

## Prática para a prova

```
create database loja;  
use loja;
```

```
CREATE TABLE clientes (  
    id SERIAL PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    telefone VARCHAR(20),  
    endereco TEXT  
);
```

```
CREATE TABLE produtos (  
    id SERIAL PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    descricao TEXT,  
    preco DECIMAL(10,2) NOT NULL,  
    estoque INTEGER NOT NULL  
);
```

```
CREATE TABLE pedidos (  
    id SERIAL PRIMARY KEY,  
    cliente_id INTEGER REFERENCES clientes(id),  
    produto_id INTEGER REFERENCES produtos(id),  
    quantidade INTEGER NOT NULL,  
    data_pedido DATE NOT NULL  
);
```

-- Inserindo registros na tabela clientes

```
INSERT INTO clientes (nome, email, telefone, endereco) VALUES  
( 'João Silva', 'joao@email.com', '1111-1111', 'Rua A, 123'),  
( 'Maria Souza', 'maria@email.com', '2222-2222', 'Rua B, 456'),  
( 'Carlos Pereira', 'carlos@email.com', '3333-3333', 'Rua C, 789'),  
( 'Ana Lima', 'ana@email.com', NULL, 'Rua D, 321'),  
( 'Bruno Castro', 'bruno@email.com', '5555-5555', 'Rua E, 654'),  
( 'Fernando Alves', 'fernando@email.com', NULL, NULL),  
( 'Tatiane Mendes', 'tatiane@email.com', '7777-7777', 'Rua G, 987'),  
( 'Diego Moreira', 'diego@email.com', '8888-8888', 'Rua H, 147'),  
( 'Larissa Rocha', 'larissa@email.com', NULL, 'Rua I, 258'),  
( 'Pedro Nogueira', 'pedro@email.com', '1010-1010', 'Rua J, 369');
```

-- Inserindo registros na tabela produtos

```
INSERT INTO produtos (nome, descricao, preco, estoque) VALUES  
( 'Notebook Dell', 'Core i7, 16GB RAM, SSD 512GB', 4500.00, 20),  
( 'Smartphone Samsung', '128GB, Tela 6.4"', 2500.00, 30),  
( 'Fone Bluetooth', 'Cancelamento de ruído', 500.00, 50),  
( 'Monitor LG', 'Full HD, 24 polegadas', 800.00, 25),
```

```
('Teclado Mecânico', 'RGB, Switch Red', 350.00, 40),
('Mouse Gamer', '16000 DPI, RGB', 250.00, 35),
('Impressora HP', 'Multifuncional Wi-Fi', 1200.00, 15),
('Cadeira Gamer', 'Reclinável, Preto/Vermelho', 950.00, 10),
('Microfone Condensador', 'USB, Profissional', 400.00, 12),
('Webcam Full HD', '1080p, Autofoco', 300.00, 18);
```

-- Inserindo registros na tabela pedidos

```
INSERT INTO pedidos (cliente_id, produto_id, quantidade, data_pedido) VALUES
(1, 1, 2, '2025-03-01'),
(2, 3, 1, '2025-03-02'),
(3, 5, 1, '2025-03-03'),
(4, 7, 3, '2025-03-04'),
(5, 9, 2, '2025-03-05'),
(NULL, 2, 1, '2025-03-06'), -- Pedido sem cliente associado
(7, NULL, 1, '2025-03-07'), -- Pedido sem produto associado
(8, 4, 2, '2025-03-08'),
(9, 6, 1, '2025-03-09'),
(10, 8, 1, '2025-03-10');
```

## Questões

-- Contar o número total de clientes

```
SELECT COUNT(*) AS total_clientes
FROM clientes;
```

-- Contar o número total de pedidos

```
SELECT COUNT(*) AS total_pedidos
FROM pedidos;
```

-- Calcular o valor total de todos os pedidos

```
SELECT SUM(p.quantidade * pr.preco) AS valor_total_pedidos
FROM pedidos p
JOIN produtos pr ON p.produto_id = pr.id;
```

-- Calcular a média de preço dos produtos

```
SELECT AVG(preco) AS media_preco_produtos
FROM produtos;
```

-- Listar todos os clientes e seus pedidos

```
SELECT c.nome, p.id, p.data_pedido
FROM clientes c
LEFT JOIN pedidos p ON c.id = p.cliente_id;
```

-- Listar todos os pedidos e seus produtos, incluindo pedidos sem produtos

```
SELECT p.id, pr.nome AS produto, p.quantidade
FROM pedidos p
LEFT JOIN produtos pr ON p.produto_id = pr.id;
```

-- Listar os produtos mais caros primeiro

```
SELECT nome, preco
FROM produtos
ORDER BY preco DESC;
```

-- Listar os produtos com menor estoque

```
SELECT nome, estoque
FROM produtos
ORDER BY estoque ASC;
```

-- Contar quantos pedidos foram feitos por cliente

```
SELECT c.nome, COUNT(p.id) AS total_pedidos
FROM clientes c
LEFT JOIN pedidos p ON c.id = p.cliente_id
GROUP BY c.id;
```

-- Contar quantos produtos diferentes foram vendidos

```
SELECT COUNT(DISTINCT p.produto_id) AS total_produtos_vendidos
FROM pedidos p;
```

-- Mostrar os clientes que não realizaram pedidos

-- Mostrar os produtos que nunca foram vendidos

-- Contar o número de pedidos feitos por dia

```
SELECT DATE(data_pedido) AS data, COUNT(id) AS total_pedidos
FROM pedidos
GROUP BY DATE(data_pedido)
ORDER BY data DESC;
```

-- Listar os produtos mais vendidos

```
SELECT pr.nome, SUM(p.quantidade) AS total_vendas
FROM pedidos p
JOIN produtos pr ON p.produto_id = pr.id
GROUP BY pr.id
ORDER BY total_vendas DESC;
```

-- Encontrar o cliente que mais fez pedidos

```
SELECT c.nome, COUNT(p.id) AS total_pedidos
FROM clientes c
JOIN pedidos p ON c.id = p.cliente_id
GROUP BY c.id
ORDER BY total_pedidos DESC
LIMIT 1;
```

-- Listar os pedidos e os clientes que os fizeram, incluindo pedidos sem clientes

```
SELECT p.id, c.nome AS cliente
FROM pedidos p
LEFT JOIN clientes c ON p.cliente_id = c.id;
```

-- Listar os produtos e o total de vendas por produto

```
SELECT pr.nome, SUM(p.quantidade) AS total_vendas
FROM pedidos p
JOIN produtos pr ON p.produto_id = pr.id
GROUP BY pr.id
ORDER BY total_vendas DESC;
```

-- Calcular a média de quantidade de produtos por pedido

-- Listar os pedidos ordenados por data (mais recentes primeiro)

```
SELECT id, data_pedido
FROM pedidos
ORDER BY data_pedido DESC;
```

-- Contar quantos clientes possuem telefone cadastrado

```
SELECT COUNT(*) AS total_clientes_com_telefone
FROM clientes
WHERE telefone IS NOT NULL;
```

-- Encontrar o cliente que gastou mais dinheiro em pedidos.

```
SELECT c.nome, SUM(p.quantidade * pr.preco) AS total_gasto
FROM clientes c
JOIN pedidos p ON c.id = p.cliente_id
JOIN produtos pr ON p.produto_id = pr.id
GROUP BY c.id
ORDER BY total_gasto DESC
LIMIT 1;
```

-- Listar os 5 produtos mais vendidos.

```
SELECT pr.nome, SUM(p.quantidade) AS total_vendas
FROM pedidos p
JOIN produtos pr ON p.produto_id = pr.id
GROUP BY pr.id
ORDER BY total_vendas DESC
LIMIT 5;
```

-- Listar os clientes que já fizeram pedidos e o número de pedidos de cada um.

```
SELECT c.nome, COUNT(p.id) AS total_pedidos
FROM clientes c
JOIN pedidos p ON c.id = p.cliente_id
GROUP BY c.id;
```

-- Encontrar a data com mais pedidos realizados

```
SELECT DATE(data_pedido) AS data, COUNT(id) AS total_pedidos
FROM pedidos
GROUP BY DATE(data_pedido)
ORDER BY total_pedidos DESC
LIMIT 1;
```

-- Calcular a média de valor gasto por pedido

```
SELECT AVG(p.quantidade * pr.preco) AS media_valor_pedido
FROM pedidos p
JOIN produtos pr ON p.produto_id = pr.id;
```