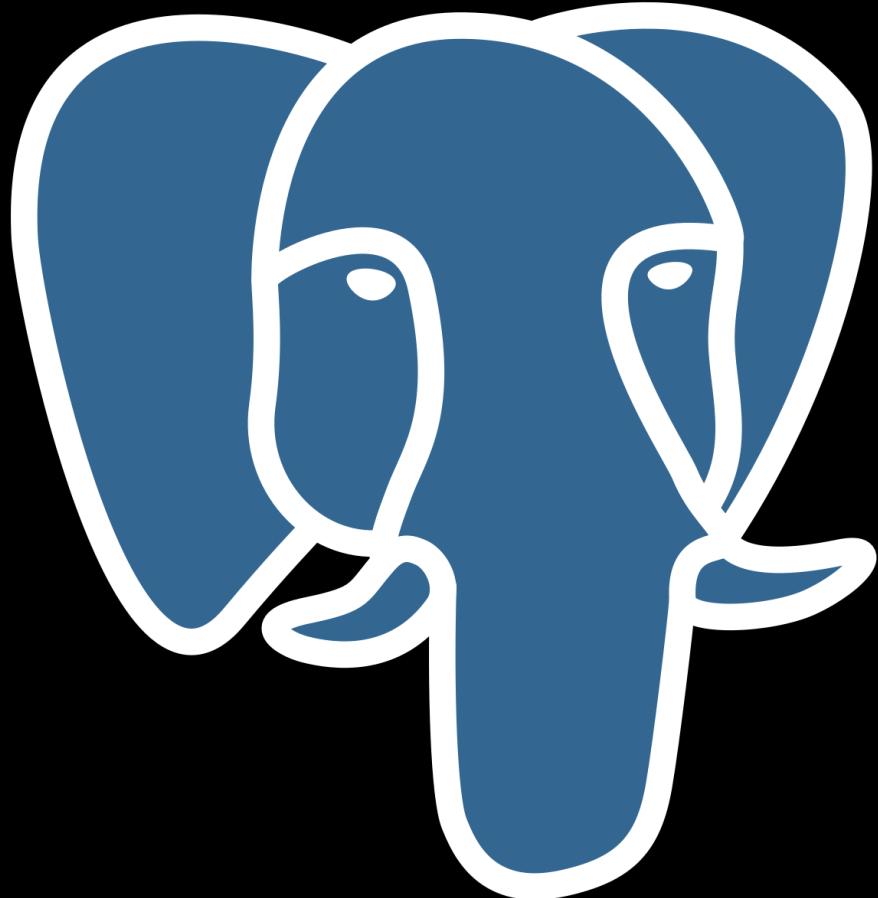


# PostgreSQL

## Aula 01



**Aluno:** Petrick Lopes

**Professor:** Fabio Souza

**Data:** 25/05/2023

# PostgreSQL -

# Plano de Estudos

- **Consultas Avançadas**
  - Tipos de dados complexos (Json, Array, Etc.)
  - **Subqueries**
  - **Relatório de Pedidos**
    - **JOIN**
    - **WHERE**
    - **LIMIT**
  - **Relatório de Pedidos Agrupado**
    - **SUM()**
    - **GROUP BY**
    - **ORDER BY**
    - **HAVING**
- **Views**
  - O que são Views
  - O que são Views Materializadas
  - Implementando as Views
  - Implementando as Views Materializadas
  - Revisão do conteúdo aprendido
- **Functions pt 1**
  - O que são functions
  - Implementando uma função
  - Usando uma função
  - Implementando uma função com parâmetros
  - Implementando uma função com parâmetro composto
  - Retorno composto
  - Parâmetros de saída
  - Revisão do conteúdo aprendido

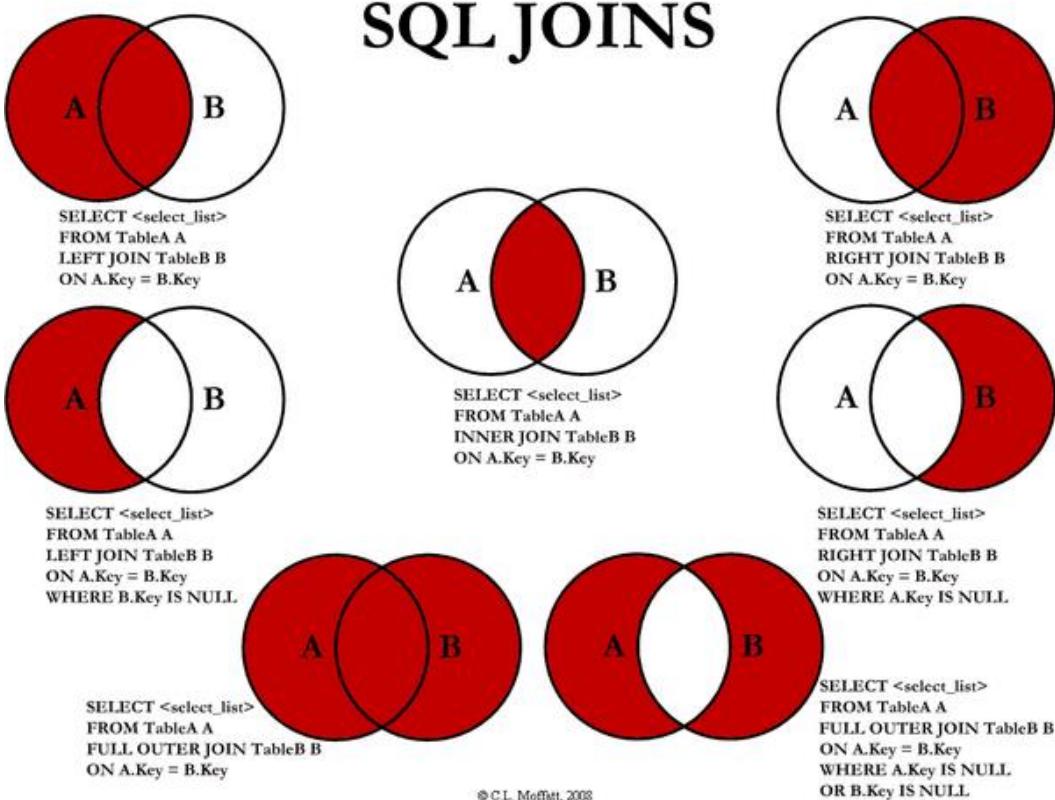
# PostgreSQL Subquery

---

## JOIN

Uma cláusula JOIN em SQL, correspondente a uma operação de junção em álgebra relacional, combina colunas de uma ou mais tabelas em um banco de dados relacional. Ela cria um conjunto que pode ser salvo como uma tabela ou usado da forma como está.

### SQL JOINS



© C.L. Moffatt, 2008

# PostgreSQL Subquery

---

## Exercicio...

- 1) Utilizando o que aprendemos sobre JOIN, faça uma query que retorne todos os clientes sem pedidos e os pedidos sem clientes.

Obs.: Será necessário “desnormalizar” a base e alterar um pedido para que ele fique sem cliente

# PostgreSQL Subquery

---

## Subquery

Uma Subquery (também conhecida como SUBCONSULTA ou SUBSELECT) é uma instrução do tipo SELECT dentro de outra instrução SQL, que efetua consultas que, de outra forma, seriam extremamente complicadas ou impossíveis de serem feitas.

```
--realizando update utilizando subquery normal
UPDATE funcionario_json func
| SET historico = jsonb_set(historico, item.path, '"2003-05-12"')
| from (SELECT ('{historico,' || index - 1 || ',''dataPromocao}'')::TEXT[] AS path,
|           id_funcionario,
|           nome,
|           item ->> 'dataPromocao' as dataPromocao
|      FROM funcionario_json,
|           jsonb_array_elements(historico -> 'historico') WITH ORDINALITY arr(item, index)
|     WHERE item ->> 'dataPromocao' IS NOT NULL
|       and nome = 'Pedro Bial') item
| WHERE item.nome = func.nome
|       and item.dataPromocao = '2023-05-06'
```

# PostgreSQL

## CTE

## CTE – Common Table Expression

No contexto do PostgreSQL, "CTE" significa "Common Table Expression" (Expressão de Tabela Comum, em português). Uma CTE é uma técnica usada para criar consultas SQL mais legíveis e reutilizáveis.

Uma CTE permite definir uma ou mais consultas dentro de uma cláusula WITH (WITH clause) e, em seguida, referenciar essas consultas na consulta principal. Isso é especialmente útil quando você precisa executar consultas complexas que envolvem subconsultas ou quando deseja dividir uma consulta grande em partes menores para facilitar a leitura e manutenção.

```
--realizando update utilizando CTE(WITH)
WITH item AS (
    SELECT ('{historico,' || index - 1 || ',"dataPromocao"}')::TEXT[] AS path,
           id_funcionario,
           nome,
           item ->> 'dataPromocao' as dataPromocao
      FROM funcionario_json,
           jsonb_array_elements(historico -> 'historico') WITH ORDINALITY arr(item, index)
     WHERE item ->> 'dataPromocao' IS NOT NULL
       and nome = 'Pedro Bial'
)
UPDATE funcionario_json func
   SET historico = jsonb_set(historico, item.path, '"2003-05-12")')
  from item
 WHERE item.nome = func.nome
   and item.dataPromocao = '2002-05-06'
```

# PostgreSQL Data Types

---

Metendo a mão nas  
subqueries...

...criando o banco de dados ecommerce  
...selecionando os registros  
...agrupando  
...limitando  
...somando

# PostgreSQL

## Data Types

---

### Desafio...

- 1) Utilizando CTE e JOIN, crie uma consulta que retorne os seguintes dados agrupados por cliente quando o valor do pedido for mario que 3000,00: "Pedido", "Cliente", "Endereco", "CEP", "Email", "Telefone", "Produto", "Modelo", "Fabricante", "Quantidade", "ValorPago".



Obrigado e  
dúvidas estou à  
disposição!

## Contato

WhatsApp / Telefone: (11) 95658-2460

E-mail: fabmasouza@hotmail.com ou fabmasouza@gmail.com

LinkedIn: <https://www.linkedin.com/in/fabiomarcelosouza/>

