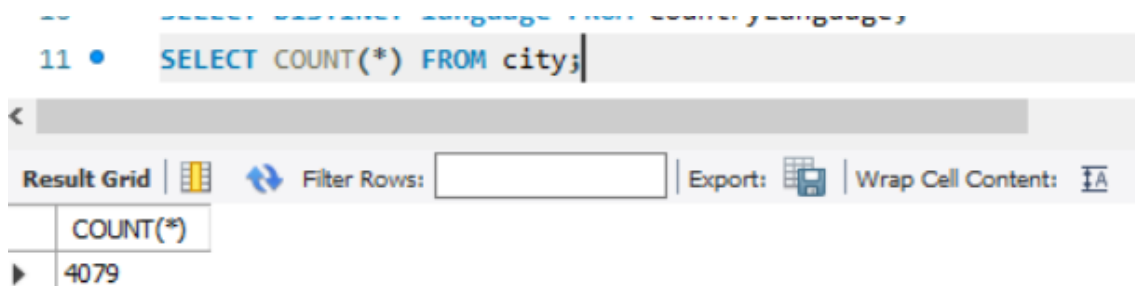


## Consultas de resumen

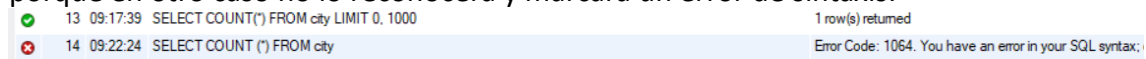
Podemos generar consultas complejas que resuman información obtenida de las tablas. Para hacer uso de estos resúmenes tenemos las funciones de columna que consiguen convertir un conjunto de registros en una información más simple, resultado de un cálculo.

Por ejemplo, una de estas funciones es **COUNT** que nos devuelve el número de registros que cumplan la condición de la consulta en la que se incluye, así si queremos conocer cuántas ciudades hay en la tabla, sin importarnos cuales sean, pondremos:

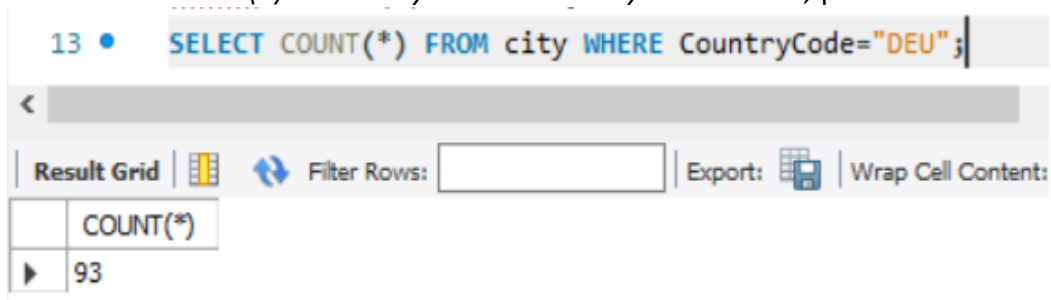
**SELECT COUNT(\*) FROM city;** y los resultados serán los de la imagen de abajo.



**Atención:** es importante señalar que **COUNT** no devuelve las filas afectadas, sino el número de ellas. En el caso anterior 4079, es decir, todas las de la tabla. También es preciso decir que **no debe haber espacios en blanco** entre **COUNT** y los paréntesis porque en otro caso no lo reconocerá y marcará un error de sintaxis:



Otro ejemplo de uso; si queremos saber el número de ciudades alemanas habremos de poner: **SELECT COUNT(\*) FROM city WHERE CountryCode="DEU";** para obtener:



La función **COUNT** hace una distinción entre dos consultas, si se le piden todos los campos de la tabla (con \*) incluirá en la cuenta los nulos si existen en esa tabla, mientras que si se le pide que cuente una columna en el resultado no tiene en cuenta los valores nulos.

Otras funciones de columna disponibles son **SUM**, **AVG**, **MIN**, **MAX** además de **COUNT**.

Vamos a ver su funcionamiento. Con **SUM(Expresión)** se nos muestra la suma de los valores indicados en el argumento. Así la consulta **SELECT SUM(Population) FROM city WHERE CountryCode="DEU";** nos dará el resultado de la suma de la población de las ciudades alemanas de la base de datos como se puede ver en la imagen de abajo:

```
14 • SELECT SUM(Population) FROM city WHERE CountryCode="DEU";
```



The screenshot shows a database interface with a toolbar containing 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar is a table with one row and one column. The column header is 'SUM(Population)' and the value in the row is '26245483'.

SUM(Population)
26245483

Con **AVG(Expresión)** será la media de los valores del parámetro la información que nos entregará, lo cual puede resultar muy útil para emplearlo en estudios estadísticos. Siguiendo con el ejemplo anterior para saber la media de población de las ciudades alemanas haremos la consulta **SELECT AVG(Population) FROM city WHERE CountryCode='DEU';** y obtendremos el resultado:

```
15 • SELECT AVG (Population) FROM city WHERE CountryCode='DEU';
```



The screenshot shows a database interface with a toolbar containing 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar is a table with one row and one column. The column header is 'AVG (Population)' and the value in the row is '282209.4946'.

AVG (Population)
282209.4946

La función **MIN(Expresión)** será la que nos calcule el mínimo de los valores. Aplicado a nuestro ejemplo: **SELECT MIN(Population) FROM city WHERE CountryCode='DEU';** para ver:

```
16 • SELECT MIN(Population) FROM city WHERE CountryCode='DEU';
```




The screenshot shows a database interface with a toolbar containing 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar is a table with one row and one column. The column header is 'MIN(Population)' and the value in the row is '89667'.

MIN(Population)
89667

Como es previsible también existe la función **MAX(Expresión)** que es la encargada de calcular el máximo de los valores. Aplicado a nuestro ejemplo: **SELECT MAX(Population) FROM city WHERE CountryCode="DEU";** para ver:

17 • `SELECT MAX(Population) FROM city WHERE CountryCode="DEU";`



The screenshot shows a database interface with a toolbar containing 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar is a table with two columns: 'MAX(Population)' and a single row with the value '3386667'.

MAX(Population)
3386667

## Agrupaciones

Con las consultas de resumen se pueden hacer agrupaciones de registros. Para ello nos valdremos de la cláusula **GROUP BY**. Si le indicamos a través de qué campo (columna) queremos agrupar nos mostrará los resultados de esa manera. La sintaxis es:

`SELECT ... [FROM ...] [WHERE ...]`

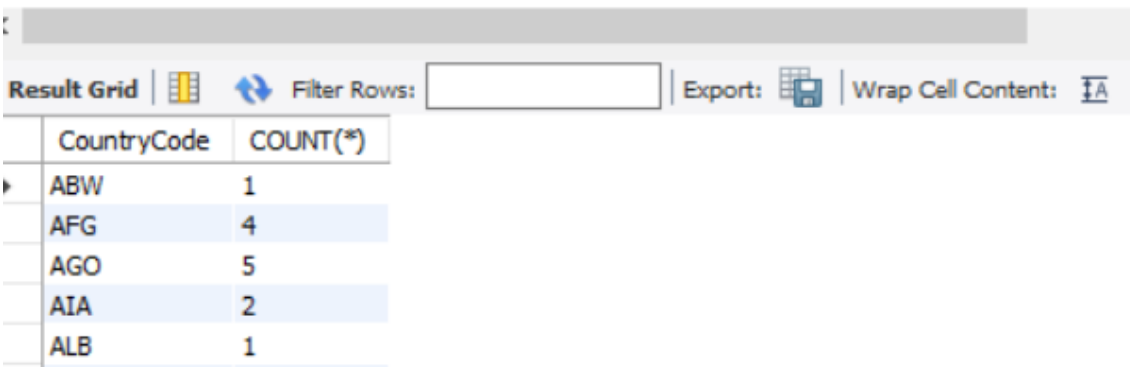
`[GROUP BY Expresión1 [, Expresión2]...]`

`[ORDER BY {nombreColumna | Expresión | Posición} [ASC | DESC], ...];`

Hay que fijarse en que *GROUP BY* va antes que *ORDER BY*.

Por ejemplo, deseamos un listado de los códigos de país con las ciudades que cada uno tiene en la base de datos, la consulta a realizar y su resultado se ven en la imagen siguiente:

21 • `SELECT CountryCode, COUNT(*) FROM city GROUP BY CountryCode;`



The screenshot shows a database interface with a toolbar containing 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar is a table with two columns: 'CountryCode' and 'COUNT(\*)'. The table contains five rows of data.

CountryCode	COUNT(*)
ABW	1
AFG	4
AGO	5
AIA	2
ALB	1

Este resultado se contrapone al que hicimos con anterioridad en el que pedíamos los códigos de países y los nombres de las ciudades (`SELECT CountryCode, Name FROM city;`) y obteníamos una fila por ciudad en la base de datos (más de 4000); con la consulta hecha ahora con *GROUP BY* hemos obtenido 232, una fila por país que es el criterio de agrupación elegido.

Podemos hacer la selección de las ciudades más pobladas de cada país sin más que escribir la consulta adecuada:

17 • `SELECT CountryCode,Name,MAX(Population) FROM city GROUP BY CountryCode;`

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
CountryCode	Name	MAX(Population)	
ABW	Oranjestad	29034	
AFG	Kabul	1780000	
AGO	Luanda	2022000	
AIA	South Hill	961	
ALB	Tirana	270000	
AND	Andorra la Vella	21189	
ANT	Willemstad	2345	
ARE	Dubai	669181	
ARG	Buenos Aires	2982146	

Pero, atención, si no se incluye el *GROUP BY* en la consulta anterior, ¿qué obtendríamos?:

18 • `SELECT CountryCode,Name,MAX(Population) FROM city;`

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
CountryCode	Name	MAX(Population)	
AFG	Kabul	10500000	

Esa respuesta es la que nos ha ofrecido el SGBD y NO es la correcta, aparentemente ha funcionado bien porque no nos ha dado un error de sintaxis, pero no es la información que queremos. Esto ha ocurrido porque las funciones de columna si se mezclan con las columnas de una tabla se ha de usar la cláusula **GROUP BY**.

**IMPORTANTE:** para cada agrupación que incluya columnas y funciones de columna (COUNT, AVG, MAX, ...) se ha de seleccionar el nombre de la columna por la que se agrupa mediante un **GROUP BY**.

## Filtros de grupos

Los filtros de grupos han de realizarse a través de la cláusula HAVING puesto que WHERE actúa antes de agrupar los registros. Es decir, si lo que pretendemos es filtrar resultados calculados mediante agrupaciones debemos usar HAVING que tiene la siguiente sintaxis:

`SELECT ... [FROM ...] [WHERE ...]`

`[GROUP BY Expresión1 [, Expresión2]...]`

`[HAVING FiltroGrupos]`

[ORDER BY {nombreColumna | Expresión | Posición} [ASC | DESC], ...];

En otras palabras, así como *WHERE* selecciona o rechaza tuplas individuales *HAVING* hace lo mismo con un grupo de registros. Con un ejemplo los veremos más claro, para seleccionar los países que tienen más de 50 ciudades en la base de datos lo tendríamos que poner como aparece en la ilustración:

20 • `SELECT CountryCode, COUNT(*) FROM city GROUP BY CountryCode HAVING COUNT(*)>50;`

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

	CountryCode	COUNT(*)
▶	ARG	57
▶	BRA	250
▶	CHN	363
▶	DEU	93
▶	ESP	59
▶	GBR	81
▶	IDN	85
▶	IND	341

Otro ejemplo, seleccionar los distritos de España que tienen más de 2 ciudades en la base de datos, y con el número de ciudades que presenta cada uno. Para resolver esta consulta habría que hacer:

29 • `SELECT District, COUNT(*) FROM city WHERE CountryCode='ESP' GROUP BY District HAVING COUNT(*)>2;`

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

	District	COUNT(*)
▶	Madrid	8
▶	Katalonia	9
▶	Valencia	4
▶	Andalusia	12
▶	Baskimaa	4
▶	Canary Islands	3
▶	Castilla and LeÃ³n	4
▶	Galicia	4