|  |
| --- |
|  |
| Cursos SQL Server 2008 R2 |
|  |
| *Transact Subconsultas* |

Cursos SQL Server 2008 R2

# Índice

* SQL Server - Transact
* Tipos de datos y variables
* Índice
* Tipos de datos
* Enteros
* Bit
* Decimales
* Aproximados
* Moneda
* Fecha / Hora
* Carácter
* Binarios
* Otros
* Variables de usuario y de Sistema

# Tipos de datos

Los tipos de datos determinan la validez de la información a introducir dentro de una columna de una tabla o de una variable de programación.

Los distintos tipos se agrupan en familias: numéricas, carácter, fecha/hora, etc.

Las versiones del producto hacen que dichos tipos vayan evolucionando y que el número de miembros de cada familia vaya ampliándose.

Vamos a exponer los más habituales.

#### Numéricos enteros

|  |  |  |
| --- | --- | --- |
| Tipo de datos | 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** |
| int | **De -2^31 (-2.147.483.648) a 2^31-1 (2.147.483.647)** | **4 bytes** |
| smallint | **De -2^15 (-32.768) a 2^15-1 (32.767)** | **2 bytes** |
| tinyint | **De 0 a 255** | **1 byte** |

#### Bit

* Tipo de datos entero que puede aceptar los valores 1, 0 o NULL.
* SQL Server optimiza el almacenamiento de las columnas de tipo **bit**. Si una tabla contiene 8 columnas o menos de tipo **bit**, éstas se almacenan como 1 byte. Si hay entre 9 y 16 columnas de tipo **bit**, se almacenan como 2 bytes, y así sucesivamente.
* Los valores de cadena TRUE y FALSE se pueden convertir en valores de tipo **bit**: TRUE se convierte en 1 y FALSE en 0.

**Bit.sql**

#### Numeric y decimal

* Tipos de datos numéricos que tienen precisión y escala fijas.
* **decimal**[ **(***p*[ **,** *s*] **)**] y **numeric**[ **(***p*[ **,** *s*] **)**] Números de precisión y escala fijas. Cuando se utiliza la precisión máxima, los valores válidos se sitúan entre -10^38 +1 y 10^38 - 1. Los sinónimos de ISO para **decimal** son de tipo **dec** y **dec(***p*, *s***)**. **numeric** es funcionalmente equivalente a **decimal**.

*p (precisión)* 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.

*s (escala)* 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 <= *s* <= *p*. Los tamaños de almacenamiento máximo varían, según la precisión.

**Decimal.sql**

#### Numéricos aproximados

Tipos de datos numéricos y aproximados que se utilizan con datos numéricos de coma flotante. Los datos de coma flotante son aproximados; por tanto, no todos los valores del intervalo del tipo de datos se pueden representar con exactitud. No es habitual utilizarlos.

|  |  |  |
| --- | --- | --- |
| Tipo de datos | Intervalo | Almacenamiento |
| float | De - 1,79E+308 a -2,23E-308, 0 y de 2,23E-308 a 1,79E+308 | Depende del valor de *n*. |
| real | De - 3,40E + 38 a -1,18E - 38, 0 y de 1,18E - 38 a 3,40E + 38 | 4 Bytes |

**float** [ **(** *n* **)** ]

Donde *n* 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 *n*, debe ser un valor entre **1** y **53**. El valor predeterminado de *n* es **53**.

|  |  |  |
| --- | --- | --- |
| *n* value | Precisión | Tamaño de almacenamiento |
| 1-24 | 7 dígitos | 4 bytes |
| 25-53 | 15 dígitos | 8 bytes |

SQL Server trata *n* como uno de dos valores posibles.Si **1**<=n<=**24**, *n* se trata como **24**. Si **25**<=n<=**53**, *n* se trata como **53**.

***Float\_Real.sql***

#### Moneda

Los tipos de datos **money** y **smallmoney** tienen una precisión de una diezmilésima de las unidades monetarias que representan.

***Money.sql***

#### 

#### Fecha y hora

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo de datos | Formato | Intervalo | Precisión | (bytes) |
| time | hh:mm:ss[.nnnnnnn] | De 00:00:00.0000000 a 23:59:59.9999999 | 100 nanosegundos | De 3 a 5 |
| date | AAAA-MM-DD | De 0001-01-01 a 9999-12-31 | 1 día | 3 |
| smalldatetime | AAAA-MM-DD hh:mm:ss | De 1900-01-01 a 2079-06-06 | 1 minuto | 4 |
| datetime | AAAA-MM-DD hh:mm:ss[.nnn] | De 1753-01-01 a 9999-12-31 | 0,00333 segundos | 8 |
| 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 |
| 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 |

#### Cadenas de caracteres

Son tipos de datos de caracteres que tienen longitud fija o variable.

* **char** [ ( *n* ) ] Datos de caracteres no Unicode de longitud fija, con una longitud de *n* bytes. *n* debe ser un valor entre 1 y 8.000. El tamaño de almacenamiento es *n* bytes.
* **varchar** [ ( *n* | **max** ) ] Datos de caracteres no Unicode de longitud variable. *n* puede ser un valor entre 1 y 8.000. **max** indica que el tamaño de almacenamiento máximo es de 2^31-1 bytes. En el caso de que no entre en la página, se almacena el puntero (24 bytes). El tamaño de almacenamiento es la longitud real de los datos especificados + 2 bytes.
* **Text** Textos de gran tamaño.  Se quitará en versiones futuras.

Recomendaciones sobre los tipos **char** o **varchar**

* Utilicemos **char** cuando los tamaños de las entradas de datos de columna tengan poca variabilidad.
* Utilicemos **varchar** cuando los tamaños de las entradas de datos de columna varíen de forma considerable.
* Utilicemos **varchar(max)** cuando los tamaños de las entradas de datos de columna varíen de forma considerable y pudieran superarse los 8.000 bytes.

#### Cadenas binarias

Tipos de datos binarios de longitud fija o variable.

Se almacenan fotos, vídeos, documentos .docx, etc.

* **binary** [ ( *n* ) ] Datos binarios de longitud fija con una longitud de *n* bytes, donde *n* es un valor que oscila entre 1 y 8.000. El tamaño de almacenamiento es de *n* bytes.
* **varbinary** [ ( *n* | **max**) ] Datos binarios de longitud variable. *n* puede ser un valor que oscila entre 1 y 8.000. **max** 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.
* **image** Se quitará en versiones futuras.

***Binary.sql***

# Otros tipos de datos

* cursor
* timestamp
* hierarchyid
* uniqueidentifier
* sql\_variant
* xml
* table

# Variables

Existen dos tipos de variables:

* **Variables de usuario**

Se declaran con la sentencia DECLARE y se nombran precedidas con el símbolo @

* **Variables de sistema**

Sus nombres comienzan con @@

En Transact son admitidas las declaraciones implícitas aunque no son recomendables.

Existe un gran nivel de compatibilidad entre los tipos de datos de las columnas y los de las variables.

# Más información

|  |  |
| --- | --- |
| C/ Miracruz, 10 (Bº de Gros) 20001 Donostia  Telf.: 943 275819  email: [seim@centroseim.com](mailto:seim@centroseim.com) |  |

Logotipo SEIM