

## Datos de muestra

## País

id	nombre	poblacion	area
1	Francia	6660000	640680
2	Alemania	80700000	357000
...	...	...	...

## Ciudad

id	nombre	pais_id	poblacion	puntuacion
1	París	1	2243000	5
2	Berlín	2	3460000	3
...	...	...	...	...

## Consulta de varias tablas

## INNER JOIN

JOIN (o explícitamente INNER JOIN) devuelve las filas que tienen valores coincidentes en ambas tablas.

SELECT ciudad.nombre , pais.nombre FROM ciudad [INNER] JOIN pais ON ciudad.pais\_id = pais.id;

Ciudad id	nombre	pais_id	País id	nombre
1	París	1	1	Francia
2	Berlín	2	2	Alemania
3	Varsovia	4	3	Islandia

## LEFT JOIN

LEFT JOIN devuelve todas las filas de la tabla izquierda con filas correspondientes de la tabla derecha. Si no hay ninguna fila coincidente, se devuelven NULL como valores de la segunda tabla.

SELECT ciudad.nombre , pais.nombre FROM ciudad LEFT JOIN pais ON ciudad.pais\_id = pais.id;

Ciudad id	nombre	pais_id	País id	nombre
1	París	1	1	Francia
2	Berlín	2	2	Alemania
3	Varsovia	4	NULL	NULL

## RIGHT JOIN

RIGHT JOIN devuelve todas las filas de la tabla derecha con filas correspondientes de la tabla izquierda. Si no hay ninguna fila coincidente, se devuelven NULL como valores de la tabla izquierda.

SELECT ciudad.nombre , pais.nombre FROM ciudad RIGHT JOIN pais ON ciudad.pais\_id = pais.id;

Ciudad id	nombre	pais_id	País id	nombre
1	París	1	1	Francia
2	Berlín	2	2	Alemania
NULL	NULL	NULL	3	Islandia

## FULL JOIN

FULL JOIN (o explícitamente FULL OUTER JOIN) devuelve todas las filas de ambas tablas - si no hay ninguna fila que coincida en la segunda tabla, se devuelven NULL.

SELECT ciudad.nombre , pais.nombre FROM ciudad FULL [OUTER] JOIN pais ON ciudad.pais\_id = pais.id;

Ciudad id	nombre	pais_id	País id	nombre
1	París	1	1	Francia
2	Berlín	2	2	Alemania
3	Varsovia	4	NULL	NULL
NULL	NULL	NULL	3	Islandia

MÁS



## Consulta de una sola tabla

Obtener todas las columnas de la tabla de países:

SELECT \* FROM país ;

Obtener las columnas de id y nombre de la tabla de ciudades:

SELECT id , nombre FROM ciudad;

Obtener los nombres de las ciudades ordenados por la columna de calificación en el orden ascendente (por defecto):

SELECT nombre FROM ciudad ORDER BY puntuacion ASC;

Obtener los nombres de las ciudades ordenados por la columna de calificación en orden descendente:

SELECT nombre FROM ciudad ORDER BY puntuacion DESC;

## Filtrado de salida

## Operadores de comparación

Obtener los nombres de las ciudades que tienen una calificación superior a 3:

SELECT nombre FROM ciudad WHERE puntuacion > 3;

Obtener los nombres de las ciudades que no son ni Berlín ni Madrid:

SELECT nombre FROM ciudad WHERE nombre != 'Berlín' AND nombre != 'Madrid';

## Operadores de texto

Obtener los nombres de las ciudades que comienzan con una "P" o terminan con una "s":

SELECT nombre FROM ciudad WHERE nombre LIKE ' P % ' OR name LIKE ' % s ';

Obtener los nombres de las ciudades que comienzan con cualquier letra seguida de ' ublin ' (como Dublín en Irlanda o Lublin en Polonia):

SELECT nombre FROM ciudad WHERE nombre LIKE ' \_ublin ';

## Otros operadores

Obtener los nombres de las ciudades que tienen una población entre 500K y 5M:

SELECT nombre FROM ciudad WHERE poblacion BETWEEN 500000 Y 5000000;

Obtener los nombres de las ciudades que no tienen un valor de clasificación:

SELECT nombre FROM ciudad WHERE puntuacion IS NOT NULL;

Obtener los nombres de las ciudades que se encuentran en países con IDs 1, 4, 7, u 8:

SELECT nombre FROM ciudad WHERE pais\_id IN (1, 4, 7, 8);

## Consulta de varias tablas

### CROSS JOIN

CROSS JOIN devuelve todas las combinaciones posibles de filas de ambas tablas. Hay dos sintaxis disponibles.

```
SELECT ciudad.nombre, pais.nombre FROM ciudad CROSS JOIN pais;
```

```
SELECT ciudad.nombre, pais.nombre FROM ciudad, pais;
```

Ciudad			País	
id	nombre	pais_id	id	nombre
1	París	1	1	Francia
1	París	1	2	Alemania
2	Berlín	2	1	Francia
2	Berlín	2	2	Alemania

### NATURAL JOIN

NATURAL JOIN unirá las tablas por todas las columnas con el mismo nombre.

```
SELECT ciudad.nombre, pais.nombre FROM ciudad
```

```
NATURAL JOIN pais;
```

Ciudad			País	
pais_id	id	nombre	nombre	id
6	6	San Marino	San Marino	6
7	7	Ciudad del Vaticano	Ciudad del Vaticano	7
5	9	Grecia	Grecia	9
10	11	Mónaco	Mónaco	10

NATURAL JOIN utilizó estas columnas para hacer coincidir las filas:

```
ciudad.id , ciudad.nombre , pais.id , pais.nombre
```

El NATURAL JOIN se utiliza muy raramente.

## Subconsultas

Una subconsulta es una consulta anidada dentro de otra consulta, o dentro de otra subconsulta.

Existen diferentes tipos de subconsultas.

### Valor único

La subconsulta más sencilla devuelve exactamente una columna y exactamente una fila. Puede ser utilizada con los operadores de comparación =, <, <=, >, o >=.

Esta consulta encuentra ciudades con la misma calificación que París:

```
SELECT nombre FROM ciudad WHERE puntuacion = (
SELECT puntuacion FROM ciudad WHERE nombre = 'París');
```

### Valores múltiples

Una subconsulta también puede devolver varias columnas o filas. Estas subconsultas pueden ser con los operadores IN, EXISTS, ALL, o ANY.

Esta consulta encuentra ciudades en países que tienen una población superior a 20M:

```
SELECT nombre FROM ciudad WHERE pais_id IN (SELECT pais_id FROM pais WHERE poblacion > 20000000);
```

### Correlación

Una subconsulta correlacionada hace referencia a las tablas introducidas en la consulta externa.

Una subconsulta depende de la consulta externa. No puede ejecutarse independientemente de la consulta externa.

Esta consulta busca las ciudades con una población superior a la media de la población del país:

```
SELECT * FROM ciudad principal WHERE poblacion >
(SELECT AVG(poblacion) FROM ciudad media WHERE media.pais_id = principal.pais_id);
```

Esta consulta encuentra los países que tienen al menos una ciudad:

```
SELECT nombre FROM pais WHERE EXISTS(SELECT * FROM ciudad WHERE pais_id = pais.id);
```

## Agregación y Agrupación

GROUP BY agrupa las filas que tienen los mismos valores en las columnas especificadas.

Calcula resúmenes (agregados) para cada combinación única de valores.

Ciudad				
id	nombre	pais_id	Ciudad	pais_id
1	París	1	1	3
101	Marsella	1	2	3
102	Lyon	1	4	2
2	Berlín	2		
103	Hamburgo	2		
104	Munich	2		
3	Varsovia	4		
105	Cracovia	4		

### Funciones de agregación

avg(expr): Valor medio de las filas del grupo.

count(expr): Cuenta de los valores de las filas del grupo.

max(expr): Valor máximo dentro del grupo.

min(expr): Valor mínimo dentro del grupo.

sum(expr): Suma de valores dentro del grupo.

### Ejemplos de consulta

Averiguar el número de ciudades:

```
SELECT COUNT(*) FROM ciudad;
```

Averiguar el número de ciudades con valoraciones no nulas:

```
SELECT COUNT(rating) FROM ciudad;
```

Averiguar el número de valores distintivos de países:

```
SELECT COUNT(DISTINCT pais_id)
```

```
FROM ciudad;
```

Averiguar la menor y la mayor población de los países:

```
SELECT MIN(poblacion), MAX(poblacion)
FROM pais;
```

Averiguar la población total de las ciudades en los respectivos países:

```
SELECT pais_id, SUM(poblacion) FROM ciudad
GROUP BY pais_id;
```

Averiguar la valoración media de las ciudades de los respectivos países si la media es superior a 3,0:

```
SELECT pais_id, AVG (rating) FROM ciudad
GROUP BY pais_id HAVING AVG(rating) > 3.0;
```

## Operaciones de conjunto

Las operaciones de conjunto se utilizan para combinar los resultados de dos o más consultas en un resultado único. Las consultas combinadas deben devolver el mismo número de columnas y tipos de datos compatibles. Los nombres de las columnas correspondientes pueden ser diferentes.

### Ciclismo

id	nombre	pais
1	YK	DE
2	ZG	DE
3	WT	PL
...	...	...

### Patinaje

id	nombre	pais
1	YK	DE
2	DF	DE
3	AK	PL
...	...	...

### UNION

UNION combina los resultados de dos conjuntos de resultados y elimina los duplicados.

UNION ALL no elimina las filas duplicadas.

Esta consulta muestra los ciclistas alemanes junto con los patinadores alemanes:

```
SELECT nombre FROM ciclismo WHERE pais = 'DE' UNION / UNION ALL SELECT nombre FROM patinaje WHERE pais = 'DE';
```



### INTERSECT

INTERSECT devuelve sólo las filas que aparecen en ambos conjuntos de resultados.

Esta consulta muestra a los ciclistas alemanes que también son patinadores alemanes al mismo tiempo:

```
SELECT nombre FROM ciclismo WHERE pais = 'DE' INTERSECT SELECT nombre FROM patinaje WHERE pais = 'DE';
```



### EXCEPT

EXCEPT devuelve sólo las filas que aparecen en el primer conjunto de resultados pero que no aparecen en el segundo.

Esta consulta muestra a los ciclistas alemanes a menos que también sean patinadores alemanes al mismo tiempo:

```
SELECT nombre FROM ciclismo WHERE pais = 'DE' EXCEPT / MINUS SELECT nombre FROM patinaje WHERE pais = 'DE';
```

