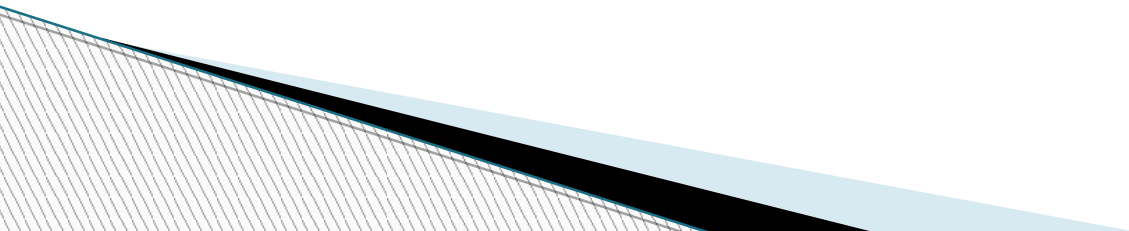


# Tipos de datos



## **int**

De  $-2^{31}$  ( $-2.147.483.648$ ) a  $2^{31}-1$  ( $2.147.483.647$ )

4 bytes

Se representan sin puntos ni comas. Ej: 12345

## **numeric y decimal (p,s)**

p: n° de dígitos, s: de ellos cuántos decimales.

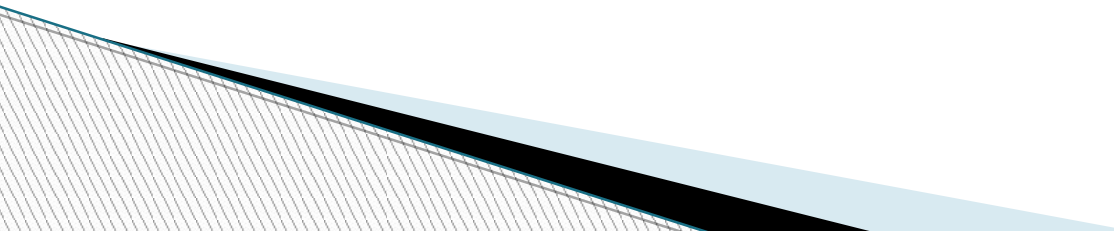
Se representan con un punto decimal: 124.56

## **money**

De  $-922,337,203,685.477,5808$  a  $922,337,203,685.477,5807$

8 bytes

Se representan con punto decimal y comas de separadores de miles.



## **float(n)**

n=24 7 dígitos de precisión; n=53 15 dígitos de precisión.

De - 1,79E+308 a -2,23E-308, 0 y de 2,23E-308 a 1,79E+308

El espacio ocupado depende del valor de n.

Se representan con punto decimal y finalizan con E y un número que representa el exponente de 10 que se aplica.

Ej: 123.458E2 que sería equivalente a 12345.8

## **datetime**

Del 1 de enero de 1753 hasta el 31 de diciembre de 9999

Intervalo de horas de 00:00:00 a 23:59:59.997

El formato habitual se escribe entre comillas 'dd/mm/yyyy hh:mn:ss'

Día (dd), mes (mm), año (yyyy), hora (hh), minutos (mn) y segundos (ss).

Ej: '15/10/2016 14:23:12'

## **char(n)**

### **varchar(n), varchar(max)**

n entre 1 y 8.000

Var es para longitud variable

El formato es un texto cualesquiera entre comillas simples

Ej: 'esto es un texto'

Los tipos de datos de SQL Server se organizan en las siguientes categorías:

- ✓ Números exactos
- ✓ Números aproximados
- ✓ Fecha y hora
- ✓ Cadena de caracteres
- ✓ Cadenas de caracteres Unicode
- ✓ Cadenas binarias
- ✓ Otros tipos de datos

En SQL Server, según las características de almacenamiento, algunos tipos de datos están designados como pertenecientes a los siguientes grupos:

- ✓ Tipos de datos de valores grandes: varchar (Max), nvarchar (Max) y varbinary (Max).
- ✓ Tipos de datos de objetos grandes: text, ntext, image, varchar (Max), nvarchar (Max), varbinary (Max) y XML.

Tipo de dato	Intervalo	Almacenamiento										
Bigint	De -2^63 (-9.223.372.036.854.775.808) a 2^63-1 (9.223.372.036.854.775.807)	8 bytes										
Bit	<p>Tipo de datos entero que puede aceptar los valores 1, 0 o NULL.</p> <p>SQL Server Database Engine (Motor de base de datos de SQL Server) optimiza el almacenamiento de las columnas de tipo <b>bit</b>.</p> <p>Si una tabla contiene 8 columnas o menos de tipo <b>bit</b>, éstas se almacenan como 1 byte. Si hay entre 9 y 16 columnas de tipo <b>bit</b>, se almacenan como 2 bytes, y así sucesivamente.</p> <p>Los valores de cadena TRUE y FALSE se pueden convertir en valores de tipo <b>bit</b>: TRUE se convierte en 1 y FALSE en 0.</p>											
Decimal	<p><b>decimal</b> [(p, [(s)])] Cuando se utiliza la precisión máxima, los valores válidos se sitúan entre - 10^38 + 1 y 10^38 - 1.</p> <p>p (precisión)</p> <p>El número total máximo de dígitos decimales que se puede almacenar, tanto a la izquierda como a la derecha del separador decimal. La precisión debe ser un valor comprendido entre 1 y la precisión máxima de 38. La precisión predeterminada es 18.</p> <p>s (escala)</p> <p>El número máximo de dígitos decimales que se puede almacenar a la derecha del separador decimal. La escala debe ser un valor comprendido entre 0 y p. Sólo es posible especificar la escala si se ha especificado la precisión. La escala predeterminada es 0; por lo tanto, 0 &lt;= s &lt;= p. Los tamaños de almacenamiento máximo varían, según la precisión.</p> <table><tr><th>Precisión</th><th>Bytes de almacenamiento</th></tr><tr><td>1 - 9</td><td>5</td></tr><tr><td>10-19</td><td>9</td></tr><tr><td>20-28</td><td>13</td></tr><tr><td>29-38</td><td>17</td></tr></table>	Precisión	Bytes de almacenamiento	1 - 9	5	10-19	9	20-28	13	29-38	17	
Precisión	Bytes de almacenamiento											
1 - 9	5											
10-19	9											
20-28	13											
29-38	17											
Int	De -2^31 (-2.147.483.648) a 2^31-1 (2.147.483.647)	4 bytes										
Money	De -922,337,203,685.477,5808 a 922,337,203,685.477,5807	8 bytes										
Numeric	Se emplea al igual que el tipo decimal. <b>Numeric</b> [(p, s)]											
Smallint	De -2^15 (-32.768) a 2^15-1 (32.767)	2 bytes										
Smallmoney	De - 214.748,3648 a 214.748,3647	4 bytes										
Tinyint	De 0 a 255	1 byte										

Tabla 3.1 Numéricos exactos.

Tipo de dato	Intervalo	Almacenamiento									
Float	<p>De <math>-1,79\text{E}+308</math> a <math>-2,23\text{E}-308</math>, 0 y de <math>2,23\text{E}-308</math> a <math>1,79\text{E}+308</math></p> <p><b>float ( n )</b></p> <p>Donde <math>n</math> es el número de bits que se utilizan para almacenar la mantisa del número float en notación científica y, por tanto, dicta su precisión y el tamaño de almacenamiento. Si se especifica <math>n</math>, debe ser un valor entre 1 y 53. El valor predeterminado de <math>n</math> es 53.</p> <table border="1"> <thead> <tr> <th>Valor de <math>n</math></th><th>Precisión</th><th>Tamaño de almacenamiento</th></tr> </thead> <tbody> <tr> <td>1-24</td><td>7 dígitos</td><td>4 bytes</td></tr> <tr> <td>25-53</td><td>15 dígitos</td><td>8 bytes</td></tr> </tbody> </table>	Valor de $n$	Precisión	Tamaño de almacenamiento	1-24	7 dígitos	4 bytes	25-53	15 dígitos	8 bytes	Depende de $n$ .
Valor de $n$	Precisión	Tamaño de almacenamiento									
1-24	7 dígitos	4 bytes									
25-53	15 dígitos	8 bytes									
Real	De $-3,40\text{E}+38$ a $-1,18\text{E}-38$ , 0 y de $1,18\text{E}-38$ a $3,40\text{E}+38$	4 Bytes									

Tabla 3.2 Numéricos aproximados.

Tipo de datos	Formato	Intervalo	Precisión	Tamaño de almacenamiento (bytes)	Precisión de fracciones de segundo definida por el usuario	Ajuste de zona horaria
time	hh:mm:ss[.nnnnnnn]	De 00:00:00.0000000 a 23:59:59.9999999	100 nanosegundos	De 3 a 5	Si	No
date	AAAA-MM-DD	De 0001-01-01 a 9999-12-31	1 día	3	No	No
smalldatetime	AAAA-MM-DD hh:mm:ss	De 1900-01-01 a 2079-06-06	1 minuto	4	No	No
datetime	AAAA-MM-DD hh:mm:ss[.nnn]	De 1753-01-01 a 9999-12-31	0,00333 segundos	8	No	No
datetime2	AAAA-MM-DD hh:mm:ss[.nnnnnnn]	De 0001-01-01 00:00:00.0000000 a 9999-12-31 23:59:59.9999999	100 nanosegundos	De 6 a 8	Si	No
datetimeoffset	AAAA-MM-DD hh:mm:ss[.nnnnnnn] [+ -]hh:mm	De 0001-01-01 00:00:00.0000000 a 9999-12-31 23:59:59.9999999 (en UTC)	100 nanosegundos	De 8 a 10	Si	Si

Tipo de dato	Intervalo
Char	<p><b>char [ ( n ) ]</b></p> <p>Datos de caracteres no Unicode de longitud fija, con una longitud de <i>n</i> bytes. <i>n</i> debe ser un valor entre 1 y 8.000. El tamaño de almacenamiento es <i>n</i> bytes.</p>
Text	<p><b>text</b></p> <p>Datos no Unicode de longitud variable de la página de códigos del servidor y con una longitud máxima de <math>2^{31}-1</math> (2.147.483.647) caracteres. Cuando la página de códigos del servidor utiliza caracteres de doble byte, el almacenamiento sigue siendo de 2.147.483.647 bytes. Dependiendo de la cadena de caracteres, el espacio de almacenamiento puede ser inferior a 2.147.483.647 bytes.</p>
Varchar	<p><b>varchar [ ( n   Max ) ]</b></p> <p>Datos de caracteres no Unicode de longitud variable. <i>n</i> puede ser un valor entre 1 y 8.000. <b>Max</b> indica que el tamaño de almacenamiento máximo es de <math>2^{31}-1</math> bytes. El tamaño de almacenamiento es la longitud real de los datos especificados + 2 bytes. Los datos especificados pueden tener una longitud de 0 caracteres.</p>

Tabla 3.4 Cadenas de caracteres.



Tipo de dato	Intervalo
Nchar	<b>nchar [( n )]</b> Datos de carácter Unicode de longitud fija, con <i>n</i> caracteres. <i>n</i> debe estar comprendido entre 1 y 4.000. El tamaño de almacenamiento es dos veces <i>n</i> bytes.
Ntext	<b>ntext</b> Datos Unicode de longitud variable con una longitud máxima de $2^{30} - 1$ (1.073.741.823) caracteres. El tamaño del almacenamiento, en bytes, es dos veces el número de caracteres especificado.
Nvarchar	<b>nvarchar [( n   Max )]</b> Datos de carácter Unicode de longitud variable. <i>n</i> puede ser un valor comprendido entre 1 y 4.000. <b>Max</b> indica que el tamaño máximo de almacenamiento es $2^{31} - 1$ bytes. El tamaño de almacenamiento en bytes es dos veces el número de caracteres especificado + 2 bytes. Los datos especificados pueden tener una longitud de 0 caracteres.

Tabla 3.5 Cadena de caracteres Unicode.

Tipo de dato	Intervalo
Binary	<b>binary [ ( <i>n</i> ) ]</b> Datos binarios de longitud fija con una longitud de <i>n</i> bytes, donde <i>n</i> es un valor que oscila entre 1 y 8.000. El tamaño de almacenamiento es de <i>n</i> bytes.
Image	Datos binarios de longitud variable desde 0 hasta $2^{31}-1$ (2.147.483.647) bytes.
Varbinary	<b>varbinary [ ( <i>n</i>   <i>Max</i> ) ]</b> Datos binarios de longitud variable. <i>n</i> puede ser un valor que oscila entre 1 y 8.000. <i>Max</i> indica que el tamaño máximo de almacenamiento es de $2^{31}-1$ bytes. El tamaño de almacenamiento es la longitud real de los datos especificados + 2 bytes. Los datos especificados pueden tener una longitud de 0 bytes.

Tabla 3.6 Cadenas binarias.

Tipo de dato	Intervalo
<b>Cursor</b>	Un tipo de datos para las variables o para los parámetros de resultado de los procedimientos almacenados que contiene una referencia a un cursor. Las variables creadas con el tipo de datos <b>cursor</b> aceptan NULL.
<b>Hierarchyid</b>	El tipo de datos del sistema de <b>hierarchyid</b> es de longitud variable. Use <b>hierarchyid</b> para representar la posición en una jerarquía. Una columna de tipo <b>hierarchyid</b> no representa automáticamente un árbol. Dependerá de la aplicación generar y asignar los valores <b>hierarchyid</b> de tal forma que la relación deseada entre las filas se refleje en los valores.
<b>Sql variant</b>	Tipo de datos que almacena valores de varios tipos de datos admitidos en SQL Server. Puede tener una longitud máxima de 8.016 bytes. Esto incluye la información y el valor de tipo base. La longitud máxima del tipo base real es 8.000 bytes.  Este no puede almacenar valores del tipo: <b>varchar (Max)</b> , <b>nvarchar (Max)</b> , <b>text</b> , <b>image</b> , <b>Sql variant</b> , <b>hierarchyid</b> , <b>varbinary (Max)</b> , <b>XML</b> , <b>ntext</b> , <b>Timestamp</b> , <b>geography</b> , <b>geometry</b> y tipos de datos definidos por el usuario.
<b>Table</b>	Es un tipo de datos especial que se puede utilizar para almacenar un conjunto de resultados para su procesamiento posterior. <b>Table</b> se utiliza principalmente para el almacenamiento temporal de un conjunto de filas devuelto como el conjunto de resultados de una función con valores de tabla.
<b>Timestamp</b>	<b>timestamp</b> es el sinónimo del tipo de datos <b>rowversion</b> y está sujeto al comportamiento de los sinónimos de tipos de datos. En las instrucciones DDL, utilice <b>rowversion</b> en lugar de <b>timestamp</b> siempre que sea posible.  Es un tipo de datos que expone números binarios únicos generados automáticamente en una base de datos. <b>rowversion</b> suele utilizarse como mecanismo para marcar la versión de las filas de la tabla. El tamaño de almacenamiento es de 8 bytes. El tipo de datos <b>rowversion</b> es simplemente un número que se incrementa y no conserva una fecha o una hora.
<b>Uniqueidentifier</b>	Es un GUID de 16 bytes. Una columna o una variable local de tipo de datos <b>uniqueidentifier</b> se puede inicializar en un valor de las siguiente formas: <ul style="list-style-type: none"> <li>Mediante la función NEWID.</li> <li>Mediante la conversión a partir de una constante de cadena con el formato xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx, donde cada x es un dígito hexadecimal en el intervalo 0-9 o a-f. Por ejemplo, 6F9619FF-8B86-D011-B42D-00C04FC964FF es un valor <b>uniqueidentifier</b> válido.</li> </ul> Con los valores <b>uniqueidentifier</b> se pueden utilizar operadores de comparación. No obstante, no se implementa la ordenación mediante la comparación de los patrones de bits de los dos valores. Las únicas operaciones que se pueden realizar con un valor <b>uniqueidentifier</b> son comparaciones (=, <>, <, >, <=, >=) y comprobaciones para NULL (IS NULL e IS NOT NULL). No es posible utilizar otros operadores aritméticos. Con el tipo de datos <b>uniqueidentifier</b> , se pueden utilizar todas las propiedades y restricciones de columna, excepto IDENTITY. La replicación de mezcla y transaccional con suscripciones de actualización utiliza columnas <b>uniqueidentifier</b> para garantizar que las filas se identifican de forma exclusiva en varias copias de la tabla.
<b>XML</b>	Es el tipo de datos que almacena datos de XML. Puede almacenar instancias de <b>XML</b> en una columna o una variable de tipo <b>XML</b> . Para obtener más información

Tabla 3.7 Otros tipos de datos.

<https://msdn.microsoft.com/es-es/library/ms187752.aspx>

