

# Creación de Bases de Datos mediante Archivos SQL

---

## 1. Generar la estructura de la base de datos

### 1.1 Usando herramientas gráficas

Herramientas como **phpMyAdmin**, **MySQL Workbench** o **DBeaver** permiten exportar fácilmente la estructura de una base de datos.

- **phpMyAdmin:**

1. Accede a tu base de datos.
2. Ve a la pestaña **Exportar**.
3. Selecciona el formato **SQL**.
4. Escoge la opción **Estructura** (para no incluir datos).
5. Guarda el archivo.

- **MySQL Workbench:**

1. Ve a **Server > Data Export**.
2. Selecciona la base de datos.
3. Marca la casilla **Dump Structure Only**.
4. Exporta el archivo.

### 1.2 Usando la línea de comandos

Si prefieres usar la línea de comandos, puedes emplear **mysqldump** o **mariadb-dump**.

- **Estructura sin datos:**

```
mysqldump -u usuario -p --no-data nombre_de_base_de_datos > estructura.sql
```

- **Estructura con la instrucción **CREATE DATABASE**:**

```
mysqldump -u usuario -p --no-data --databases nombre_de_base_de_datos > estructura.sql
```

Este archivo incluirá algo como:

```
CREATE DATABASE `nombre_de_base_de_datos`;  
USE `nombre_de_base_de_datos`;  
  
CREATE TABLE `tabla1` (...);
```

## 1.3 Escribiendo la estructura manualmente

Si necesitas mayor control, puedes crear un archivo SQL manualmente.

Ejemplo:

```
CREATE DATABASE IF NOT EXISTS nombre_de_base_de_datos;
USE nombre_de_base_de_datos;

CREATE TABLE empleados (
  id INT PRIMARY KEY AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL,
  salario DECIMAL(10, 2) NOT NULL
);

CREATE TABLE departamentos (
  id INT PRIMARY KEY AUTO_INCREMENT,
  nombre VARCHAR(100) NOT NULL
);
```

---

## 2. Generar un archivo SQL con datos

Para exportar los datos de las tablas sin la estructura:

```
mysqldump -u usuario -p --no-create-info nombre_de_base_de_datos > datos.sql
```

Este archivo incluirá únicamente las instrucciones `INSERT INTO` para insertar los datos en las tablas.

---

## 3. Restauración completa: Estructura + Datos

### 3.1 Generar un archivo combinado

Si necesitas un archivo que incluya tanto la estructura como los datos:

```
mysqldump -u usuario -p nombre_de_base_de_datos > backup_completo.sql
```

Este archivo contendrá:

- Instrucciones `CREATE DATABASE` (si se usó `--databases`).
- `CREATE TABLE`.
- `INSERT INTO`.

### 3.2 Restaurar desde el archivo combinado

### Caso 1: El archivo incluye **CREATE DATABASE**

Si el archivo contiene la instrucción **CREATE DATABASE**, puedes restaurarlo con:

```
mysql -u usuario -p < backup_completo.sql
```

### Caso 2: El archivo NO incluye **CREATE DATABASE**

En este caso, deberás crear la base de datos antes de restaurar:

1. Accede al cliente MySQL:

```
mysql -u usuario -p
```

2. Crea la base de datos y usa el archivo para restaurarla:

```
CREATE DATABASE nombre_de_base_de_datos;  
USE nombre_de_base_de_datos;  
SOURCE /ruta/al/backup_completo.sql;
```

---

## 4. Cargar datos desde un archivo CSV

Si los datos están en un archivo CSV, puedes usar **LOAD DATA INFILE**.

### Ejemplo

Archivo **empleados.csv**:

```
1,Juan Perez,25000.50  
2,Ana López,30000.00  
3,Carlos Ruiz,28000.75
```

Comando para cargar los datos en la tabla **empleados**:

```
LOAD DATA INFILE '/ruta/empleados.csv'  
INTO TABLE empleados  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\n';
```

---

## 5. Buenas prácticas

### 1. Separar estructura y datos:

- Genera un archivo para la estructura (`estructura.sql`) y otro para los datos (`datos.sql`) para mayor flexibilidad.

### 2. Verifica permisos:

- El usuario que ejecuta los comandos debe tener permisos para crear bases de datos y tablas, así como para insertar datos.

### 3. Controla conflictos:

- Antes de cargar datos, verifica que las tablas tengan la estructura correcta y que no existan restricciones como claves únicas que generen errores.

### 4. Automatiza el proceso:

- Usa scripts o herramientas para automatizar la creación de bases de datos y la carga de datos.

---

Con esta guía, puedes crear y gestionar bases de datos mediante archivos SQL de manera efectiva.