

SQL Basics Cheat Sheet

SQL, or *Structured Query Language*, é uma linguagem para administrar banco de dados. Ela permite que você selecionem dados específicos para e criar relatórios complexos. Atualmente, o SQL é uma linguagem universal de dados. Ele é usado em praticamente todas as tecnologias que processam dados.

SAMPLE DATA (DADOS DE AMOSTRA)

COUNTRY			
id	name	population	area
1	France	66600000	640680
2	Germany	80700000	357000
...

CITY				
id	name	country_id	population	rating
1	Paris	1	2243000	5
2	Berlin	2	3460000	3
...

CONSULTA DE UMA ÚNICA TABELA

Obtém todas as colunas da tabela de **country**:

```
SELECT * FROM country;
```

Obtenha as colunas id e name da tabela **city**:

```
SELECT id, name FROM city;
```

Obtém nomes de cidades classificados pela coluna de **rating** na ordem ascendente padrão:

```
SELECT name FROM city ORDER BY rating [ASC];
```

Obtém nomes de cidades classificadas pela coluna de **rating** na ordem DESCENDENTE:

```
SELECT name FROM city ORDER BY rating DESC;
```

ALIASES

COLUNAS

```
SELECT name AS city_name FROM city;
```

TABELAS

```
SELECT co.name, ci.name FROM city AS ci  
JOIN country AS co ON ci.country_id = co.id;
```

FILTERING THE OUTPUT (FILTRAGEM DA SAÍDA)

OPERADORES DE COMPARAÇÃO

Obter nomes de cidades que tenham uma classificação acima de 3:

```
SELECT name FROM city WHERE rating > 3;
```

Obtenha nomes de cidades que não sejam nem Berlim nem Madri:

```
SELECT name FROM city WHERE name != 'Berlin'  
AND name != 'Madrid';
```

OPERADORES DE TEXTO

Busca de nomes de cidades que começam com um "P" ou terminam com um "s":

```
SELECT name FROM city WHERE name LIKE 'P%'  
OR name LIKE '%s';
```

Busque nomes de cidades que começam com qualquer letra seguida de "ublin" (como Dublin na Irlanda ou Lublin na Polônia):

```
SELECT name FROM city WHERE name LIKE '_ublin';
```

OUTROS OPERADORES

Obter nomes de cidades com população entre 500 mil e 5 milhões de habitantes:

```
SELECT name FROM city WHERE population BETWEEN 500000 AND  
5000000;
```

Obtém nomes de cidades que não perdem um valor de classificação:

```
SELECT name FROM city WHERE rating IS NOT NULL;
```

Obtém nomes de cidades que estão em países com IDs 1, 4, 7 ou 8:

```
SELECT name FROM city WHERE country_id IN (1, 4, 7, 8);
```


CONSULTA A VÁRIAS TABELAS

INNER JOIN

JOIN (ou explicitamente **INNER JOIN**) retorna linhas que têm valores correspondentes em ambas as tabelas.

```
SELECT city.name, country.name FROM city [INNER] JOIN country ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
3	Warsaw	4	3	Iceland

LEFT JOIN

LEFT JOIN retorna todas as linhas da tabela da esquerda com as linhas correspondentes da tabela da direita. Se não houver nenhuma linha correspondente, NULLs serão retornados como valores da segunda tabela.

```
SELECT city.name, country.name FROM city LEFT JOIN country ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
3	Warsaw	4	NULL	NULL

RIGHT JOIN

RIGHT JOIN retorna todas as linhas da tabela da direita com as linhas correspondentes da tabela da esquerda. Se não houver nenhuma linha correspondente, NULLs serão retornados como valores da tabela da esquerda.

```
SELECT city.name, country.name FROM city RIGHT JOIN country ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
NULL	NULL	NULL	3	Iceland

FULL JOIN

FULL JOIN (ou explicitamente **FULL OUTER JOIN**) retorna todas as linhas de ambas as tabelas - se não houver nenhuma linha correspondente na segunda tabela, NULLs serão retornados.

```
SELECT city.name, country.name FROM city FULL [OUTER] JOIN country ON city.country_id = country.id;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
2	Berlin	2	2	Germany
3	Warsaw	4	NULL	NULL
NULL	NULL	NULL	3	Iceland

CROSS JOIN

O **CROSS JOIN** retorna todas as combinações possíveis de linhas de ambas as tabelas. Há duas sintaxes disponíveis.

```
SELECT city.name, country.name FROM city CROSS JOIN country;
```

```
SELECT city.name, country.name FROM city, country;
```

CITY			COUNTRY	
id	name	country_id	id	name
1	Paris	1	1	France
1	Paris	1	2	Germany
2	Berlin	2	1	France
2	Berlin	2	2	Germany

JUNÇÃO NATURAL

O **NATURAL JOIN** unirá as tabelas por todas as colunas com o mesmo nome.

```
SELECT city.name, country.name FROM city NATURAL JOIN country;
```

CITY			COUNTRY	
country_id	id	name	name	id
6	6	San Marino	San Marino	6
7	7	Vatican City	Vatican City	7
5	9	Greece	Greece	9
10	11	Monaco	Monaco	10

O **NATURAL JOIN** usou essas colunas para fazer a correspondência entre as linhas:

`city.id`, `city.name`, `country.id`, `country.name`.

O **NATURAL JOIN** é muito raramente usado na prática.

AGGREGATION AND GROUPING

(AGREGAÇÃO E AGRUPAMENTO)

O **GROUP BY** **agrupa** as linhas que têm os mesmos valores nas colunas especificadas.

Ele calcula resumos (agregados) para cada combinação exclusiva de valores.

CITY				
id	name	country_id		
1	Paris	1		
101	Marseille	1		
102	Lyon	1		
2	Berlin	2		
103	Hamburg	2		
104	Munich	2		
3	Warsaw	4		
105	Cracow	4		

→

CITY		
country_id	count	
1	3	
2	3	
4	2	

FUNÇÕES AGREGADAS

- **avg(expr)** - valor médio das linhas dentro do grupo
- **count(expr)** - contagem de valores para linhas dentro do grupo
- **max(expr)** - valor máximo dentro do grupo
- **min(expr)** - valor mínimo dentro do grupo
- **sum(expr)** - soma dos valores dentro do grupo

CONSULTAS DE EXEMPLO

Descubra o número de cidades:

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

Descubra o número de valores de países distintos:

```
SELECT COUNT(DISTINCT country_id) FROM city;
```

Descubra a população total das cidades em seus respectivos países:

```
SELECT country_id, SUM(population) FROM city GROUP BY country_id;
```

Descubra o número de cidades com classificações não nulas:

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

Descubra a menor e a maior população de países:

```
SELECT MIN(population), MAX(population) FROM country;
```

Descubra a classificação média das cidades em seus respectivos países se a média for superior a 3,0:

```
SELECT country_id, AVG(rating) FROM city GROUP BY country_id HAVING AVG(rating) > 3.0;
```

SUBQUERIES

Uma subconsulta é uma consulta que está aninhada em outra consulta ou em outra subconsulta. Há diferentes tipos de subconsultas.

VALOR ÚNICO

A subconsulta mais simples retorna exatamente uma coluna e exatamente uma linha. Ela pode ser usada com os operadores de comparação `=`, `<`, `<=`, `>` ou `>=`.

Essa consulta encontra cidades com a mesma classificação de Paris:

```
SELECT name FROM city WHERE rating = ( SELECT rating FROM city WHERE name = 'Paris');
```

VALORES MÚLTIPLOS

Uma subconsulta também pode retornar várias colunas ou várias linhas. Essas subconsultas podem ser usadas com os operadores `IN`, `EXISTS`, `ALL` ou `ANY`.

Essa consulta encontra cidades em países que têm uma população acima de 20 milhões de habitantes:

```
SELECT name FROM city WHERE country_id IN (SELECT country_id FROM country WHERE population > 20000000);
```

CORRELACIONADO

Uma subconsulta correlacionada refere-se às tabelas introduzidas na consulta externa. Uma subconsulta correlacionada depende da consulta externa. Ela não pode ser executada independentemente da consulta externa.

Essa consulta encontra cidades com uma população maior do que a população média do país:

```
SELECT * FROM city main_city  HERE population > (SELECT
AVG(population) FROM city average_city

WHERE average_city.country_id = main_city.country_id);
```

Essa consulta encontra países que têm pelo menos uma cidade:

```
SELECT name FROM country WHERE EXISTS ( SELECT * FROM
city WHERE country_id = country.id);
```


OPERAÇÕES DE CONFIGURAÇÃO

As operações de conjunto são usadas para combinar os resultados de duas ou mais consultas em um único resultado. As consultas combinadas devem retornar o mesmo número de colunas e tipos de dados compatíveis. Os nomes das colunas correspondentes podem ser diferentes.

CYCLING		
id	name	country
1	YK	DE
2	ZG	DE
3	WT	PL
...

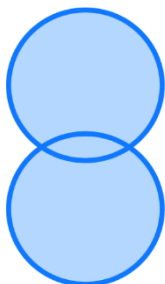
SKATING		
id	name	country
1	YK	DE
2	DF	DE
3	AK	PL
...

UNIÃO

O **UNION** combina os resultados de dois conjuntos de resultados e remove as duplicatas. O **UNION ALL** não remove as linhas duplicadas.

Essa consulta exibe ciclistas alemães e skatistas alemães:

```
SELECT name FROM cycling
WHERE country = 'DE'
UNION / UNION ALL
SELECT name FROM skating
WHERE country = 'DE';
```



INTERSECÇÃO

INTERSECT retorna apenas as linhas que aparecem em ambos os conjuntos de resultados.

Essa consulta exibe ciclistas alemães que também são skatistas alemães ao mesmo tempo:

```
SELECT name FROM cycling
WHERE country = 'DE'
INTERSECT
SELECT name FROM skating
WHERE country = 'DE';
```

EXCEPT

EXCEPT retorna apenas as linhas que aparecem no primeiro conjunto de resultados, mas não aparecem no segundo conjunto de resultados.

Essa consulta exibe ciclistas alemães, a menos que eles também sejam skatistas alemães ao mesmo tempo:

```
SELECT name FROM cycling WHERE country = 'DE' EXCEPT / MINUS SELECT name FROM skating WHERE country = 'DE';
```