

# Scripts SQL - Usos y Aplicaciones

## Scripts SQL

Los scripts SQL son una serie de comandos o un programa que se ejecutará en un servidor SQL.

Los scripts SQL son útiles para realizar cambios complejos en la base de datos y se pueden utilizar para crear, modificar o eliminar objetos de la base de datos, como tablas, vistas, procedimientos almacenados y funciones.

## Aplicaciones de Scripts SQL

Aquí hay algunas de las cosas que puedes hacer con scripts SQL:

- **Crear tablas**  
Puedes usar scripts SQL para crear nuevas tablas en tu base de datos. Esto es útil cuando necesitas agregar nueva funcionalidad a tu aplicación o cuando deseas almacenar nuevos tipos de datos.
- **Eliminar tablas**  
Los scripts SQL a menudo tienen comandos para eliminar tablas de las bases de datos. Esto es especialmente importante antes de los comandos de creación de tablas para asegurarte de que no exista ya una tabla con el mismo nombre en la base de datos.
- **Insertar datos**  
Los scripts SQL también se pueden utilizar para insertar datos en tus tablas. Esto es útil cuando necesitas poblar tu base de datos con datos de prueba o cuando deseas importar datos de una fuente externa.
- **Actualizar datos**  
Puedes usar scripts SQL para actualizar datos existentes en tus tablas. Esto es útil cuando necesitas corregir errores o actualizar registros según los cambios en los requisitos del negocio.
- **Eliminar datos**  
Los scripts SQL también se pueden utilizar para eliminar datos de tus tablas. Esto es útil cuando necesitas eliminar registros antiguos u obsoletos de tu base de datos.
- **Crear vistas**  
Las vistas son tablas virtuales que te permiten consultar datos de múltiples tablas como si fueran una sola tabla. Puedes usar scripts SQL para crear vistas que simplifiquen consultas complejas y faciliten el trabajo con tus datos.
- **Crear procedimientos almacenados**  
Los procedimientos almacenados son declaraciones SQL precompiladas que se pueden ejecutar bajo demanda. Puedes usar scripts SQL para crear procedimientos almacenados que encapsulen lógica de negocio compleja y faciliten la gestión de tu base de datos.
- **Crear disparadores**  
Los disparadores son tipos especiales de procedimientos almacenados que se ejecutan automáticamente en respuesta a ciertos eventos, como una operación de inserción, actualización o eliminación. Puedes usar scripts SQL para crear disparadores que apliquen reglas de negocio y mantengan la integridad de los datos.

## Ejemplo: Creación de Tablas

Ejecutemos un script que contenga los comandos CREATE TABLE para todas las tablas en un conjunto de datos dado, en lugar de crear cada tabla manualmente escribiendo los comandos DDL en el editor SQL.

Tenga en cuenta los siguientes puntos sobre estos scripts.

1. Los scripts SQL son básicamente un conjunto de comandos SQL compilados en un solo archivo.
2. Cada comando debe terminarse con un delimitador o terminador. A menudo, el delimitador predeterminado es un punto y coma ;.
3. Se recomienda mantener la extensión del archivo como .sql.
4. Al importar este archivo en la interfaz de phpMyAdmin, los comandos en el archivo se ejecutan secuencialmente.

Considere el siguiente script

```
DROP TABLE IF EXISTS PATIENTS;
DROP TABLE IF EXISTS MEDICAL_HISTORY;
DROP TABLE IF EXISTS MEDICAL_PROCEDURES;
DROP TABLE IF EXISTS MEDICAL_DEPARTMENTS;
DROP TABLE IF EXISTS MEDICAL_LOCATIONS;
CREATE TABLE PATIENTS (
  PATIENT_ID CHAR(9) NOT NULL,
  FIRST_NAME VARCHAR(15) NOT NULL,
  LAST_NAME VARCHAR(15) NOT NULL,
  SSN CHAR(9),
  BIRTH_DATE DATE,
  SEX CHAR,
  ADDRESS VARCHAR(30),
  DEPT_ID CHAR(9) NOT NULL,
  PRIMARY KEY (PATIENT_ID)
);
CREATE TABLE MEDICAL_HISTORY (
  MEDICAL_HISTORY_ID CHAR(9) NOT NULL,
  PATIENT_ID CHAR(9) NOT NULL,
  DIAGNOSIS_DATE DATE,
  DIAGNOSIS_CODE VARCHAR(10),
  MEDICAL_CONDITION VARCHAR(100),
  DEPT_ID CHAR(9),
  PRIMARY KEY (MEDICAL_HISTORY_ID)
);
CREATE TABLE MEDICAL_PROCEDURES (
  PROCEDURE_ID CHAR(9) NOT NULL,
  PROCEDURE_NAME VARCHAR(30),
  PROCEDURE_DATE DATE,
  PATIENT_ID CHAR(9) NOT NULL,
  DEPT_ID CHAR(9),
```

```

PRIMARY KEY (PROCEDURE_ID)
);
CREATE TABLE MEDICAL_DEPARTMENTS (
DEPT_ID CHAR(9) NOT NULL,
DEPT_NAME VARCHAR(15),
MANAGER_ID CHAR(9),
LOCATION_ID CHAR(9),
PRIMARY KEY (DEPT_ID)
);
CREATE TABLE MEDICAL_LOCATIONS (
LOCATION_ID CHAR(9) NOT NULL,
DEPT_ID CHAR(9) NOT NULL,
LOCATION_NAME VARCHAR(50),
PRIMARY KEY (LOCATION_ID, DEPT_ID)
);

```

Este script incorpora comandos para primero eliminar cualquier tabla con los nombres mencionados en la base de datos. Después de eso, el script contiene comandos para crear 5 tablas diferentes. Todos estos comandos se ejecutan secuencialmente en la interfaz.

El contenido de este archivo se puede guardar en un formato de archivo .sql y ejecutarse en la interfaz de phpMyAdmin. Esto se puede hacer seleccionando primero la base de datos, subiendo el script SQL en el espacio proporcionado y ejecutándolo, como se muestra en la imagen a continuación.

Server: mysql:3306 » Database: CVD

Structure SQL Search Query Export Import Operations Privileges Routines Events

## Importing into the database "CVD"

**File to import:**

File may be compressed (gzip, bzip2, zip) or uncompressed.  
A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: Choose File CVD\_Datab...es\_Script.sql (Max: 2,048KiB)

You may also drag and drop a file on any page.

Character set of the file: utf-8

**Partial import:**

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it ca*

Skip this number of queries (for SQL) starting from the first one: 0

**Other options:**

☒ Enable foreign key checks

**Format:**

SQL

**Format-specific options:**

SQL compatibility mode: NONE

☒ Do not use AUTO\_INCREMENT for zero values

Tras la ejecución exitosa de cada declaración en secuencia, aparece una nota en la interfaz, como se muestra en la imagen a continuación. También es prudente notar que las tablas creadas son ahora visibles en la estructura de árbol a la izquierda bajo la base de datos seleccionada.

phpMyAdmin

Recent Favorites

- New
- CVD
  - New
  - MEDICAL\_DEPARTMENTS
  - MEDICAL\_HISTORY
  - MEDICAL\_LOCATIONS
  - MEDICAL\_PROCEDURES
  - PATIENTS
- information\_schema
- mysql
- performance\_schema
- sys

Server: mysql:3306 » Database: CVD

Structure SQL Search Query Export Import Operations Privileges Routines Events

✓ Import has been successfully finished, 15 queries executed. (CVD\_Database\_Create\_Tables\_Script.sql)

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0065 seconds.)

DROP TABLE IF EXISTS PATIENTS

[Edit inline] [Edit] [Close]

⚠ Note: #1051 Unknown table 'CVD.PATIENTS'

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0038 seconds.)

DROP TABLE IF EXISTS MEDICAL\_HISTORY

[Edit inline] [Edit] [Close]

⚠ Note: #1051 Unknown table 'CVD.MEDICAL\_HISTORY'

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0038 seconds.)

DROP TABLE IF EXISTS MEDICAL\_PROCEDURES

[Edit inline] [Edit] [Close]

⚠ Note: #1051 Unknown table 'CVD.MEDICAL\_PROCEDURES'

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0044 seconds.)

DROP TABLE IF EXISTS MEDICAL\_DEPARTMENTS

[Edit inline] [Edit] [Close]

⚠ Note: #1051 Unknown table 'CVD.MEDICAL\_DEPARTMENTS'

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0161 seconds.)

DROP TABLE IF EXISTS MEDICAL\_LOCATIONS

[Edit inline] [Edit] [Close]

Console

Puedes hacer clic en cualquiera de las tablas para ver su Definición de Tabla (su lista de columnas, tipos de datos, etc.). La imagen a continuación muestra la estructura de la tabla PACIENTES.

phpMyAdmin

Recent Favorites

- New
- CVD
  - New
  - MEDICAL\_DEPARTMENTS
  - MEDICAL\_HISTORY
  - MEDICAL\_LOCATIONS
  - MEDICAL\_PROCEDURES
  - PATIENTS
- information\_schema
- mysql
- performance\_schema
- sys

Server: mysql:3306 » Database: CVD » Table: PATIENTS

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	PATIENT_ID	char(9)	utf8mb4_0900_ai_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 2	FIRST_NAME	varchar(15)	utf8mb4_0900_ai_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 3	LAST_NAME	varchar(15)	utf8mb4_0900_ai_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 4	SSN	char(9)	utf8mb4_0900_ai_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 5	BIRTH_DATE	date			Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 6	SEX	char(1)	utf8mb4_0900_ai_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 7	ADDRESS	varchar(30)	utf8mb4_0900_ai_ci		Yes	NULL			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>
<input type="checkbox"/> 8	DEPT_ID	char(9)	utf8mb4_0900_ai_ci		No	None			<a href="#">Change</a> <a href="#">Drop</a> <a href="#">More</a>

☐ Check all
 With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#)

**Autor(es)**

Abhishek Gagneja



**Skills Network**