Banco de Dados II - Semana 20 Prof. Gaio Data: 04/08/2025

Criação da Tabela cliente para o BD Loja

A tabela pedido atualmente armazena o nome do cliente em uma coluna cliente_nome. O ideal em um modelo relacional é ter uma tabela dedicada para cliente para evitar redundância de dados e facilitar a integridade e o gerenciamento das informações dos clientes.

SQL

```
CREATE TABLE cliente (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE,
    telefone VARCHAR(20)
);
```

Na criação da tabela Cliente podemos ver uma restrição que ainda não havíamos utilizado: UNIQUE. Essa propriedade é útil quando temos um atributo que não é chave, mas que não deve se repetir, ou seja, o e-mail é algo único e nunca será igual para dois clientes diferentes.

Inserção de Clientes de Exemplo

Para popular (inserir dados) a nova tabela cliente, aqui estão 10 registros que podem ser usados.

SQL

```
INSERT INTO cliente (nome, email, telefone) VALUES
('Carlos Silva', 'carlos.silva@email.com', '(11) 98765-4321'),
('Ana Pereira', 'ana.pereira@email.com', '(11) 99876-5432'),
('Beatriz Costa', 'beatriz.costa@email.com', '(11) 97654-3210'),
('João Mendes', 'joao.mendes@email.com', '(11) 96543-2109'),
('Fernanda Lima', 'fernanda.lima@email.com', '(11) 95432-1098'),
('Lucas Martins', 'lucas.martins@email.com', '(11) 94321-0987'),
('Juliana Almeida', 'juliana.almeida@email.com', '(11) 93210-9876'),
('Ricardo Souza', 'ricardo.souza@email.com', '(11) 92109-8765'),
('Daniela Rocha', 'daniela.rocha@email.com', '(11) 91098-7654'),
('Mariana Santos', 'mariana.santos@email.com', '(11) 90987-6543');
```

Alteração da Tabela pedido

Para usar a nova tabela cliente, a tabela pedido precisa ser alterada. Primeiro, adicione a chave estrangeira id_cliente e, em seguida, remova a coluna cliente_nome.

SQL

```
-- Adicionar a nova coluna para a chave estrangeira
ALTER TABLE pedido ADD COLUMN id_cliente INT;

-- Atualizar os pedidos existentes com base no nome do cliente
UPDATE pedido
JOIN cliente ON pedido.cliente_nome = cliente.nome
SET pedido.id_cliente = cliente.id;

-- Adicionar a restrição de chave estrangeira
ALTER TABLE pedido ADD FOREIGN KEY (id_cliente) REFERENCES cliente(id);

-- Remover a coluna antiga
ALTER TABLE pedido DROP COLUMN cliente_nome;

-- Ajustar a coluna para não permitir valores nulos se necessário
ALTER TABLE pedido MODIFY COLUMN id_cliente INT NOT NULL;
```

---

2. Exercícios de Views

O objetivo desses exercícios é praticar a criação de views no MySQL, utilizando as tabelas do banco de dados loja.sql e as alterações propostas.
Instruções: Para cada exercício, escreva a instrução SQL CREATE VIEW e, em seguida, uma consulta SELECT para testar a View recém-criada.

Exercício 1: vw_produtos_completo

Crie uma view que liste todos os produtos, incluindo o nome da categoria e o nome da marca. A view deve ter as seguintes colunas: produto_id, produto_nome, categoria_nome, marca_nome.
Dica: Use JOINs entre as tabelas produto, categoria e marca.

Exercício 2: vw_estoque_baixo

Crie uma view que mostre os produtos com quantidade em estoque menor ou igual a 5. A view deve listar produto_nome, tamanho, cor e quantidade.
Dica: Use JOINs entre produto e estoque, e adicione uma cláusula WHERE para filtrar a quantidade.

Exercício 3: vw_vendas_por_cliente

Crie uma view que liste cada cliente e o número total de pedidos que ele fez. A view deve exibir cliente_nome e total_pedidos.
Dica: Use JOINs entre cliente e pedido e a função de agregação COUNT() com GROUP BY.

Exercício 4: vw_detalhes_pedidos

Crie uma view que exiba uma lista detalhada de todos os itens de pedidos. As colunas devem incluir: pedido_id, data_pedido, cliente_nome, produto_nome, tamanho, cor, quantidade_vendida, e valor_unitario.
Dica: Este é o exercício mais complexo. Você precisará de múltiplos JOINs entre pedido, pedido_item, estoque, produto e cliente.

Exercício 5: vw_faturamento_por_categoria

Crie uma view que calcule o faturamento total para cada categoria de produto. A view deve ter as colunas: categoria_nome e faturamento_total.
Dica: Combine JOINs (entre pedido_item, estoque, produto e categoria), a função de agregação SUM() e a cláusula GROUP BY.

Exercício 6: vw_vendas_recents_cliente_carlos

Crie uma view que mostre todos os pedidos feitos pelo cliente 'Carlos Silva'. A view deve ter pedido_id, data_pedido, valor_total e produto_nome.
Dica: Use JOINs para conectar as tabelas necessárias e uma cláusula WHERE para filtrar pelo nome do cliente. Se você seguiu as instruções de alteração de tabela, use a tabela cliente e a chave estrangeira id_cliente.

Exercício 7: vw_produtos_disponiveis

Crie uma view que liste todos os produtos com quantidade em estoque maior que 0. As colunas devem ser produto_nome, tamanho, cor, quantidade. Esta view deve ser usada para um catálogo de produtos online.
Dica: Use um JOIN entre produto e estoque e a cláusula WHERE para a quantidade.
