

Módulo: MySQL

TIPOS DE DATOS

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 dato nos sirve para nuestro modelo.

Cada tipo de dato tiene un tamaño determinado y cuanto más precisión pongamos a este tipo de dato, más **rápido** y **performante** va a funcionar MySQL.

012345

Tipos numéricos

2020-05-09

Tipos de fecha

Ad lorem ipsum

Tipos de texto

A large, solid blue diagonal shape that starts from the top right corner and extends towards the bottom left, creating a dynamic background element.

1.

DATOS DE TIPO NUMÉRICO

NUMÉRICOS SIN DECIMALES

TINYINT

-128 a 128, 0 a 255

SMALLINT

-32768 a 32767, 0 a 65535

MEDIUMINT

-8388608 a 8388607, 0 a 16777215

INT

-2147483648 to 214748364, 0 a 4294967295

BIGINT

-9223372036854775808 a 9223372036854775807

0 a 18446744073709551615

NUMÉRICOS **CON DECIMALES**

FLOAT

Permite almacenar **pequeños números decimales**

DOUBLE

Permite almacenar **grandes números decimales**

DECIMAL

Permite almacenar **grandes números decimales de punto fijo**

BOOLEANS

MySQL guarda los booleanos por detrás como un **cero** o como un **uno**. Por cuestiones de performance, no se recomienda utilizar este tipo de dato en MySQL.

En caso de querer guardar valores "verdaderos" y "falsos" podemos usar el tipo de dato **tinyint** y usar el 0 para representar el **false** y el 1 para representar el **true**.

A thick, solid blue diagonal stripe runs from the top right corner towards the bottom left, separating the white background on the left from the solid blue background on the right.

2.

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**

TIME

Almacena solamente la hora en formato **HH:MM:SS**

DATETIME

Corresponde a una representación completa de fecha y hora, es decir algo como el 03 de enero de 1967 a las 8:00 am, lo que se almacena del siguiente modo **1967-01-03 08:00:00**

A thick, bright blue diagonal stripe runs from the top right corner towards the bottom left, separating the white background on the left from a solid blue background on the right.

3.

DATOS DE TIPO
TEXTO

DATOS DE TIPO **TEXTO**

CHAR(num)

El número entre paréntesis va a indicar la **cantidad exacta** de caracteres.

Ejemplo: char(100) → 100 caracteres **siempre**

VARCHAR(num)

El número entre paréntesis va a indicar la **cantidad máxima** de caracteres.

Ejemplo: varchar(100) → 100 caracteres **como máximo**

TEXT

Determina un dato de tipo texto sin límite de caracteres. Se suele usar para post de un blog, por ejemplo.

4.

CONSTRAINTS

Son **restricciones** a nivel tabla. Se especifican vía DDL.
El servidor las analiza a la hora de modificar registros.

CONSTRAINTS

UNIQUE KEYS

Una unique key es una restricción que solo permite valores únicos para uno (o múltiples) campos.

NULL / NOTNULL

NULL Significa que el valor para ese campo no existe o no se conoce. NULL NO es vacío ' ' (blank) ni cero (0).

DEFAULT

Se usa para definir un **valor** por defecto para una columna. Este valor se le va a agregar a cada registro nuevo siempre y cuando no se especifique otro valor que lo sobrescriba.

CONSTRAINTS

AUTO_INCREMENT

Genera un número único y lo incrementa automáticamente con cada nuevo registro en la tabla. Se suele usar a menudo para el campo **ID**.