

Bases de Datos

Unidad 3: Diseño Físico de Bases de Datos

UNIDAD 3: DISEÑO FÍSICO DE BASES DE DATOS

1. Características físicas del almacenamiento de la base de datos
2. Herramientas gráficas para la implementación de la base de datos.
3. El lenguaje de definición de datos.
4. Creación modificación y eliminación de bases de datos.
5. Tipos de datos. Valores. Operadores
6. Administración de tablas
 - 6.1 Sintaxis de la instrucción CREATE TABLE
 - 6.2 Propiedades de tablas
 - 6.3 Modificación de tablas
 - 6.4 Eliminación de tablas
7. El lenguaje de control de datos.
8. Vistas
9. Usuarios y privilegios

Tipos de datos:

Vamos a ver de forma resumida los tipos de datos que podemos usar en MySQL para las columnas de las tablas. Estos se pueden clasificar en:

- ☐ Numéricos
- ☐ Cadenas de caracteres
- ☐ Cadenas de bytes o binarias
- ☐ Fecha y hora
- ☐ Booleanos
- ☐ Enumerados
- ☐ Conjuntos

Tipos de datos numéricos:

TINYINT	Enteros entre -128 y +127. Sin signo entre 0 y 255. Ocupan 1 byte
SMALLINT	Enteros entre -32768 y +32767. Sin signo entre 0 y 65535. Ocupan 2 bytes
MEDIUMINT	Enteros entre aproximadamente -8 millones y + 8 millones. Sin signo aproximadamente entre 0 y 16 millones. Ocupan 3 bytes.
INT, INTEGER	Enteros entre aproximadamente -2 mil millones y +2 mil millones. Sin signo aproximadamente entre 0 y 4 mil millones. Ocupan 4 bytes.
BIGINT	Enteros entre aproximadamente -10^{19} y $+10^{19}$. Sin signo aproximadamente entre 0 y 2×10^{19}
FLOAT	Reales en coma flotante de precisión simple (6dígitos). Admite negativos entre -3.4×10^{38} y -1.2×10^{-38} , 0 y positivos entre 1.2×10^{-38} y $3.4 \times 10^{+38}$
DOUBLE, REAL	Reales en coma flotante de precisión doble (12dígitos). Permite números negativos entre -1.8×10^{308} y -2.2×10^{-308} , 0 y positivos entre 2.2×10^{-308} y $1.8 \times 10^{+308}$
DECIMAL, NUMERIC	Número en coma fija (con una posición fija de la coma decimal. Por defecto sirve para números de hasta 10 cifras en la parte entera sin cifras decimales. La variante DECIMAL(M,D) permite especificar en M el número total de dígitos y en D el número de decimales. Por ejemplo con (3,2) el número positivo más grande que podremos almacenar con esa definición será 9.99 ya que son tres dígitos y dos de ellos serán utilizados como decimales.

Numéricos. Modificadores

- ❑ Todos los enteros se pueden definir en la forma TIPO(N) donde N indicará el número de cifras con que se presenta o edita el número.
- ❑ Todos los reales se pueden definir en la forma TIPO(N,D) donde N indicará el número total de cifras con que se presenta o edita el número (de 0 a 24) y D es el número de decimales .
- ❑ Todos los tipos numéricos admiten los modificadores UNSIGNED y ZEROFILL. UNSIGNED especifica que el entero es sin signo y ZEROFILL que un número que ocupa N cifras se muestra en pantalla rellenando con ceros las cifras no significativas del número.

Ejemplos de definición de columnas:

```
numPrimitiva TINYINT UNSIGNED;  
numLoteria INT(5) UNSIGNED ZEROFILL;  
pesoAtomico DOUBLE;  
tempMedia DECIMAL(4,2);  
precioUnidad FLOAT;
```

Tipos de datos cadenas de caracteres:

CHAR(N)	Cadena de longitud fija de N caracteres. Cualquier valor que se almacene ocupará lo correspondiente a N caracteres. Si se cargan menos caracteres se rellena con espacios por la derecha. Admite hasta 255 caracteres
VARCHAR(N)	Cadena de longitud variable hasta un máximo de N caracteres. Si se carga una cadena con menos de N caracteres, ocupará tanto espacio como necesiten los caracteres cargados (no se rellena con espacios). Admite hasta 65535 caracteres.
TINYTEXT	Igual que VARCHAR para cadenas de hasta 255 caracteres.
TEXT	Parecido a VARCHAR. Tiene algunas pequeñas diferencias, como la imposibilidad de contener valores por defecto o formar parte de un índice. En general es más conveniente usar VARCHAR por compatibilidad con otros SGBD. Hasta 65535 caracteres.
MEDIUMTEXT	Igual que VARCHAR para cadenas de hasta 16 millones de caracteres.
LONGTEXT	Igual que VARCHAR para cadenas de hasta 4 mil millones de caracteres.

Ejemplos de definición de columnas de tipos cadenas de caracteres:

nombreCiclo VARCHAR(80),

dniProfesor CHAR(9);

codPostal CHAR(5);

signoQuiniela CHAR; -- Equivalente a usar CHAR(1)

codPais CHAR(2);

argPelicula TEXT; -- Se puede y es más recomendable usar VARCHAR

Tipos de datos. Cadenas de bytes o binarias

Permiten almacenar secuencias de bytes, por ejemplo el contenido de ficheros. También permiten almacenar cadenas de texto, en cuyo caso, al comparar se diferencia entre mayúsculas y minúsculas. No es adecuado definir una columna para cargar en ella el contenido de un fichero. Para ese caso es mejor definirla para que contenga un texto con el nombre y ubicación del fichero en el disco.

BINARY(N)	Cadena de longitud fija de N bytes. Cualquier valor que se almacene ocupará lo correspondiente a N bytes. Si se cargan menos caracteres se rellena con espacios por la derecha. Admite hasta 255 caracteres
VARBINARY(N)	Similar a VARCHAR para cadenas binarias.
TINYBLOB(N)	Similar a TINYTEXT para cadenas binarias.
BLOB(N)	Similar a TEXT para cadenas binarias.
MEDIUMBLOBN)	Similar a MEDIUMTEXT para cadenas binarias.
LOBLOB(N)	Similar a LONGTEXT para cadena binarias.

Tipos de datos. Fecha y hora

DATE	Permite almacenar fechas en el formato 'aaaa-mm-dd'. Se pueden usar otros separadores. El rango soportado es desde 1000-1-1 hasta 9999-12-31
TIME	Permite almacenar datos de tipo hora con el formato 'hh:mm:ss'. Se pueden usar otros separadores. El rango soportado es desde -838:59:59 hasta +838:59:59.
DATETIME	Permite almacenar datos con fecha y hora con el formato: 'aaaa-mm-dd hh:mm:ss'
TIMESTAMP	<p>Permite almacenar datos con fecha y hora con el formato: 'aaa-mm-dd hh:mm:ss'</p> <p>El rango de representación es entre 1970-01-01 00:00:00 y 2037-12-31 23:59:59. Es útil para registrar cuando se producen operaciones de inserción y modificación sobre columnas de este tipo. Reciben por defecto la fecha y hora actuales cuando no se carga en ellas ningún valor.</p>

Tipos de datos. Booleanos

En MySQL se tiene el tipo BOOLEAN para representar valores booleanos (verdadero o falso).

La realidad es que el dato que se almacena en un BOOLEAN es un tipo TINYINT(1).

El valor 0 almacenado representa **false** y el valor 1 representa **true**.

Para hacer referencia a los valores que tiene un BOOLEAN podemos usar indistintamente 0 o false y 1 o true, aunque es mejor usar false y true.

Tipos de datos. Enumerados

❑ Es un tipo de dato que puede contener uno de entre un conjunto de textos definidos en la declaración del dato. Un dato de tipo enumerado se define como `ENUM('cad1', 'cad2',, 'cadN')`

❑ Para definir una columna dia para contener los días de la semana, haremos:

Dia

ENUM('lunes', 'martes', 'miercoles', 'jueves', 'viernes', 'sabado', 'domingo')

❑ Realmente en una columna ENUM se almacenan los **valores índice** del dato guardado comprendidos entre 1 y el número de elementos de la enumeración.

❑ Un dato ENUM se puede manejar indistintamente con los valores definidos en la enumeración o con los índices.

❑ Los datos enumerados se ordenan por el índice

Tipos de datos. Conjuntos

❑ Es un tipo de datos que puede contener varios valores o ninguno de entre un conjunto de textos definidos en la declaración del dato. Un dato de tipo conjunto se define como SET('cad1', 'cad2',, 'cadN')

❑ Para definir una columna Formato para contener el formato de letra fuente haremos:

formato SET('negrilla','subrayado','cursiva')

❑ Al insertar valores en una columna del tipo anterior podemos insertar:

'negrilla'

'cursiva'

'negrilla,cursiva'

❑ Si se insertan dos o más valores del conjunto, los valores se han de escribir respetando el orden en que se definieron en el conjunto. Sería inválida la inserción de 'cursiva,negrilla'

❑ Los valores no válidos que se traten de insertar se ignoran.

❑ Para comprobar si un dato SET contiene un determinado grupo de valores se usa la función FIND_IN_SET. También se puede usar el operador LIKE adecuadamente.

Representación de valores literales

Cadenas de caracteres:

- Entre comillas dobles o entre comillas simples
- Para representar comillas dentro de un literal cadena de caracteres se tienen que preceder de \
- Caracteres especiales se tienen que preceder de \

Numéricos:

- El separador de parte entera y decimal de un número es el carácter punto.
- Los valores correspondientes a numéricos como flotante se pueden representar en notación exponencial (por ejemplo, 2.7562e+12).
- Se pueden representar valores hexadecimales (precedidos de 0x, por ejemplo, 0x3A24FF).

Booleanos:

Se representan con true o false

Valores nulos (sin valor asignado): Se representan con **NULL**.

Operadores

En las instrucciones SQL podemos usar un amplio número de operadores. De ellos, los más importantes son:

Operadores de comparación y pertenencia

De igualdad, desigualdad: = !=

Mayor que, mayor o igual que: > >=

Menor que, menor o igual que: < <=

Es nulo, no es nulo: IS NULL IS NOT NULL

Pertenencia a un rango: BETWEEN 1 AND 100

Pertenencia a un conjunto: IN(1,2,4,8)

Operadores lógicos

Y Lógico: AND

nota >=5 AND nota <=10

O lógico: OR

nota>10 OR nota <0

Negación: NOT

NOT(x>=5)