Tipos de datos

DigitalHouse >
Coding School

Certified Tech Developer
The Ultimate Degree

Índice

1 | Definición

Tipos de datos

Los datos o atributos de cada registro de una tabla tienen que ser de un tipo de dato concreto.

Cuando diseñamos una base de datos tenemos que pensar qué tipo de datos requerimos para nuestro modelo.

Cada tipo de dato tiene un tamaño determinado y cuanta más precisión apliquemos en su definición, más **rápido** y **performante** va a funcionar MySQL.

Tipo de datos			
Texto	Numérico Entero	Numérico Decimal	Fecha
Juan Pérez	**12345**	**120,50**	**2021-03-15**

2 | Datos de tipo texto

Datos de tipo texto

Almacenan datos **alfanuméricos** y **símbolos**.

CHAR(*n*)

n → 1 a 255 caracteres.
Se recomienda utilizar en cadenas de texto de **longitud poco variable**.
Ej.: La **longitud** de la palabra "*HOLA*" es **4** y se define: **Char(*4*)**.

VARCHAR(*n*)

n → 1 a 21.845 caracteres.
Se recomienda utilizar en cadenas de texto de **longitud muy variable**.
Ej.: La **longitud** de la palabra "*HASTA LUEGO*" es **11** y se define: **Varchar(11)**.

Nota: La letra ***n*** indica la **longitud máxima de caracteres** a utilizar.

Datos de tipo texto

Almacenan datos **alfanuméricos** y **símbolos**.

TINYTEXT
0 a 255 caracteres.

MEDIUMTEXT
0 a 16.777.215 caracteres.

TEXT
0 a 4.294.967.295 caracteres.

LONGTEXT
0 a 18.446.744.073.709.551.615 caracteres.

La longitud máxima permitida depende de la configuración del protocolo cliente-servidor y la memoria disponible.

3 | Datos de tipo numérico

Datos de tipo numérico entero

TINYINT
-128 a 127 (sin signo de 0 a 255)

SMALLINT
-32.768 a 32.767 (sin signo de 0 a 65.535)

MEDIUMINT
-8.388.608 a 8.388.607 (sin signo de 0 a 16.777.215)

INT
-2.147.483.648 a 2.147.483.647 (sin signo de 0 a 4.294.967.295)

BIGINT
-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807

(sin signo de 0 a 18.446.744.073.709.551.615)

Datos de tipo numérico decimal

FLOAT(*n*,*d*)

Almacenan números de coma flotante pequeño.
Tienen precisión simple para la parte decimal (máx. 7 dígitos).
n → 1 a 24 dígitos (incluyendo la parte decimal).
d → 0 a 7 dígitos dependiendo de cuánto se asigne en ***n***.

DOUBLE(*n*,*d*)

Almacenan números de coma flotante grande.
Tienen precisión doble para la parte decimal (máx. 15 dígitos).
n → 25 a 53 dígitos (incluyendo la parte decimal).
d → 0 a 15 dígitos dependiendo de cuánto se asigne en ***n***.

Nota: Las letras ***n*** y ***d*** indican la **longitud máxima de dígitos** a utilizar.

Datos de tipo numérico decimal

FLOAT(*24,7*)

1234567890123456,*1234567*

Parte entera (17 dígitos)

Parte decimal (7 dígitos)

DOUBLE(*53,15*)

12345678901234567890123456789012345678,*123456789012345*

Parte entera (38 dígitos)

Parte decimal (15 dígitos)

4 | Datos de tipo fecha

Datos de tipo fecha

A la hora de almacenar fechas, hay que tener en cuenta que MySQL no comprueba de una manera estricta si una fecha es válida o no.

DATE

Almacena solamente la fecha en formato **YYYY-MM-DD**.
Valores permitidos: '0001-01-01' a '9999-12-31'.
La fecha se debe colocar entre **comillas** simples o dobles y se separa por **guiones**. Ejemplo: '2021-05-15'.

Datos de tipo fecha

TIME

Almacena solamente la hora en formato **HH:MM:SS**.
Valores permitidos: '00:00:00' a '23:59:59'.
La hora se debe colocar entre **comillas** simples o dobles y se separa por **dos puntos**. Ejemplo: '11:50:55'.

DATETIME

Almacena la fecha y hora en formato **YYYY-MM-DD HH:MM:SS**.
Valores permitidos: '0001-01-01 00:00:00' a '23:59:59 9999-12-31'.
La fecha se debe colocar entre **comillas** simples o dobles, un espacio y la hora que se debe separar por **dos puntos**. Ejemplo: '2021-05-15 11:50:55'.

5 | Datos de tipo boolean

Datos de tipo boolean

BOOLEAN

MySQL guarda los booleanos como un **cero** o como un **uno**. Por cuestiones de performance, este tipo de dato viene desactivado y no se recomienda su uso.

En el caso de querer guardar valores "verdaderos" y "falsos", se recomienda utilizar el tipo de dato **tinyint(*1*)**, donde:

- **0** para representar el **false**.
- **1** para representar el **true**.


DigitalHouse>
Coding School