

Lista de Exercícios - Administração de Banco de Dados

Baseado no esquema relacional da base de dados "Loja Digital" (tabelas: clientes, pedidos e produtos).

5 Exercícios de Criação de Views (Tabelas Virtuais Lógicas)

1. **Consolidação de Vendas:** Crie uma view chamada vw_resumo_vendas que utilize JOINS para exibir o nome do cliente, o nome do produto, a quantidade comprada e o total (quantidade * preco).
2. **Auditoria de Estoque:** Crie a view vw_estoque_critico para retornar apenas os registros da tabela de produtos cujo estoque seja menor que 10.
3. **Ocultação Estrutural:** Crie uma view baseada na tabela clientes que exponha apenas as colunas nome e email, protegendo colunas mais sensíveis (como o ID) e fornecendo apenas os dados necessários para leitura.
4. **Simplificação de Leitura:** Crie uma view que relacione pedidos e produtos para exibir o ID do pedido e o respectivo preço do produto, aplicando a simplificação de JOINS e evitando duplicação de dados em disco.
5. **Catálogo Simplificado:** Crie uma view que exponha exclusivamente o nome e o preço da tabela produtos, servindo como uma tabela virtual para consultas de setores que não devem visualizar a quantidade em estoque.

5 Exercícios de Criação de Stored Procedures

1. **Transação de Venda:** Crie a procedure sp_novo_pedido recebendo parâmetros IN (ID do cliente, ID do produto e quantidade). A rotina deve usar um bloco BEGIN/END para inserir o pedido na tabela pedidos e, em seguida, fazer o UPDATE para baixar o estoque na tabela produtos.
2. **Atualização de Valores:** Crie a procedure sp_atualiza_preco que receba o ID de um produto e seu novo preço como parâmetros de entrada (IN), e realize o UPDATE internamente.
3. **Uso de Retorno (OUT):** Crie uma rotina pré-compilada que receba o ID de um produto e utilize um parâmetro OUT para retornar à aplicação qual é a quantidade atual em estoque daquele item.
4. **Modificação em Trânsito (INOUT):** Escreva uma Stored Procedure que receba o preço de um produto através de um parâmetro INOUT. A procedure deve calcular um desconto nesse valor e modificar a própria variável para retornar o resultado.
5. **Aplicação de Delimitadores:** Crie uma procedure simples para inserir um novo cliente na tabela clientes, garantindo a alteração temporária do delimitador padrão do MySQL (para

// ou \$\$) para evitar que a instrução seja finalizada prematuramente.



2 Exercícios de DCL (Segurança Granular)

1. **Criação e Concessão de Menor Privilégio:** Crie o usuário externo 'vendedor'@'%' com uma senha forte. Em seguida, ao invés de dar acesso à tabela bruta, conceda o privilégio de SELECT exclusivamente na view loja_db.vw_resumo_vendas.
2. **Remoção de Privilégios e Recarga:** Simule a correção de um privilégio acidental executando o comando REVOKE para remover a permissão de DELETE de toda a base (loja_db.*) do usuário 'vendedor'@'%. Após isso, escreva o comando responsável por forçar a recarga dos privilégios na memória (FLUSH PRIVILEGES).