

Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

Creamos una base de datos llamada “continentes”.

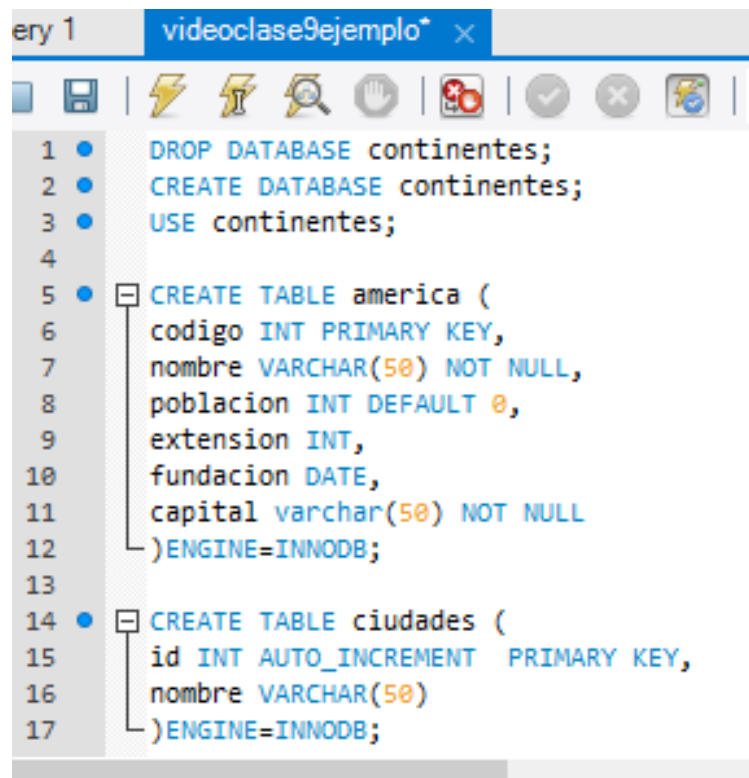
Después creamos una tabla llamada “**america**”.

Asignamos **PRIMARY KEY**

Asignamos campos **NOT NULL**

Asignamos **DEFAULT**

También creamos una tabla llamada “**ciudades**”



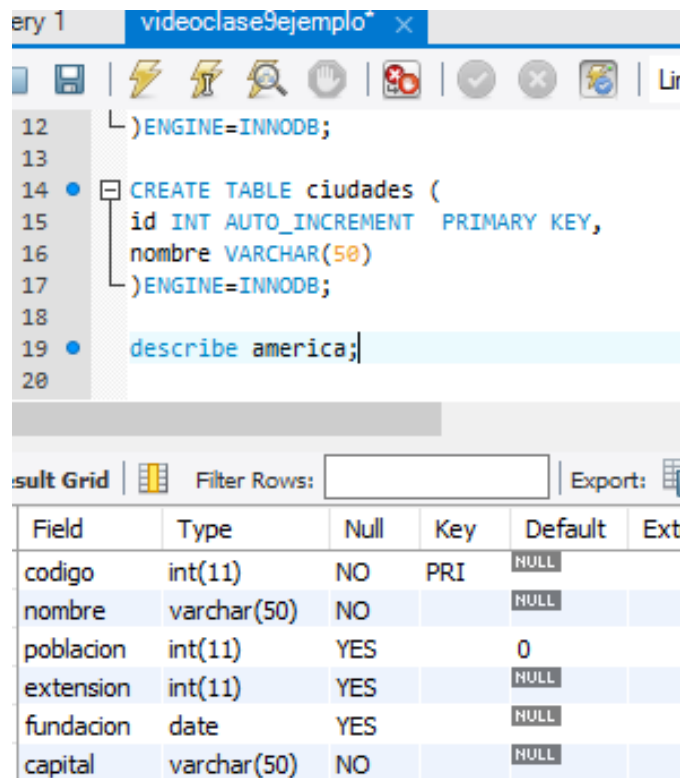
```
1 DROP DATABASE continentes;
2 CREATE DATABASE continentes;
3 USE continentes;
4
5 CREATE TABLE america (
6     codigo INT PRIMARY KEY,
7     nombre VARCHAR(50) NOT NULL,
8     poblacion INT DEFAULT 0,
9     extension INT,
10    fundacion DATE,
11    capital varchar(50) NOT NULL
12 )ENGINE=INNODB;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

Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

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

¿Qué observamos?



The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for saving, executing, and viewing results. The SQL editor contains the following code:

```
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
```

Below the editor is the 'Result Grid' section, which displays the output of the 'describe america;' query. It includes a 'Filter Rows' input and an 'Export' button. The grid itself has columns for Field, Type, Null, Key, Default, and Extra.

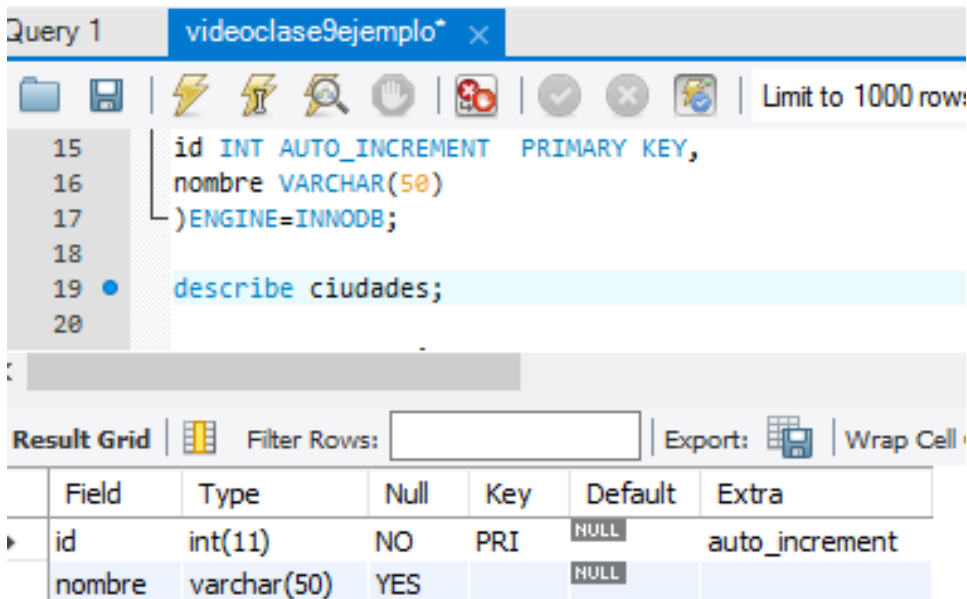
Field	Type	Null	Key	Default	Extra
codigo	int(11)	NO	PRI	NULL	
nombre	varchar(50)	NO		NULL	
poblacion	int(11)	YES		0	
extension	int(11)	YES		NULL	
fundacion	date	YES		NULL	
capital	varchar(50)	NO		NULL	

Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

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

¿Qué observamos?



The screenshot shows the MySQL Workbench interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' button. The query editor displays the following SQL code:

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

Below the query editor is the 'Result Grid' section, which includes a 'Filter Rows' input field and an 'Export' button. The result grid displays the following table structure:

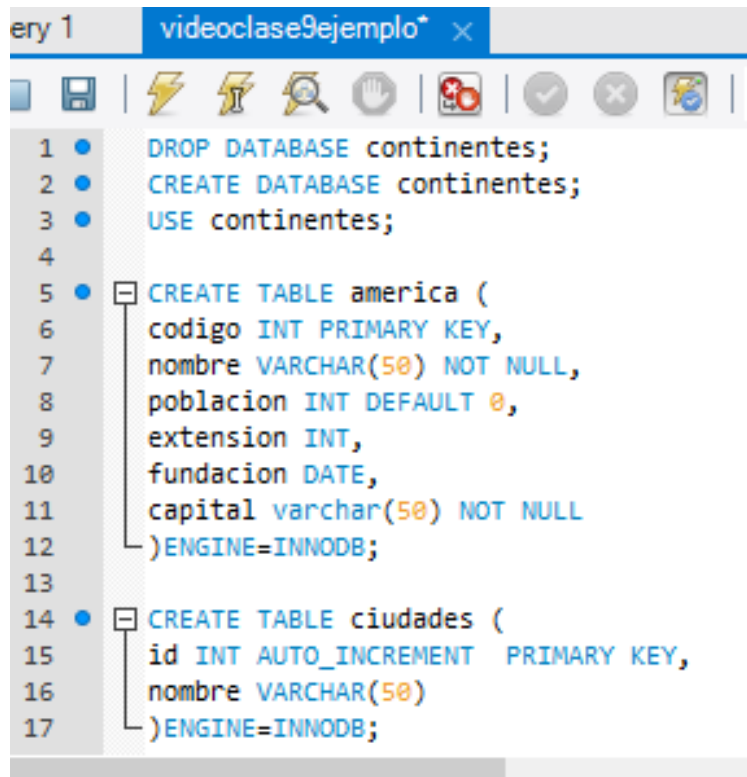
	Field	Type	Null	Key	Default	Extra
▶	id	int(11)	NO	PRI	NULL	auto_increment
	nombre	varchar(50)	YES		NULL	

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”

¿Cómo podemos hacerlo?



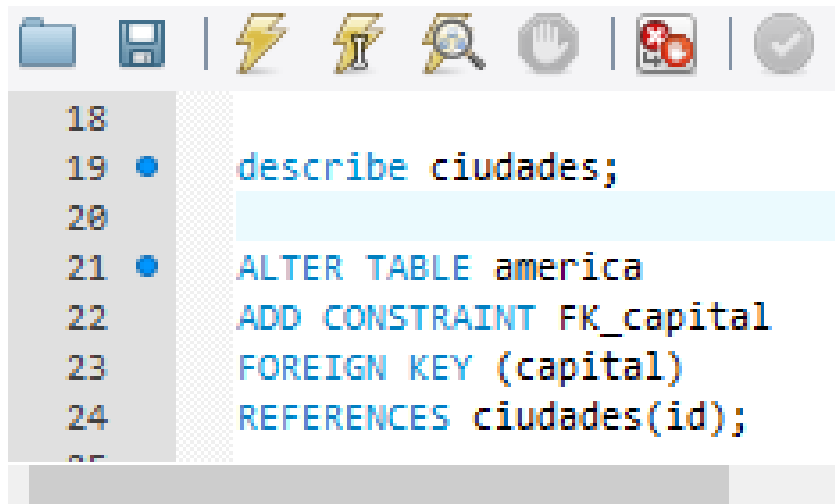
```
1 DROP DATABASE continentes;
2 CREATE DATABASE continentes;
3 USE continentes;
4
5 CREATE TABLE america (
6     codigo INT PRIMARY KEY,
7     nombre VARCHAR(50) NOT NULL,
8     poblacion INT DEFAULT 0,
9     extension INT,
10    fundacion DATE,
11    capital varchar(50) NOT NULL
12 )ENGINE=INNODB;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

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?



The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for file operations (folder, save), execution (lightning bolt, lightning bolt with cursor), search (magnifying glass), undo (hand), redo (hand with plus), and a status bar (checkmark). The main window displays a SQL script with line numbers 18 through 25. Line 19 contains 'describe ciudades;', and line 21 contains 'ALTER TABLE america'. Lines 22 through 25 contain the foreign key constraint definition: 'ADD CONSTRAINT FK_capital FOREIGN KEY (capital) REFERENCES ciudades(id);'.

```
18  
19 describe ciudades;  
20  
21 ALTER TABLE america  
22 ADD CONSTRAINT FK_capital  
23 FOREIGN KEY (capital)  
24 REFERENCES ciudades(id);  
25
```

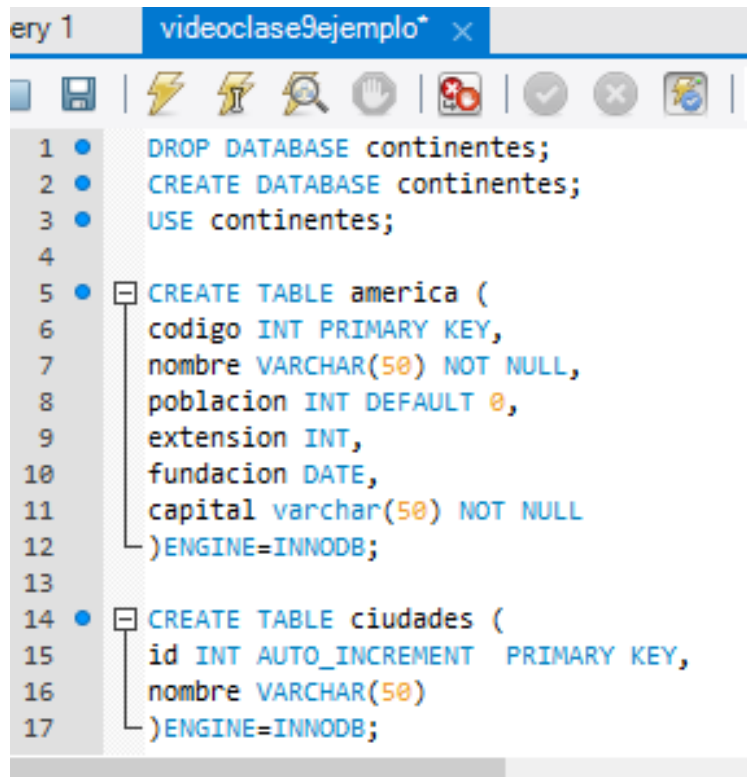
✓	64	01:53:39	describe ciudades	Script completed	0.00
✗	65	01:53:39	ALTER TABLE america ADD CONSTRAINT FK_capital FOREIGN KEY (capital) R...	Error Code: 1215. Cannot add foreign key constraint	0.62

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?



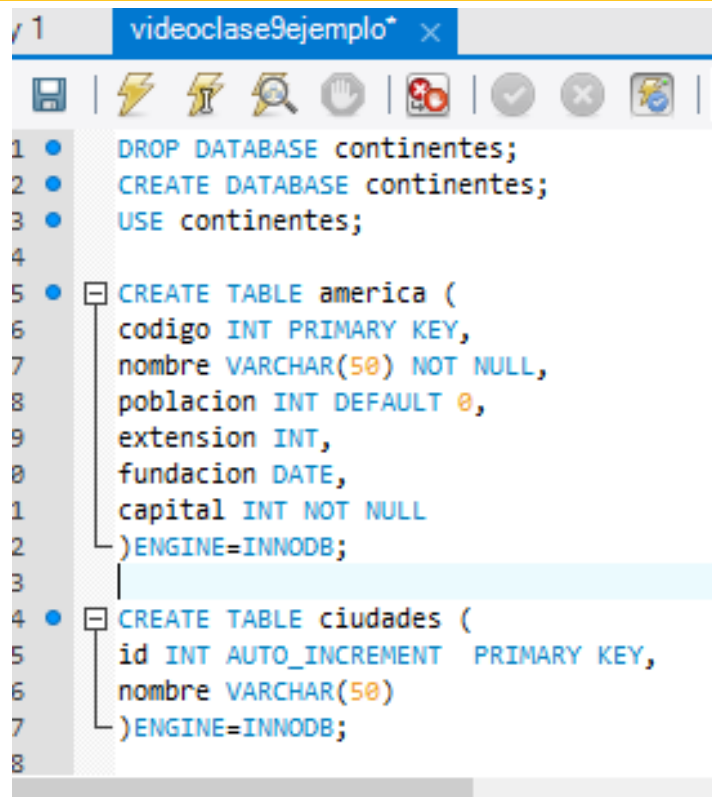
```
1 DROP DATABASE continentes;
2 CREATE DATABASE continentes;
3 USE continentes;
4
5 CREATE TABLE america (
6     codigo INT PRIMARY KEY,
7     nombre VARCHAR(50) NOT NULL,
8     poblacion INT DEFAULT 0,
9     extension INT,
10    fundacion DATE,
11    capital varchar(50) NOT NULL
12 )ENGINE=INNODB;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

Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

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

¿Nos permitirá ahora asignar a capital como clave ajena?



```
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 Definition Language (DDL)

EJEMPLOS en WORKBENCH

Ahora sí nos permite asignar como clave ajena a “capital”, dado que coincide el tipo de datos con el original, “id” de la tabla “ciudades”

Nos fijamos que en la descripción de la tabla, ¿muestra algo diferente en “capital”?

The screenshot shows the Oracle Workbench interface. The top pane displays a SQL script with the following queries:

```
21 ALTER TABLE america
22 ADD CONSTRAINT FK_capital
23 FOREIGN KEY (capital)
24 REFERENCES ciudades(id);
25
26 describe america;
```

The middle pane shows the 'Result Grid' for the 'describe america;' query. It displays the table structure for 'america'.

	Field	Type	Null	Key	Default	Extra
▶	codigo	int(11)	NO	PRI	NULL	
	nombre	varchar(50)	NO		NULL	
	poblacion	int(11)	YES		0	
	extension	int(11)	YES		NULL	
	fundacion	date	YES		NULL	
	capital	int(11)	NO	MUL	NULL	

The bottom pane shows the 'Action Output' for the executed queries:

#	Time	Action	Message
✓ 87	02:08:00	CREATE TABLE ciudades (i...	0 row(s) affected
✓ 88	02:08:00	ALTER TABLE america AD...	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓ 89	02:08:01	describe america	6 row(s) returned

Data Definition Language (DDL)

EJEMPLOS en WORKBENCH

Lo vemos más de cerca...

Field	Type	Null	Key	Default
codigo	int(11)	NO	PRI	NULL
nombre	varchar(50)	NO		NULL
poblacion	int(11)	YES		0
extension	int(11)	YES		NULL
fundacion	date	YES		NULL
capital	int(11)	NO	MUL	NULL

Query 1 | videoclase9ejemplo* x

```
21 ALTER TABLE america
22 ADD CONSTRAINT FK_capital
23 FOREIGN KEY (capital)
24 REFERENCES ciudades(id);
25
26 describe america;
```

- 87 02:08:00 CREATE TABLE ciudades (i... 0 row(s) affected
- 88 02:08:00 ALTER TABLE america AD... 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
- 89 02:08:01 describe america 6 row(s) returned

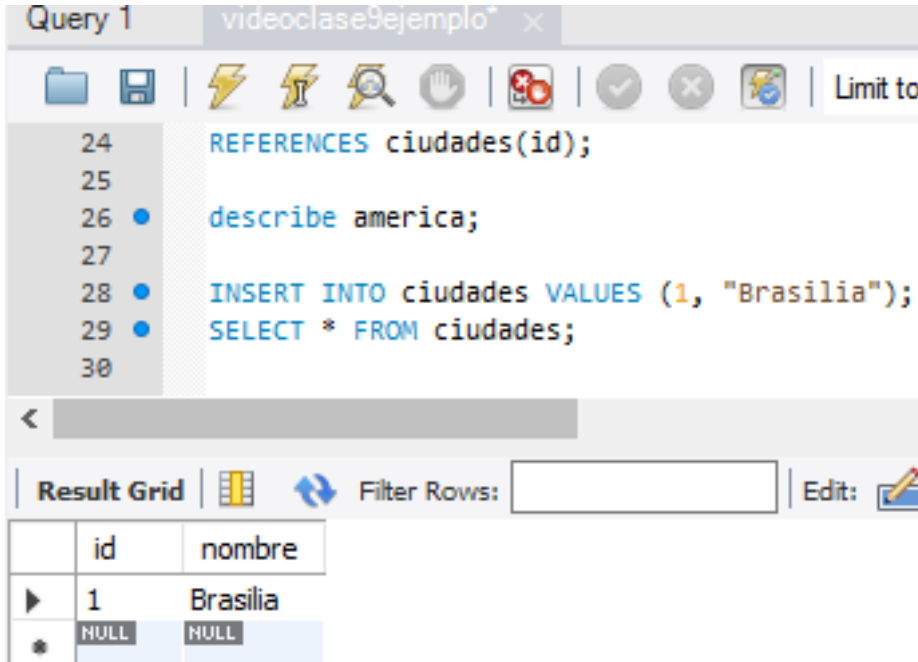
Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

INSERT INTO

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**



The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for file operations, execution, and window management. The query window, titled 'Query 1', contains the following SQL code:

```
24 REFERENCES ciudades(id);
25
26 describe america;
27
28 INSERT INTO ciudades VALUES (1, "Brasilia");
29 SELECT * FROM ciudades;
30
```

Below the query window is the 'Result Grid' section, which displays the output of the last executed query (SELECT * FROM ciudades). The grid shows the following data:

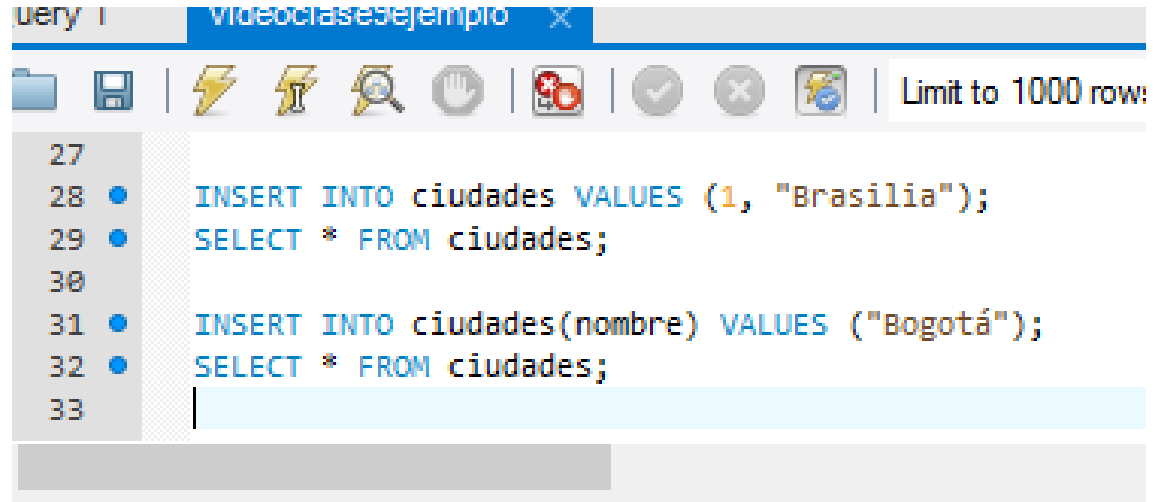
	id	nombre
▶	1	Brasilia
*	NULL	NULL

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.



The screenshot shows the Oracle SQL Developer interface. At the top, there's a toolbar with icons for file operations, execution, and undo/redo. Below the toolbar, a query window titled 'Query 1' and 'Videoclasesejemplo' contains the following SQL code:

```
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
```

The code demonstrates two ways to insert data into the 'ciudades' table. The first line (28) uses a full VALUES clause with two arguments. The second line (31) uses a column list (nombre) followed by a single value in the VALUES clause. Both are followed by a SELECT statement to verify the data.

¿Qué pasará?

En ambos casos se ejecutará. Cuando no pones el atributo al lado de la tabla, en este caso nombre, debes poner todos los atributos de esa fila entre paréntesis. Y si pones (nombre) tendrás que especificar esos atributos.

Data Manipulation Language (DML)

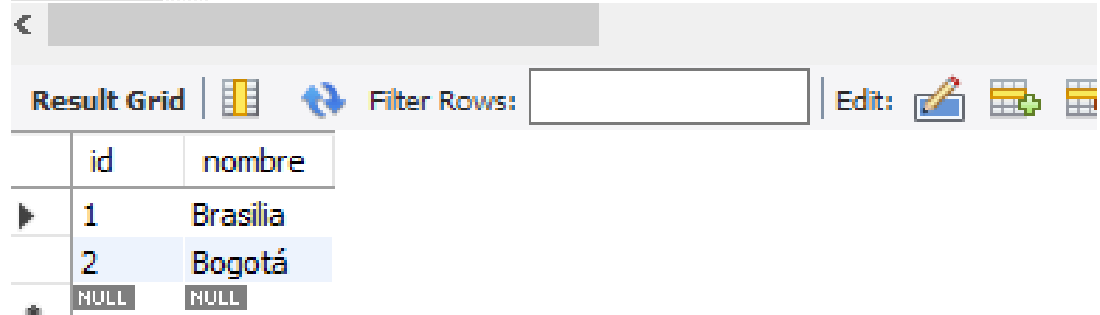
EJEMPLOS en WORKBENCH

INSERT INTO

¡LO ACEPTA!

¿por qué?

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



The screenshot shows the Oracle Workbench interface. At the top, the SQL editor contains two lines of code: `INSERT INTO ciudades(nombre) VALUES ("Bogotá");` and `SELECT * FROM ciudades;`. Below the editor is the 'Result Grid' tab, which displays the results of the SQL query. The grid has two columns: 'id' and 'nombre'. It shows two rows of data: the first row has '1' in the 'id' column and 'Brasilia' in the 'nombre' column; the second row has '2' in the 'id' column and 'Bogotá' in the 'nombre' column. The second row is highlighted. Below the grid, there are 'NULL' values for both columns, indicating that the query returned all rows from the table.

	id	nombre
▶	1	Brasilia
	2	Bogotá
★	NULL	NULL

Data Manipulation Language (DML)

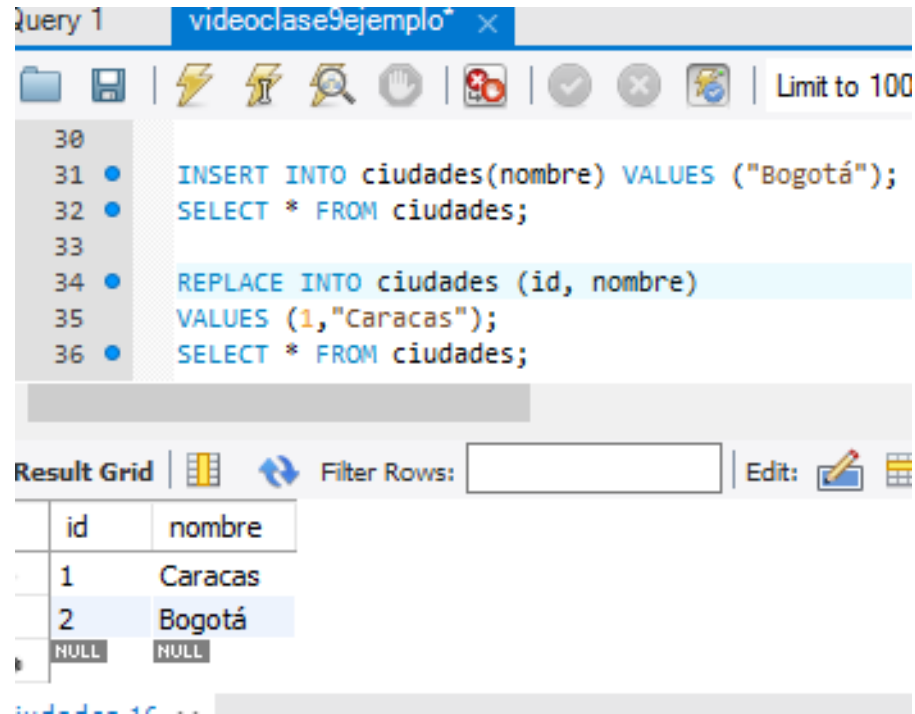
EJEMPLOS en WORKBENCH

REPLACE Reemplaza toda la fila.

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 Oracle SQL Developer interface. The top pane displays a SQL query in the 'Query 1' window. The query is as follows:

```
Query 1: videoclase9ejemplo* x
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 bottom pane shows the 'Result Grid' with the following data:

id	nombre
1	Caracas
2	Bogotá
NULL	NULL

Below the table, the text 'ciudades 16' is visible.

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 as 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;
```

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**”

Query 1 | videoclase9ejemplo* x

Limit to 1000 rows

```
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 | Filter Rows: | Edit: | Export/In

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

El 1 hace referencia a ciudades (caracas).
Capital es FK

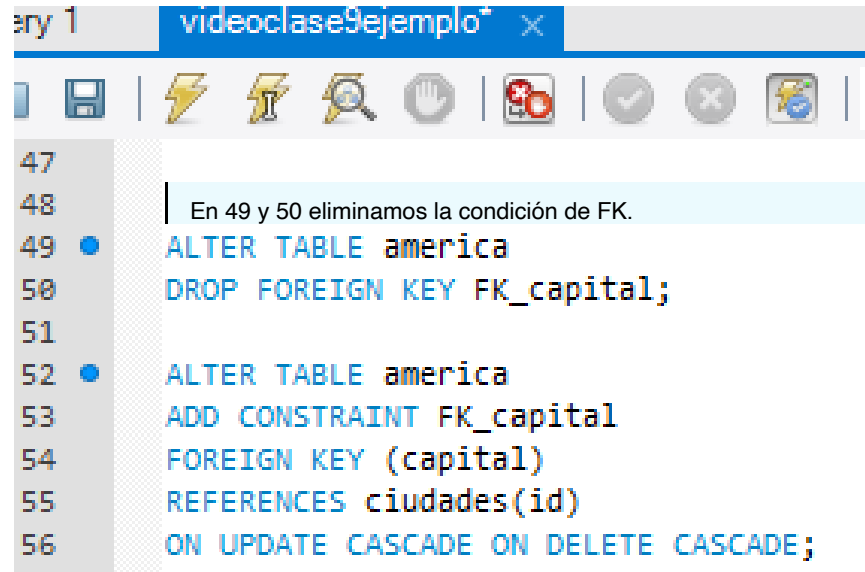
Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

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. No eliminamos "capital", solo deja de ser FK.

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



```
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;
```

En 49 y 50 eliminamos la condición de FK.

A partir de 53 lo volvemos a asignar como FK pero respecto a id de la tabla ciudades.

Se va a reflejar en cascada al actualizar porque si hubiese mas tablas que referencien hacia ciudades id todas se van a cambiar en cascada, todas se van a cambiar. DELETE CASCADE: Si elimináramos el registro, debería hacerse también en todos los registros donde esté referenciado. Todo debe propagarse.

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

¿Qué obtendremos al ejecutar esto?

57

58



```
UPDATE ciudades SET id=99 WHERE id=1;
```

59



```
SELECT * FROM america;
```

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

El cambio en la tabla referenciada “ciudades” se ha propagado en la otra tabla, en el atributo que es clave ajena

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

Result Grid						
Filter Rows:						
	codigo	nombre	poblacion	extension	fundacion	capital
▶	1	Venezuela	35000000	1000000	1815-04-11	99
•	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid						
Filter Rows:						
	codigo	nombre	poblacion	extension	fundacion	capital
▶	1	Venezuela	35000000	1000000	1815-04-11	1
•	NULL	NULL	NULL	NULL	NULL	NULL

antes era 1

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Creamos un ejemplo con una tabla nueva, la tabla "granciudad"

Result Grid

	id	nombre	poblacion
▶	1	madrid	4000000
	2	barcelona	2000000
	3	valencia	1000000
*	NULL	NULL	NULL

granciudad 1 x

Output

Action Output

#	Time	Action
✓ 121	11:38:21	INSERT INTO granciudad (nomt
✓ 122	11:38:21	select *from granciudad LIMIT 0

```
1 drop database if exists ejemplo;
2 create database ejemplo;
3 USE ejemplo;
4
5 create table granciudad (
6     id INT PRIMARY KEY auto_increment,
7     nombre VARCHAR(10),
8     poblacion INT
9 ) Engine=InnoDB;
10
11
12
13 ALTER TABLE granciudad auto_increment = 1;
14
15 INSERT INTO granciudad (nombre, poblacion) VALUES
16 ("madrid", 4000000),
17 ("barcelona", 2000000),
18 ("valencia", 1000000);
```

id será entero y PK con autoincremento

Se establece el autincremento a partir del 1.

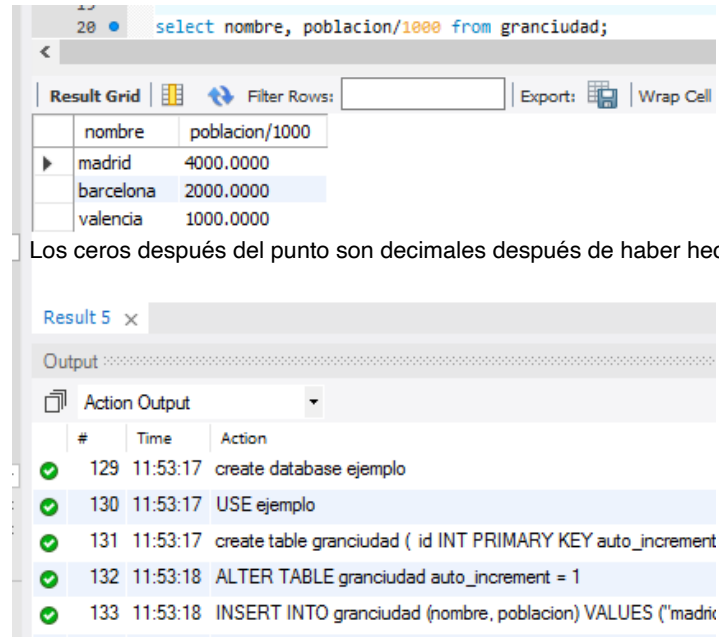
Insertamos 3 registros a la vez.

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Podemos hacer cálculos en las sentencias SELECT

Hemos realizado la división de la población de las ciudades para obtener la población en miles de habitantes.



The screenshot shows the Oracle Workbench interface. At the top, a SQL query is entered in the editor: `select nombre, poblacion/1000 from granciudad;`. Below the editor, the 'Result Grid' tab is active, displaying the results of the query in a table format. The table has two columns: 'nombre' and 'poblacion/1000'. The data rows are: 'madrid' with '4000.0000', 'barcelona' with '2000.0000', and 'valencia' with '1000.0000'. Below the result grid, the 'Output' tab is active, showing a list of executed SQL statements with their execution times and status (all successful, indicated by green checkmarks).

	nombre	poblacion/1000
▶	madrid	4000.0000
	barcelona	2000.0000
	valencia	1000.0000

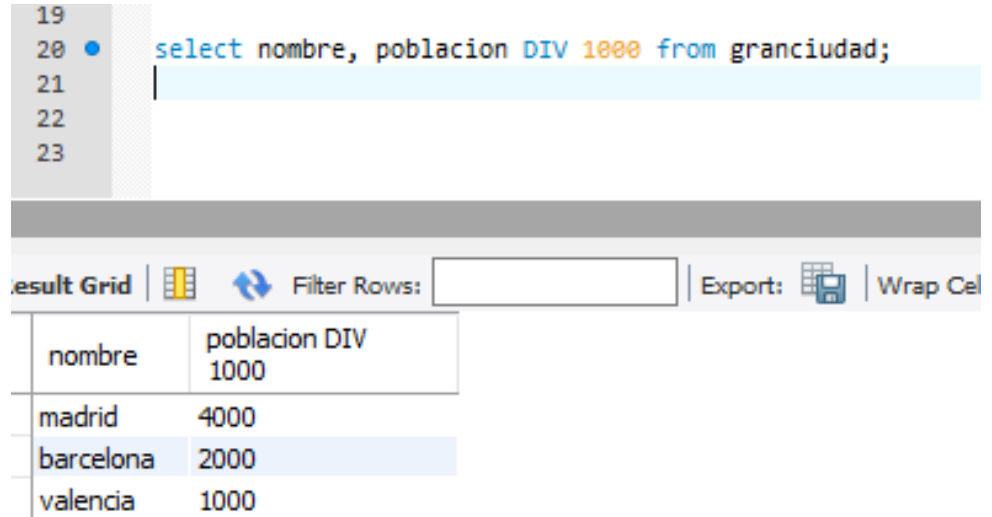
#	Time	Action
✓ 129	11:53:17	create database ejemplo
✓ 130	11:53:17	USE ejemplo
✓ 131	11:53:17	create table granciudad (id INT PRIMARY KEY auto_increment
✓ 132	11:53:18	ALTER TABLE granciudad auto_increment = 1
✓ 133	11:53:18	INSERT INTO granciudad (nombre, poblacion) VALUES ('madrid

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Podemos hacer cálculos en las sentencias SELECT

En este caso, hemos usado el DIV para hacer una división sin decimales



The screenshot shows the Oracle SQL Developer interface. The SQL Editor at the top contains the query: `select nombre, poblacion DIV 1000 from granciudad;`. Below the editor, the 'Result Grid' tab is active, displaying the results of the query in a table format. The table has two columns: 'nombre' and 'poblacion DIV 1000'. The data rows are 'madrid' (4000), 'barcelona' (2000), and 'valencia' (1000). The 'barcelona' row is highlighted in blue. The interface also shows a 'Filter Rows' field and an 'Export' button.

nombre	poblacion DIV 1000
madrid	4000
barcelona	2000
valencia	1000

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Podemos hacer cálculos en las sentencias SELECT

En este caso, además de usar el DIV, renombramos con el AS la columna a nuestro gusto.

```
19  
20 • select nombre, poblacion DIV 1000 AS poblacion_en_miles from granciudad;  
21
```

	nombre	poblacion_en_miles
▶	madrid	4000
	barcelona	2000
	valencia	1000

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

USO del LIMIT

INSERTAMOS nuevas ciudades en nuestra tabla, en este caso Sevilla y Bilbao.

```
18 (valencia, 1000000);
19
20 • INSERT INTO granciudad (nombre, poblacion) VALUES
21 ("bilbao", 800000),
22 ("sevilla", 900000);
23
24 • select * from granciudad;
25
26
27
```

Result Grid

	id	nombre	poblacion
▶	1	madrid	4000000
	2	barcelona	2000000
	3	valencia	1000000
	4	bilbao	800000
	5	sevilla	900000
	6	NULL	NULL
	7	NULL	NULL

granciudad 7 x

Output

Action Output

#	Time	Action
✓ 132	11:53:18	ALTER TABLE granciudad auto_increment = 1
✓ 133	11:53:18	INSERT INTO granciudad (nombre, poblacion) VALUES ("max", 1000000);

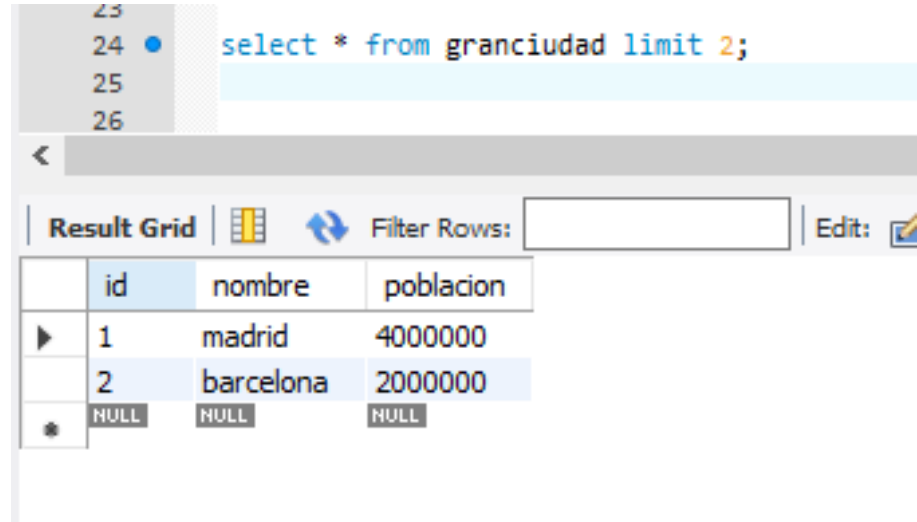
Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

USO del LIMIT

Si ponemos limit 2 , solo nos aparecerán las 2 primeras ciudades

¿Qué pasará si ponemos limit 2,3?



The screenshot shows the Oracle SQL Developer interface. The SQL Editor at the top contains the query: `select * from granciudad limit 2;`. Below the editor, the 'Result Grid' tab is active, displaying the results of the query. The grid has four columns: 'id', 'nombre', and 'poblacion'. The first two rows are highlighted in blue, corresponding to the first two rows of the data. The third row is a summary row with 'NULL' values. The interface also includes a 'Filter Rows' search bar and an 'Edit' button.

	id	nombre	poblacion
▶	1	madrid	4000000
	2	barcelona	2000000
*	NULL	NULL	NULL

Data Manipulation Language (DML)

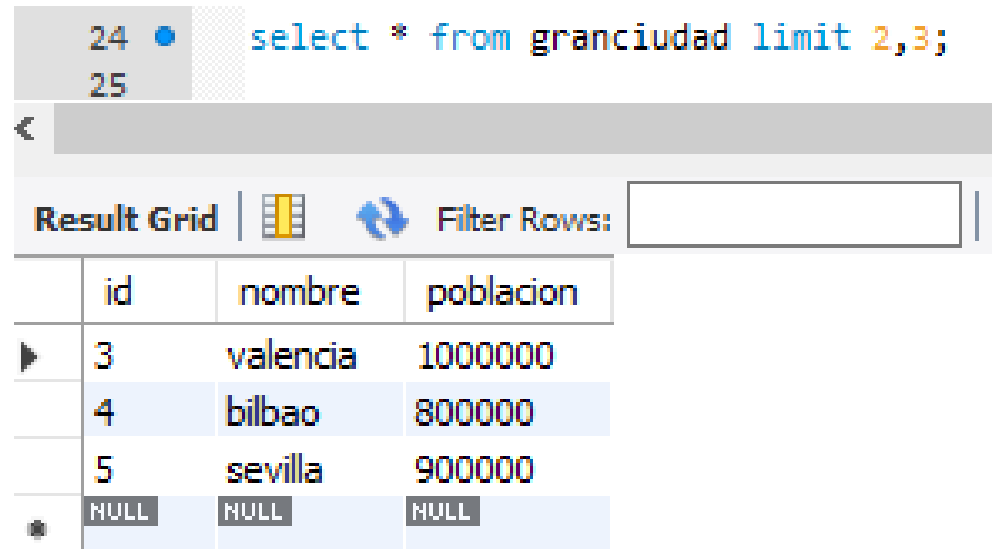
EJEMPLOS en WORKBENCH

USO del LIMIT

Como podemos observar, nos devuelve 3 registros

Por tanto...

¿Qué pasará si ponemos limit 3,2?



The screenshot shows the Oracle SQL Developer interface. At the top, a SQL editor contains the query: `select * from granciudad limit 2,3;`. Below the editor, the 'Result Grid' tab is active, displaying the results of the query in a table format. The table has three columns: 'id', 'nombre', and 'poblacion'. It contains three data rows and one row with NULL values. The interface also includes a 'Filter Rows' search bar and a refresh button.

	id	nombre	poblacion
▶	3	valencia	1000000
	4	bilbao	800000
	5	sevilla	900000
⊙	NULL	NULL	NULL

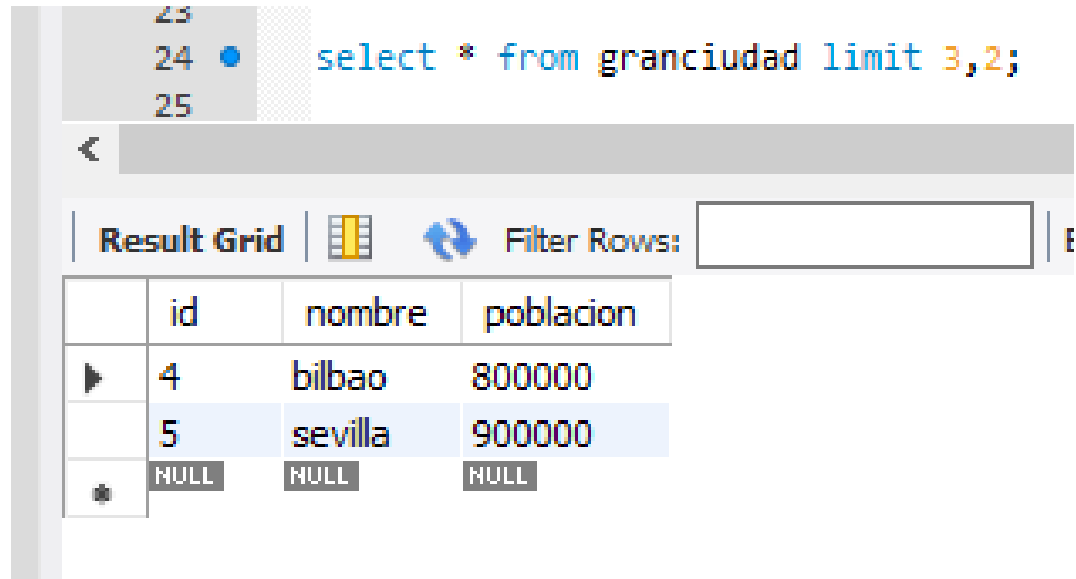
Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

USO del LIMIT

Como podemos observar, solo salen 2 registros.

Y el primer registro en mostrarse es el que va después del 3ero.



The screenshot shows the Oracle SQL Developer interface. At the top, a SQL query is entered in the editor: `select * from granciudad limit 3,2;`. Below the editor, the 'Result Grid' tab is active, displaying the results of the query. The grid has four columns: 'id', 'nombre', and 'poblacion'. The first two rows are highlighted in blue, representing the results returned by the query. The third row is a placeholder with NULL values.

	id	nombre	poblacion
▶	4	bilbao	800000
	5	sevilla	900000
✱	NULL	NULL	NULL

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Vamos a crear en la misma base de datos otra tabla, la tabla pais

```
26
27 • create table pais(
28
29     id_P INT PRIMARY KEY auto_increment,
30     nombre_P VARCHAR(10)
31
32 )Engine=InnoDB;
33
34 • ALTER TABLE pais auto_increment = 100;
35
36 • INSERT INTO pais (nombre_P) VALUES
37     ("España"),
38     ("Francia"),
39     ("Argentina");
40
41 • select * from pais;
```

<

Result Grid   Filter Rows: Edit: 

	id_P	nombre_P
▶	100	España
	101	Francia
	102	Argentina
*	NULL	NULL

pais 16 x

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Modificamos la tabla granciudad

```
25  
26 ● ALTER TABLE granciudad ADD pais INT;  
27 ● select * FROM granciudad;  
28  
29
```

< Result Grid   Filter Rows: Edit:

	id	nombre	poblacion	pais
▶	1	madrid	4000000	NULL
	2	barcelona	2000000	NULL
	3	valencia	1000000	NULL
	4	bilbao	800000	NULL
	5	sevilla	900000	NULL
✱	NULL	NULL	NULL	NULL

granciudad 18 ✕

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Hacemos un describe de la tabla granciudad y comprobamos las nuevas características

```
28  
29 describe granciudad;  
30  
31
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nombre	varchar(10)	YES		NULL	
poblacion	int(11)	YES		NULL	
pais	int(11)	YES		NULL	

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Añadimos la restricción de clave ajena de “país” en la tabla granciudad respecto del atributo id_P de la tabla país

(país como campo, no como tabla)

Podemos observar como ha modificado granciudad su estructura.

```
28
29 • ALTER TABLE granciudad ADD CONSTRAINT fk_pais_granciudad FOREIGN KEY (pais) REFERENCES pais(id_P);
30 • describe granciudad;
31
32
33
```

La clave ajena la ponemos en la tabla gran ciudad aunque se refiera a id_P de la tabla país.

< | Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Field	Type	Null	Key	Default	Extra
▶	id	int(11)	NO	PRI	NULL	auto_increment
	nombre	varchar(10)	YES		NULL	
	poblacion	int(11)	YES		NULL	
	pais	int(11)	YES	MUL	NULL	

¿Dónde está la diferencia?

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Realizamos un Select de granciudad.

¿Qué observamos?

```
29 • ALTER TABLE granciudad ADD CONSTRAINT fk_pais_granciudad FOREIGN KEY (pais) REFERENCES pais(id_P);
30 • select * from granciudad;
31
32
33
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	nombre	poblacion	pais
1	madrid	4000000	NULL
2	barcelona	2000000	NULL
3	valencia	1000000	NULL
4	bilbao	800000	NULL
5	sevilla	900000	NULL
NULL	NULL	NULL	NULL

anciudad 22 x

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Damos valor a la clave ajena
PAIS

```
29 • ALTER TABLE granciudad ADD CONSTRAINT fk_pais_granciudad FOREIGN KEY (pais) REFERENCES pais(id_P);
30 |
31 • UPDATE granciudad SET pais=100 WHERE id =1;
32 • UPDATE granciudad SET pais=100 WHERE id =2;
33 • UPDATE granciudad SET pais=100 WHERE id =3;
34 • UPDATE granciudad SET pais=100 WHERE id =4;
35 • UPDATE granciudad SET pais=100 WHERE id =5;
36 • select * from granciudad;
```

< Result Grid Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	id	nombre	poblacion	pais
▶	1	madrid	4000000	100
	2	barcelona	2000000	100
	3	valencia	1000000	100
	4	bilbao	800000	100
	5	sevilla	900000	100
*	NULL	NULL	NULL	NULL

granciudad 24 ×

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Insertamos nuevas ciudades en granciudad

101 Francia

102 Argentina

Haciendo referencia a la otra tabla.

```
28
29 ALTER TABLE granciudad ADD CONSTRAINT fk_pais_granciudad FOREIGN KEY (pais) REFERENCES pais(id_P);
30
31 UPDATE granciudad SET pais=100 WHERE id =1;
32 UPDATE granciudad SET pais=100 WHERE id =2;
33 UPDATE granciudad SET pais=100 WHERE id =3;
34 UPDATE granciudad SET pais=100 WHERE id =4;
35 UPDATE granciudad SET pais=100 WHERE id =5;
36 select * from granciudad;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: IA

	id	nombre	poblacion	pais
▶	1	madrid	4000000	100
	2	barcelona	2000000	100
	3	valencia	1000000	100
	4	bilbao	800000	100
	5	sevilla	900000	100
	6	B.Aires	15000000	102
	7	mendoza	2000000	102
	8	paris	7000000	101
	9	lyon	500000	101
*	NULL	NULL	NULL	NULL

granciudad 28 x

Output

Estamos referenciando con la clave primaria de la otra tabla.
Por eso ponemos un número en lugar del país al que hace referencia.

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Query 1 | videoclipase9ejemplo* x

Limit to 1000 rows

```
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 | Filter Rows: | Edit: | Export/Im

	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

PARTIMOS DE ESTA BASE DE DATOS

Creamos una base de datos de una biblioteca con varias tablas

```
1 • drop database biblioteca;
2 • create database biblioteca;
3 • USE biblioteca;
4
5
6 • CREATE TABLE autor (
7     cod_autor INT auto_increment,
8     nombre VARCHAR(40),
9     primary key(cod_autor)
10 );
11 • CREATE TABLE libro (
12
13     cod INT auto_increment,
14     titulo varchar(50),
15     autor varchar(50),
16     editorial varchar(50),
17     primary key(cod)
18 ) engine=innnoDB ;
```

Data Manipulation Language (DML)

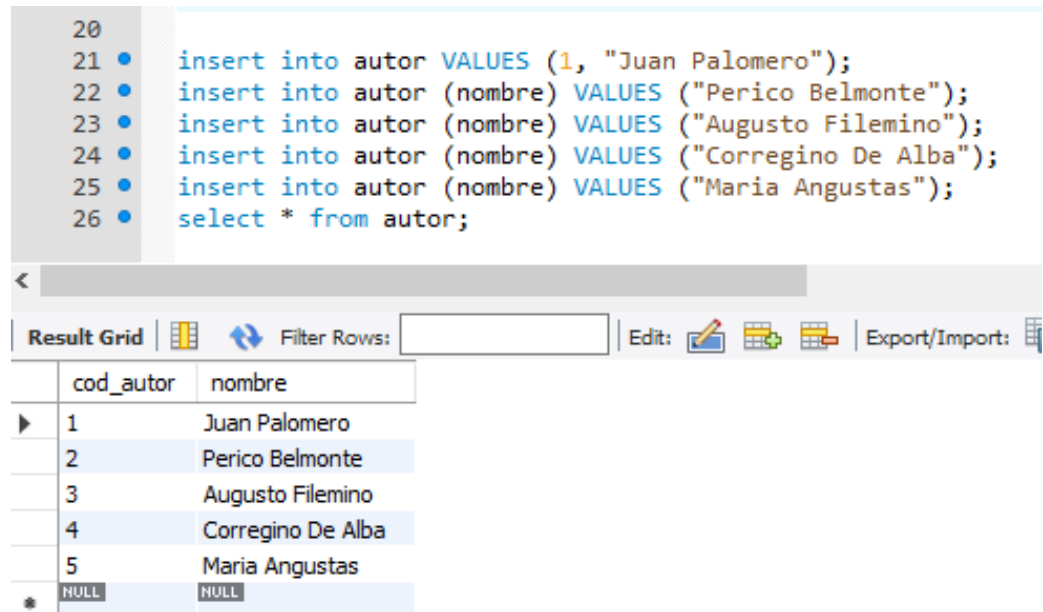
EJEMPLOS en WORKBENCH

INSERTAMOS DATOS

Insertamos datos en la base de datos en la biblioteca.

Y mostramos la tabla

No se puede referenciar a otra tabla con un registro que no existe.
Si referenciamos con 8, y la otra tabla solo tiene hasta el 5, da error.



The screenshot displays the Oracle SQL Developer interface. The top pane shows a SQL script with the following commands:

```
20  
21 • insert into autor VALUES (1, "Juan Palomero");  
22 • insert into autor (nombre) VALUES ("Perico Belmonte");  
23 • insert into autor (nombre) VALUES ("Augusto Filemino");  
24 • insert into autor (nombre) VALUES ("Corregino De Alba");  
25 • insert into autor (nombre) VALUES ("Maria Angustas");  
26 • select * from autor;
```

The bottom pane shows the 'Result Grid' for the executed query. It contains a table with two columns: 'cod_autor' and 'nombre'.

cod_autor	nombre
1	Juan Palomero
2	Perico Belmonte
3	Augusto Filemino
4	Corregino De Alba
5	Maria Angustas
NULL	NULL

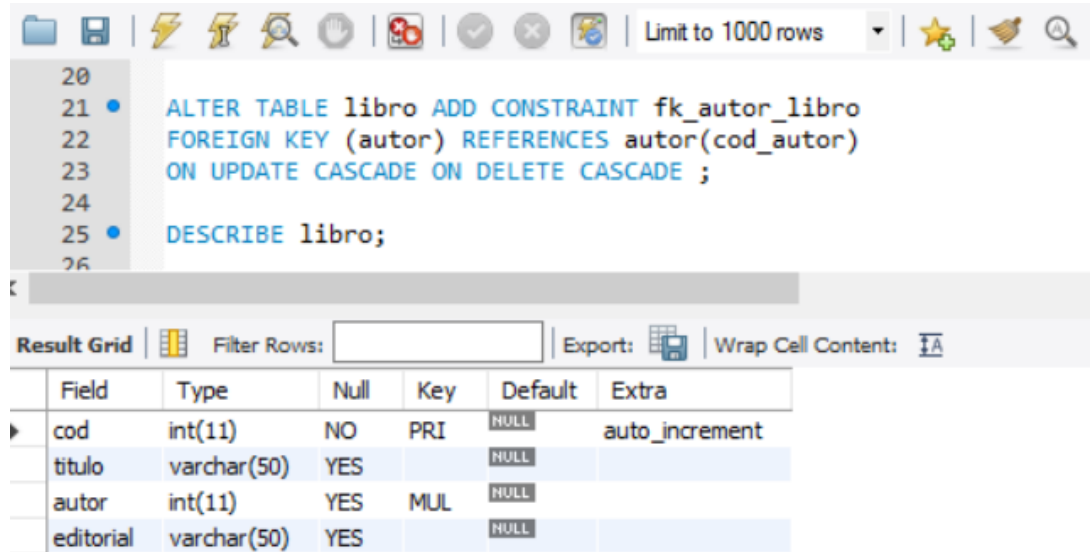
Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

CLAVE AJENA

Relacionamos las 2 tablas
Con una clave ajena

No aparece el código que lo ejecuta, pero cambiamos el atributo autor a INT para que sea del mismo tipo que cod de la tabla autor y pueda ser clave ajena de cod de la tabla autor.



The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for file operations, execution, and search, along with a 'Limit to 1000 rows' dropdown. The SQL editor displays the following code:

```
20  
21 • ALTER TABLE libro ADD CONSTRAINT fk_autor_libro  
22 FOREIGN KEY (autor) REFERENCES autor(cod_autor)  
23 ON UPDATE CASCADE ON DELETE CASCADE ;  
24  
25 • DESCRIBE libro;  
26
```

Below the editor is the 'Result Grid' section, which includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The grid displays the structure of the 'libro' table:

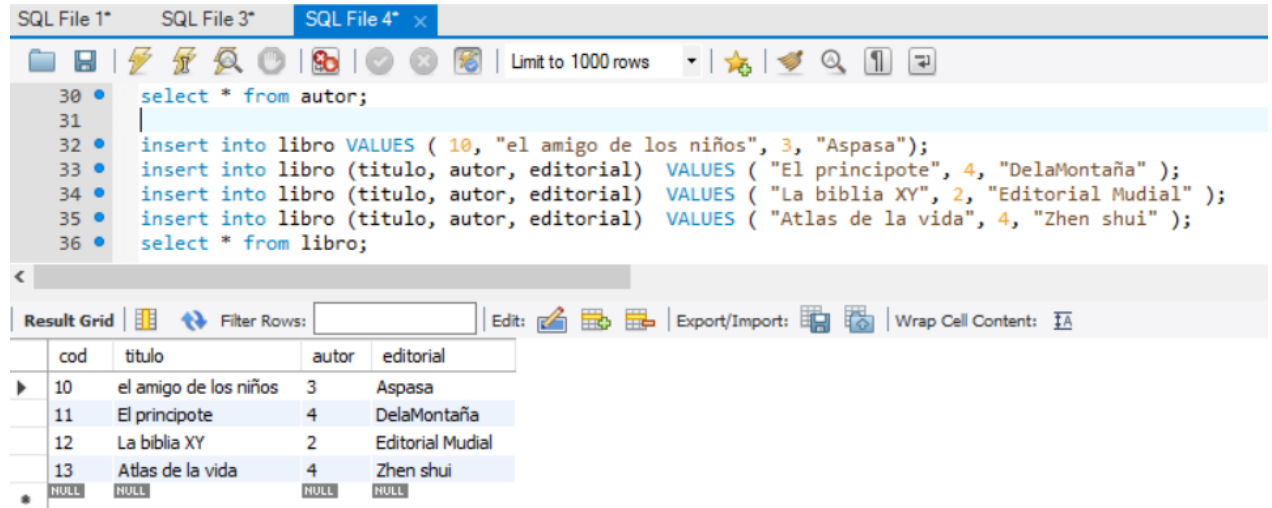
Field	Type	Null	Key	Default	Extra
cod	int(11)	NO	PRI	NULL	auto_increment
titulo	varchar(50)	YES		NULL	
autor	int(11)	YES	MUL	NULL	
editorial	varchar(50)	YES		NULL	

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

INSERTAMOS DATOS

Insertamos datos
en la otra tabla



The screenshot shows the Oracle SQL Developer interface. The top pane displays a SQL script with the following queries:

```
30 • select * from autor;  
31 |  
32 • insert into libro VALUES ( 10, "el amigo de los niños", 3, "Aspasa");  
33 • insert into libro (titulo, autor, editorial) VALUES ( "El principote", 4, "DelaMontaña" );  
34 • insert into libro (titulo, autor, editorial) VALUES ( "La biblia XY", 2, "Editorial Mudial" );  
35 • insert into libro (titulo, autor, editorial) VALUES ( "Atlas de la vida", 4, "Zhen shui" );  
36 • select * from libro;
```

The bottom pane shows the 'Result Grid' for the last query. It displays a table with 5 columns: 'cod', 'titulo', 'autor', and 'editorial'. The data rows are as follows:

	cod	titulo	autor	editorial
▶	10	el amigo de los niños	3	Aspasa
	11	El principote	4	DelaMontaña
	12	La biblia XY	2	Editorial Mudial
	13	Atlas de la vida	4	Zhen shui
*	NULL	NULL	NULL	NULL

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

UNION

¿Qué pasa si usamos esta sentencia?

```
36  
37 • select * from libro  
38 UNION  
39 select * from autor;
```


Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

UNION

ERROR!!!

Error por no tener el mismo número de campos.

```
36  
37 • select * from libro  
38 UNION  
39 select * from autor;
```

❌ 164 16:23:36 select * ... Error Code: 1222. The used SELECT statements have a different number of columns

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

UNION

Ahora sí funciona
porque coincide en columnas.

```
37 • select cod, titulo from libro
38 UNION
39 select * from autor;
```

Result Grid |   Filter Rows: | Export

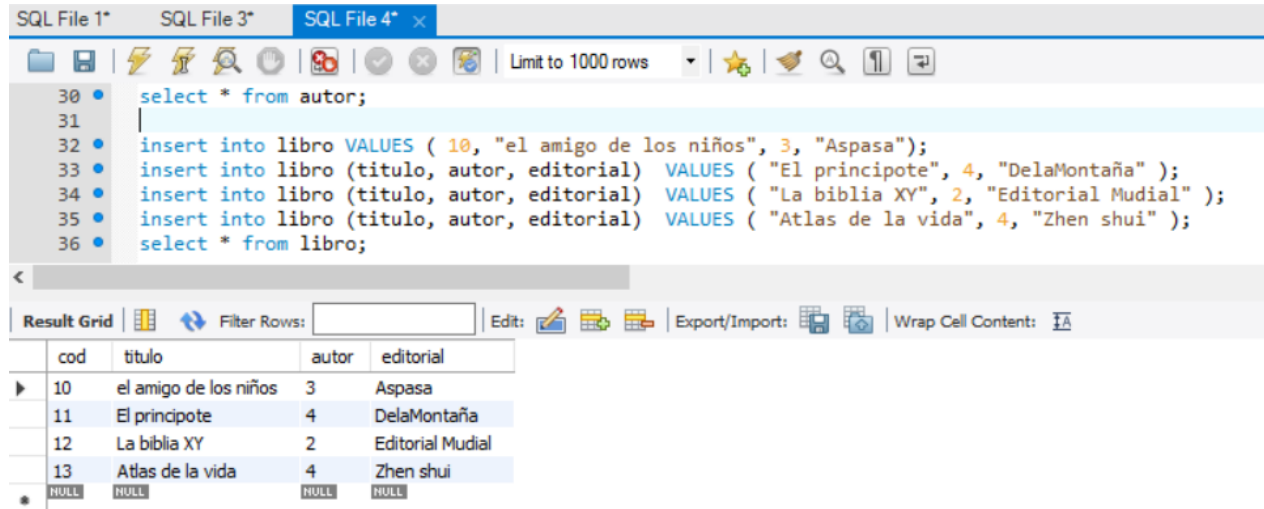
	cod	titulo
10		el amigo de los niños
11		El principote
12		La biblia XY
13		Atlas de la vida

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

¿?

¿Cómo haríamos para seleccionar un listado de los libros en cuyo título exista la palabra “de”?



The screenshot shows the Oracle SQL Developer interface. The top pane displays a SQL script with the following queries:

```
30 • select * from autor;
31 |
32 • insert into libro VALUES ( 10, "el amigo de los niños", 3, "Aspasa");
33 • insert into libro (titulo, autor, editorial) VALUES ( "El principote", 4, "DelaMontaña" );
34 • insert into libro (titulo, autor, editorial) VALUES ( "La biblia XY", 2, "Editorial Mudial" );
35 • insert into libro (titulo, autor, editorial) VALUES ( "Atlas de la vida", 4, "Zhen shui" );
36 • select * from libro;
```

The bottom pane shows the 'Result Grid' for the last query, displaying a list of books. The grid has columns for 'cod', 'titulo', 'autor', and 'editorial'.

cod	titulo	autor	editorial
10	el amigo de los niños	3	Aspasa
11	El principote	4	DelaMontaña
12	La biblia XY	2	Editorial Mudial
13	Atlas de la vida	4	Zhen shui
* NULL	NULL	NULL	NULL

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

LIKE

Usamos la cláusula **LIKE** !!!

```
40  
41 • SELECT *  
42 FROM libro  
43 WHERE titulo LIKE "%de%";  
44
```

Expresión regular, que sirve para pedir que busque la palabra "de". Los % significan que nos da igual lo que haya antes y después del "de".

Result Grid				
Filter Rows: Edit:				
	cod	titulo	autor	editorial
▶	10	el amigo de los niños	3	Aspasa
	13	Atlas de la vida	4	Zhen shui
*	NULL	NULL	NULL	NULL

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

¿Cómo podemos saber cuántos libros ha escrito el autor “Corregino de Alba”?

¿Qué comando debemos usar?

TABLA AUTOR

	cod_autor	nombre
▶	1	Juan Palomero
	2	Perico Belmonte
	3	Augusto Filemino
	4	Corregino De Alba
	5	Maria Angustas
✱	NULL	NULL

TABLA LIBROS

Result Grid		Filter Rows:			Edit
	cod	titulo	autor	editorial	
▶	10	el amigo de los niños	3	Aspasa	
	11	El principote	4	DelaMontaña	
	12	La biblia XY	2	Editorial Mudial	
	13	Atlas de la vida	4	Zhen shui	
✱	NULL	NULL	NULL	NULL	

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

COUNT (*)

LO PODEMOS SABER CON LA
CLÁUSULA COUNT (*)

Cuenta cuantos registros cumplen la restricción.

Restricción.

```
45  
46 • SELECT count(*) from libro where autor=4;
```

<

Result Grid |  Filter Rows: | Export:  | Wrap Cell

	count(*)
▶	2

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

¿?

¿Y SI QUEREMOS QUE TAMBIÉN APAREZCA EL NOMBRE DEL AUTOR?

Result Grid	Filter Rows:
cod_autor	nombre
1	Juan Palomero
2	Perico Belmonte
3	Augusto Filemino
4	Corregino De Alba
5	Maria Angustas
NULL	NULL

Result Grid



Filter Rows:

Edit

	cod	titulo	autor	editorial
▶	10	el amigo de los niños	3	Aspasa
	11	El principote	4	DelaMontaña
	12	La biblia XY	2	Editorial Mudial
	13	Atlas de la vida	4	Zhen shui
✱	NULL	NULL	NULL	NULL

USAREMOS EL COMANDO **INNER JOIN**

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

INNER JOIN

	cod_autor	nombre
▶	1	Juan Palomero
	2	Perico Belmonte
	3	Augusto Filemino
	4	Corregino De Alba
	5	Maria Angustas
*	NULL	NULL

	cod	titulo	autor	editorial
▶	10	el amigo de los niños	3	Aspasa
	11	El principote	4	DelaMontaña
	12	La biblia XY	2	Editorial Mudial
	13	Atlas de la vida	4	Zhen shui
*	NULL	NULL	NULL	NULL

AL COMANDO **INNER JOIN**, AL CUAL SE LE SUMINISTRAN DOS TABLAS

```
45  
46 • SELECT autor.nombre, count(*) as num_libros  
47 from libro INNER JOIN autor |  
48 ON autor.cod_autor = libro.autor  
49 where autor=4;
```

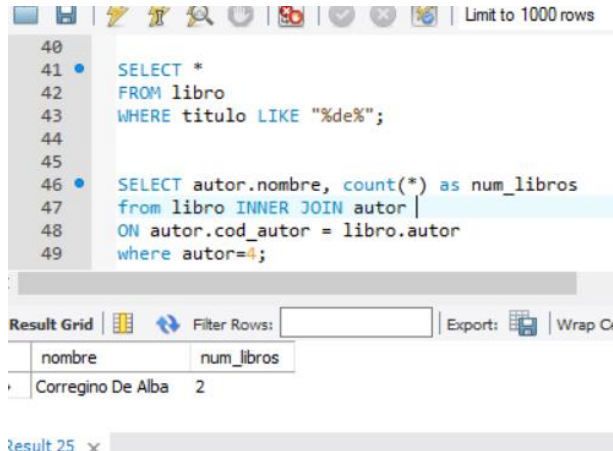
	nombre	num_libros
▶	Corregino De Alba	2

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

¿Y cómo pondríamos la cantidad de libros que ha escrito CADA autor?

¿?



The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for file operations, editing, and execution. Below the toolbar, two SQL queries are listed in a numbered list:

```
40 • SELECT *
41 FROM libro
42 WHERE titulo LIKE "%de%";
43
44
45 • SELECT autor.nombre, count(*) as num_libros
46 from libro INNER JOIN autor
47 ON autor.cod_autor = libro.autor
48 where autor=4;
49
```

Below the queries is the 'Result Grid' section. It has a 'Filter Rows' input field, an 'Export' button, and a 'Wrap C' checkbox. The grid displays the results of the second query, showing the author's name and the number of books they have written. The first row shows 'Corregino De Alba' with 2 books.

nombre	num_libros
Corregino De Alba	2

At the bottom, there is a 'result 25' tab with a close button.

cod	titulo
10	el amigo de los niños
11	El principote
12	La biblia XY
13	Atlas de la vida

	cod_autor	nombre
▶	1	Juan Palomero
	2	Perico Belmonte
	3	Augusto Filemino
	4	Corregino De Alba
	5	Maria Angustas
*	NULL	NULL

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

GROUP BY

Agrupamos por Autor con GROUP BY
y usamos COUNT

```
57  
58 • SELECT autor.nombre, COUNT(*)  
59      from autor inner join libro  
60      ON libro.autor = autor.cod_autor  
61      GROUP BY (autor);  
62
```

Result Grid |   Filter Rows: | Export: 

nombre	COUNT(*)
Perico Belmonte	1
Augusto Filemino	1
Corregino De Alba	2

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

GROUP BY

HAY EQUIVALENCIA ENTRE
INNER JOIN Y LA COMA (,)

```
57
58 • SELECT autor.nombre, COUNT(*)
59   from autor inner join libro
60   ON libro.autor = autor.cod_autor
61   GROUP BY (autor);
62
```

Result Grid | Filter Rows: | Export:

nombre	COUNT(*)
Perico Belmonte	1
Augusto Filemino	1
Corregino De Alba	2

```
58 • SELECT autor.nombre, COUNT(*)
59   from autor, libro
60   WHERE libro.autor = autor.cod_autor
61   GROUP BY (autor);
62
```

Result Grid | Filter Rows: | Export: | W

nombre	COUNT(*)
Perico Belmonte	1
Augusto Filemino	1
Corregino De Alba	2

Result 40 x

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

INNER JOIN VS COMA (,)

```
57
58 • SELECT autor.nombre, COUNT(*)
59   from autor inner join libro
60   ON libro.autor = autor.cod_autor
61   GROUP BY (autor);
62
```

Result Grid | Filter Rows: | Export:

nombre	COUNT(*)
Perico Belmonte	1
Augusto Filemino	1
Corregino De Alba	2

```
58 • SELECT autor.nombre, COUNT(*)
59   from autor, libro
60   WHERE libro.autor = autor.cod_autor
61   GROUP BY (autor);
62
```

Result Grid | Filter Rows: | Export: | W

nombre	COUNT(*)
Perico Belmonte	1
Augusto Filemino	1
Corregino De Alba	2

Result 40 x