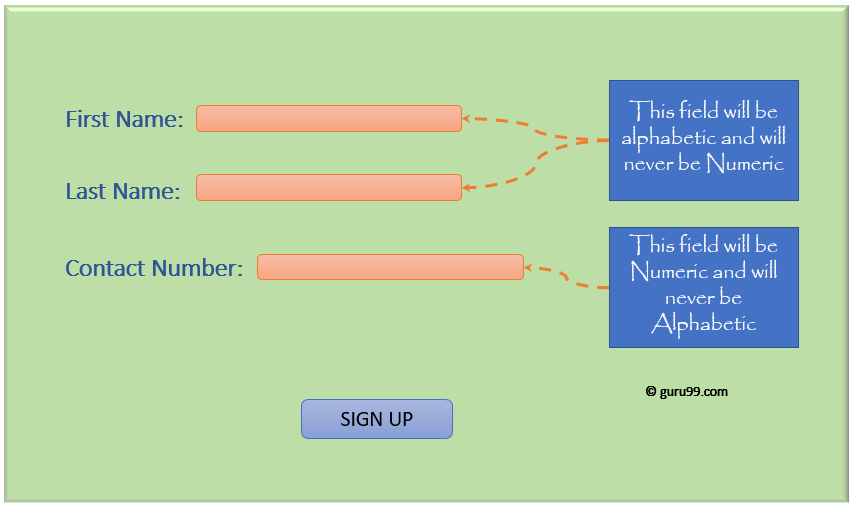
In this tutorial, you will learn MS SQL data types with examples:

## Why use DataTypes?

Let's, take a sample of simple **Sign up page** of website application.Three input fields are First Name, Last Name & Contact number.

Here we should note that in real time:

* **"First/Last Name"** will always be **alphabetic**.
* **"Contact"** will always be **numeric**.



* From the above picture it worth defining **"First/Last Name"** as a **character**and"**Contact"** as an **integer**.

It is evident that in any application, all fields have one or the other type of data. E.g., numeric, alphabetic, date, and many more.

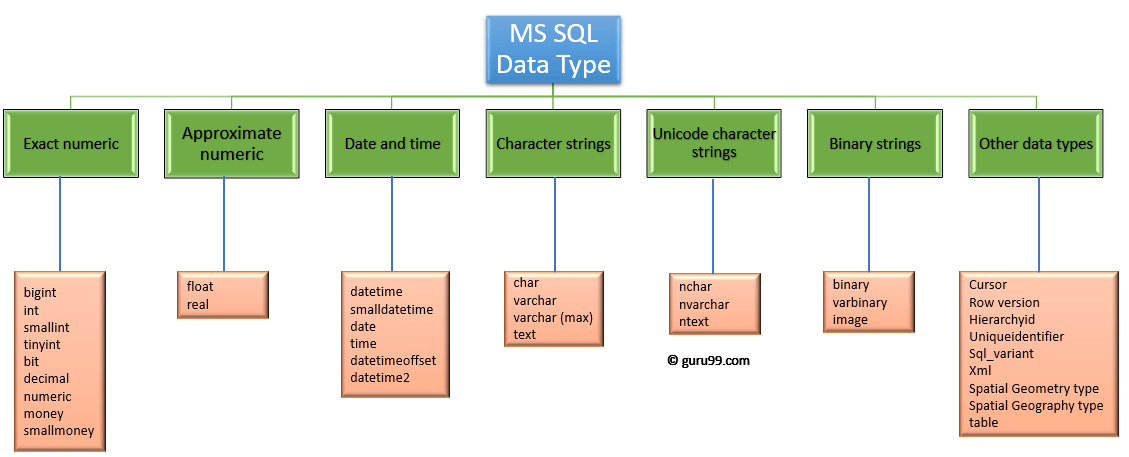
Also, note that different datatype has different memory requirement. Therefore, it makes more sense to define the column or variable with the data type it will hold for efficient use of memory.

## Data type available in MS SQL Server

Here is MS SQL server data types list:

[MS SQL server](https://www.guru99.com/ms-sql-server-tutorial.html) support following categories of Data type:

* Exact numeric
* Approximate numeric
* Date and time
* Character strings
* Unicode character strings
* Binary strings
* Other data types

MS SQL Datatypes

### Exact Numeric Data Types in SQL

Exact numeric has nine types of sub data types in SQL server.

Exact Numeric Data Types

| **Data Type** | **Description** | **Lower limit** | **Upper limit** | **Memory** |
| --- | --- | --- | --- | --- |
| bigint | It stores whole numbers in the range given | −2^63 (−9,223,372, 036,854,775,808) | 2^63−1 (−9,223,372, 036,854,775,807) | **8 bytes** |
| int | It stores whole numbers in the range given | −2^31 (−2,147, 483,648) | 2^31−1 (−2,147, 483,647) | **4 bytes** |
| smallint | It stores whole numbers in the range given | −2^15 (−32,767) | 2^15 (−32,768) | **2 bytes** |
| tinyint | It stores whole numbers in the range given | 0 | 255 | **1 byte** |
| bit | It can take 0, 1, or NULL values. | 0 | 1 | **1 byte/8bit column** |
| decimal | Used for scale and fixed precision numbers | −10^38+1 | 10^381−1 | **5 to 17 bytes** |
| numeric | Used for scale and fixed precision numbers | −10^38+1 | 10^381−1 | **5 to 17 bytes** |
| money | Used monetary data | −922,337, 203, 685,477.5808 | +922,337, 203, 685,477.5807 | **8 bytes** |
| smallmoney | Used monetary data | −214,478.3648 | +214,478.3647 | **4 bytes** |

**Exact Numeric data types in SQL server with Examples:**

**Query:**

DECLARE @Datatype\_Int INT = 2

PRINT @Datatype\_Int

**Output:**2

**Syntax:** Decimal (P,S)

Here,

* P is precision
* S is scale

**Query:**

DECLARE @Datatype\_Decimal DECIMAL (3,2) = 2.31

PRINT @Datatype\_Decimal

**Output:**2.31

### Approximate Numeric Data Types in SQL

SQL Approximate Numeric category includes floating point and real values. These datatypes in SQL are mostly used in scientific calculations.

Approximate Numeric Data Type

| **Data Type** | **Description** | **Lower limit** | **Upper limit** | **Memory** | **Precision** |
| --- | --- | --- | --- | --- | --- |
| **float(n)** | Used for a floating precision number | −1.79E+308 | 1.79E+308 | Depends on the value of n | 7 Digit |
| **real** | Used for a floating precision number | −3.40E+38 | 3.40E+38 | 4 bytes | 15 Digit |

Syntax: FLOAT [(n)]

Here, **n** is the number of bits that are used to store the mantissa of the **float**number in scientific notation. By default, the value of n is 53.

When the user defines a data type like float, **n** should be a value between 1 and 53.

SQL Server treats **n** as one of two possible values. If 1<=n<=24, n is treated as 24. If 25<=n<=53, n is treated as 53.

**Example Query:**

DECLARE @Datatype\_Float FLOAT(24) = 22.1234

PRINT @Datatype\_Float

**Output:**22.1234

### Date and Time Data Types in SQL

It stores data of type Date and time.

Date and Time Data Type

| **Data Type** | **Description** | **Storage size** | **Accuracy** | **Lower Range** | **Upper Range** |
| --- | --- | --- | --- | --- | --- |
| **DateTime** | Used for specifying a date and time from January 1, 1753 to December 31, 9999. It has an accuracy of 3.33 milliseconds. | 8 bytes | Rounded to increments of .000, .003, .007 | 1753-01-01 | 9999-12-31 |
| **smalldatetime** | Used for specifying a date and time from January 1, 0001 to December 31, 9999. It has an accuracy of 100 nanoseconds | 4 bytes, fixed | 1 minute | 1900-01-01 | 2079-06-06 |
| **date** | Used to store only date from January 1, 0001 to December 31, 9999 | 3 bytes, fixed | 1 day | 0001-01-01 | 9999-12-31 |
| **time** | Used for storing only time only values with an accuracy of 100 nanoseconds. | 5 bytes | 100 nanoseconds | 00:00:00.0000000 | 23:59:59.9999999 |
|  |  |  |  |  |  |
| **datetimeoffset** | Similar to datatime but has a time zone offset | 10 bytes | 100 nanoseconds | 0001-01-01 | 9999-12-31 |
| **datetime2** | Used for specifying a date and time from January 1, 0001 to December 31, 9999 | 6 bytes | 100 nanoseconds | 0001-01-01 | 9999-12-31 |

**Example Query :**

DECLARE @Datatype\_Date DATE = '2030-01-01'

PRINT @Datatype\_Date

**Output:**'2030-01-01'

### Character Strings Data Types in SQL

This category is related to a character type. It allows the user to define the data type of character which can be of fixed and variable length. It has four kinds of data types. Below are the character string SQL server data types with examples.

Character Strings Data Types

| **Data Type** | **Description** | **Lower limit** | **Upper limit** | **Memory** |
| --- | --- | --- | --- | --- |
| **char** | It is a character string with a fixed width. It stores a maximum of 8,000 characters. | 0 chars | 8000 chars | n bytes |
| **varchar** | This is a character string with variable width | 0 chars | 8000 chars | n bytes + 2 bytes |
| **varchar (max)** | This is a character string with a variable width. It stores a maximum of 1,073,741,824 characters. | 0 chars | 2^31 chars | n bytes + 2 bytes |
| **text** | This is a character string with a variable width. It stores a maximum 2GB of text data. | 0 chars | 2,147,483,647 chars | n bytes + 4 bytes |

**Example Query :**

DECLARE @Datatype\_Char VARCHAR(30) = 'This is Character Datatype'

PRINT @Datatype\_Char

**Output:**This is Character Datatype

### Unicode Character Strings Data Types in SQL

This category store the full range of Unicode character which uses the UTF-16 character encoding.

Unicode Character String Data Types

| **Data Type** | **Description** | **Lower limit** | **Upper limit** | **Memory** |
| --- | --- | --- | --- | --- |
| **nchar** | It is a Unicode string of fixed width | 0 chars | 4000 chars | 2 times n bytes |
| **nvarchar** | It is a unicode string of variable width | 0 chars | 4000 chars | 2 times n bytes + 2 bytes |
| **ntext** | It is a unicode string of variable width | 0 chars | 1,073,741,823 char | 2 times the string length |

**Example Query:**

DECLARE @Datatype\_nChar VARCHAR(30) = 'This is nCharacter Datatype'

PRINT @Datatype\_nChar

**Output:**This is nCharacter Datatype

### Binary String Data Types in SQL

This category contains a binary string of fixed and variable length.

Binary String Data Types

| **Data Type** | **Description** | **Lower limit** | **Upper limit** | **Memory** |
| --- | --- | --- | --- | --- |
| **binary** | It is a fixed width binary string. It stores a maximum of 8,000 bytes. | 0 bytes | 8000 bytes | n bytes |
| **varbinary** | This is a binary string of variable width. It stores a maximum of 8,000 bytes | 0 bytes | 8000 bytes | The actual length of data entered + 2 bytes |
| **image** | This is a binary string of variable width. It stores a maximum of 2GB. | 0 bytes | 2,147,483,647 bytes |  |

Example Query:

DECLARE @Datatype\_Binary BINARY(2) = 12;

PRINT @Datatype\_Binary

**Output:**0x000C

### Other Datatypes in SQL

These are other different SQL server datatypes with the description below-

| **Data Type** | **Description** |
| --- | --- |
| **Cursor** | Its output is a column of **sp\_cursor\_list** and **sp\_describe\_cursor.** It returns the name of the cursor variable. |
| **Row version** | It version stamps table rows. |
| **Hierarchyid** | This datatype represents a position in the hierarchy |
| **Uniqueidentifier** | Conversion from a character expression. |
| **Sql\_variant** | It stores values of SQL server supported Datatypes. |
| **XML** | It stores XML data in a column. |
| **Spatial Geometry type** | It represents data in a flat coordinate system. |
| **Spatial Geography type** | It represents data in the round-earth coordinate system. |
| **table** | It stores a result set for later processing. |