Chuleta de SQL

LearnSQL.es

SQL, o *Structured Query Language (lenguaje de consulta estructurada)*, es un lenguaje para comunicarse con bases de datos. Se utiliza para seleccionar datos específicos y crear informes complejos. Hoy en día, SQL es un lenguaje de datos universal y se utiliza en prácticamente todas las tecnologías que procesan datos.

CONTENTS

Esta chuleta ha sido elaborada por **LearnSQL.es** como parte de su programa de formación en SQL. Conozca nuestras otras **chuletas** de SQL.

Chuleta de SQL

LearnSQL.es

EJEMPLO DE DATOS

PAIS

id	nombre	poblacion	superficie
1	Francia	66600000	640680
2	Alemania	80700000	357000
...	...	...	...

CIUDAD

id	nombre	id_pais	poblacion	clasificacion
1	París	1	2243000	5
2	Berlín	2	3460000	3
...	...	...	...	...

Esta chuleta ha sido elaborada por **LearnSQL.es** como parte de su programa de formación en SQL. Conozca nuestras otras **chuletas** de SQL.

Chuleta de SQL

LearnSQL.es

CONSULTAR A UNA SOLA TABLA

Recuperar todas las columnas de la tabla `pais`:

```
SELECT *
FROM pais;
```

Recuperar las columnas `id` y `nombre` de la tabla `ciudad`:

```
SELECT id, nombre
FROM ciudad;
```

Recuperar los nombres de las ciudades ordenados por la columna `clasificacion` en orden ascendente (ASC):

```
SELECT nombre
FROM ciudad
ORDER BY clasificacion [ASC];
```

Recuperar los nombres de las ciudades ordenados por la columna `clasificacion` en orden descendente (DESC):

```
SELECT nombre
FROM ciudad
ORDER BY clasificacion DESC;
```

ALIAS

COLUMNAS

```
SELECT nombre AS nombre_ciudad
FROM ciudad;
```

TABLAS

```
SELECT pa.nombre, ci.nombre
FROM ciudad AS ci
JOIN pais AS pa
  ON ci.id_pais = pa.id;
```

Esta chuleta ha sido elaborada por **LearnSQL.es** como parte de su programa de formación en SQL. Conozca nuestras otras **chuletas** de SQL.

Chuleta de SQL

LearnSQL.es

FILTRAR RESULTADOS

OPERADORES DE COMPARACIÓN

Recuperar los nombres de las ciudades cuya clasificación sea superior a 3:

```
SELECT nombre
FROM ciudad
WHERE clasificacion > 3;
```

Recuperar los nombres de las ciudades que no sean ni Berlín ni Madrid:

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

OPERADORES DE TEXTO

Recuperar los nombres de las ciudades que empiecen por "P" o terminen por "s":

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

Recuperar los nombres de las ciudades que empiecen por cualquier letra seguida de "adrid" (como Madrid, de España):

```
SELECT nombre
FROM ciudad
WHERE nombre LIKE '_adrid';
```

Chuleta de SQL

LearnSQL.es

OTROS OPERADORES

Recuperar los nombres de las ciudades con poblaciones comprendidas entre 500.000 y 5 millones de habitantes:

```
SELECT nombre
FROM ciudad
WHERE poblacion BETWEEN 500000 AND 5000000;
```

Recuperar los nombres de las ciudades que tienen un valor en la clasificación:

```
SELECT nombre
FROM ciudad
WHERE clasificacion IS NOT NULL;
```

Recuperar los nombres de las ciudades situadas en países cuyo ID es 1, 4, 7 u 8:

```
SELECT nombre
FROM ciudad
WHERE id_pais IN (1, 4, 7, 8);
```

Chuleta de SQL

LearnSQL.es

CONSULTAR A VARIAS TABLAS

INNER JOIN

JOIN (o explícitamente **INNER JOIN**) devuelve las filas en las que coinciden los valores en ambas tablas.

```
SELECT ciudad.nombre, pais.nombre
FROM ciudad
[INNER] JOIN pais
  ON ciudad.id_pais = pais.id;
```

CIUDAD			PAIS	
id	**nombre**	**id_pais**	**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 las filas correspondientes de la tabla derecha. Si no coincide ninguna fila de la tabla derecha, devuelve **NULL** como valores de la tabla derecha.

```
SELECT ciudad.nombre, pais.nombre
FROM ciudad
LEFT JOIN pais
  ON ciudad.id_pais = pais.id;
```

CIUDAD			PAIS	
id	**nombre**	**id_pais**	**id**	**nombre**
1	París	1	1	Francia
2	Berlín	2	2	Alemania
3	Varsovia	4	NULL	NULL

Esta chuleta ha sido elaborada por **LearnSQL.es** como parte de su programa de formación en SQL. Conozca nuestras otras **chuletas** de SQL.

Chuleta de SQL

LearnSQL.es

RIGHT JOIN

`RIGHT JOIN` devuelve todas las filas de la tabla derecha con las filas correspondientes de la tabla izquierda. Si no coincide ninguna fila de la tabla izquierda, devuelve **`NULL`** como valores de la tabla izquierda.

```
SELECT ciudad.nombre, pais.nombre
FROM ciudad
RIGHT JOIN pais
  ON ciudad.id_pais = pais.id;
```

CIUDAD			PAIS	
id	**nombre**	**id_pais**	**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 coincide ninguna fila de la otra tabla, devuelve valores **`NULL`**.

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

CIUDAD			PAIS	
id	**nombre**	**id_pais**	**id**	**nombre**
1	París	1	1	Francia
2	Berlín	2	2	Alemania
3	Varsovia	4	NULL	NULL
NULL	NULL	NULL	3	Islandia

Esta chuleta ha sido elaborada por **LearnSQL.es** como parte de su programa de formación en SQL. Conozca nuestras otras **chuletas** de SQL.

Chuleta de SQL

LearnSQL.es

CROSS JOIN

CROSS JOIN devuelve todas las combinaciones posibles de las filas de las dos tablas. Se puede usar con dos sintaxis diferentes.

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

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

CIUDAD			PAIS	
id	**nombre**	**id_pais**	**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 une las tablas por todas las columnas que tienen el mismo nombre.

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

CIUDAD			PAIS	
id_pais	**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 las siguientes columnas para emparejar las filas: **ciudad.id**, **ciudad.nombre**, **pais.id**, **pais.nombre**.

En la práctica, **NATURAL JOIN** rara vez se utiliza.

Esta chuleta ha sido elaborada por **LearnSQL.es** como parte de su programa de formación en SQL. Conozca nuestras otras **chuletas** de SQL.

Chuleta de SQL

LearnSQL.es

AGREGAR Y AGRUPAR

GROUP BY **agrupa** las filas que tienen los mismos valores en columnas especificadas. Genera resúmenes (agregados) para cada combinación única de valores.

CIUDAD		
id	**nombre**	**id_pais**
1	París	1
101	Marsella	1
102	Lyon	1
2	Berlín	2
103	Hamburgo	2
104	Múnich	2
3	Varsovia	4
105	Cracovia	4

CIUDAD	
id_pais	**cantidad**
1	3
2	3
4	2

FUNCIONES DE AGREGADO

- `avg(expr)` - valor medio de las filas del grupo
- `count(expr)` - número de valores de las filas del grupo
- `max(expr)` - valor máximo del grupo
- `min(expr)` - valor mínimo del grupo
- `sum(expr)` - suma de los valores del grupo

Esta chuleta ha sido elaborada por **LearnSQL.es** como parte de su programa de formación en SQL. Conozca nuestras otras **chuletas** de SQL.

Chuleta de SQL

LearnSQL.es

EJEMPLOS DE CONSULTAS

Conocer el número de ciudades:

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

Conocer el número de ciudades con una valoración que no sea nula:

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

Conocer el número de valores distintos de los países:

```
SELECT COUNT(DISTINCT id_pais)
FROM ciudad;
```

Conocer los países con menor y mayor población:

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

Conocer la población total de las ciudades en sus respectivos países:

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

Conocer la clasificación media de las ciudades en sus respectivos países, si la media es superior a 3,0:

```
SELECT id_pais, AVG(clasificacion)
FROM ciudad
GROUP BY id_pais
HAVING AVG(clasificacion) > 3.0;
```

Chuleta de SQL

LearnSQL.es

SUBCONSULTAS

Una subconsulta es una consulta anidada en otra consulta o en otra subconsulta. Existen diferentes tipos de subconsultas.

VALOR ÚNICO

Es la subconsulta más sencilla, que devuelve exactamente una columna y una fila. Puede utilizarse con los operadores de comparación =, <, <=, > o >=.

La siguiente consulta se utiliza para buscar ciudades con la misma clasificación que París:

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

VARIOS VALORES

Las subconsultas también pueden devolver varias columnas o varias filas. En estas subconsultas pueden utilizarse los operadores `IN`, `EXISTS`, `ALL` o `ANY`.

Esta consulta busca ciudades en países con más de 20 millones de habitantes:

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

Esta chuleta ha sido elaborada por **LearnSQL.es** como parte de su programa de formación en SQL. Conozca nuestras otras **chuletas** de SQL.

Chuleta de SQL

LearnSQL.es

SUBCONSULTA CORRELACIONADA

Las subconsultas correlacionadas consultan a las tablas insertadas en la consulta externa. Las subconsultas correlacionadas dependen de la consulta externa. No pueden ejecutarse de forma independiente, sin la consulta externa.

Esta consulta busca las ciudades cuya población sea superior a la población media del país:

```
SELECT *
FROM ciudad ciudad_principal
WHERE poblacion > (
  SELECT AVG(poblacion)
  FROM ciudad ciudad_media
  WHERE ciudad_media.id_pais =
ciudad_principal.id_pais
);
```

Esta consulta busca países con, al menos, una ciudad:

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

Chuleta de SQL

LearnSQL.es

OPERACIONES DE CONJUNTOS

Las operaciones de conjuntos se utilizan para combinar los resultados de dos o más consultas en un único resultado. 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 excluye los duplicados. **UNION ALL** no excluye las filas duplicadas.

Esta consulta muestra los ciclistas alemanes y los patinadores alemanes:

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

Esta chuleta ha sido elaborada por **LearnSQL.es** como parte de su programa de formación en SQL. Conozca nuestras otras **chuletas** de SQL.

Chuleta de SQL

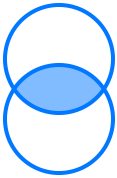
LearnSQL.es

INTERSECT

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

Esta consulta muestra los ciclistas alemanes que también son patinadores alemanes:

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

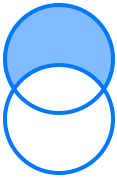


EXCEPT

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

Esta consulta muestra los ciclistas alemanes, siempre que no sean patinadores alemanes:

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



Esta chuleta ha sido elaborada por **LearnSQL.es** como parte de su programa de formación en SQL. Conozca nuestras otras **chuletas** de SQL.

Apréndalo todo en LearnSQL.es