

Funções Sugeridas para o Sistema de Banco de Dados (Banco de Dados II - 2025.1)

Este documento detalha as funções a serem desenvolvidas para o trabalho final da disciplina de Banco de Dados II, conforme as regras estabelecidas pelo professor Dr. Thiago Elias. O sistema deverá privilegiar o banco de dados, contendo um mínimo de 7 a 10 tabelas e abundantes movimentações, operando sem interface gráfica. Todas as funcionalidades de cadastro, alteração, remoção e movimentação devem ser realizadas utilizando-se de funções e triggers. O trabalho pode ser desenvolvido em dupla ou individualmente.

Prioridade 1: Funções Obrigatórias (Manipulação de Dados Genérica)

De acordo com as regras do trabalho, três funções são obrigatórias: uma para cadastramento, uma para remoção e uma para alteração de dados. Essas funções deverão receber o nome da tabela a ser manipulada como um dos parâmetros. A função de remoção deve, adicionalmente, escolher uma estratégia para manter a integridade referencial dos dados e evitar mensagens de erro do SGBD aos usuários.

1. Função para Cadastramento de Dados (INSERT)

- **Detalhe:** Esta função será genérica para inserir novos registros em qualquer tabela do banco de dados, conforme a exigência de uma função para cadastramento.
- **Parâmetros:** Os parâmetros repassados para as funções devem ser os mais próximos do usuário.
 - **p_nome_tabela:** TEXT (Nome da tabela onde os dados serão inseridos)
 - **p_colunas:** TEXT[] (Array de nomes das colunas onde os dados serão inseridos)
 - **p_valores:** TEXT[] (Array de valores correspondentes às colunas, na ordem correta)

2. Função para Alteração de Dados (UPDATE)

- **Detalhe:** Esta função permitirá atualizar registros existentes em qualquer tabela, cumprindo a exigência de uma função para alteração de dados.
- **Parâmetros:** Os parâmetros repassados para as funções devem ser os mais próximos do usuário.
 - **p_nome_tabela:** TEXT (Nome da tabela a ser atualizada)
 - **p_colunas_set:** TEXT[] (Array de nomes das colunas a serem atualizadas)
 - **p_valores_set:** TEXT[] (Array de novos valores para as colunas, na ordem correta)

- **p_coluna_condicao:** TEXT (Nome da coluna a ser usada na cláusula WHERE para identificar o registro)
- **p_valor_condicao:** TEXT (Valor da coluna de condição para identificar o registro)

3. Função para Remoção de Dados (DELETE)

- **Detalhe:** Esta função será responsável por remover registros de qualquer tabela, garantindo a integridade referencial e evitando mensagens de erro do SGBD ao usuário, conforme as diretrizes. A estratégia de integridade pode ser um "soft delete" (marcar como inativo) ou uma verificação e tratamento de dependências.
- **Parâmetros:** Os parâmetros repassados para as funções devem ser os mais próximos do usuário.
 - **p_nome_tabela:** TEXT (Nome da tabela de onde os dados serão removidos)
 - **p_coluna_condicao:** TEXT (Nome da coluna a ser usada na cláusula WHERE para identificar o registro a ser removido)
 - **p_valor_condicao:** TEXT (Valor da coluna de condição para identificar o registro a ser removido)

Prioridade 2: Funções de Relatórios e Consultas Específicas

Estas funções visam explorar as movimentações no banco de dados e gerar relatórios úteis, conforme a necessidade de "movimentações" abundantes no banco de dados.

4. Função para retornar o(s) funcionário(s) que mais vendeu(ram) em um determinado mês/ano.

- **Detalhe:** Esta função agregará os valores de vendas por funcionário e retornará o(s) que tiver(em) o maior volume no período especificado.
- **Parâmetros:**
 - **p_mes:** INT (Mês da consulta)
 - **p_ano:** INT (Ano da consulta)

5. Função para retornar a(s) loja(s) que mais vendeu(ram) em um determinado mês/ano.

- **Detalhe:** Esta função calculará o total de vendas por loja para o período especificado.
- **Parâmetros:**
 - **p_mes:** INT (Mês da consulta)
 - **p_ano:** INT (Ano da consulta)

6. Função para retornar o histórico de compras de um cliente específico.

- **Detalhe:** Apresenta todas as vendas associadas a um cliente, incluindo detalhes dos itens adquiridos.

- **Parâmetros:**
 - `p_cod_cliente`: INT (Código do cliente)
- 7. **Função para retornar a quantidade de carros em estoque para uma determinada marca, modelo ou tipo.**
 - **Detalhe:** Fornece um panorama do estoque de carros, permitindo filtros por marca, tipo ou nome do carro.
 - **Parâmetros:**
 - `p_nome_marca`: TEXT (Opcional - Nome da marca do carro)
 - `p_nome_tipo`: TEXT (Opcional - Nome do tipo de carro)
 - `p_nome_carro`: TEXT (Opcional - Nome do modelo do carro)
- 8. **Função para retornar a lista de vendas realizadas por um funcionário específico.**
 - **Detalhe:** Lista todas as vendas que um determinado funcionário realizou, incluindo os detalhes das vendas.
 - **Parâmetros:**
 - `p_cod_funcionario`: INT (Código do funcionário)
- 9. **Função para calcular o valor total gasto por um cliente em todas as suas compras.**
 - **Detalhe:** Retorna a soma de todos os `VALOR_TOTAL` das vendas atribuídas a um cliente.
 - **Parâmetros:**
 - `p_cod_cliente`: INT (Código do cliente)
- 10. **Função para listar todos os carros disponíveis para venda em uma loja específica.**
 - **Detalhe:** Exibe os carros que uma determinada loja possui em estoque, considerando a tabela `LOJA_CARRO`.
 - **Parâmetros:**
 - `p_cod_loja`: INT (Código da loja)
- 11. **Função para verificar se um funcionário bateu a meta mensal.**
 - **Detalhe:** Verifica se a `QTD_VENDIDA_NO_MES` de um funcionário atingiu ou ultrapassou sua `META_MENSAL`.
 - **Parâmetros:**
 - `p_cod_funcionario`: INT (Código do funcionário)
- 12. **Função para listar os clientes (Pessoa Física ou Jurídica) com base em um critério.**
 - **Detalhe:** Permite buscar clientes por nome, CPF ou CNPJ, e opcionalmente filtrar por tipo de cliente.
 - **Parâmetros:**
 - `p_termo_busca`: TEXT (Opcional - Termo para buscar no nome, CPF ou CNPJ)

- **p_tipo_cliente**: TEXT (Opcional - 'FISICA' ou 'JURIDICA' para filtrar por tipo de cliente)

13. Função para retornar detalhes de uma venda específica, incluindo os itens vendidos.

- **Detalhe:** Fornece uma visão completa de uma transação de venda, unindo informações de **VENDA** e **ITEM_VENDA**.
- **Parâmetros:**
 - **p_cod_venda**: INT (Código da venda)