

MSc Sandir Rodrigues Campos

# Banco de Dados 2

## Aula 9

# FAZER

- Crie uma view chamada `clientes_com_pedidos_medios` que mostre:
  - `id_cliente`
  - `Nome`
  - `media_valor_pedido` (média do valor dos pedidos de cada cliente)
  - Mas só mostre clientes com mais de 2 pedidos registrados.

# Exercícios com views

Exercício 1: Criar uma view que retorne os pedidos com os nomes dos produtos e os nomes dos clientes

Crie as seguintes tabelas:

- `CREATE TABLE clientes ( id INT PRIMARY KEY, nome VARCHAR(255), email VARCHAR(255));`
- `CREATE TABLE pedidos ( id INT PRIMARY KEY, cliente_id INT, data_pedido DATE, valor DECIMAL(10, 2), FOREIGN KEY (cliente_id) REFERENCES clientes(id));`
- `CREATE TABLE produtos ( id INT PRIMARY KEY, nome VARCHAR(255), preco DECIMAL(10, 2));`
- `CREATE TABLE itens_pedidos ( id INT PRIMARY KEY, pedido_id INT, produto_id INT, quantidade INT, FOREIGN KEY (pedido_id) REFERENCES pedidos(id), FOREIGN KEY (produto_id) REFERENCES produtos(id));`

Insira alguns dados:

- `INSERT INTO clientes (id, nome, email) VALUES (1, 'João', 'joao@example.com'), (2, 'Maria', 'maria@example.com'), (3, 'Pedro', 'pedro@example.com');`
- `INSERT INTO pedidos (id, cliente_id, data_pedido, valor) VALUES (1, 1, '2022-01-01', 100.00), (2, 1, '2022-01-15', 200.00), (3, 2, '2022-02-01', 50.00);`
- `INSERT INTO produtos (id, nome, preco) VALUES (1, 'Produto 1', 10.00), (2, 'Produto 2', 20.00), (3, 'Produto 3', 30.00);`
- `INSERT INTO itens_pedidos (id, pedido_id, produto_id, quantidade) VALUES (1, 1, 1, 2), (2, 1, 2, 1), (3, 2, 3, 3), (4, 3, 1, 1);`

Crie uma view que retorne os pedidos com os nomes dos produtos e os nomes dos clientes:

# Exercícios com views

Exercício 2: Criar uma view que retorne os produtos com os nomes dos fornecedores e os preços médios

Crie as seguintes tabelas:

- `CREATE TABLE fornecedores ( id INT PRIMARY KEY, nome VARCHAR(255), email VARCHAR(255));`
- `CREATE TABLE produtos ( id INT PRIMARY KEY, nome VARCHAR(255), preco DECIMAL(10, 2), fornecedor_id INT, FOREIGN KEY (fornecedor_id) REFERENCES fornecedores(id));`

Insira alguns dados:

- `INSERT INTO fornecedores (id, nome, email) VALUES (1, 'Fornecedor 1', 'fornecedor1@example.com'), (2, 'Fornecedor 2', 'fornecedor2@example.com'), (3, 'Fornecedor 3', 'fornecedor3@example.com');`
- `INSERT INTO produtos (id, nome, preco, fornecedor_id) VALUES (1, 'Produto 1', 10.00, 1), (2, 'Produto 2', 20.00, 1), (3, 'Produto 3', 30.00, 2), (4, 'Produto 4', 40.00, 3), (5, 'Produto 5', 50.00, 3);`

Crie uma view que retorne os produtos com os nomes dos fornecedores e os preços médios:

# Exercícios com views

Exercício 3: Criar uma view que retorne os pedidos com os nomes dos clientes e os valores totais, agrupados por mês

Crie as seguintes tabelas:

- CREATE TABLE clientes ( id INT PRIMARY KEY, nome VARCHAR(255), email VARCHAR(255));
- CREATE TABLE pedidos ( id INT PRIMARY KEY, cliente\_id INT, data\_pedido DATE, valor DECIMAL(10, 2), FOREIGN KEY (cliente\_id) REFERENCES clientes(id));

Insira alguns dados:

- INSERT INTO clientes (id, nome, email) VALUES (1, 'João', 'joao@example.com'), (2, 'Maria', 'maria@example.com'), (3, 'Pedro', 'pedro@example.com');
- INSERT INTO pedidos (id, cliente\_id, data\_pedido, valor) VALUES (1, 1, '2022-01-01', 100.00), (2, 1, '2022-01-15', 200.00), (3, 2, '2022-02-01', 50.00), (4, 3, '2022-03-01', 150.00), (5, 1, '2022-03-15', 250.00);

Crie uma view que retorne os pedidos com os nomes dos clientes e os valores totais, agrupados por mês:

# Exercícios com views

Exercício 4: Criar uma view que retorne os produtos com os nomes dos fornecedores e os preços médios, considerando apenas os produtos com preço maior que 50

Crie as seguintes tabelas:

- `CREATE TABLE fornecedores ( id INT PRIMARY KEY, nome VARCHAR(255), email VARCHAR(255));`
- `CREATE TABLE produtos ( id INT PRIMARY KEY, nome VARCHAR(255), preco DECIMAL(10, 2), fornecedor_id INT, FOREIGN KEY (fornecedor_id) REFERENCES fornecedores(id));`

Insira alguns dados:

- `INSERT INTO fornecedores (id, nome, email) VALUES (1, 'Fornecedor 1', 'fornecedor1@example.com'), (2, 'Fornecedor 2', 'fornecedor2@example.com'), (3, 'Fornecedor 3', 'fornecedor3@example.com');`
- `INSERT INTO produtos (id, nome, preco, fornecedor_id) VALUES (1, 'Produto 1', 10.00, 1), (2, 'Produto 2', 20.00, 1), (3, 'Produto 3', 60.00, 2), (4, 'Produto 4', 40.00, 3), (5, 'Produto 5', 70.00, 3);`

Crie uma view que retorne os produtos com os nomes dos fornecedores e os preços médios, considerando apenas os produtos com preço maior que 50:

# SUBCONSULTAS

- Uma **subconsulta** (ou **subquery**) é uma **consulta dentro de outra consulta**. Ela é usada para **obter dados intermediários** que vão ser usados pela consulta principal.

# Pra que serve?

- Filtrar resultados
- Fazer comparações complexas
- Obter dados agregados (como somas, médias)
- Usar valores temporários em INSERT, UPDATE ou DELETE

# Tipos de Subconsultas

## 1. Subconsulta em cláusula WHERE (mais comum)

- Usada para **filtrar** resultados com base no resultado da subconsulta.

*Lista os produtos cujo preço é maior que a média de todos os produtos*

```
SELECT nome, preco
```

```
FROM produtos
```

```
WHERE preco > (SELECT AVG(preco) FROM produtos);
```

# Tipos de Subconsultas

## 2. Subconsulta com IN / NOT IN

Verifica se um valor está **dentro de uma lista** gerada pela subconsulta.

*Lista os clientes que já fizeram pedidos*

```
SELECT nome  
FROM clientes  
WHERE id_cliente IN (SELECT id_cliente FROM pedidos);
```

# Tipos de Subconsultas

## 3. Subconsulta com EXISTS / NOT EXISTS

Checa se a subconsulta retorna **algum resultado**.

*-- Lista os clientes que fizeram pelo menos um pedido*

```
SELECT nome  
FROM clientes c  
WHERE EXISTS (SELECT 1 FROM pedidos p WHERE p.id_cliente =  
c.id_cliente);
```

# Tipos de Subconsultas

## 4. Subconsulta em **SELECT** (também chamada de escalar)

Retorna um **único valor** e pode ser usada como uma **coluna**.

*-- Lista os produtos e a média geral de preços (na mesma linha)*

```
SELECT nome, preco,  
       (SELECT AVG(preco) FROM produtos) AS media_precos  
FROM produtos;
```

# Tipos de Subconsultas

## 5. Subconsulta em FROM (subquery como tabela temporária)

- Você pode dar um nome para a subconsulta e usá-la como se fosse uma tabela.

*-- Subconsulta calcula a média, e a consulta principal filtra*

```
SELECT *  
FROM (  
    SELECT nome, preco FROM produtos  
) AS tabela_temporaria  
WHERE preco > 100;
```

# Tipos de Subconsultas

## 6. Subconsulta em cláusulas INSERT, UPDATE, DELETE

- Pode usar subquery para inserir ou alterar dados com base em outras tabelas.

*-- Inserir clientes VIPs em outra tabela*

*INSERT INTO clientes\_vip (id\_cliente, nome)*

*SELECT id\_cliente, nome*

*FROM clientes*

*WHERE gasto\_total > (SELECT AVG(gasto\_total) FROM clientes);*

# Exercícios

- Cenário Base (use como referência para todos os exercícios):
  - Tabelas:
    - clientes(id\_cliente, nome, cidade)
    - pedidos(id\_pedido, id\_cliente, data\_pedido, valor\_total)
    - produtos(id\_produto, nome\_produto, preco)
    - itens\_pedido(id\_pedido, id\_produto, quantidade)
  - Popule com o mínimo de 8 tuplas cada tabela

# Exercícios

- Exercício 1 — Subconsulta Correlacionada com EXISTS Enunciado: Liste os nomes dos clientes que compraram pelo menos 3 produtos diferentes em algum pedido.
- Exercício 2 — Subconsulta com IN + Agregação Enunciado: Liste os nomes dos clientes cujo valor total de pedidos é superior ao valor total de pedidos dos clientes da cidade 'São Paulo'.
- Exercício 3 — Subconsulta no FROM (Tabela Derivada) Enunciado: Crie uma subconsulta que calcula o total de vendas por mês. A consulta principal deve retornar somente os meses onde o total de vendas foi superior a R\$ 10.000.
- Exercício 4 — Subconsulta no UPDATE Enunciado: Atualize os preços dos produtos aumentando em 5% apenas aqueles cujo preço esteja abaixo da média dos preços de todos os produtos.
- Exercício 5 — Subconsulta com NOT IN Enunciado: Liste os nomes dos produtos que nunca foram vendidos em nenhum pedido.
- Exercício 6 — Subconsulta no SELECT com cálculo Enunciado: Liste os pedidos (ID e valor) junto com a média de valor dos pedidos feitos pelo mesmo cliente. (Dica: subconsulta correlacionada no SELECT)