

Desafio SQL - Controle de Estoque de uma Distribuidora de Combustíveis

Contexto

Uma distribuidora de combustíveis precisa manter um controle rigoroso do seu estoque, vendas e compras de produtos como gasolina, etanol e diesel. Para isso, é necessário desenvolver um banco de dados relacional que permita armazenar informações sobre fornecedores, clientes, produtos, transações de compra e venda, bem como registros de estoque. O objetivo deste desafio é criar um banco de dados SQL completo, desde o levantamento dos requisitos até a implementação de consultas avançadas.

Requisitos do Desafio

1. Interpretação e Modelagem do Banco de Dados

- 1.1 Leia atentamente o contexto acima e liste pelo menos **seis tabelas** essenciais para o funcionamento do sistema. (Se quiser acrescentar mais tabelas fica critério do time)
 - 1.2 Elabore um **diagrama entidade-relacionamento (DER)** representando a estrutura do banco de dados.
 - 1.3 Defina os atributos de cada tabela, garantindo integridade referencial e aplicando normalização adequada.
-

2. Criação do Banco de Dados e Estruturas (DDL - Data Definition Language)

- 2.1 Utilize comandos SQL para criar o banco de dados DistribuidoraCombustiveis.
 - 2.2 Crie as seis tabelas principais do sistema, assegurando o uso de **chaves primárias e estrangeiras, restrições de integridade e tipos de dados adequados**. As tabelas recomendadas são:
 - **Fornecedores** (ID, Nome, CNPJ, Telefone, Endereço)
 - **Clientes** (ID, Nome, CPF/CNPJ, Telefone, Endereço)
 - **Produtos** (ID, Nome, Tipo, Preço Unitário, Unidade de Medida)
 - **Estoque** (ID, Produto_ID, Quantidade, Data_Ultima_Atualizacao)
 - **Compras** (ID, Fornecedor_ID, Produto_ID, Quantidade, Data_Compra, Valor_Total)
 - **Vendas** (ID, Cliente_ID, Produto_ID, Quantidade, Data_Venda, Valor_Total)
-

3. Inserção de Dados (DML - Data Manipulation Language)

- 3.1 Insira pelo menos **cinco registros em cada tabela**, garantindo diversidade nos dados.
 - 3.2 Certifique-se de que os valores inseridos nas tabelas **respeitam as chaves estrangeiras** e a lógica do negócio.
-

4. Consultas e Relatórios (DQL - Data Query Language)

4.1 Selecione todos os produtos disponíveis no estoque e suas respectivas quantidades.

4.2 Liste todas as compras feitas de um fornecedor específico dentro de um determinado período.

4.3 Exiba os clientes que mais compraram combustível no último mês.

4.4 Calcule o valor total das vendas de cada tipo de combustível no último trimestre.

4.5 Identifique os produtos que estão abaixo de um estoque mínimo definido (por exemplo, menos de 100 litros).

5. Criação de Views

5.1 Crie uma **view** chamada **vw_relatorio_vendas**, que exiba de maneira consolidada os seguintes dados:

- Nome do cliente
- Nome do produto
- Quantidade comprada
- Data da venda
- Valor total da venda

5.2 Crie uma **view** chamada **vw_estoque_atual**, que exiba o nome do produto e sua quantidade atual no estoque.

6. Procedures e Functions

6.1 Crie uma **stored procedure** chamada **sp_registrar_compra** para registrar automaticamente uma nova compra e atualizar o estoque.

6.2 Crie uma **stored function** chamada **fn_calcular_valor_venda** que, ao receber como entrada a quantidade e o preço do produto, retorne o valor total da venda.

6.3 Crie uma **stored function** chamada **fn_percentual_vendas_produto**, que retorne o percentual de vendas de um produto específico em relação ao total de vendas de todos os produtos.

7. Triggers

7.1 Crie um **trigger** chamado **tr_atualizar_estoque_venda**, que subtraia automaticamente do estoque a quantidade de produtos vendidos ao inserir um novo registro na tabela de vendas.

7.2 Crie um **trigger chamado tr_atualizar_estoque_compra**, que adicione automaticamente ao estoque a quantidade de produtos comprados ao inserir um novo registro na tabela de compras.

7.3 Crie uma **tabela chamada Monitoramento_Banco** para registrar todas as operações realizadas nas tabelas do banco de dados. A tabela deve conter os seguintes campos:

- ID (chave primária)
- Ação realizada (INSERT, UPDATE, DELETE)
- Nome da tabela afetada
- Data e hora da ação
- Usuário responsável

7.4 Crie um **trigger chamado tr_monitoramento_banco**, que insira automaticamente um registro na tabela Monitoramento_Banco sempre que um registro for inserido, atualizado ou excluído em qualquer tabela principal do banco de dados.

Entrega do Desafio

- O código SQL completo deve ser armazenado em um arquivo .sql, contendo todos os comandos de criação do banco, inserção de dados e consultas.
- O diagrama entidade-relacionamento deve ser entregue em formato .png ou .pdf.
- As respostas das consultas devem ser apresentadas e explicadas pelos integrantes do grupo no final da aula com todos os arquivos.