

## Comandos SQL:

<b>CREATE TABLE;</b>	Crea una tabla nueva.
<b>DROP TABLE;</b>	Elimina tabla. Comando peligroso. No se puede eliminar una tabla si está vinculada a otra.
<b>SHOW TABLES;</b>	Lista todas las tablas.
<b>CREATE DATABASE;</b>	Crea una nueva base de datos.
<b>USE DATABASE;</b>	Selecciona una base de datos.

- ★ Útil para corregir errores. Añadir al principio del archivo .sql (que almacena consultas para crear una base de datos) para que a cada **Run on active connection** se eliminen las tablas existentes y se vuelvan a crear:

```
DROP DATABASE IF EXISTS nombredelabase;  
CREATE DATABASE nombredelabase;  
USE nombredelabase;
```

- ★ Las claves foráneas en una tabla se añaden al final de la misma.

```
CREATE TABLE Pedidos (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    FECHA DATE,  
    TOTAL DECIMAL(10,2),  
    CLIENTE_ID INT,  
    FOREIGN KEY (CLIENTE_ID) REFERENCES Cliente(ID)  
);
```

- ★ Un campo no es lo mismo que una clave. Los campos son visibles, las claves indican de qué manera la base de datos interactúa con los datos. Un campo puede ser una clave, pero una clave no tiene por qué ser un campo.

Un campo de ID en una tabla intermedia como clave primaria, sería una columna visualizable. Una clave primaria compuesta, no.

Lo más recomendable es que la tabla tenga su propio identificador, en vez de uno compuesto.

→ **Tabla intermedia con ID:**

```
CREATE TABLE Incripcion (  
    ID INT AUTO_INCREMENT PRIMARY KEY, → Clave primaria  
    única  
    ESTUDIANTE_ID INT,  
    CURSO_ID INT,  
    FECHA_INSCRIPCION DATE,  
    FOREIGN KEY (ESTUDIANTE_ID) REFERENCES  
    Estudiante(ID),  
    FOREIGN KEY (CURSO_ID) REFERENCES Curso(ID)  
);
```

→ **Tabla intermedia sin ID y clave primaria compuesta:**

```
CREATE TABLE Incripcion (  
    ESTUDIANTE_ID INT,  
    CURSO_ID INT,  
    FECHA_INSCRIPCION DATE,  
    PRIMARY KEY (ESTUDIANTE_ID, CURSO_ID), → Clave primaria  
    compuesta  
    FOREIGN KEY (ESTUDIANTE_ID) REFERENCES  
    Estudiante(ID),  
    FOREIGN KEY (CURSO_ID) REFERENCES Curso(ID)  
);
```

- ★ No es necesario insertar los datos de una fila que contenga AUTO\_INCREMENT. A menos que se trate de insertar valores de claves foráneas en una tabla intermedia.

```
CREATE TABLE Department (  
    ID INT AUTO_INCREMENT PRIMARY KEY,  
    DEPARTMENT_NAME VARCHAR(50) NOT NULL  
);
```

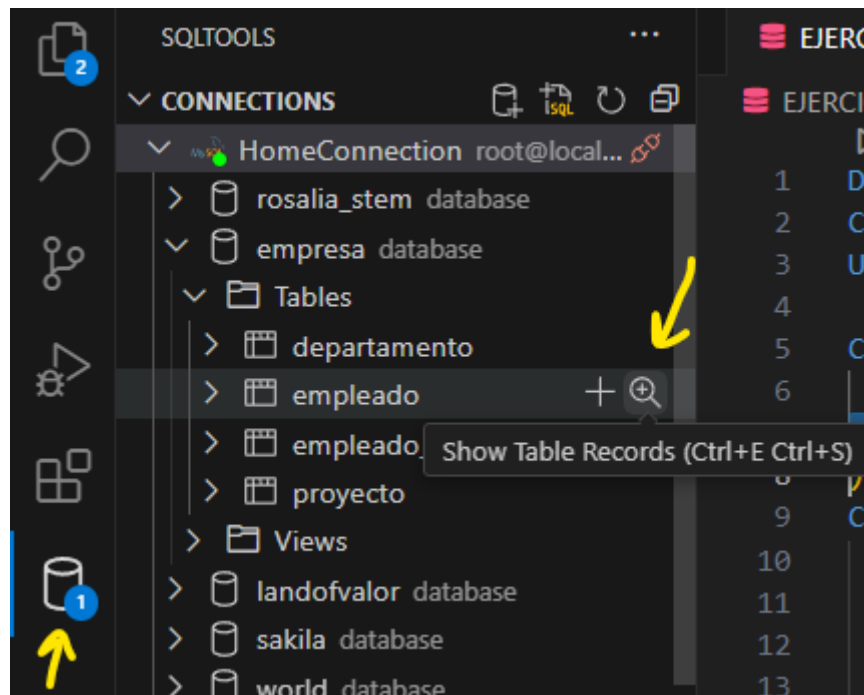
```
INSERT INTO Department (DEPARTMENT_NAME)  
VALUES  
    ('Engineering'),  
    ('Product'),  
    ('QA'),  
    ('Sales');
```

- ★ El orden de las tablas afecta. Si hay una tabla con claves foráneas que hace referencia a una tabla creada debajo, dará error.

El enunciado para el ejercicio de EmpresaLogistica.sql lista “Empleado” antes que “Departamento”. Sin embargo, la tabla “Departamento” se tiene que crear antes que la tabla “Empleado”, ya que “Empleado” está vinculado con “Departamento” a través de una clave foránea. Si “Departamento” no existe, “Empleado” dará error al no existir la tabla “Departamento” a la que quiere hacer referencia.

```
EmpresaLogistica.sql
Run on active connection | Select block
1 DROP DATABASE IF EXISTS EmpresaLogistica;
2 CREATE DATABASE EmpresaLogistica;
3 USE EmpresaLogistica;
4
5
6 CREATE TABLE Departamento (
7     id_departamento INT AUTO_INCREMENT PRIMARY KEY,
8     nombre_departamento VARCHAR(50) NOT NULL UNIQUE
9
10 );
11
12 CREATE TABLE Empleado (
13     id_empleado INT AUTO_INCREMENT PRIMARY KEY,
14     nombre VARCHAR(50) NOT NULL,
15     apellido VARCHAR(50) NOT NULL,
16     salario DECIMAL,
17     fecha_contratacion DATE,
18     id_departamento INT,
19     FOREIGN KEY (id_departamento) REFERENCES Departamento(id_d
20
21 );
22
23
24 CREATE TABLE Producto (
25     id_producto INT AUTO_INCREMENT PRIMARY KEY,
26     nombre_producto VARCHAR(50) NOT NULL,
27     precio DECIMAL,
28     stock DECIMAL DEFAULT 0,
29     fecha_adicion DATE
30
31 );
32
33 CREATE TABLE Cliente (
34     id_cliente INT AUTO_INCREMENT PRIMARY KEY,
35     nombre_cliente VARCHAR(50) NOT NULL,
36     correo VARCHAR(50) UNIQUE,
37     telefono CHAR(15)
38
39 );
```

- ★ Para verificar que las tablas se han creado correctamente: **Run on active connection**.
- ★ Para verificar que la introducción de datos (INSERT INTO) se ha aplicado correctamente:



- ★ Para verificar las consultas con SELECT: **Run on active connection** en el archivo .sql específico para este propósito.