

tipos de datos en mySQL

Uno de los conceptos básicos que debemos tener presente siempre a la hora de trabajar con bases de datos **mySQL** (o cualquier otra) es el tipo de datos que podemos utilizar para introducir registros en cada una de las tablas de la base de datos.

¿Tipos de Datos?

Creo que este concepto es muy fácil de entender. La información podemos representarla por medio de símbolos numéricos, alfanuméricos, formatos de fecha, hora, binarios, etc. Todas estas clases o divisiones son tipos de datos. Si tenemos un número que nos indica la cantidad de dinero que disponemos en una cuenta bancaria en Suiza diremos que tenemos un tipo de datos numérico. Nuestra fecha de cumpleaños es un tipo de dato de fecha, etc.

mySQL distingue una serie de tipos de datos que podremos utilizar a la hora de crear los campos que formarán nuestras tablas. Veamos a continuación estos tipos:

Tipo Texto (Char(x), Varchar(x), Text, TinyText, MediumText, LongText)

Char(x)	Tipo de datos que admite caracteres alfanuméricos. La longitud de este campo varía entre 1-255 y está delimitado a la longitud especificada entre paréntesis (x) en el momento de la creación del campo de la tabla. Al introducir datos en este campo siempre se solicitará el número de caracteres especificados. Si creamos un campo con Char(5) deberemos introducir cinco caracteres cada vez que incluyamos un dato en ese campo. Si incluimos menos, mySQL rellenará los caracteres que faltan hasta el número indicado con espacios.
Varchar(x)	Tipo de datos que admite caracteres alfanuméricos. Su uso es similar a Char(x) . A la hora de definir un campo de datos Varchar deberemos especificar el número máximo de caracteres que podrá aceptar en la entrada de datos, donde x es un número entre 1-255. A diferencia de Char , este tipo de datos es variable en su longitud, admitiendo entradas inferiores a la establecida.
Text, TinyText, MediumText, LongText	Mediante la declaración de este tipo de datos se admiten la inclusión de cadenas alfanuméricas "case-insensitive" de longitudes variables. TinyText admite un máximo de 255 caracteres, Text admite 65.535, MediumText permite introducir textos de hasta 16.777.215 caracteres, LongText nos ofrece la posibilidad de incluir un máximo de 4.294.967.295 caracteres. Estos campos no necesitan de especificaciones de longitud a la hora de ser declarados.

Tipo Binario (Blob, TinyBlob, MediumBlob, LongBlob)

Blob	Un tipo de datos Blob es un objeto binario que puede almacenar cualquier tipo de datos o información, desde un archivo de texto con todo su formato (se diferencia en esto de el tipo Text) hasta imágenes, archivos de sonido o video, etc. Al igual que el tipo Text , Blob admite hasta 65.535 caracteres.
TinyBlob, MediumBlob, LongBlob	Son datos del mismo tipo que el anterior pero que varían en cuanto a su tamaño, así TinyBlob admite hasta 255 caracteres máximo, MediumBlob acepta tamaños de hasta 16.777.215 de caracteres y LongBlob 4.294.967.295 caracteres (como vemos estos tamaños se corresponden con los de TinyText , MediumText y LongText).

Tipo numérico (TinyInt, SmallInt, MediumInt, Int, BigInt, Float, Double, Decimal)

Int	Este es un tipo de datos numéricos de tipo entero. Este tipo de datos guarda valores enteros (no decimales) entre -2.147.483.648 y 2.147.483.647.
TinyInt, SmallInt, MediumInt, BigInt	Son tipos de datos numéricos enteros (no decimal). TinyInt agrupa un rango de números entre -128 y 127. SmallInt alcanza desde -32.768 hasta 32.767. MediumInt tiene un rango comprendido entre -8.388.608 y 8.388.607. Finalmente el tipo de datos BigInt ocupa un rango numérico entre -9.223.372.036.854.775.808 hasta 9.223.372.036.854.775.807.
Float (M,D)	Número de coma flotante de precisión simple. El valor del argumento M nos indica el número de dígitos decimales que se van a utilizar para representar el número. Así, un valor de 5 nos permitirá representar números comprendidos entre -99 y 99 (Numeros expresados en binario con 5 dígitos y signo). El valor del argumento D nos indica el número de posiciones decimales que se van a utilizar en la representación del número. Así, una representación tipo Float (5,2) nos permitirá incluir números entre -99,99 y 99,99. El rango de los números de coma flotante de precisión simple es de -3,402823466E+38 a -1,175494351E-38, 0, y 1,175494351E-38 hasta 3,402823466E+38.
Double (M,D)	Número de coma flotante de precisión doble. Es un tipo de datos igual al anterior cuya única diferencia es el rango numérico que abarca, siendo este el comprendido entre 1,7976931348623157E+308 hasta -2,2250738585072014E-308, 0, y 2,2250738585072014E-308 to 1,7976931348623157E+308
Decimal (M,D)	Su uso es similar al de los anteriores, pero, en este caso, D puede tener valor 0. El rango de este número es el mismo que el de número con coma flotante de precisión doble.

Tipo Fecha-Hora (Date, DateTime, TimeStamp, Time, Year)

Date	Formato de Fecha. Su representación es en formato de fecha numérica del tipo 'YYYY-MM-DD' (Año con cuatro dígitos, Mes con dos dígitos, día con dos dígitos). Su rango es '1000-01-01' (1 de enero del año 1000, en el cual yo era aún muy pequeño) hasta '9999-12-31' (31 de diciembre del 9999, que ya veremos que pasa después de las uvas)
DateTime	Es una combinación de formato de fecha y hora conjuntamente. Su representación es 'YYYY-MM-DD HH:MM:SS' (Año con cuatro dígitos, Mes con dos dígitos, día con dos dígitos, hora con dos dígitos, minutos con dos dígitos, segundos con dos dígitos). El rango que soporta este formato es de '1000-01-01 00:00:00' (las 00 horas, 00 minutos, 00 segundos del 1 de enero del año 1000, que no se yo con que reloj podían medir esto) hasta '9999-12-31 23:59:59' (las 23 horas, 59 minutos, 59 segundos del 31 de diciembre del año 9999, es decir, justo antes de las campanadas y una vez que han acabado los cuartos).
TimeStamp(N)	Este es un tipo de datos muy particular. Necesita de un argumento N que puede ser uno de estos números; 14, 12, 10, 8, 6, 4, 2. N representa el número de dígitos que se utilizarán para representar un valor de fecha y hora comprendido desde el inicio del año 1970 hasta algún momento del año 2037. Así: TimeStamp(14): YYYYMMDDHHMMSS (Año 4 dígitos + mes + día + hora + minutos + segundos 2 dígitos) TimeStamp(12): YYMMDDHHMMSS (Año 2 dígitos + mes + día + hora + minutos + segundos 2 dígitos) TimeStamp(10): YYMMDDHHMM (Año + mes + día + hora + minutos 2 dígitos) TimeStamp(8): YYMMDDHH (Año + mes + día + hora 2 dígitos) TimeStamp(6): YYMMDD (Año + mes + día 2 dígitos) TimeStamp(4): YYMM (Año + mes 2 dígitos) TimeStamp(2): YY (Año 2 dígitos)
Time	Tipo de datos con formato de Hora. MySQL muestra valores de hora con

	formato ' HH:MM:SS '
Year(D)	Tipo de datos con formato de año. Su representación puede ser ' YYYY ' (año con formato de 4 dígitos) o ' YY ' (año con formato de 2 dígitos) donde el valor del argumento D puede ser 4 o 2 respectivamente.

Este ha sido un breve repaso a los tipos de datos que podemos utilizar a la hora de crear campos de datos en tablas de bases de datos mySQL. Recomiendo (yo y todos) analizar profundamente el tipo de datos que se van a utilizar en cada campo ya que de esta manera podemos mejorar el rendimiento de nuestra base de datos en un porcentaje muy grande.