

SQL

SQL - manipulação de dados

SQL - manipulação de dados

Introdução ao SQL ():

Tipos de relacionamento entre tabelas:

2. Intermediário
3. **Joining Data in SQL**

Introdução ao SQL ():

Tipos de relacionamento entre tabelas:

- one to many
- one to one
- many to many

1. Comandos básicos

```
# busca os dados da coluna patrons na tabela names
SELECT name, [...]
FROM patrons

# RENAME COLUMNS
SELECT name as first_name, year_hired (muda o nome de name e chama name e year_hired)
FROM employees

# VALORES DIFERENTES
SELECT DISTINCT year_hired
FROM employees

# SALVA A VISUALIZAÇÃO
CREATE VIEW employee_hire_years AS
SELECT id, name, year_hired
FROM employees;
```

```
# FLAVORS, principais diferenças
Os flavors são muito parecido com algumas expressões diferentes (LIMIT e TOP)
```

PostgreSQL -

SQL Server -

2. Intermediário

```
# COUNT()
SELECT COUNT (birthdate) AS count_birth_dates
FROM people

# COUNT() multiple fields
SELECT COUNT (birthdate) AS count_birth_dates, COUNT(name) AS count_names
FROM people

# Usando * com COUNT()
SELECT COUNT(*) AS total_records
FROM people

# COUNT() with DISTINCT
SELECT COUNT(DISTINCT birthdate) AS count_distinct_birthdates
FROM people

# Filtering numbers
WHERE color = 'green'
WHERE release_year >= 1960
WHERE release_year <> 1960

WHERE color = 'yellow' OR length = 'short';
WHERE color = 'yellow' AND length = 'short';
WHERE buttons BETWEEN 1 AND 5;

WHERE (release_year = 1994 OR release_year = 1995)
      AND (certification = 'PG' OR certification = 'R');

WHERE release_year IN (1920, 1930, 1940);

# Filtering TEXT
#LIKE é usado para buscar nomes com o valor indicado
WHERE name LIKE 'Ade%';      #nomes que começam com Ade...
WHERE name LIKE 'Ev_';      #nomes que iniciam com Ev +1

#NOT LIKE é usado para buscar palavras sem o valor indicado
WHERE name NOT LIKE 'A.%';   #nomes sem A. na composição

#IS NULL and IS NOT NULL
WHERE birthdate IS NULL;     #valores onde birthdate is null
WHERE birthdate IS NOT NULL; #valores onde birthdate is not null

# Summarizing data
SELECT AVG(budget)           #calcula a média
SELECT SUM(budget)           #calcula soma
SELECT MIN(budget)           #calcula a min   (LOWEST)
SELECT MAX(budget)           #calcula a máx   (HIGHEST)

SELECT AVG(budget) Where release_year = 2010;
SELECT ROUND(AVG(budget), 2) #limita em duas casas decimais

SELECT COUNT(deathdate) * 100.0 / COUNT(*) AS percentage_dead

# Sorting results
      #ORDER BY (ordena)
ORDER BY budget;
ORDER BY budget ASC;
ORDER BY wins ASC, Imdb_score DESC;

WHERE budget IS NOT NULL
ORDER BY budget DESC;
```

```

# GROUP BY (agrupa)
SELECT certification, language, COUNT(title) as title_count
FROM films
GROUP BY certification, language
ORDER BY title_count

SELECT release_year, country, MAX(budget) AS max_budget
FROM films
GROUP BY release_year, country
ORDER BY release_year, country;

#agrupando com média
SELECT region, AVG(gdp_percapita) AS avg_gdp
FROM countries AS c
LEFT JOIN economies AS e
USING(code)
WHERE year = 2010
GROUP BY region;

#HAVING
SELECT certification, COUNT(title) AS title_count
FROM films
WHERE certification IN ('G','PG','PG-13')
GROUP BY certification
HAVING COUNT (tittle) > 500      # seleciona valores >500 na contagem
ORDER BY title_count DESC
LIMIT 3;

```

3. Joining Data in SQL

3.1 - INNER JOIN

key é usado pra identificar e relacionar as tabelas

Prime ministers, meet in presidents

The presidents table

country	continent	president
Egypt	Africa	Abdel Fattah el-Sisi
Portugal	Europe	Marcelo Rebelo de Sousa
USA	North America	Joe Biden
Uruguay	South America	Luis Lacalle Pou
Pakistan	Asia	Arif Alvi
Chile	South America	Gabriel Boric
India	Asia	Ram Nath Kovind

The prime_ministers table

country	continent	prime_minister
Egypt	Africa	Mostafa Madbouly
Portugal	Europe	António Costa
Pakistan	Asia	Shehbaz Sharif
United Kingdom	Europe	Boris Johnson
India	Asia	Narendra Modi
Australia	Oceania	Scott Morrison
Norway	Europe	Jonas Gahr Støre
Brunei	Asia	Hassanal Bolkiah
Oman	Asia	Haitham bin Tarik

For larger tables though, it's not as feasible to identify these records by visual inspection.



```

#
SELECT prime_minister.country, prime_ministers.continent, prime_minister, president
FROM prime_ministers
INNER JOIN presidents
ON prime_minister.country = presidents.country;

OR

SELECT p1.country, p1.continent, prime_minister, president
FROM prime_minister AS p1
INNER JOIN presidents AS p2
ON p1.country = p2.country

#Simplificado juntando 2 tabelas completas
SELECT *
FROM cities #tab1
INNER JOIN countries #tab2
ON cities.country_code = countries.code; #colunas que se relacionam

# Simplificado juntando apenas linhas de cada tabela
SELECT cities.name AS city, countries.name AS country, region #colunas de todas as tabelas(cities, countries, region)
FROM cities
INNER JOIN countries
ON cities.country_code = countries.code;

# Resumindo nomenclaturas
SELECT c.code AS country_code, #seleciona e muda o nome da coluna
       name,
       year,
       inflation_rate
FROM countries AS c #tab1 e chama de c
INNER JOIN economies AS e #tab2 e chama de e
ON c.code = e.code;

# Resumindo com USING
SELECT c.name AS country, l.name AS language, official
FROM countries AS c
INNER JOIN languages AS l
USING(code); # qnd as duas colunas tem o mesmo nome basta referenciar com USING

# juntando 2 tabelas por 2 referencias

SELECT *
FROM left_table #tab1
INNER JOIN right_table #tab2
ON left_table.id = right_table.id #colunas que se relacionam
AND left_table.date = right_table.date; #colunas que se relacionam

```

```

# MULTIPLE JOINS

#Simplificado juntando 3 tabelas completas
SELECT *
FROM left_table #tab1
INNER JOIN right_table #tab2
ON left_table.id = right_table.id; #colunas que se relacionam
INNER JOIN another_table
ON left_table.id = another_table.id;

# Juntando 3 tabelas com col. que se repetem
SELECT name, e.year, fertility_rate, unemployment_rate #col year se repete em 2 tabelas
FROM countries AS c
INNER JOIN populations AS p
ON c.code = p.country_code
INNER JOIN economies AS e
USING (code)
AND e.year = p.year; #join também por year

```

LEFT JOIN - prioriza left e relaciona aos pares de right com valores null

```
SELECT p1.country, prime_minister, president
FROM prime_minister AS p1
LEFT JOIN presidents AS p2
USING (country)
```

#RIGHT JOIN - prioriza right e relaciona aos pares em left com valores null

```
SELECT p1.country, prime_minister, president
FROM prime_minister AS p1
RIGHT JOIN presidents AS p2
USING (country)
```

#FULL JOIN - junta todas os records

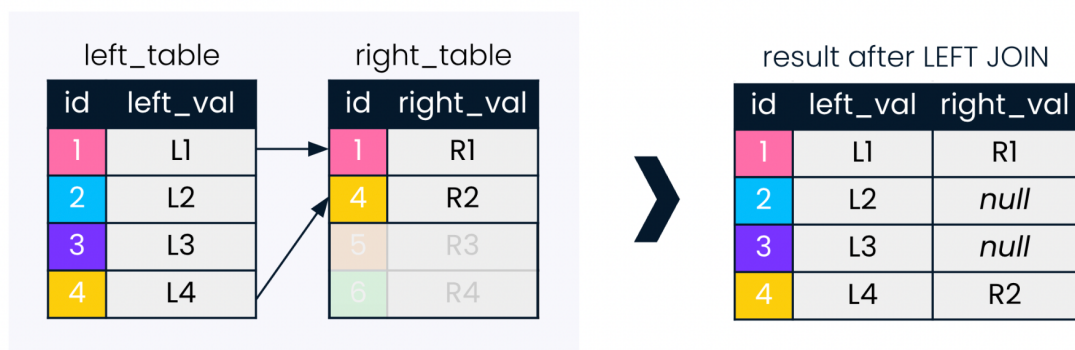
```
SELECT left_table.id as L_id
       right_table.id as R_id
FROM left_table
FULL JOIN right_table
WHERE left_table.id = 'alguma coisa'
   OR left_table.id IS NULL
USING (id)
```

```
SELECT
  c1.name AS country,
    region,
  l.name AS language,
    basic_unit,
    frac_unit
FROM countries as c1
FULL JOIN languages as l
USING(code)
FULL JOIN currencies AS c2
USING(code)
WHERE region LIKE 'M%esia';
```

LEFT JOIN diagram

Pressione **Esc** para sair do modo tela cheia

Diagram for a **LEFT JOIN** ON the **id** field



INNER JOIN returns only records corresponding to ids 1 and 4, whereas **LEFT JOIN** keeps all

3.3 Cross Join

```
# Cria todas as possibilidades com os identificadores informados
# 3 linhas em 2 identificadores geram 9 possibilidades

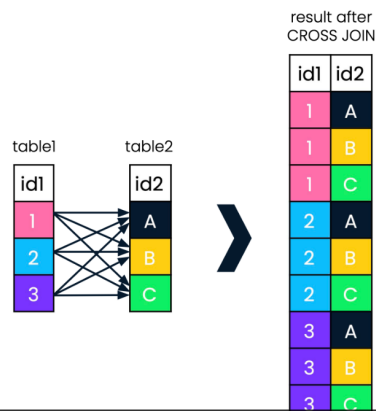
SELECT id1, id2
FROM table1
CROSS JOIN table2;
WHERE table1.id1 IN ('asia')           #filtros para reduzi o volume
      AND table2.id2 IN ('south america');
```

```
# cria a relação de datas 2010 e 2015 da tabela populations com ela mesma
SELECT
  p1.country_code,
  p1.size AS size2010,
  p2.size AS size2015
FROM populations AS p1
INNER JOIN populations AS p2
USING(country_code)
WHERE p1.year = 2010
      AND p1.year = p2.year - 5;
```

CROSS JOIN diagram

Pressione **Esc** para sair do modo tela cheia

CROSS JOIN creates all possible combinations of two tables.



In this diagram we have two tables named table1 and table2, with one field each: id1 and id2, respectively.

datacamp

JOINING DATA IN SQL

3.4 SELF JOINS

```
# cria relações internas

SELECT
  p1.country
  p2.country
  p1.continent
FROM prime_ministers as p1
INNER JOIN prime_ministers as p2      # a tab se junta com ela mesma
ON p1.continent = p2.continent        # critério de junção
   AND p1.country <> p2.country;       # critério de junção
```

3.5 Ser theory for SQL joins

```
# UNION - UNE TABELAS EXCLUINDO DUPLICADOS
# número e número de colunas deve ser identico
SELECT monarch AS leader, country
FROM monarchs
UNION
SELECT prime_minister, country
FROM prime_ministers
ORDER BY country, leader          #prime_minister entra em leader
LIMIT 10

# UNION ALL - UNE TABELAS COM DUPLICADAS
SELECT *
FROM left_table
UNION ALL
SELECT *
FROM right_table

# INTERSECT - interseção entre tabelas
# inner join na mesma situação traria linhas duplicadas
SELECT col1, col2
FROM left_table
INTERSECT
SELECT col1, col2
FROM right_table

#EXCEPT - só valores da tab1 não presentes em tab2
SELECT col1, col2
FROM left_table
EXCEPT
SELECT col1, col2
FROM right_table
```

3.6 Subquerying with semi joins and anti joins

```
# semi-join
SELECT country
FROM states
WHERE indep_year < 1800;

# Subqueries
# seleciona presidentes
SELECT president, country, continent
FROM presidents
WHERE country IN                #tem q ter o IN para começar a subq
    (SELECT country
     FROM states
     WHERE indep_year <1800);
    AND code NOT IN
    (SELECT code
     FROM currencies);

# subqueries inside select, cria uma contagem de monarcas por continentes
SELECT DISTINCT continent,
    (SELECT COUNT (*)
     FROM monarchs
     WHERE states.continent = monarch.continent) AS monarch_count
FROM states

# subquerie filtra life_expectancy > 1.15 em 2015
SELECT *
FROM populations
WHERE life_expectancy > 1.15 *
```

```

        (SELECT AVG(life_expectancy)
         FROM populations
         WHERE year = 2015)
        AND year = 2015;

    # subquerie filtra a presença do nome nas duas tabelas
    SELECT name, country_code, urbanarea_pop
    FROM cities
    WHERE name IN
        (SELECT capital
         FROM countries)
    ORDER BY urbanarea_pop DESC;

# Subqueries inside FROM
SELECT continent, MAX(indep_year) AS most_recent
FROM states
GROUP BY continent;

# Anti Join
# seleciona presidentes da america com independencia < 1800
SELECT president, country
FROM presidents LIKE '%America'
WHERE country NOT IN
    (SELECT country
     FROM states
     WHERE indep_year <1800);

```

3.2 defining relationships

- one to many
- one to one
- many to many