

18 - Tipo de dato (numérico)

Ya explicamos que al crear una tabla debemos elegir la estructura adecuada, esto es, definir los campos y sus tipos más precisos, según el caso.

Para almacenar valores NUMERICOS SQL Server dispone de varios tipos.

Para almacenar valores ENTEROS, por ejemplo, en campos que hacen referencia a cantidades, usamos:

- 1) integer o int: su rango es de -2000000000 a 2000000000 aprox. El tipo "integer" tiene subtipos:
 - smallint: Puede contener hasta 5 dígitos. Su rango va desde -32000 hasta 32000 aprox.
 - tinyint: Puede almacenar valores entre 0 y 255.
 - bigint: De -9000000000000000000 hasta 9000000000000000000 aprox.

Para almacenar valores numéricos EXACTOS con decimales, especificando la cantidad de cifras a la izquierda y derecha del separador decimal, utilizamos:

- 2) decimal o numeric (t,d): Pueden tener hasta 38 dígitos, guarda un valor exacto. El primer argumento indica el total de dígitos y el segundo, la cantidad de decimales.

Por ejemplo, si queremos almacenar valores entre -99.99 y 99.99 debemos definir el campo como tipo "decimal(4,2)". Si no se indica el valor del segundo argumento, por defecto es "0". Por ejemplo, si definimos "decimal(4)" se pueden guardar valores entre -9999 y 9999.

El rango depende de los argumentos, también los bytes que ocupa.
Se utiliza el punto como separador de decimales.

Si ingresamos un valor con más decimales que los permitidos, redondea al más cercano; por ejemplo, si definimos "decimal(4,2)" e ingresamos el valor "12.686", guardará "12.69", redondeando hacia arriba; si ingresamos el valor "12.682", guardará "12.67", redondeando hacia abajo.

Para almacenar valores numéricos APROXIMADOS con decimales utilizamos:

- 3) float y real: De 1.79E+308 hasta 1.79E+38. Guarda valores aproximados.
- 4) real: Desde 3.40E+308 hasta 3.40E+38. Guarda valores aproximados.

Para almacenar valores MONETARIOS empleamos:

- 5) money: Puede tener hasta 19 dígitos y sólo 4 de ellos puede ir luego del separador decimal; entre -9000000000000000.5808 aprox y 9000000000000000.5807.

- 6) smallmoney: Entre -200000.3648 y 200000.3647 aprox.

Para todos los tipos numéricos:

- si intentamos ingresar un valor fuera de rango, no lo permite.
- si ingresamos una cadena, SQL Server intenta convertirla a valor numérico, si dicha cadena consta solamente de dígitos, la conversión se realiza, luego verifica si está dentro del rango, si es así, la ingresa, sino, muestra un mensaje de error y no ejecuta la sentencia. Si la cadena contiene caracteres que SQL Server no puede convertir a valor numérico, muestra un mensaje de error y la sentencia no se ejecuta.

Por ejemplo, definimos un campo de tipo decimal(5,2), si ingresamos la cadena '12.22', la convierte al

valor numérico 12.22 y la ingresa; si intentamos ingresar la cadena '1234.56', la convierte al valor numérico 1234.56, pero como el máximo valor permitido es 999.99, muestra un mensaje indicando que está fuera de rango. Si intentamos ingresar el valor '12y.25', SQL Server no puede realizar la conversión y muestra un mensaje de error.

Es importante elegir el tipo de dato adecuado según el caso, el más preciso. Por ejemplo, si un campo numérico almacenará valores positivos menores a 255, el tipo "int" no es el más adecuado, conviene el tipo "tinyint", de esta manera usamos el menor espacio de almacenamiento posible.

Si vamos a guardar valores monetarios menores a 200000 conviene emplear "smallmoney" en lugar de "money".

Tipo	Bytes de almacenamiento
int	4
smallint	2
tinyint	1
bigint	8
decimal	2 a 17
float	4 u 8
real	4 u 8
money	8
smallmoney	4

Servidor de SQL Server instalado en forma local.

Ingresemos el siguiente lote de comandos en el SQL Server Management Studio:

```
if object_id('libros') is not null
    drop table libros;

create table libros(
    codigo smallint identity,
    titulo varchar(40) not null,
    autor varchar(30),
    editorial varchar(15),
    precio smallmoney,
    cantidad tinyint
);

go

-- Intentemos ingresar un valor fuera del rango definido, una cantidad
-- que supera el rango del tipo "tinyint", el valor 260 (genera error):
insert into libros (titulo,autor,editorial,precio,cantidad)
    values('El aleph','Borges','Emece',25.60,260);

-- Intentamos ingresar un precio que supera el rango del tipo "smallmoney",
-- el valor 250000 (genera error):
insert into libros (titulo,autor,editorial,precio,cantidad)
    values('El aleph','Borges','Emece',250000,100);

-- Intentamos ingresar una cadena que SQL Server no pueda convertir a valor
```

```
-- numérico en el campo "precio" (genera error):
insert into libros (titulo,autor,editorial,precio,cantidad)
  values('Uno','Richard Bach','Planeta','a50.30',100);

-- Ingresamos una cadena en el campo "cantidad" (lo convierte a valor numérico):
insert into libros (titulo,autor,editorial,precio,cantidad)
  values('Uno','Richard Bach','Planeta',50.30,'100');

select * from libros;
```