

MP02-A: BASES DE DATOS
UNIDAD FORMATIVA 2
LENGUAJES SQL:
DML Y DDL



Lenguajes de la BBDD

DML Data Manipulation Language

DDL Data Definition Language

DCL Data Control Language

TCL Transaction Control Language

Lenguajes de la BBDD

Servidor



Cliente



Data Definition Language

CREATE DATABASE;
SHOW DATABASES;
USE prueba;

Para crear una base de datos se usa una sentencia **CREATE DATABASE**:

mysql>CREATE DATABASE prueba;
Query OK, 1 row affected (0.03 sec)

Ejemplo real.

Para saber cuantas bases de datos existen en nuestro sistema usamos la sentencia **SHOW DATABASES**:

```
mysql>SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| mysql | } Ue de BD qe hay en  
| prueba | el servidor.  
| test |  
+-----+  
3 rows in set (0.00 sec)
```

Paso 1
Sirve para ver cuantas base de datos hay en el sistema.

Paso 2
Para usar una base de datos en particular:

mysql>USE prueba;
Database changed

Aunque hayamos creado la base de datos luego tenemos qe visualizarla.

Data Definition Language

CREATE TABLE ...

Veamos ahora la sentencia **CREATE TABLE** que sirve para crear tablas.

La sintaxis de esta sentencia es muy compleja, pero empezaremos con ejemplos sencillos. En su forma mas simple, la sentencia **CREATE TABLE** creara una tabla con las columnas que indiquemos. Crearemos, como ejemplo, una tabla que nos permitirá almacenar nombres de personas y sus fechas de nacimiento. Deberemos indicar el nombre de la tabla y los nombres y tipos de las columnas:

```
mysql> USE prueba  
Database changed  
mysql>CREATE TABLE gente (nombre VARCHAR(40), fecha DATE);  
Query OK, 0 rows affected (0.53 sec)
```

Crea la tabla en la BD elegida.

nombre tabl*a*

nombre *columna de texto*

fecha *columna de fecha*

fecha DATE *especificar que tipo de dato es cada atributo*

Data Definition Language

SHOW TABLES;

Podemos consultar cuantas tablas y que nombres tienen en una base de datos, usando la sentencia **SHOW TABLES**:

```
mysql>SHOW TABLES;  
+-----+  
| Tables_in_prueba |  
+-----+  
| gente |  
+-----+  
1 row in set (0.01 sec)
```

Igual
que el
Show database
pero muestra solo
las tablas de la
base de datos que tenemos
elijido

Data Definition Language

DROP TABLE ...

Para borrar una tabla usamos **DROP TABLE** de manera sencilla:

```
mysql> show tables;  
+-----+  
| Tables_in_mundo | - nombre de la BD  
+-----+  
| lista |  
| naciones |  
| prueba |  
| telef |  
+-----+  
4 rows in set (0.00 sec)
```

Annotations:
- A yellow bracket groups the table names 'lista', 'naciones', 'prueba', and 'telef'.
- A yellow note next to the bracket says: 'tablas que hay en esa BD.'

Borrar tablas

```
mysql> DROP TABLE telef;  
Query OK, 0 rows affected (0.32 sec)
```

¿Qué nos devolverá la siguiente sentencia?

```
mysql> show tables;
```

Data Definition Language

DROP TABLE ...

Para borrar una tabla usamos **DROP TABLE** de manera sencilla:

```
mysql> show tables;
+-----+
| Tables_in_mundo |
+-----+
| lista |
| naciones |
| prueba |
| telef |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> DROP TABLE telef;
Query OK, 0 rows affected (0.32 sec)
mysql> show tables;
+-----+
| Tables_in_mundo |
+-----+
| lista |
| naciones |
| prueba |
+-----+
3 rows in set (0.00 sec)
Query OK, 0 rows affected (0.08 sec)
```

} tablas que hay tras
borrar la tabla telef

Data Definition Language – Modificar estructura de una tabla

ALTER TABLE ... RENAME... *cambiar nombre*
ALTER TABLE ... DROP COLUMN... *borrar columna*
ALTER TABLE ... DROP PRIMARY KEY... *elimina clave primaria*
ALTER TABLE ... ADD...
ALTER TABLE ... ADD...AFTER...
ALTER TABLE ... ADD...FIRST;
ALTER TABLE ... ADD...PRIMARY KEY...
ALTER TABLE ... MODIFY COLUMN...

modificar las tablas

Para modificar una tabla podemos usar el comando
ALTER TABLE:

Por ejemplo, si queremos cambiar el nombre de la tabla:

```
mysql>ALTER TABLE ciudad6 RENAME graciudad;  
Query OK, 0 rows affected (0.03 sec)
```

Para eliminar una columna de una tabla:

```
mysql>ALTER TABLE graciudad DROP COLUMN poblacion;  
Query OK, 0 rows affected (1.28 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

Ariberto
recientemente
introducido

Data Definition Language – Modificar estructura de una tabla

ALTER TABLE ... DROP PRIMARY KEY...

Elimina la Clave primaria de una tabla

Ejemplo

```
mysql>ALTER TABLE graciudad DROP PRIMARY KEY;  
Query OK, 1 row affected (1.20 sec)  
Records: 1 Duplicates: 0 Warnings: 0
```

Data Definition Language – Modificar estructura de una tabla

ALTER TABLE ... ADD...

Añade un atributo a la tabla
Columna

Ejemplo

```
mysql>ALTER TABLE graciudad ADD fecha date;  
Query OK, 1 row affected (1.20 sec)  
Records: 1 Duplicates: 0 Warnings: 0
```

The diagram consists of two yellow circles. The first circle contains the word 'fecha'. An arrow points from the text 'nueva columna' to this circle. The second circle contains the word 'date;'. An arrow points from the text 'Tipo de dato.' to this circle.

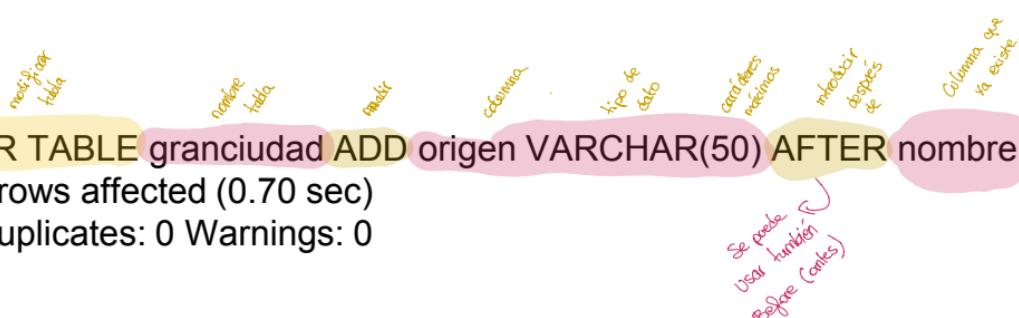
Data Definition Language – Modificar estructura de una tabla

ALTER TABLE ... ADD...AFTER

Añade un atributo a la tabla después de otro específico

Ejemplo

```
mysql>ALTER TABLE graciudad ADD origen VARCHAR(50) AFTER nombre;  
Query OK, 0 rows affected (0.70 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```



Data Definition Language – Modificar estructura de una tabla

ALTER TABLE ... ADD... FIRST

min 13:26

Añade un **atributo** a la tabla pero lo pone el primero
colombia

Ejemplo

mysql> ALTER TABLE graciudad ADD id INT FIRST;
Query OK, 0 rows affected (0.72 sec)
Records: 0 Duplicates: 0 Warnings: 0

nombre de tabla columna tipo de dato la introduce primera posición

Data Definition Language – Modificar estructura de una tabla

ALTER TABLE ... ADD PRIMARY KEY...

Añade una clave primaria a una tabla

Ejemplo

mysql>ALTER TABLE graciudad ADD PRIMARY KEY(id);

nombre de
la columna ya existente
a la que pertenece
la clave primaria

Para eliminarla sería:

alter table graciudad drop primary key;

Data Definition Language – Modificar estructura de una tabla

ALTER TABLE ... MODIFY COLUMN...

Reestructurar una columna

Ejemplo

```
mysql>ALTER TABLE graciudad MODIFY COLUMN id INT auto_increment;
```

nombre de
la columna
a modificar

la columna
ahora será
entero y auto
incremental

Será un número
entero que se irá
incrementando de uno en uno según
se vayan añadiendo
registros.

Data Definition Language – Modificar estructura de una tabla

ALTER TABLE ... auto_increment = ...

Establecer el inicio del contador de la clave primaria

Ejemplo

```
mysql> alter table graciudad auto_increment = 40;  
Query OK, 0 rows affected (0.08 sec)
```

← Decímos a partir
de qué número empezamos
a contar

Data Definition Language – Modificar estructura de una tabla

ALTER TABLE ... set insert_ID = ...

Establecer el inicio del contador de la clave primaria

Ejemplo

```
mysql> set insert_ID=50;  
Query OK, 0 rows affected (0.00 sec)
```

lo mismo que el anterior
pero de otra forma

Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

Creamos una base de datos llamada “continentes”.

Después creamos una tabla llamada “america”.

Asignamos **PRIMARY KEY**

Asignamos campos **NOT NULL**

Asignamos **DEFAULT**

También creamos una tabla llamada “ciudades”

The screenshot shows a MySQL Workbench interface with a query editor titled "videoclase9ejemplo*". The code pane contains the following DDL statements:

```
1 • DROP DATABASE continentes;
2 • CREATE DATABASE continentes;
3 • USE continentes;
4
5 • CREATE TABLE america (
6     codigo INT PRIMARY KEY,
7     nombre VARCHAR(50) NOT NULL,
8     poblacion INT DEFAULT 0,
9     extension INT,
10    fundacion DATE,
11    capital varchar(50) NOT NULL
12 )ENGINE=INNODB;
13
14 • CREATE TABLE ciudades (
15     id INT AUTO_INCREMENT PRIMARY KEY,
16     nombre VARCHAR(50)
17 )ENGINE=INNODB;
```

Handwritten annotations in yellow highlight specific parts of the code:

- A callout points to the `DEFAULT 0` in the `poblacion` column of the `america` table, with the text: "DEFALT → Se establece el tipo de dato por defecto, en ese caso sera 0 si no declararamos nada."
- A callout points to the `NOT NULL` constraint in the `capital` column of the `america` table, with the text: "Potemos esto en esas columnas que pensamos que nunca podra estar vacia."

In the bottom right corner, there is a logo for "ILERNA Online".

Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

Realizamos un **DESCRIBE** a la tabla **america**

este comando muestra la tabla

¿Qué observamos?

El **SELECT* FROM [tabla]** es para ver los registros de la tabla, no las especificaciones.

Siparemos que un campo no puede ser null, es que jamás pueden ser valores nulos.

```
Query 1  videoclase9ejemplo* x
12   )ENGINE=INNODB;
13
14 •   CREATE TABLE ciudades (
15     id INT AUTO_INCREMENT PRIMARY KEY,
16     nombre VARCHAR(50)
17   )ENGINE=INNODB;
18
19 •   describe america;
20
```

Field	Type	Null	Key	Default	Extra
codigo	int(11)	NO	PRI	NULL	
nombre	varchar(50)	NO		NULL	
poblacion	int(11)	YES		0	no admite nulls
extension	int(11)	YES		NULL	por defecto NULLamente
fundacion	date	YES		NULL	
capital	varchar(50)	NO		NULL	

Da clave Primaria nunca acepta nulos

Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

Realizamos un **DESCRIBE** a la tabla **ciudades**

¿Qué observamos?

↳ Una descripción de la tabla

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL code is visible:

```
15 id INT AUTO_INCREMENT PRIMARY KEY,  
16 nombre VARCHAR(50)  
17 )ENGINE=INNODB;  
18  
19 • describe ciudades;  
20
```

The cursor is positioned at line 19, which contains the `describe ciudades;` command. Below the editor is the Result Grid, which displays the structure of the `ciudades` table:

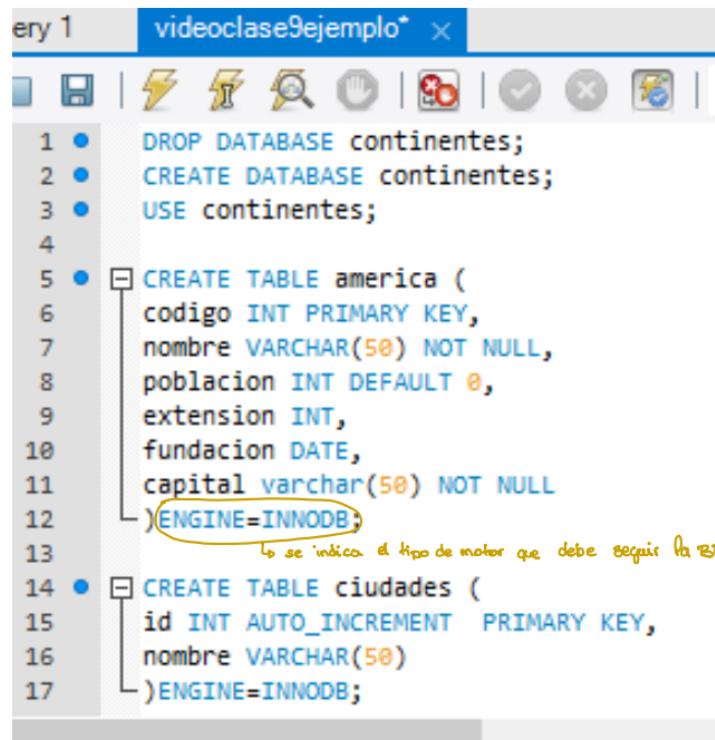
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nombre	varchar(50)	YES		NULL	

Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

Ahora deseamos convertir el atributo “capital” de la tabla “america” en una **FOREIGN KEY** (clave ajena) de la tabla “ciudades” *→ Hay varias formas de hacerlo*

¿Cómo podemos hacerlo?



```
Query 1  videoclase9ejemplo* x
 1 • DROP DATABASE continentes;
 2 • CREATE DATABASE continentes;
 3 • USE continentes;
 4
 5 • CREATE TABLE america (
 6     codigo INT PRIMARY KEY,
 7     nombre VARCHAR(50) NOT NULL,
 8     poblacion INT DEFAULT 0,
 9     extension INT,
10     fundacion DATE,
11     capital varchar(50) NOT NULL
12     )ENGINE=INNODB;
13
14 • CREATE TABLE ciudades (
15     id INT AUTO_INCREMENT PRIMARY KEY,
16     nombre VARCHAR(50)
17     )ENGINE=INNODB;
```

*Lo se indica el tipo de motor que debe seguir la BD * no es necesario ponerlo. Para esta asegurara.*

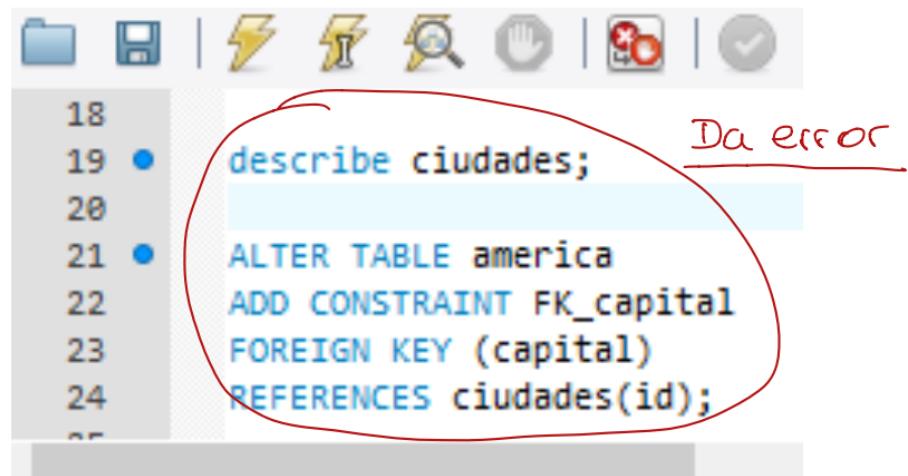
Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

Hemos usado un **ALTER TABLE** para añadir una clave ajena a la tabla **america** pero...

¿Por qué nos da error?

Porque según hemos declarado antes capital es tipo de dato Varchar , y la id de ciudades es un tipo entero . por lo que debemos corregir el tipo de dato y poner los dos del mismo tipo de dato.



```
describe ciudades;
ALTER TABLE america
ADD CONSTRAINT FK_capital
FOREIGN KEY (capital)
REFERENCES ciudades(id);
```

65 01:53:39 ALTER TABLE america ADD CONSTRAINT FK_capital FOREIGN KEY (capital) R... Error Code: 1215. Cannot add foreign key constraint 0.62

SIEMPRE la Foreign Key y la primary key sea del mismo tipo de dato

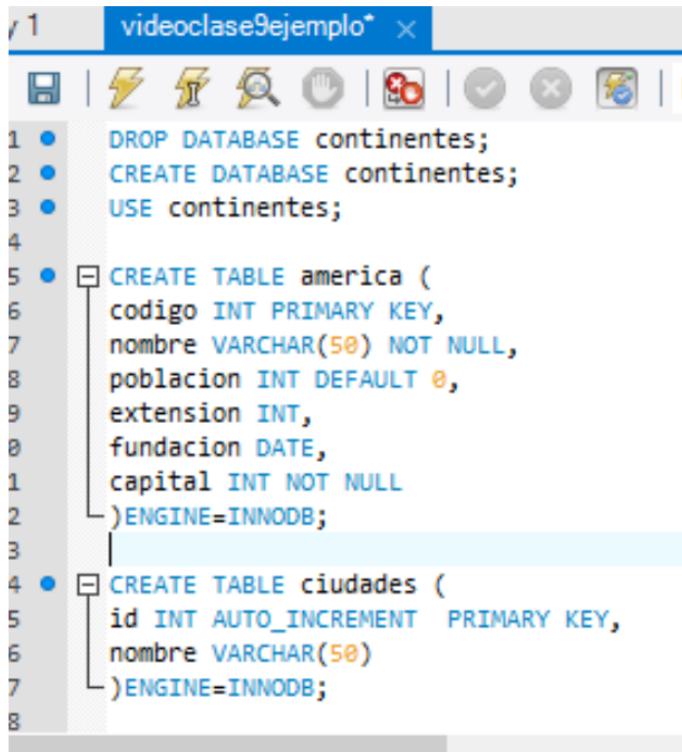


Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

Cambiamos el tipo de atributo de “capital”, ahora es un **INT**, al igual que lo es “**id**” en la tabla “ciudades”

¿Nos permitirá ahora asignar a capital como clave ajena?



```
/ 1 videoclase9ejemplo* ×
 1 • DROP DATABASE continentes;
 2 • CREATE DATABASE continentes;
 3 • USE continentes;
 4
 5 • ┌─ CREATE TABLE america (
 6   codigo INT PRIMARY KEY,
 7   nombre VARCHAR(50) NOT NULL,
 8   poblacion INT DEFAULT 0,
 9   extension INT,
10   fundacion DATE,
11   capital INT NOT NULL
12   )ENGINE=INNODB;
13
14 • ┌─ CREATE TABLE ciudades (
15   id INT AUTO_INCREMENT PRIMARY KEY,
16   nombre VARCHAR(50)
17   )ENGINE=INNODB;
```

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

INSERT INTO

sirve para introducir un registro.

Introducimos un valor a la tabla “ciudades” con el comando **INSERT**

A continuación realizamos una visualización del contenido de la tabla “ciudades” con el comando **SELECT**

Query 1 videoclase9ejemplo

REFERENCES ciudades(id);

describe america;

Para introducir un registro nombre tabla siempre va con insert into Valores del registro

INSERT INTO ciudades VALUES (1, "Brasilia");

SELECT * FROM ciudades;

Result Grid | Filter Rows: Edit:

	id	nombre
▶	1	Brasilia
*	NULL	NULL

↳ siempre hay que meter un valor a no ser que el campo admite valores nulos.

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

INSERT INTO

Ahora intentamos introducir otro registro (ósea otra fila) en la tabla "ciudades", pero en este caso probaremos el **INSERT** sin introducirle todos los campos.

```
Query | Videoclassesejemplo | X
Limit to 1000 rows
27
28 • INSERT INTO ciudades VALUES (1, "Brasilia");
29 • SELECT * FROM ciudades;
30
31 • INSERT INTO ciudades(nombre) VALUES ("Bogotá");
32 • SELECT * FROM ciudades;
33 |
```

¿Qué pasará?

↳ lo aceptará ya que el ID está declarado como auto incrementa.

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

INSERT INTO

¡LO ACEPTA!

¿por qué?

Porque id es autoincrement

```
30
31 • INSERT INTO ciudades(nombre) VALUES ("Bogotá");
32 • SELECT * FROM ciudades;
33
```

Result Grid | Filter Rows: | Edit:

	id	nombre
▶	1	Brasilia
	2	Bogotá
◀	NULL	NULL

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

REPLACE

Ahora deseamos cambiar “Brasilia” por “Caracas” como la ciudad de id=1

Es decir, queremos reemplazar Brasilia por Caracas.

Para ello, usamos el comando **REPLACE**

The screenshot shows the MySQL Workbench interface. The query editor tab is titled "Query 1" and contains the following SQL code:

```
Query 1  videoclase9ejemplo* ×
30
31 • INSERT INTO ciudades(nombre) VALUES ("Bogotá");
32 • SELECT * FROM ciudades;
33
34 • REPLACE INTO ciudades (id, nombre)
35 •   VALUES (1,"Caracas");
36 • SELECT * FROM ciudades;
```

The line "REPLACE INTO ciudades (id, nombre)" is highlighted with a red arrow pointing to it from the left. The result grid below shows the data in the "ciudades" table:

	id	nombre
1	1	Caracas
2	2	Bogotá
	NULL	NULL

The "Result Grid" tab is selected at the bottom of the interface.

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Si partimos de la tabla ciudades con este contenido

	id	nombre
▶	1	Caracas
◀	2	Bogotá
*	NULL	NULL

¿Qué pasará si introducimos las sentencias?

```
38 • ALTER TABLE ciudades AUTO_INCREMENT = 10;
39 • INSERT INTO ciudades(nombre) VALUES ("Santiago de Chile");
40 • INSERT INTO ciudades VALUES (20, "Lima");
41 • INSERT INTO ciudades(nombre) VALUES ("Quito");
42 • SELECT * FROM ciudades;
```

d auto increment
eumentura para esa linea
en lo

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Como podemos observar, se ejecutan todas de manera natural y la numeración de las id deja de ser secuencial

```
38 • ALTER TABLE ciudades AUTO_INCREMENT = 10;
39 • INSERT INTO ciudades(nombre) VALUES ("Santiago de Chile");
40 • INSERT INTO ciudades VALUES (20, "Lima");
41 • INSERT INTO ciudades(nombre) VALUES ("Quito");
42 • SELECT * FROM ciudades;
```

Result Grid | Filter Rows: [] | Edit: [] | Export

id	nombre
1	Caracas
2	Bogotá
10	Santiago de Chile
20	Lima
21	Quito

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Agregamos un registro a la tabla “america”

The screenshot shows the MySQL Workbench interface with a query editor and a result grid.

Query Editor:

```
Query 1  videoclase9ejemplo* ×
41 • INSERT INTO ciudades(nombre) VALUES ("Quito");
42 • SELECT * FROM ciudades;
43
44 • INSERT INTO america
45     VALUES (1, "Venezuela", 35000000, 1000000, "1815-04-11",1);
46 • SELECT * FROM america;
```

Result Grid:

	codigo	nombre	poblacion	extension	fundacion	capital
▶	1	Venezuela	35000000	1000000	1815-04-11	1
*	NULL	NULL	NULL	NULL	NULL	NULL

Data Manipulation Language (DML)

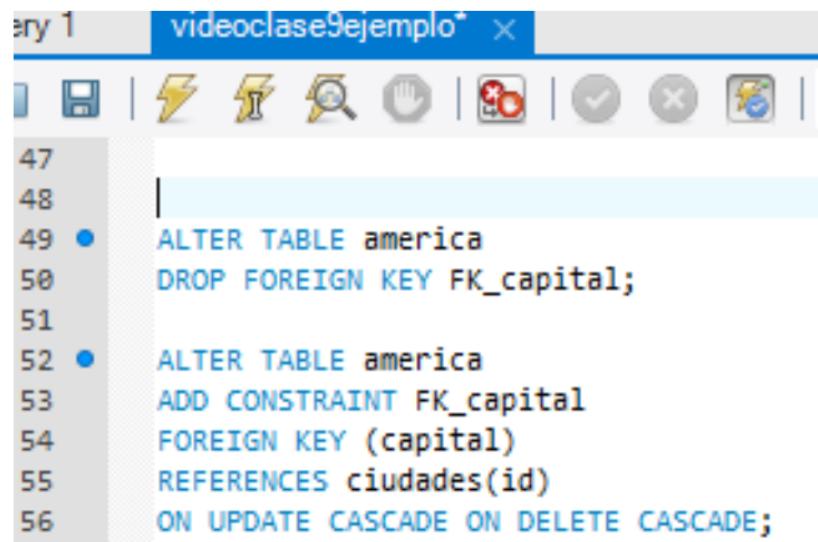
EJEMPLOS en WORKBENCH

VIDEOCLASE 8

Ahora rehacemos la restricción de **clave ajena** que teníamos en la tabla “**america**”.

Creamos de nuevo la clave ajena de modo que si en la tabla origen (**ciudades**) hay una modificación o eliminación del atributo referenciado, este cambio se verá reflejado en la tabla “**america**”, concretamente en el atributo “**capital**”, que es la clave ajena.

Como se observa, para crear de nuevo la clave ajena hemos eliminado la antigua.



```
Query 1 | videoclase9ejemplo* x
47
48
49 • ALTER TABLE america
50      DROP FOREIGN KEY FK_capital;
51
52 • ALTER TABLE america
53      ADD CONSTRAINT FK_capital
54          FOREIGN KEY (capital)
55              REFERENCES ciudades(id)
56              ON UPDATE CASCADE ON DELETE CASCADE;
```

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

¿Qué obtendremos al ejecutar esto?

Cambiaremos
el ID de la
tabla anterior.
(Pasaría de
ser 1 a 99)

```
57
58 • UPDATE ciudades SET id=99 WHERE id=1;
59 • SELECT * FROM america;
```