

# Tipos de datos en MySQL

---

Antes de empezar debes acostumbrarte a los tipos de datos que maneja MySQL (que son similares a los de otros SGBD). Los más habituales son:

## Tipos de datos para texto:

- **VARCHAR(n)**: Pueden contener cualquier carácter. Longitud variable con un tamaño máximo de n bytes, siendo el límite para n de 65536. La ventaja de este tipo es que sólo consume en memoria el número de caracteres que realmente se use.
- **CHAR(n)**: Pueden contener cualquier carácter.

## Tipos de datos para fechas y horas:

- **DATE**: una fecha con formato 2020-12-31
- **TIME**: una hora con formato 21:50:05
- **DATETIME**: combinación de ambas con un espacio en medio: 2020-12-31 21:50:05
- **YEAR**: un año, como 2021

## Tipos numéricos enteros:

- **TINYINT**: un número entero pequeño de un byte (8 bits), por tanto será entre -128 y 127
- **SMALLINT**: dos bytes (aproximadamente entre -32.000 y 32.000)
- **MEDIUMINT**: tres bytes (aproximadamente entre -8.000.000 y 8.000.000)
- **INT**: cuatro bytes (aproximadamente entre -2.000.000.000 y 2.000.000.000)
- **BIGINT**: ocho bytes (entre  $-2^{63}$  y  $2^{63}$ )

No utilices valores entre paréntesis en estos tipos.

En todos estos casos, podemos marcar los campos como **UNSIGNED** (casilla UN en MySQL Workbench). Al hacer esto sólo se consideran números positivos y por tanto se duplican los posibles valores positivos que podemos almacenar: por ejemplo un valor de tipo TINYINT sin signo podrá valer entre 0 y 255.

## Tipos numéricos con decimales:

- **FLOAT**: número decimal de 4 bytes
- **DOUBLE**: número decimal de 8 bytes
- **DECIMAL(5,2)**: este tipo sirve cuando queremos un número exacto de cifras decimales. Por ejemplo estos valores indican que se almacenan números de 5 dígitos y 2 de ellos son decimales, admitiendo únicamente valores del estilo 567,43. Este tipo se suele usar para valores monetarios.

## Tipo de datos booleano:

Este tipo no existe como tal en MySQL: en su lugar se almacena como TINYINT, con valores de 1 (true) o 0 (false). Cuando rellenes valores en la tabla de un campo de este estilo, no podrás usar las palabras “true” o “false”: sin embargo estas palabras sí se podrán usar al utilizar sentencias SQL, y MySQL las traducirá automáticamente a 1 y 0.