**README - Projeto de Banco de Dados**

Este repositório contém o código SQL de um projeto de banco de dados que gerencia informações relacionadas a usuários, produtos, transações e logs em um contexto de estoque de produtos. Este README fornece uma visão geral do esquema do banco de dados, incluindo suas tabelas, procedimentos armazenados, cursores, gatilhos, inserções de dados de exemplo e algumas transações de teste.

**Estrutura do Banco de Dados**

O banco de dados é composto por várias tabelas, cada uma responsável por armazenar um tipo específico de informação. Aqui está um resumo das principais tabelas:

* **USUARIO**: Armazena informações sobre os usuários do sistema.
* **TELEFONE**: Mantém os números de telefone dos usuários.
* **FABRICANTE**: Contém detalhes sobre os fabricantes dos produtos.
* **COR**: Armazena as cores dos produtos.
* **MODELO**: Mantém informações sobre os modelos dos produtos.
* **CATEGORIA**: Define as categorias dos produtos.
* **PRODUTOS**: Armazena os detalhes dos produtos, incluindo informações sobre modelo, categoria e fornecedor.
* **TRANSACAO**: Registra as transações relacionadas ao estoque de produtos.
* **REL\_USER\_TRANS**: Relaciona usuários e transações.
* **REL\_TRANS\_PROD**: Relaciona transações e produtos.
* **USER\_LOG**: Registra informações de log relacionadas aos usuários.
* **ACTION\_LOG**: Registra ações realizadas no sistema.

Além das tabelas, o banco de dados também inclui stored procedures, cursores e gatilhos para manipulação de dados e lógica de negócios.

**Procedimentos Armazenados**

Os procedimentos armazenados fornecem uma maneira de encapsular lógica de negócios que pode ser reutilizada em várias partes do sistema. Aqui estão alguns exemplos de procedimentos armazenados neste banco de dados:

* **AddProduct**: Adiciona um novo produto ao banco de dados.
* **UpdateProductStock**: Atualiza a quantidade em estoque de um produto.
* **GetProductDetails**: Obtém os detalhes de um produto pelo seu ID.
* **AddTransaction**: Adiciona uma nova transação ao banco de dados.
* **GetTransactionsByUser**: Obtém as transações de um usuário pelo seu ID.
* **insertActionLog**: Insere uma ação no log de ações.
* **FetchUsersWithPhones**: Busca todos os usuários com seus números de telefone.
* **CalculateTotalStockValue**: Calcula o valor total dos produtos em estoque.
* **UpdateProductPricesByCategory**: Atualiza os preços dos produtos em uma determinada categoria.
* **GenerateTransactionReport**: Gera um relatório de transações com detalhes do produto.
* **IdentifyInactiveUsers**: Identifica usuários inativos.

**Gatilhos**

Os gatilhos são usados para automatizar ações no banco de dados em resposta a determinados eventos. Aqui estão alguns exemplos de gatilhos neste banco de dados:

* **validate\_cpf\_before\_insert**: Valida o CPF antes de inserir um usuário na tabela USUARIO.
* **log\_user\_creation**: Registra a criação de usuário em uma tabela separada.
* **update\_product\_quantity**: Atualiza a quantidade de produtos na tabela PRODUTOS após uma transação.
* **validate\_stock**: Valida o estoque do produto antes de uma transação.
* **update\_category\_on\_model\_change**: Atualiza a categoria do produto quando o modelo é alterado.

**Testes e Transações de Exemplo**

O banco de dados inclui inserções de dados de exemplo e algumas transações de teste para demonstrar o funcionamento das stored procedures e gatilhos. Essas transações podem ser úteis para entender como as diferentes partes do sistema interagem e como os dados são manipulados.

**Como Utilizar**

Para utilizar este projeto, siga estas etapas:

1. Crie um banco de dados no seu sistema de gerenciamento de banco de dados MySQL.
2. Execute o script SQL fornecido neste repositório para criar as tabelas, procedimentos armazenados, cursores e gatilhos.
3. Insira os dados de exemplo fornecidos no script SQL para popular o banco de dados.
4. Experimente executar as stored procedures e testar os gatilhos com transações de exemplo para entender melhor o funcionamento do sistema.