



# Base de Datos (MySQL)

**Presentación 02 – Conceptos iniciales**





## Contenido:

- Tablas
- Tipos de datos (numéricos, texto, fecha)
- Creación de una base de datos
- Creación de una tabla
- Eliminación de una tabla
- Restricciones de tablas
- Modificadores de campos de tablas
- Otras restricciones y comandos en MySQL
- Comentarios
- Enum y Set



## Tablas

- Son el **objeto principal de una base de datos**, ya que en ellas se almacenarán los datos.
- Son objetos compuestos por una estructura que almacena datos relacionados acerca de algún objeto/entidad en general.
- Las tablas tienen un **nombre** y debe **ser único** en toda la base datos.
- Están compuestas por registros **(filas)** y **campos (columnas)**.
- Los registros y campos pueden estar en diferentes órdenes.
- Una base de datos puede contener varias tablas.



## Tipos de datos

Al diseñar tablas dentro de una base de datos, se debe especificar el tipo de datos y la capacidad que podrá almacenar cada campo.

MySQL admite dos tipos de datos: números y cadenas de caracteres. Además de estos dos grandes grupos, admite otros tipos de datos especiales, como formatos de fecha, lógicos, etc. (No utilizaremos todos los tipos de datos en este curso).

Una correcta elección debe procurar que la tabla esté correctamente dimensionada en su capacidad, que destine un tamaño apropiado a la longitud de los datos y que apunte a lograr la máxima velocidad de ejecución de consultas posible.



## Caracteres o cadenas de texto

Las cadenas de texto se utilizan para almacenar una serie de caracteres, palabras y/o frases de texto en donde cada carácter es lo mismo que un byte.

|                 |  |
|-----------------|--|
| <b>CHAR</b>     | Este tipo se utiliza para almacenar cadenas de <b>longitud fija</b> . Su longitud abarca desde <b>1 a 255 caracteres</b> . Un campo CHAR ocupará siempre el máximo de longitud que le hayamos asignado, aunque el tamaño del dato sea menor.                     |
| <b>VARCHAR</b>  | Al igual que el anterior se utiliza para almacenar cadenas, en el mismo rango de <b>1-255 caracteres</b> , pero en este caso, de <b>longitud variable</b> . VARCHAR solo almacena la longitud del dato, permitiendo que el tamaño de la base de datos sea menor. |
| <b>TINYTEXT</b> | Texto de <b>longitud variable</b> que puede tener <b>hasta 255 caracteres</b> .  |



## Caracteres o cadenas de texto

|                   |  |
|-------------------|--|
| <b>TEXT</b>       | Texto de <b>longitud variable</b> que puede tener <b>hasta 65.535 caracteres</b> .   |
| <b>MEDIUMTEXT</b> | Texto de <b>longitud variable</b> que puede tener <b>hasta 16.777.215 caracteres</b> .   |
| <b>LONGTEXT</b>   | Texto de <b>longitud variable</b> que puede tener <b>hasta 4.294.967.295 caracteres</b> .  |
| <b>BLOB</b>       | <b>Dato binario</b> que puede almacenar <b>archivos o texto</b> . En este caso, los tipos TINYBLOB, BLOB, MEDIUMBLOB y LONGBLOB son idénticos a sus homólogos TEXT, con la diferencia de que las búsquedas en un tipo <b>BLOB tienen en cuenta las mayúsculas y minúsculas</b> . |



## Datos numéricos

En este tipo de campos pueden almacenarse sólo números, positivos o negativos, enteros o decimales, en notación hexadecimal, científica o decimal.

Los tipos numéricos tipo INTEGER admiten los atributos SIGNED y UNSIGNED. En el primer caso, indica que pueden tener valor negativo y en el segundo, sólo positivo.

Los tipos numéricos pueden además usar el atributo ZEROFILL, en cuyo caso los números se completarán hasta la máxima capacidad disponible con ceros.



## Datos numéricos

|                   |   |
|-------------------|---|
| <b>BIT o BOOL</b> | Para un <b>número entero</b> que puede ser <b>0</b> ó <b>1</b> .  |
| <b>TINYINT</b>    | Es un <b>número entero</b> con rango de <b>valores válidos desde -128 a 127</b> .<br>Si se configura como <b>unsigned</b> (sin signo), el rango de valores es de <b>0 a 255</b> .         |
| <b>SMALLINT</b>   | Para <b>números enteros</b> , con rango desde <b>-32.768 a 32.767</b> .<br>Si se configura como <b>unsigned</b> , <b>0 a 65.535</b> .   |
| <b>MEDIUMINT</b>  | Para <b>números enteros</b> . El rango de valores va desde <b>-8.388.608 a 8.388.607</b> .<br>Si se configura como <b>unsigned</b> , <b>0 a 16.777.215</b> .                              |
| <b>INT</b>        | Para almacenar <b>números enteros</b> , en un rango de <b>-2.147.463.848 a 2.147.483.647</b> .<br>Si configuramos este dato como <b>unsigned</b> , el rango es <b>0 a 4.294.967.295</b> . |





## Datos numéricos

|                    |  |
|--------------------|--|
| <b>BIGINT</b>      | <b>Número entero</b> con rango de valores desde <b>-9.223.372.036.854.775.808</b> a <b>9.223.372.036.854.775.807</b> .<br><i>Unsigned</i> , desde <b>0</b> a <b>18.446.744.073.709.551.615</b> .   |
| <b>FLOAT (m,d)</b> | Representa <b>números decimales</b> . Podemos especificar <b>cuántos dígitos (m)</b> pueden utilizarse (término también conocido como <i>ancho de pantalla</i> ), y cuántos en la parte <b>decimal (d)</b> . <i>MySQL</i> redondeará el decimal para ajustarse a la capacidad. |
| <b>DOUBLE</b>      | <b>Número de coma flotante de precisión doble</b> . Es un tipo de datos igual al anterior cuya única <b>diferencia es el rango numérico</b> que abarca.  |
| <b>DECIMAL</b>     | Almacena los <b>números como cadenas de texto</b> .  |



## Datos de fecha

|          |  |
|----------|--|
| DATE     | Para almacenar <b>fechas</b> . El rango de valores va desde <b>0000-00-00 a 9999-12-31</b> .<br>El formato por defecto es <b>YYYY-MM-DD</b> .  |
| DATETIME | Combinación de <b>fecha y hora</b> . El rango de valores va desde el <b>1 de enero del 1001</b> a las <b>0 horas, 0 minutos y 0 segundos al 31 de diciembre del 9999</b> a las <b>23 horas, 59 minutos y 59 segundos</b> .<br>El formato de almacenamiento es de <b>año-mes-día horas:minutos:segundos</b> . |
| TIME     | Almacena <b>una hora</b> . El rango de horas va desde <b>-838 horas, 59 minutos y 59 segundos a 838, 59 minutos y 59 segundos</b> .<br>El formato de almacenamiento es de <b>'HH:MM:SS'</b>  |
| YEAR     | Almacena <b>un año</b> . El rango de valores permitidos va desde el <b>año 1901 al año 2155</b> .<br>El campo puede tener <b>tamaño dos o tamaño 4</b> , depende de si queremos almacenar el año con dos, o cuatro dígitos.  |



## Creación de una base de datos

Para crear una base de datos (por ejemplo, con el nombre negocio), la instrucción SQL a ejecutar será:

```
CREATE DATABASE negocio;
```

## Creación de una tabla

La sentencia CREATE TABLE se utiliza para crear una tabla con las columnas (campos) que se indiquen.

```
CREATE TABLE facturas(  
letra CHAR(1) not null,  
numero INT not null,  
id_cliente INT not null,  
fecha DATE not null,  
monto FLOAT not null  
);
```



## Eliminar una tabla

Para eliminar una tabla (por ejemplo, con el nombre productos) de una base de datos, la instrucción SQL a ejecutar será:

```
DROP TABLE productos;  
o  
DROP TABLE IF EXISTS productos;
```

La cláusula IF EXISTS devuelve una advertencia en caso de que no exista la tabla a eliminar.



## Restricciones de las tablas

### Puntos claves

- Los nombres de las tablas deben ser únicos en la base de datos.
- Los nombres de las columnas debe ser únicos en la tabla.
- No podrán existir dos registros con el mismo valor de la clave primaria.

### Columnas no descomponibles

- Son aquellas columnas que contienen cierta información que no puede almacenarse en dos o más columnas.
- Son fáciles de actualizar.
- Son fáciles de consultar.
- Mejoran la implementación de la integridad de los datos.



## Modificadores aplicables a los campos de una tabla

- **NOT NULL:** no permite valores nulos; la carga del dato en ese campo será obligatoria. Es decir, el campo no puede quedar vacío.
- **DEFAULT:** permite especificar un valor predeterminado.
- **CLAVE PRIMARIA (PRIMARY KEY):** una tabla suele tener una columna, o una combinación de columnas, cuyos valores identifican de forma única a cada registro de la tabla.

Estas columnas se denominan claves principales de la tabla y exigen la integridad de entidad de la tabla (un solo registro con ese valor de indicador único). Cuando cree o modifique una tabla, puede crear una clave primaria mediante la definición de una restricción PRIMARY KEY.



## PRIMARY KEY

Una tabla sólo puede tener una restricción PRIMARY KEY y ninguna columna a la que se aplique una restricción PRIMARY KEY puede aceptar valores NULL.

Cuando se especifica una restricción del tipo PRIMARY KEY en una tabla, el motor de base de datos exige la unicidad de los datos mediante la creación de un índice único para las columnas de la clave principal.

Este índice también permite un acceso rápido a los datos cuando se utiliza la clave principal en las consultas. De esta forma, las claves principales que se eligen deben seguir las reglas para crear índices únicos.

Si se define una restricción PRIMARY KEY para más de una columna, puede haber valores duplicados dentro de la misma columna, pero cada combinación de valores de todas las columnas de la definición de la restricción PRIMARY KEY debe ser única.



## Otras restricciones de campos

**NOT NULL:** indica que una columna debe ser de carga obligatoria. **NULL** constituye un valor en sí (valor desconocido). No es vacío en un campo de tipo texto, ni 0 en un campo numérico.

**UNIQUE:** indica que la columna no tendrá ningún valor repetido. Similar a PRIMARY KEY, pero a diferencia de esta última, UNIQUE permite un valor nulo, y puede haber varias columnas de este tipo en una tabla.

**BINARY:** indica que los valores en esta columna se almacenarán en modo binario, respetando mayúsculas y minúsculas.

**UNSIGNED:** indica que esta columna no usará un byte para el signo, es decir, permitirá almacenar números positivos únicamente.

**ZERO FILL:** indica que el contenido del campo se completará con ceros (siempre que sea numérico).

**AUTO\_INCREMENT:** el motor de base de datos incrementará automáticamente su valor. Una tabla sólo puede tener un campo autoincremental y éste tiene que formar parte de la PK (PRIMARY KEY).





## Otros comandos en MySQL

Comando **SHOW DATABASES**: muestra el catálogo de base de datos del servidor.

Comando **USE**: pone en uso una base de datos del servidor. Todos los comandos SQL que se ejecuten, se llevarán a cabo sobre la base de datos en uso.

Comando **SHOW TABLES**: muestra el catálogo o listado de tablas de la base de datos activa.

Comando **DESCRIBE**: devuelve la descripción de campos y detalles de una tabla (estructura física).

Comando **SHOW CHARSET**: muestra los CHARSET (juegos de caracteres).

Comando **SHOW COLLATION**: Muestra los COLLATIONS instalados.



## Comentarios

Los comentarios permiten escribir texto que no será interpretado como parte de las sentencias SQL, útiles para documentar y comentar acciones realizadas por las sentencias.

Se pueden utilizar las siguientes formas de escribir comentarios:

**# Esto es un comentario de 1 línea**  
**(Propia de MySQL)**

**/\* Esto es un comentario de 1 o más líneas \*/**  
**(Soportada en MySQL y otros motores)**

**-- Esto es un comentario de 1 línea**  
**(Soportada en MySQL y otros motores)**



## Enum

El tipo de datos denominado ENUM presenta las siguientes características:

- Sólo puede contener un valor.
- Se puede definir una lista de hasta 65.535 valores distintos.
- Si se permiten valores null, éste será el valor predeterminado; si no se define un valor predeterminado con DEFAULT, será el primer valor de la lista.
- Cada valor de la lista es numerado con un índice (comenzando en 1).

```
CREATE TABLE medida (  
  medida ENUM('pequeño', 'mediano', 'grande') NOT NULL DEFAULT 'mediano'  
);
```



## Set

Otro tipo de dato es SET, el cual posee las siguientes características:

- 0, 1 ó varios valores.
- Se puede definir una lista de hasta 64 valores distintos.
- Los valores no pueden contener comas, ya que los valores asignados en la lista están separados por ese carácter.
- Cada valor de la lista representa un bit de la cadena de bits del campo.
- El valor decimal del campo determina los bits, al marcar los valores que contiene el campo. De manera que, si todos los bits están a 1, significa que ese campo contiene todos los valores.