

Definiendo cómo es almacenada la información.

- **CREATE DATABASE** se utiliza para crear una nueva base de datos vacía.
- **DROP DATABASE** se utiliza para eliminar completamente una base de datos existente.
- **CREATE TABLE** se utiliza para crear una nueva tabla, donde la información se almacena realmente.
- **ALTER TABLE** se utiliza para modificar una tabla ya existente.
- **DROP TABLE** se utiliza para eliminar por completo una tabla existente.
- **TRUNCATE TABLE** elimina los datos de la tabla (menos los nombres de las columnas).

Manipulando los datos.

- **SELECT** se utiliza cuando quieres leer (o seleccionar) tus datos.
- **INSERT** se utiliza cuando quieres añadir (o insertar) nuevos datos.
- **UPDATE** se utiliza cuando quieres cambiar (o actualizar) datos existentes.
- **DELETE** se utiliza cuando quieres eliminar (o borrar) datos existentes.
- **REPLACE** se utiliza cuando quieres añadir o cambiar (o reemplazar) datos nuevos o ya existentes.
- **TRUNCATE** se utiliza cuando quieres vaciar (o borrar) todos los datos de la plantilla.

Debemos tener instalado XAMPP en nuestros equipos.

Una vez instalado y corriendo el servicio de MySQL en el equipo, podemos abrir el CMD y ejecutar las siguientes instrucciones:

```
cd C:\xampp\mysql\bin  
mysql.exe -h localhost -u root -p
```

Cuando nos pide contraseña, damos a Intro, normalmente root no tiene contraseña.

Instrucciones:

Instrucción para devolver o mostrar las Bases de Datos que tengo creadas:

```
show databases;
```

```
MariaDB [(none)]> show databases;  
+-----+  
| Database |  
+-----+  
| bddlibreria |  
| hospital |  
| information_schema |  
| mp_hospital |  
| mysql |  
| nueva-bbdd |  
| performance_schema |  
| phpmyadmin |  
| test |  
+-----+  
9 rows in set (0.001 sec)
```

Crear una nueva base de datos

```
CREATE DATABASE mydb;
```

Seleccione la base de datos para trabajar

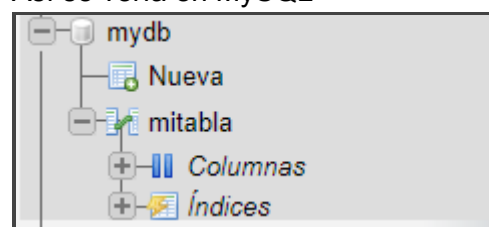
```
USE mydb;
```

Crear una tabla nueva con atributos y características

```
;
```

```
CREATE TABLE clientes ( id INT auto_increment PRIMARY KEY,  
id_cliente int(3), Nombre varchar(20), Apellidos varchar(50),  
direccion varchar(50));
```

Así se vería en MySQL



Modificar tablas

- Añadir columna a una tabla:

```
ALTER TABLE `members` ADD COLUMN `credit_card_number` VARCHAR(25);
```

- Borrar columna de una tabla:

```
ALTER TABLE `members` DROP COLUMN `credit_card_number`;
```

- Modificar atributos de una columna de una tabla:

```
ALTER TABLE `members` MODIFY `fullname` char (50) NOT NULL;
```

- Cambiar nombre de una columna de una tabla:

```
ALTER TABLE tabla1 CHANGE id id_alumno INT NOT NULL;
```

Modificar el valor de la columna con propiedad autoincrement para que comiencen por el valor 15000:

```
ALTER TABLE nombre_tabla AUTO_INCREMENT=15000;
```

Más ejemplos

```
ALTER TABLE vendors
```

```
ADD COLUMN email VARCHAR(100) NOT NULL,
```

```
ADD COLUMN hourly_rate decimal(10,2) NOT NULL;
```

```
ALTER TABLE t2 ADD c INT UNSIGNED NOT NULL AUTO_INCREMENT,  
ADD PRIMARY KEY (c);
```

Como mover una columna de posición:

```
alter table Empleados modify column Apellido varchar() after  
Nombre;
```

Renombrar tabla

```
RENAME TABLE `current_table_name` TO `new_table_name`;
```

Mostrar columnas/atributos de una tabla

```
SHOW COLUMNS FROM YOUR_TABLE;
```

Modificar atributos de una columna ya creada (aunque se cambie un solo atributo hay que poner en la sentencia todos los atributos)

```
ALTER TABLE mitabla MODIFY nombreColumna VARCHAR(50) NOT NULL;
```

Insertar nuevos registros en la tabla

```
INSERT INTO mitabla VALUES ( 1, 'Will' );
```

Si todo va nos devuelve este mensaje

```
Query OK, 0 rows affected (0.016 sec)
```

Inserto más

```
INSERT INTO mitabla VALUES ( 2, 'Marry' );
```

```
INSERT INTO mitabla VALUES ( 3, 'Dean' );
```

Insertar nuevos registros en una tabla con una sola línea de comando

```
INSERT INTO mitabla VALUES (1, 'Will'), ( 2, 'Marry' ), (3,  
'Dean');
```

Así se vería en Workbench

The screenshot shows the MySQL Workbench interface. On the left, the 'Information' tab displays the table structure for 'mitabla':

Table:	mitabla
Columns:	
id	int(11) PK
nombre	varchar(20)

On the right, the 'Result Grid' tab shows the data in the table:

	id	nombre
▶	1	Will
	2	Marry
	3	Dean
*	NULL	NULL

Below the grid, there is a tab labeled 'mitabla 1' and an 'Output' section.

[SELECT](#) se utiliza cuando quieres leer (o seleccionar) tus datos.

```
SELECT id, nombre FROM mitabla WHERE id = 1;
```

The terminal shows the result of the query:

id	nombre
1	Will

[UPDATE](#) se utiliza cuando quieres cambiar (o actualizar) datos existentes.

```
UPDATE mitabla SET nombre = 'Willy' WHERE id = 1;
```

Query OK, 0 rows affected (0.000 sec)

Rows matched: 0 Changed: 0 Warnings: 0

```
SELECT id, nombre FROM mitabla;
```

The terminal shows the result of the query:

id	nombre
1	Willy
2	Marry
3	Dean

Para ver todos los registros de una tabla:

```
SELECT * FROM mitabla;
```

[DELETE](#) se utiliza cuando quieres eliminar (o borrar) datos existentes.

```
DELETE FROM mitabla WHERE id = 1;  
Query OK, 1 row affected (0.003 sec)
```

id	nombre
2	Marry
3	Dean

```
SELECT id, nombre FROM mitabla;
```

[DELETE CON ALIAS](#)

```
delete <alias> from <table> <alias> where <alias>.<field>
```

```
-- delete from both tables  
-- this can be used as well:
```

```
DELETE t, p  
t.*, p.* FROM topics AS t  
JOIN posts AS p  
ON t.id = p.topic_id WHERE t.id = 5599 ;
```

ELIMINAR DUPLICADOS

```
delete p2 from Person p1, Person p2 where p1.Email = p2.Email  
AND p2.ID > p1.ID;
```

[DROP DATABASE](#) se utiliza para eliminar completamente una base de datos existente.

```
DROP DATABASE mydb;
```

```
Query OK, 1 row affected (0.018 sec)
```

```
SELECT count(1) from mitabla; da el número de registros en la tabla
```

```
ERROR 1046 (3D000): No database selected
```

Como se ha eliminado la base de datos el comando SELECT no devuelve nada

ANEXO RESUMEN

- **Mostrar bases**

```
show databases;
```

- **Mostrar tablas**

```
show tables;
```

- **Mostrar atributos tabla**

```
show columns from nombretabla;
```

```
describe alumnos;
```

Field	Type	Null	Key	Default	Extra
id	int(20)	NO	PRI	NULL	auto_increment
nombre	varchar(20)	NO		NULL	
apellido	varchar(20)	NO		NULL	
sexo	enum('F','M')	NO		NULL	
edad	int(2)	NO		NULL	
ciudad	varchar(40)	YES		NULL	
pcia	varchar(4)	YES		NULL	

- **Conecta**

```
USE nombrebd;
```

- **Modificar tabla**

```
ALTER TABLE nombretabla ADDCOLUMN "DNI" varchar (10);
```

- **Creo Tabla**

```
CREATE TABLE nombretabla (id INT PRIMARY KEY, nombre  
VARCHAR(20));
```

- **Insertar Tabla**

```
INSERT INTO nombretabla VALUES (1,"luis");  
INSERT INTO nombretable (nombre) VALUES ('Luis');
```

- **Consulta Registros Tabla**

```
SELECT FROM nombretabla;
```

- **Modificar**

```
update nombretabla SET nombre="Juan" WHERE" id=1;
```

- **Eliminar Registro**

```
DELETE FROM nombretabla WHEREid=1;
```

- **Eliminar todos los registros de la tabla** (menos los nombres de las columnas)

```
TRUNCATE TABLE nombreBD.tabla;
```

- **Eliminar Tabla**

```
DROP TABLE nombretabla;
```

- **Eliminar Base**

```
DROP DATABASE nombrebdd;
```