



## ¿Qué es MySQL?

MySQL es un sistema de gestión de bases de datos (SGBD) relacional, de código abierto y muy popular en la actualidad. Está diseñado para ser un SGBD rápido, eficiente y escalable, que puede manejar grandes cantidades de datos. Es muy utilizado en aplicaciones web y aplicaciones empresariales, debido a su capacidad para trabajar con múltiples usuarios al mismo tiempo y su integración con lenguajes de programación como PHP, Python, Java y C#.

### Conceptos básicos:

- **Modelo de datos:** Es una representación abstracta de los datos que se almacenarán en una base de datos. Define los tipos de datos que se pueden almacenar, las relaciones entre ellos y las restricciones de integridad que deben cumplirse.
- **Tablas:** Son estructuras de datos que almacenan información en una base de datos relacional. Una tabla está compuesta por filas y columnas, donde cada columna representa un campo de la tabla y cada fila representa un registro o entrada en la tabla.
- **Relaciones:** Son la forma en que las tablas se conectan entre sí. Las relaciones se establecen mediante el uso de claves foráneas y claves primarias que se relacionan entre sí. Por ejemplo, una tabla de clientes puede tener una relación con una tabla de pedidos a través de una clave foránea que relaciona los registros de la tabla de pedidos con los registros de la tabla de clientes.
- **Claves:** Las claves son campos o combinaciones de campos que se utilizan para identificar de forma única cada registro en una tabla. Las claves pueden ser claves primarias, que identifican de forma única cada registro en una tabla, o claves foráneas, que establecen una relación con otra tabla.
- **Consultas:** Son comandos que se utilizan para recuperar información de una base de datos. Las consultas pueden ser simples o complejas y se pueden utilizar para filtrar, ordenar y agrupar datos de una o varias tablas en función de ciertos criterios.

- **Tipos de datos:** En SQL, los tipos de datos son una forma de especificar qué tipo de información se almacena en una columna de una tabla. Cada tipo de dato tiene un rango y una precisión diferente, y elegir el tipo de datos correcto para cada columna es esencial para una buena gestión de la base de datos.

A continuación, se presentan algunos de los tipos de datos más comunes en SQL:

- **CHAR(n):** Almacena cadenas de caracteres de longitud fija de hasta n caracteres.
- **VARCHAR(n):** Almacena cadenas de caracteres de longitud variable de hasta n caracteres.
- **TEXT:** Almacena cadenas de caracteres de longitud variable de hasta 65,535 caracteres
- **INT:** Almacena números enteros en un rango específico, generalmente de -2,147,483,648 a 2,147,483,647.
- **FLOAT(p):** Almacena números de punto flotante con una precisión de p dígitos.
- **DOUBLE(p):** Almacena números de punto flotante de doble precisión con una precisión de p dígitos.
- **DATE:** Almacena fechas en el formato AAAA-MM-DD.
- **TIME:** Almacena horas en el formato HH:MM:SS.
- **DATETIME:** Almacena fechas y horas en el formato AAAA-MM-DD HH:MM:SS.
- **BOOLEAN:** Almacena valores booleanos, que pueden ser verdadero o falso.

## Consultas frecuentes


- Mostrar todas las bases de datos disponibles:

```
1 SHOW DATABASES
```

Database
didacus
information_schema
mi_wordpress
mysql
performance_schema
phpmyadmin
respaldo
test

- Crear una base de datos:

```
1 CREATE DATABASE nueva_bd
```

 nueva\_bd

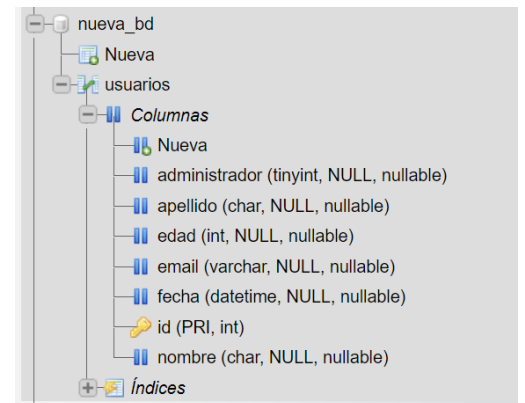
- Seleccionar una base de datos:

```
1 USE nueva_bd
```

- Crear una tabla:

Ejecutar la(s) consulta(s) SQL en la base de datos nueva\_bd: ?

```
1 CREATE TABLE usuarios (
2   id INT NOT NULL AUTO_INCREMENT,
3   nombre CHAR(15),
4   apellido CHAR(15),
5   edad INT(3) UNSIGNED CHECK (edad <= 119),
6   email VARCHAR(50),
7   fecha DATETIME,
8   administrador BOOLEAN,
9   PRIMARY KEY (id)
10 );
```



- Mostrar todas las tablas de una base de datos:

```
1 SHOW TABLES;
```

Tables\_in\_nueva\_bd

usuarios

- Mostrar información sobre una tabla:

```
1 DESCRIBE usuarios;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nombre	char(15)	YES		NULL	
apellido	char(15)	YES		NULL	
edad	int(3) unsigned	YES		NULL	
email	varchar(50)	YES		NULL	
fecha	datetime	YES		NULL	
administrador	tinyint(1)	YES		NULL	

- Insertar datos en una tabla:

```
1 INSERT
2 INTO usuarios (nombre,apellido,edad,email,fecha, administrador)
3 VALUES ('Daniel', 'Rondon', 27, 'daniel@email.com','2023-04-05 12:30:45',false);
```

✓ 1 fila insertada.  
La Id de la fila insertada es: 1 (La consulta tardó 0,0004 segundos.)

- Seleccionar todos los datos de una tabla:

```
1 SELECT * FROM usuarios;
```

	id	nombre	apellido	edad	email	fecha	administrador
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	Daniel	Rondon	27	daniel@email.com	2023-04-05 12:30:45	0
<input type="checkbox"/> Seleccionar todo Para los elementos que están marcados: <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar <input type="checkbox"/> Exportar							
<input type="checkbox"/> Mostrar todo   Número de filas: 25   Filtrar filas: <input type="text" value="Buscar en esta tabla"/>							

- Seleccionar datos específicos de una tabla:

```
1 SELECT nombre,email FROM usuarios
```

	nombre	email
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	Daniel	daniel@email.com
<input type="checkbox"/> Seleccionar todo Para los elementos que están marcados: <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar <input type="checkbox"/> Exportar		
<input type="checkbox"/> Mostrar todo   Número de filas: 25   Filtrar filas: <input type="text" value="Buscar en esta tabla"/>		

- Filtrar datos por una condición:

```
1 SELECT * FROM usuarios WHERE edad > 18;
```

	id	nombre	apellido	edad	email	fecha	administrador
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1	Daniel	Rondon	27	daniel@email.com	2023-04-05 12:30:45	0
<input type="checkbox"/> Seleccionar todo Para los elementos que están marcados: <input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar <input type="checkbox"/> Exportar							
<input type="checkbox"/> Mostrar todo   Número de filas: 25   Filtrar filas: <input type="text" value="Buscar en esta tabla"/>							

- Ordenar datos en orden ascendente(ASC) o descendente(DESC):

```
1 SELECT * FROM usuarios ORDER BY edad ASC;
```

				id	nombre	apellido	edad	email	fecha	administrador
<input type="checkbox"/>	Editar	Copiar	Borrar	2	Manuel	Durán	17	mduran@correo.com	2023-04-06 01:23:05	1
<input type="checkbox"/>	Editar	Copiar	Borrar	1	Daniel	Rondon	27	daniel@email.com	2023-04-05 12:30:45	0

☐ Seleccionar todo Para los elementos que están marcados: ☐ Editar ☐ Copiar ☐ Borrar ☐ Exportar

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

- Actualizar un registro en una tabla:

```
1 UPDATE usuarios SET nombre = 'Diego' WHERE id = 1;
```

✓ 1 fila afectada. (La consulta tardó 0,0004 segundos.)

```
UPDATE usuarios SET nombre = 'Diego' WHERE id = 1;
```

				id	nombre	apellido	edad	email	fecha	administrador
<input type="checkbox"/>	Editar	Copiar	Borrar	1	Diego	Rondon	27	daniel@email.com	2023-04-05 12:30:45	0
<input type="checkbox"/>	Editar	Copiar	Borrar	2	Manuel	Durán	17	mduran@correo.com	2023-04-06 01:23:05	1

☐ Seleccionar todo Para los elementos que están marcados: ☐ Editar ☐ Copiar ☐ Borrar ☐ Exportar

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

- Eliminar un registro en una tabla:

```
1 DELETE FROM usuarios WHERE id = 1;
```

✓ 1 fila afectada. (La consulta tardó 0,0011 segundos.)

```
DELETE FROM usuarios WHERE id = 1;
```

				id	nombre	apellido	edad	email	fecha	administrador
<input type="checkbox"/>	Editar	Copiar	Borrar	2	Manuel	Durán	17	mduran@correo.com	2023-04-06 01:23:05	1

☐ Seleccionar todo Para los elementos que están marcados: ☐ Editar ☐ Copiar ☐ Borrar ☐ Exportar

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

- Unir dos o más tablas:

```
1 SELECT * FROM usuarios JOIN ventas ON usuarios.email = ventas.email;
```

	id	producto	precio	email	fecha	concretado
<input type="checkbox"/>	2	Tv Smart	399000	mduran@correo.com	2023-02-01 06:45:25	1

☐ Seleccionar todo
 Para los elementos que están marcados:
 ☐ Editar
 ☐ Copiar
 ☐ Borrar
 ☐ Exportar

Tabla - ventas

	id	nombre	apellido	edad	email	fecha	administrador
<input type="checkbox"/>	2	Manuel	Durán	17	mduran@correo.com	2023-04-06 01:23:05	1

☐ Seleccionar todo
 Para los elementos que están marcados:
 ☐ Editar
 ☐ Copiar
 ☐ Borrar
 ☐ Exportar

☐ Mostrar todo
 Número de filas: 25
 Filtrar filas:

Tabla - usuarios

id	nombre	apellido	edad	email	fecha	administrador	id	producto	precio	email	fecha	concretado
2	Manuel	Durán	17	mduran@correo.com	2023-04-06 01:23:05	1	2	Tv Smart	399000	mduran@correo.com	2023-02-01 06:45:25	1

Unión de tablas (solo para lectura)

- Establecer llave primaria

Ejecutar la(s) consulta(s) SQL en la base de datos nueva\_bd:

```

1 CREATE TABLE usuarios (
2   id INT NOT NULL AUTO_INCREMENT,
3   nombre CHAR(15),
4   apellido CHAR(15),
5   edad INT(3) UNSIGNED CHECK (edad <= 119),
6   email VARCHAR(50),
7   fecha DATETIME,
8   administrador BOOLEAN,
9   PRIMARY KEY (id)
10 );
    
```

- Establecer una llave foránea

```

1 CREATE TABLE ventas (
2   id INT NOT NULL AUTO_INCREMENT,
3   producto VARCHAR(80),
4   precio INT(7) UNSIGNED CHECK (precio <= 1000000),
5   email VARCHAR(50),
6   fecha DATETIME,
7   concretado BOOLEAN,
8   PRIMARY KEY(id),
9   FOREIGN KEY(id) REFERENCES usuarios(id)
10 );
    
```

## Condicionales

Los condicionales se utilizan en las consultas para filtrar los resultados según un conjunto de criterios especificados.

Hay varios tipos de condicionales que se pueden utilizar, estos son:

- **WHERE:** Se utiliza para filtrar los registros de una tabla según una o más condiciones especificadas.

```
1 SELECT * FROM usuarios WHERE edad > 18;
```

- **AND:** Se utiliza para combinar dos o más condiciones y sólo devuelve las filas que cumplen todas las condiciones.

```
1 SELECT * FROM `usuarios` WHERE edad > 16 AND nombre = 'Manuel';
```

- **OR:** Se utiliza para combinar dos o más condiciones y devuelve las filas que cumplen al menos una de las condiciones.

```
1 SELECT * FROM `usuarios` WHERE edad > 18 OR administrador = true;
```

- **BETWEEN:** Se utiliza para filtrar los registros cuyo valor se encuentra dentro de un rango determinado.

```
1 SELECT * FROM `usuarios` WHERE edad BETWEEN 15 AND 18;
```

- **IN:** Se utiliza para filtrar los registros cuyo valor se encuentra en una lista de valores especificados.

```
1 SELECT * FROM `usuarios` WHERE edad IN (15,16,17,18,19);
```

- **LIKE:** Se utiliza para buscar patrones en una columna de tipo cadena.

```
1 SELECT * FROM ventas WHERE producto LIKE 'TV Smart';
```

- **IS NULL / IS NOT NULL:** Se utiliza para buscar filas con valores nulos o no nulos.

```
1 SELECT * FROM ventas WHERE producto IS NULL;  
2 SELECT * FROM email WHERE email IS NOT NULL;
```