

Desafio Banco de Dados – SQL II

Script das tabelas (SQLite/Dbeaver)

-- Clientes

```
CREATE TABLE customers (  
    customer_id INTEGER PRIMARY KEY,  
    nome TEXT NOT NULL,  
    email TEXT UNIQUE,  
    dt_cadastro DATE  
);
```

-- Pedidos

```
CREATE TABLE orders (  
    order_id INTEGER PRIMARY KEY,  
    customer_id INTEGER,  
    dt_pedido DATE,  
    valor_total DECIMAL(10,2),  
    status TEXT CHECK(status IN ('Pendente','Pago','Cancelado')),  
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```

-- Produtos

```
CREATE TABLE products (  
    product_id INTEGER PRIMARY KEY,  
    nome TEXT NOT NULL,  
    preco DECIMAL(10,2),  
    categoria TEXT  
);
```

-- Itens do Pedido

```
CREATE TABLE order_items (  
    order_item_id INTEGER PRIMARY KEY,  
    order_id INTEGER,  
    product_id INTEGER,  
    quantidade INTEGER,  
    preco_unitario DECIMAL(10,2),  
    FOREIGN KEY (order_id) REFERENCES orders(order_id),  
    FOREIGN KEY (product_id) REFERENCES products(product_id)  
);
```

1. Clientes

```
INSERT INTO customers (customer_id, nome, email, dt_cadastro) VALUES  
(6, 'Mariana Costa', 'mariana@gmail.com', '2024-03-12'),  
(7, 'Ricardo Almeida', 'ricardo@uol.com', '2024-06-05'),
```

```
(8, 'Fernanda Lima', 'fernanda@yahoo.com', '2024-07-01'),  
(9, 'Gabriel Rocha', 'gabriel@hotmail.com', '2024-08-18'),  
(10, 'Juliana Mendes', 'juliana@gmail.com', '2024-09-02'),  
(1, 'João Pereira', 'joao@gmail.com', '2024-01-10'),  
(2, 'Ana Souza', 'ana@yahoo.com', '2024-02-15'),  
(3, 'Carlos Lima', 'carlos@hotmail.com', '2024-03-05'),  
(4, 'Beatriz Oliveira', 'bia@outlook.com', '2024-04-20'),  
(5, 'Pedro Santos', 'pedro@gmail.com', '2024-05-01');
```

2. Produtos

```
INSERT INTO products (product_id, nome, preco, categoria) VALUES  
(1, 'Notebook Lenovo', 3500.00, 'Eletrônicos'),  
(2, 'Smartphone Samsung', 2500.00, 'Eletrônicos'),  
(3, 'Camisa Polo', 120.00, 'Roupas'),  
(4, 'Tênis Nike', 400.00, 'Roupas'),  
(5, 'Livro SQL Avançado', 90.00, 'Livros'),  
(6, 'Cafeteira Elétrica', 250.00, 'Casa'),  
(7, 'Fone Bluetooth', 350.00, 'Eletrônicos'),  
(8, 'Calça Jeans', 180.00, 'Roupas'),  
(9, 'Tablet Apple iPad', 4500.00, 'Eletrônicos'),  
(10, 'Monitor Dell 27"', 1800.00, 'Eletrônicos'),  
(11, 'Batedeira Planetária', 900.00, 'Casa'),  
(12, 'Livro Python Avançado', 110.00, 'Livros'),  
(13, 'Jaqueta de Couro', 750.00, 'Roupas'),  
(14, 'Tênis Adidas', 380.00, 'Roupas'),  
(15, 'Smartwatch Garmin', 2200.00, 'Eletrônicos'),  
(16, 'Aspirador de Pó', 600.00, 'Casa');
```

3. Pedidos

```
INSERT INTO orders (order_id, customer_id, dt_pedido, valor_total, status) VALUES  
(101, 1, '2024-06-01', 3620.00, 'Pago'),  
(102, 1, '2024-07-10', 120.00, 'Pago'),  
(103, 2, '2024-07-15', 2500.00, 'Pago'),  
(104, 2, '2024-08-20', 0.00, 'Cancelado'),  
(105, 3, '2024-08-25', 490.00, 'Pago'),  
(106, 3, '2024-09-05', 90.00, 'Pago'),  
(107, 4, '2024-09-10', 250.00, 'Pago'),  
(108, 4, '2024-09-15', 400.00, 'Pago'),  
(109, 5, '2024-09-20', 3500.00, 'Pendente'),  
(110, 6, '2024-06-20', 750.00, 'Pago'),  
(111, 6, '2024-07-22', 900.00, 'Pago'),  
(112, 7, '2024-08-05', 4500.00, 'Pago'),  
(113, 7, '2024-08-25', 2200.00, 'Pago'),  
(114, 8, '2024-09-01', 380.00, 'Pago'),  
(115, 8, '2024-09-10', 1800.00, 'Pago'),
```

```
(116, 9, '2024-09-12', 110.00, 'Pago'),  
(117, 9, '2024-09-22', 600.00, 'Pago'),  
(118, 10, '2024-09-25', 0.00, 'Cancelado'),  
(119, 10, '2024-09-28', 350.00, 'Pendente');
```

4. Itens dos Pedidos

```
INSERT INTO order_items (order_item_id, order_id, product_id, quantidade,  
preco_unitario) VALUES
```

```
-- Pedido 101 (João)
```

```
(1001, 101, 1, 1, 3500.00),  
(1002, 101, 5, 1, 120.00),
```

```
-- Pedido 102 (João)
```

```
(1003, 102, 3, 1, 120.00),
```

```
-- Pedido 103 (Ana)
```

```
(1004, 103, 2, 1, 2500.00),
```

```
-- Pedido 104 (Ana) - cancelado
```

```
(1005, 104, 8, 1, 180.00),
```

```
-- Pedido 105 (Carlos)
```

```
(1006, 105, 4, 1, 400.00),  
(1007, 105, 5, 1, 90.00),
```

```
-- Pedido 106 (Carlos)
```

```
(1008, 106, 5, 1, 90.00),
```

```
-- Pedido 107 (Beatriz)
```

```
(1009, 107, 6, 1, 250.00),
```

```
-- Pedido 108 (Beatriz)
```

```
(1010, 108, 4, 1, 400.00),
```

```
-- Pedido 109 (Pedro - pendente)
```

```
(1011, 109, 1, 1, 3500.00),
```

```
-- Pedido 110 (Mariana)
```

```
(1012, 110, 13, 1, 750.00),
```

```
-- Pedido 111 (Mariana)
```

```
(1013, 111, 11, 1, 900.00),
```

```
-- Pedido 112 (Ricardo)
```

```
(1014, 112, 9, 1, 4500.00),
```

-- Pedido 113 (Ricardo)
(1015, 113, 15, 1, 2200.00),

-- Pedido 114 (Fernanda)
(1016, 114, 14, 1, 380.00),

-- Pedido 115 (Fernanda)
(1017, 115, 10, 1, 1800.00),

-- Pedido 116 (Gabriel)
(1018, 116, 12, 1, 110.00),

-- Pedido 117 (Gabriel)
(1019, 117, 16, 1, 600.00),

-- Pedido 118 (Juliana - cancelado)
(1020, 118, 5, 1, 90.00),
(1021, 118, 8, 1, 260.00),

-- Pedido 119 (Juliana - pendente)
(1022, 119, 7, 1, 350.00);

Exercícios

1. CTE (Common Table Expressions)

1. Liste o valor médio gasto por cliente usando uma CTE para calcular o total de cada cliente.
2. Use uma CTE recursiva para gerar uma sequência de datas (últimos 12 meses) e depois faça um LEFT JOIN com os pedidos para mostrar meses sem vendas.

2. Window Functions

1. Para cada cliente, mostre o valor do pedido e o rank dos pedidos (maior → menor).
2. Calcule a média móvel de 3 pedidos para cada cliente.
3. Mostre o valor do primeiro e do último pedido de cada cliente usando FIRST_VALUE e LAST_VALUE.

3. Estruturas de apoio

1. Crie uma View que mostre o faturamento diário consolidado.
2. Crie uma Tabela temporária com os clientes que fizeram compras nos últimos 30 dias.

4. Joins Avançados

1. Liste todos os produtos e indique se foram ou não vendidos (LEFT JOIN).
2. Traga os clientes que compraram todos os produtos de uma categoria (pode usar NOT EXISTS).

3. Monte um relatório de clientes que compraram um produto, mas nunca outro (ex.: compraram da categoria "Eletrônicos", mas nunca da categoria "Roupas").

5. Manipulação de Strings e Conversão

1. Mostre apenas o domínio do e-mail dos clientes (parte depois do @).
2. Converta o nome dos clientes para MAIÚSCULAS.
3. Crie uma coluna calculada que concatene o nome do cliente com o ID do pedido (ex.: "João - Pedido 123").

6. Manipulação de Datas

1. Mostrar apenas o ano da data de cadastro dos clientes
2. Calcular quantos dias se passaram desde o pedido
3. Adicionar 7 dias à data do pedido (ex.: prazo de entrega)

7. Performance e Otimização

1. Crie um índice em orders (customer_id, dt_pedido) e explique quando ele será usado.
2. Compare uma query que faz JOIN com EXISTS E analise qual é mais performática.
3. Identifique consultas que podem ser reescritas usando CTE materializada para melhorar a performance.

8. Tratamento de Casos Específicos

1. Use CASE WHEN para classificar pedidos em: "Baixo" (<100), "Médio" (100-500), "Alto" (>500).
2. Crie uma coluna calculada que mostre se o cliente está Ativo (pedido nos últimos 6 meses) ou Inativo.
3. Em um relatório de pedidos, mostre NULL como "Sem valor informado".