

Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

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/Imp

	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

¿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: | Edit:

	codigo	nombre	poblacion	extension	fundacion	capital
▶	1	Venezuela	35000000	1000000	1815-04-11	99
•	NULL	NULL	NULL	NULL	NULL	NULL

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

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 ×

Output

Action Output

#	Time	Action
✓ 121	11:38:21	INSERT INTO granciudad (nom
✓ 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
7     id INT PRIMARY KEY auto_increment,
8     nombre VARCHAR(10),
9     poblacion INT
10
11 )Engine=InnoDB;
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);
19
```

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 graciudad;`. Below the editor, the 'Result Grid' tab is active, displaying the results of the query. The grid has two columns: 'nombre' and 'poblacion/1000'. The results are as follows:

nombre	poblacion/1000
madrid	4000.0000
barcelona	2000.0000
valencia	1000.0000

Below the result grid, the 'Output' tab is visible, showing the 'Action Output' log. The log contains the following entries:

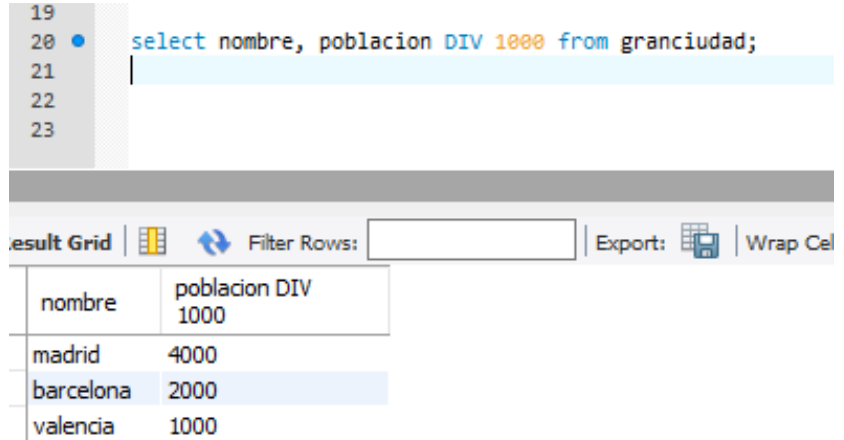
#	Time	Action
✓ 129	11:53:17	create database ejemplo
✓ 130	11:53:17	USE ejemplo
✓ 131	11:53:17	create table graciudad (id INT PRIMARY KEY auto_increment
✓ 132	11:53:18	ALTER TABLE graciudad auto_increment = 1
✓ 133	11:53:18	INSERT INTO graciudad (nombre, poblacion) VALUES ('madrid

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 query results in a table. The table has two columns: 'nombre' and 'poblacion DIV 1000'. The results are as follows:

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 granciedad (nombre, poblacion) VALUES
21    ("bilbao", 800000),
22    ("sevilla", 900000);
23
24  • select * from granciedad;
```

Result Grid

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

granciedad 7 ×

Output

Action Output

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

Data Manipulation Language (DML)

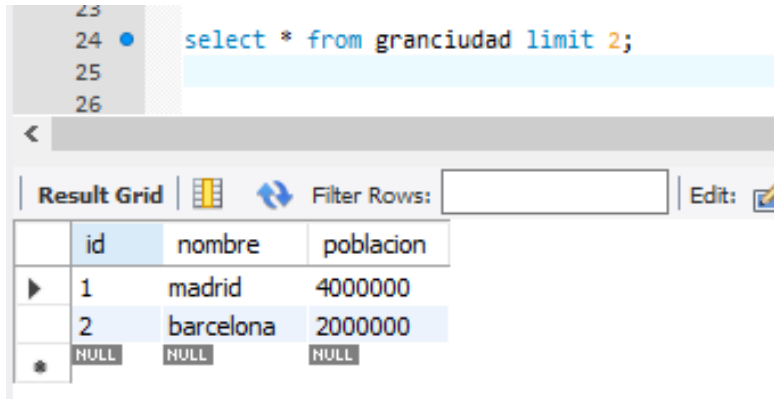
EJEMPLOS en WORKBENCH

USO del LIMIT

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

¿Qué pasará si ponemos limit 2,3?

Nos mostrará a partir del segundo registro (en este caso empezaría en 3), 3 registros. El primer valor nos muestra a partir de dónde cuenta y la segunda cuántos valores tiene que mostrar.



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

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

Data Manipulation Language (DML)

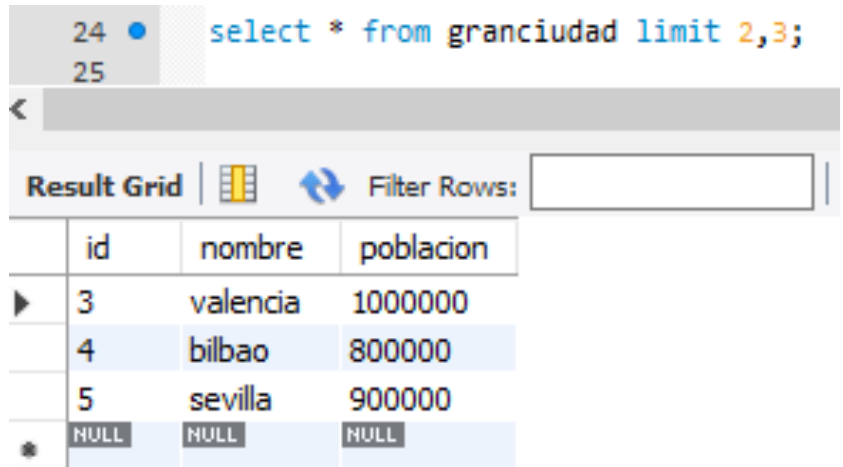
EJEMPLOS en WORKBENCH

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 statement is entered in the editor: `select * from granciudad limit 2,3;`. Below the editor, the 'Result Grid' tab is active, displaying the results of the query. The results are shown in a table with four columns: 'id', 'nombre', and 'poblacion'. The first three rows are highlighted in blue, representing the results returned by the query. The fourth row is a placeholder with 'NULL' values. The 'Filter Rows' button is visible to the right of the table.

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

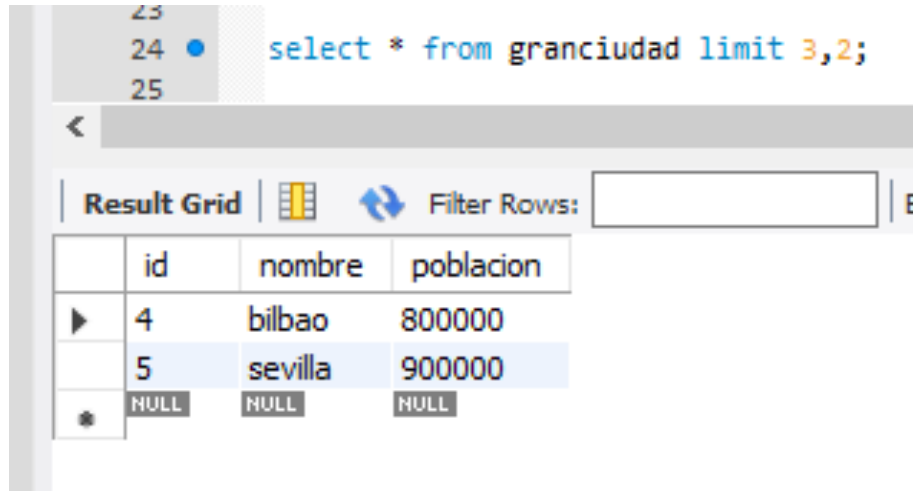
Data Manipulation Language (DML)

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 row is highlighted with a blue background and contains the values 4, bilbao, and 800000. The second row also has a blue background and contains 5, sevilla, and 900000. The third row is not highlighted and contains NULL, NULL, and NULL. A scrollbar on the right indicates that there are more rows in the result set than are currently visible.




	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 ×



Data Manipulation Language (DML)

EJEMPLOS en WORKBENCH

Modificamos la tabla granciudad

Aquí vemos que podemos insertar nuevos campos de en una tabla que ya está rellena. Aunque hasta que no tenga valores estará rellena de valores nulos

```
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 x

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
```

<

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

	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

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
```

< 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

```
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

	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