

# Tipos de campos en MySQL

MYSQL TIENE HABILITADOS DIVERSOS TIPOS DE CAMPOS QUE EN UNA PRIMERA APROXIMACIÓN PODRÍAN CLASIFICARSE EN TRES GRUPOS:

CAMPOS NUMÉRICOS

CAMPOS DE FECHA

CAMPOS DE CADENAS DE CARACTERES

CAMPOS NUMÉRICOS

MYSQL SOPORTA LOS TIPOS NUMÉRICOS EXACTOS (INTEGER, NUMERIC, DECIMAL, Y SMALLINT) Y LOS TIPOS NUMÉRICOS APROXIMADOS (FLOAT, DOUBLE PRECISION Y REAL).

LOS CAMPOS QUE CONTIENEN NÚMEROS ENTEROS ADMITEN EL PARÁMETRO UNSIGNED, QUE IMPLICA QUE NO ADMITA SIGNOS, POR LO QUE SOLO ACEPTARÍA ENTEROS POSITIVOS.

TODOS LOS CAMPOS NUMÉRICOS ADMITEN EL PARÁMETRO ZEROFILL CUYA FUNCIÓN ES COMPLETAR EL CAMPO CON CEROS A LA IZQUIERDA HASTA SU LONGITUD MÁXIMA.

## Tipos de campos numéricos enteros

ESTOS SON LOS DISTINTOS TIPOS DE CAMPOS NUMÉRICOS ENTEROS QUE ADMITE MYSQL. LOS PARÁMETROS SEÑALADOS ENTRE CORCHETES SON OPCIONALES.

**TINYINT [(M)] [UNSIGNED] [ZEROFILL]**

NÚMERO ENTERO MUY PEQUEÑO. CON LA OPCIÓN UNSIGNED PUEDE TOMAR VALORES ENTRE 0 Y 255. EN CASO CONTRARIO, PUEDE ESTAR COMPRENDIDO ENTRE -128 Y 127.

EL PARÁMETRO ZEROFILL SÓLO TIENE SENTIDO JUNTO CON LA OPCIÓN UNSIGNED YA QUE NO ES HABITUAL RELLENAR LOS NÚMEROS NEGATIVOS CON CEROS A LA IZQUIERDA DEL SIGNO.

EL VALOR POR DEFECTO DE PARÁMETRO M (NÚMERO DE CIFRAS) ES 4 SI NO ESTÁ ACTIVADA LA OPCIÓN UNSIGNED. SI ESTA OPCIÓN ESTUVIERA ACTIVADA EL VALOR POR DEFECTO SERÍA M=3. PARA VALORES DE M > VALOR POR DEFECTO REAJUSTA EL TAMAÑO AL VALOR POR DEFECTO.

SI SE ASIGNA A M UN VALOR MENOR QUE CUATRO LIMITA EL NÚMERO DE CARACTERES AL TAMAÑO ESPECIFICADO CONSIDERANDO EL SIGNO SÓLO EN LOS NÚMEROS NEGATIVOS.

POR EJEMPLO, SI M=3 ADMITIRÍA 148, PERO SI INTENTAMOS INSERTAR -148 RECORTARÍA POR LA IZQUIERDA Y SOLO INSERTARÍA -14.

SI INTENTAMOS INSERTAR UN VALOR FUERA DE RANGO REGISTRARÍA EL VALOR DENTRO DEL RANGO MÁS PRÓXIMO A ÉL.

P. EJ.: SI TRATAMOS DE INSERTAR EL VALOR 437 ESCRIBIRÍA 127 Ó 255, ESTE ÚLTIMO EN EL CASO DE TENER LA OPCIÓN UNSIGNED.

SI PRETENDIÉRAMOS INSERTAR -837 CON LA OPCIÓN UNSIGNED ESCRIBIRÍA 0 Y SIN ELLA PONDRÍA -128.

EL TAMAÑO DE UN CAMPO TINYINT ES DE 1 BYTE.

#### **SMALLINT [(M)] [UNSIGNED] [ZEROFILL]**

NÚMERO ENTERO PEQUEÑO. CON LA OPCIÓN **UNSIGNED** PUEDE TOMAR VALORES ENTRE 0 Y 65 535. EN CASO CONTRARIO, PUEDE ESTAR COMPRENDIDO ENTRE -32 768 Y 32 767.

SON VÁLIDOS LOS COMENTARIOS HECHOS PARA TINYINT, EXCEPTO LOS RELATIVOS A LOS VALORES POR DEFECTO DE M QUE EN ESTE CASO SERÍAN 6 Ó 5. SU TAMAÑO ES DE 2 BYTES.

#### **MEDIUMINT [(M)] [UNSIGNED] [ZEROFILL]**

NÚMERO ENTERO MEDIANO. CON LA OPCIÓN **UNSIGNED** PUEDE TOMAR VALORES ENTRE 0 Y 16 777 215. EN CASO CONTRARIO, PUEDE ESTAR COMPRENDIDO ENTRE -8 388 608 Y 8 388 607.

TAMBIÉN SON VÁLIDOS LOS COMENTARIOS HECHOS PARA TINYINT, EXCEPTO LOS RELATIVOS AL VALOR POR DEFECTO DE M QUE EN ESTE CASO SERÍAN 8. SU TAMAÑO ES DE 3 BYTES.

#### **INT [(M)] [UNSIGNED] [ZEROFILL]**

NÚMERO ENTERO. CON LA OPCIÓN **UNSIGNED** PUEDE TOMAR VALORES ENTRE 0 Y 4 294 967 295. EN CASO CONTRARIO, PUEDE ESTAR COMPRENDIDO ENTRE -2 147 483 648 Y 2 147 483 647.

SON VÁLIDOS TODOS LOS COMENTARIOS DE LOS CASOS ANTERIORES. SU TAMAÑO ES DE 4 BYTES.

#### **INTEGER [(M)] [UNSIGNED] [ZEROFILL]**

ES UN SINÓNIMO DE INT

#### **BIGINT [(M)] [UNSIGNED] [ZEROFILL]**

NÚMERO ENTERO GRANDE. CON LA OPCIÓN **UNSIGNED** PUEDE TOMAR VALORES ENTRE 0 Y 18 446 744 073 709 551 615. EN CASO CONTRARIO, PUEDE ESTAR COMPRENDIDO ENTRE -9 223 372 036 854 775 808 Y 9 223 372 036 854 775 807 647, PERO AL USARLO DESDE PHP ESTARÁ SUJETO A LAS LIMITACIONES MÁXIMAS DE LOS VALORES NUMÉRICOS DE ESTE.

SON VÁLIDOS TODOS LOS COMENTARIOS DE LOS CASOS ANTERIORES. SU TAMAÑO ES DE 8 BYTES.

## **NÚMEROS DE COMA FLOTANTE**

POR LA ESTRUCTURA BINARIA DE LOS MICROPROCESADORES Y HABIDA CUENTA DE QUE ALGUNOS NÚMEROS NO ENTEROS -SIN IR MÁS LEJOS, EL 0.1- REQUERIRÍAN INFINITOS CARACTERES BINARIOS PARA SU REPRESENTACIÓN EXACTA, SE HACE NECESARIO INTRODUCIR UN REDONDEO EN SU TRATAMIENTO INFORMÁTICO Y COMO CONSECUENCIA DE ELLO ASUMIR QUE SE GENERAN ERRORES DE MEDIDA.

ESTA CIRCUNSTANCIA OBLIGÓ AL TRATAMIENTO DE LOS NÚMEROS DECIMALES MEDIANTE EL LLAMADO STANDAR DE ARITMÉTICA DE PUNTO FLOTANTE, UN ALGORITMO DEFINIDO POR LA IEEE (INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS) QUE UNIFICÓ LOS PROCESOS DE REPRESENTACIÓN DE NÚMEROS EN ORDENADORES CON LO QUE SON UNIFORMEMENTE CONTROLABLES LOS ERRORES INTRODUCIDOS.

EL STANDAR DE ARITMÉTICA DE PUNTO FLOTANTE ESTABLECIÓ DOS NIVELES DE PRECISIÓN: PRECISIÓN SIMPLE, EN LA QUE TODO NÚMERO DEBE SER ALMACENADO EN 32 BITS (4 BYTES)

DOBLE PRECISIÓN, EN LA QUE LOS NÚMEROS SE ALMACENAN EN 64 BITS (8 BYTES).

MYSQL ADMITE LOS SIGUIENTES TIPOS DE NÚMEROS DE COMA FLOTANTE:

### **FLOAT(x) [ZEROFILL]**

NÚMERO DE COMA FLOTANTE. IGNORA LA OPCIÓN UNSIGNED, PERO SÍ ACEPTA ZEROFILL, POR LO QUE DEBE PRESTARSE ATENCIÓN A ESTAS OPCIONES YA QUE NO SERÍA DEMASIADO HABITUAL UNA PRESENTACIÓN COMO ESTA: 000-3.47

EL VALOR DE X ESPECIFICA LA PRECISIÓN. SI  $x \leq 24$  SERÁ DE PRECISIÓN SIMPLE. CUANDO  $24 < x \leq 53$  LO CONVERTIRÁ AUTOMÁTICAMENTE A DOBLE PRECISIÓN.

CUANDO NO SE ESPECIFICA EL VALOR DE X CONSIDERA EL CAMPO COMO DE PRECISIÓN SIMPLE. SU TAMAÑO ES DE 4 BYTES SI  $x \leq 24$  Y DE 8 BYTES CUANDO  $24 < x \leq 53$

### **FLOAT [(M,D)] [ZEROFILL]**

NÚMERO DE COMA FLOTANTE DE PRECISIÓN SIMPLE. SON VÁLIDOS LOS COMENTARIOS RELATIVOS A LAS OPCIONES UNSIGNED Y ZEROFILL DEL CASO ANTERIOR.

TOMA VALORES EN LOS INTERVALOS SIGUIENTES:

-3.402823466E+38 A -1.175494351E-38

0 Y

1.175494351E-38 A 3.402823466E+38.

M ES LA ANCHURA MÁXIMA DE VISUALIZACIÓN Y D ES EL NÚMERO DE DECIMALES. SI  $M > 24$  SE CONVIERTE AUTOMÁTICAMENTE A DOBLE PRECISIÓN

FLOAT SIN ARGUMENTOS REPRESENTA UN NÚMERO DE COMA FLOTANTE Y PRECISIÓN SIMPLE.

### **DOUBLE [(M,D)] [ZEROFILL]**

NÚMERO DE COMA FLOTANTE DE DOBLE PRECISIÓN. SIGUEN SIENDO VÁLIDOS LOS COMENTARIOS RELATIVOS A LAS OPCIONES UNSIGNED Y ZEROFILL DEL CASO ANTERIOR.

TOMA VALORES EN LOS INTERVALOS SIGUIENTES:

-1.7976931348623157E+308 A -2.2250738585072014E-308

0 Y

2.2250738585072014E-308 A 1.7976931348623157E+308

M ES LA ANCHURA MÁXIMA DE VISUALIZACIÓN Y D ES EL NÚMERO DE DECIMALES.

**DOUBLE SIN ARGUMENTOS REPRESENTA UN NÚMERO DE COMA FLOTANTE Y PRECISIÓN DOBLE.**

**REAL [(M,D)] [ZEROFILL]**

**ES SINÓNIMO DE DOUBLE.**

**DECIMAL [(M[,D])] [ZEROFILL]**

**ES UN NÚMERO DE COMA FLOTANTE Y DOBLE PRECISIÓN QUE SE ALMACENA COMO UN CAMPO DE TIPO CHAR.**

**EL VALOR ES GUARDADO COMO UNA CADENA DONDE CADA CARÁCTER REPRESENTA UNA CIFRA. LA COMA Y EL SIGNO MENOS DE LOS NÚMEROS NEGATIVOS NO SON TENIDOS EN CUENTA EN EL VALOR DE M -ANCHURA MÁXIMA DE VISUALIZACIÓN- AUNQUE SI SE RESERVA -AUTOMÁTICAMENTE- ESPACIO PARA ELLOS EN CAMPO.**

**SI D VALE 0 NO TENDRÁ PARTE DECIMAL. LOS NÚMEROS TOMAN VALORES EN EL MISMO INTERVALO ESPECIFICADO PARA DOUBLE.**

**LOS VALORES POR DEFECTO DE M Y D SON RESPECTIVAMENTE 10 Y 0. OCUPAN M+2 BYTES SI D > 0; M+1 BYTES SI D = 0 Ó D+2 BYTES SI M < D**

**NUMERIC(M,D) [ZEROFILL]**

**SE COMPORTA DE FORMA IDÉNTICA A DECIMAL**

## Campos de fecha

**MYSQL DISPONE DE CAMPOS ESPECÍFICOS PARA EL ALMACENAMIENTO DE FECHAS. SON LOS SIGUIENTES:**

**DATE**

**RECOGE UNA FECHA DENTRO DEL INTERVALO 01-01-1000 A 31-12-9999. MYSQL GUARDA LOS VALORES DATE CON FORMATO AAAA-MM-DD (AÑO-MES-DÍA) . SU TAMAÑO ES DE 3 BYTES.**

**DATETIME**

**RECOGE UNA COMBINACIÓN DE FECHA Y HORA DENTRO DEL INTERVALO 00:00:00 DEL DÍA 01-01-1000 Y LAS 23:59:59 DEL DÍA 31-12-9999. MYSQL GUARDA LOS VALORES DATETIME CON FORMATO AAAA-MM-DD HH:MM:SS (AÑO-MES-DÍA HORA:MINUTOS:SEGUNDOS) . SU TAMAÑO ES DE 8 BYTES.**

**TIME**

RECOGE UNA HORA DENTRO DEL INTERVALO -838:59:59 A 838:59:59. MYSQL GUARDA LOS VALORES TIME CON FORMATO HH:MM:SS(HORAS:MINUTOS:SEGUNDOS) . SU TAMAÑO ES DE 3 BYTES.

### YEAR O YEAR(2) O YEAR(4)

RECOGE UN AÑO EN FORMATO DE CUATRO CIFRAS (YEAR O YEAR(4)) O EN FORMATO DE DOS CIFRAS (YEAR(2)) DENTRO DEL INTERVALO 1901 A 2155 EN EL CASO DE CUATRO CIFRAS O DE 1970 A 2069 SI SE TRATA DE DOS CIFRAS. SU TAMAÑO ES DE 1 BYTE.

### TIMESTAMP [(M)]

RECOGE UN TIEMPO UNIX. EL INTERVALO VÁLIDO VA DESDE 01-01-1970 00:00:00 A CUALQUIER FECHA DEL AÑO 2037.

EL PARÁMETRO M PUEDE TOMAR LOS VALORES: 14 (VALOR POR DEFECTO), 12, 8, O 6 QUE SE CORRESPONDEN CON LOS FORMATOS AAAAMMDDHHMMSS, AAMMDDHHMMSS, AAAAMMDD, O AAMMDD. SI SE LE ASIGNA LA OPCIÓN NUL GUARDARÁ LA HORA ACTUAL. CUANDO SE ASIGNA 8 O 14 COMO PARÁMETROS ES CONSIDERADO COMO UN NÚMERO Y PARA LAS DEMÁS OPCIONES COMO UNA CADENA.

INDEPENDIEMENTE DEL VALOR DEL PARÁMETRO, UN CAMPO TIMESTAMP SIEMPRE OCUPA 4 BYTES.

## CAMPOS TIPO CADENA DE CARACTERES

### CHAR (M) [BINARY]

ES UNA CADENA DE TAMAÑO FIJO QUE SE COMPLETA A LA DERECHA POR ESPACIOS SI ES NECESARIO.

EL PARÁMETRO M PUEDE VALER DE 1 A 255 CARACTERES.

LOS ESPACIOS FINALES SON SUPRIMIDOS CUANDO LA CADENA ES INSERTADA EN EL REGISTRO.

LOS VALORES DE TIPO CHAR SON ELEGIDOS Y COMPARADOS SIN TENER EN CUENTA MAYÚSCULAS / MINÚSCULAS Y UTILIZAN EL JUEGO DE CARÁCTERES POR DEFECTO.

SE PUEDE UTILIZAR EL OPERADOR BINARY PARA HACER LA CADENA SENSIBLE A MAYÚSCULAS / MINÚSCULAS.

SE PUEDE UTILIZAR UN CAMPO TIPO CHAR(0) CON EL ATRIBUTO NULL PARA ALMACENAR UNA VALOR BOOLEANO. EN ESTE CASO OCUPARÁ UN SOLO BYTE Y PODRÁ TENER ÚNICAMENTE DOS VALORES: NUL O "".

SU TAMAÑO ES DE M BYTES SIENDO  $1 \leq M \leq 255$  .

### VARCHAR(M) [BINARY]

ES UNA CADENA DE CARACTERES DE LONGITUD VARIABLE. SU TAMAÑO MÁXIMO -ESPECIFICADO EN EL PARÁMETRO **M**- PUEDE ESTAR COMPRENDIDO ENTRE 1 Y 255 CARACTERES. CON LA OPCIÓN **BINARY** ES CAPAZ DE DISCRIMINAR ENTRE MAYÚSCULAS / MINÚSCULAS.

### **TINYBLOB o TINYTEXT**

**TINYBLOB** Y **TINYTEXT** SON CADENAS DE CARACTERES DE LONGITUD VARIABLE CON UN TAMAÑO MÁXIMO DE 255 ( $2^8 - 1$ ) CARACTERES.

LA DIFERENCIA ENTRE AMBAS ES QUE **TINYBLOB** DISCRIMINA ENTRE MAYÚSCULAS / MINÚSCULAS, MIENTRAS QUE **TINYTEXT** NO LO HACE.

NINGUNO DE LOS CAMPOS **BLOB** Y **TEXT** ADMITE VALORES POR DEFECTO

LAS VERSIONES DE **MYSQL** ANTERIORES A 3.23.2 PERMITEN UTILIZAR ESTOS CAMPOS PARA INDEXAR.

SI SE INTENTA GUARDAR EN UN CAMPO DE ESTE TIPO UNA CADENA DE MAYOR LONGITUD QUE LA ESPECIFICADA SOLAMENTE SE GUARDARÁN LOS **M** PRIMEROS CARACTERES DE LA CADENA.

### **BLOB o TEXT**

**BLOB** Y **TEXT** SON CADENAS DE CARACTERES DE LONGITUD VARIABLE CON UN TAMAÑO MÁXIMO DE 65535 ( $2^{16} - 1$ ) CARACTERES.

LA DIFERENCIA ENTRE AMBAS ES QUE **BLOB** SI DISCRIMINA ENTRE MAYÚSCULAS / MINÚSCULAS, MIENTRAS QUE **TEXT** NO LO HACE.

NINGUNO DE LOS CAMPOS: **BLOB** Y **TEXT** ADMITE VALORES POR DEFECTO

### **MEDIUMBLOB o MEDIUMTEXT**

**MEDIUMBLOB** Y **MEDIUMTEXT** SON CADENAS DE CARACTERES DE LONGITUD VARIABLE CON UNA LONGITUD MÁXIMA DE 16.777.215 ( $2^{24} - 1$ ) CARACTERES.

SON VÁLIDAS LAS ESPECIFICACIONES HECHAS EN EL APARTADO ANTERIOR.

EL TAMAÑO MÁXIMO DE LOS CAMPOS DE ESTE TIPO ESTÁ SUJETO A LIMITACIONES EXTERNAS TALES COMO LA MEMORIA DISPONIBLE Y EL TAMAÑO DEL BUFFER DE COMUNICACIÓN SERVIDOR/CLIENTE.

### **LOBLOB o LONGTEXT**

SU ÚNICA DIFERENCIA CON LA ANTERIOR ES EL TAMAÑO MÁXIMO DE LA CADENA, QUE EN ESTE CASO ES 4.294.967.295 ( $2^{32} - 1$ ) CARACTERES.

### **ENUM('VALOR1', 'VALOR2', ...)**

ES UNA CADENA DE CARACTERES QUE CONTIENE UNO SOLO DE LOS VALORES DE LA LISTA (**VALOR1**, **VALOR2**, ETC. ETC.).

A LA HORA DE INSERTAR UN NUEVO REGISTRO EN UNA TABLA, EL VALOR A ESPECIFICAR PARA UN CAMPO DE ESTE TIPO HA DE SER UNA CADENA QUE CONTenga UNO DE LOS VALORES ESPECIFICADOS EN LA TABLA. SI SE TRATARA DE INSERTAR UN VALOR DISTINTO DE ELLOS INSERTARÍA UNA CADENA VACÍA.

**SET('VALOR1','VALOR2','VALOR3'...)**

ES UNA CADENA DE CARACTERES FORMADOS POR LA UNIÓN DE NINGUNO, UNO O VARIOS DE LOS VALORES DE UNA LISTA. EL MÁXIMO DE ELEMENTOS ES 64.

LOS VALORES QUE DEBEN ESCRIBIRSE EN LOS CAMPOS DE ESTE TIPO HAN DE SER NUMÉRICOS, EXPRESADOS EN FORMA BINARIA O EN FORMA DECIMAL.

EN EL SUPUESTO DE QUE CONTUVIERA TRES VALORES LOS POSIBLES VALORES A INSERTAR EN UN CAMPO DE ESTE TIPO A LA HORA DE AÑADIR UN REGISTRO SERÍAN LOS SIGUIENTES.

INCLUIR VALORES			CÓDIGO VALOR			CADENA	EQUIV.
VAL1	VAL2	VAL3	VAL1	VAL2	VAL3	BINARIA	DECIMAL
Si	Sí	Sí	1	1	1	111	7
Si	Sí	No	1	1	0	011	3
Si	No	Sí	1	0	1	101	5
No	Sí	Sí	0	1	1	110	6
No	No	Sí	0	0	1	100	4
No	Sí	NO	0	1	0	010	2
Si	No	No	1	0	0	001	1
No	No	No	0	0	0	000	0