PUC Minas

Gestão Centralizada de Estoque

6602.1.01-8 – Banco de Dados

Arthur Rezende de Almeida

Gabriel Falk Prado Bonifácio de Menezes Soares Dias

Humberto Faria Menezes Samora

Samuel Valadão Pereira

Belo Horizonte, novembro de 2024

Sumário

1. Introdução

Este relatório apresenta o desenvolvimento de um sistema de gestão centralizada de estoque para empresas com múltiplas filiais ou pertencentes ao mesmo grupo. A solução foi projetada para garantir flexibilidade, escalabilidade e integridade dos dados. Ela integra funcionalidades como controle de fornecedores, rastreamento de movimentações, e gerenciamento de produtos e estoques, oferecendo suporte eficiente à cadeia logística.

O documento está estruturado em seis seções principais: análise de objetivos, análise de requisitos, modelagem do domínio, projeto do banco de dados, interação com o usuário e consultas, e conclusão. Cada seção aborda um aspecto específico do sistema, detalhando sua funcionalidade e relação com os stakeholders.

2. Análise de Objetivos

2.1. Descrição do Problema

O problema identificado está relacionado à dificuldade de gerenciar unidades de estoque em várias filiais ou empresas, especialmente no rastreamento de **movimentações** e no controle de informações sobre fornecedores e lotes de produtos. A falta de organização e rastreabilidade pode levar a perdas, atrasos, e falhas na tomada de decisões logísticas.

2.2. Situação Atual

Os stakeholders envolvidos incluem:

- **Administração Central:** Gerencia o estoque e controla fornecedores.
- **Filiais/Empresas:** Solicitam produtos de outra unidade, ou do estoque central, e fornecem dados sobre demandas.
- **Equipe de Logística:** É responsável por movimentar produtos e registrar transferências, que podem ser:
    - Interna: entre unidades.
    - Entrada: Recebimento de produtos dos fornecedores.
    - Saída: Vendas, devoluções ou condições extraordinárias (ex.: doações).
    - Perda: Extravio, acidentes logísticos, furto etc.

As dependências estão relacionadas ao envio de informações sobre estoque, movimentações entre unidades e controle de fornecedores e lotes.

2.3. Análise de Alternativas

Os principais objetivos dos stakeholders incluem:

- A administração deseja rastrear produtos, controlar lotes e identificar fornecedores.
- As filiais querem consultar a disponibilidade de produtos.
- A equipe de logística busca simplificar o registro de transferências.

Soluções propostas:

- Banco de dados relacional com tabelas para produtos, empresas/filiais, estoque, movimentações e fornecedores.
- Consultas e relatórios automatizados para controle e análise de dados.

3. Análise de Requisitos

3.1. Detalhamento da Solução

O sistema permitirá:

- Registro de produtos, fornecedores e lotes.
- Controle de movimentações entre o estoque central e as filiais.
- Geração de relatórios analíticos, como movimentações por categoria ou lotes próximos da validade.

3.2. Tabela de Requisitos

ID	Requisito Funcional	Prioridade
RF01	Registrar produtos no sistema	Alta
RF02	Consultar estoque central	Alta
RF03	Registrar movimentações de produtos	Alta
RF04	Gerar relatório de movimentações por período	Média
RF05	Gerar relatórios por categoria de produtos movimentados a cada mês de um ano especificado	Alta
RF06	Identificar qual filial realizou a maior quantidade de movimentações de saída de produtos em um período específico	Alta
RF07	Consultar produtos disponíveis apenas no estoque central	Média
RF08	Registrar e gerenciar informações detalhadas sobre fornecedores	Alta
RF09	Consultar produtos organizados por fornecedor	Alta
RF10	Gerar relatório de lotes próximos à data de validade	Alta
RF11	Listar movimentações de entrada agrupadas por empresa/filial	Alta

RF12	Listar movimentações de saída agrupadas por empresa/filial	Alta
RF13	Listar movimentações de perda agrupadas por empresa/filial	Média

Tabela 1 - Requisitos Funcionais

4. Modelagem do Domínio

Diagrama de Classes

A modelagem do domínio é representada pelas seguintes classes principais:

- **Empresas/Filiais:** Representam os locais de armazenamento ou distribuição.
- **Produtos:** Contêm informações detalhadas sobre os itens disponíveis no sistema.
- **Fornecedores:** Gerenciam as entidades que fornecem produtos à empresa.
- **Estoque:** Controla as quantidades, lotes e validade de produtos em diferentes localizações.
- **Movimentações:** Registram transferências de produtos entre empresas e filiais, bem como entradas, saídas e perdas. Cada movimentação é categorizada pelo campo "tipo", que identifica a natureza do evento.

Restrições de Integridade

- **Entidade:** Cada tabela possui uma chave primária única (PRIMARY KEY) que garante a identificação inequívoca de cada registro.
- **Referencial:** Relacionamentos entre tabelas são garantidos por chaves estrangeiras (FOREIGN KEY) para manter a consistência dos dados.
- **Domínio:** Garantem que os valores inseridos em colunas específicas estejam dentro de um conjunto válido.
- **Funcional:** Algumas combinações de colunas são únicas para evitar duplicidade indesejada. Exemplo:
  Na tabela produtos_fornecedores, a combinação produto_id e fornecedor_id é única, impedindo que o mesmo produto seja associado ao mesmo fornecedor mais de uma vez.
- **Regras de Negócio Incorporadas:** Dados críticos como data_fabricacao e data_validade em estoque devem seguir a lógica de temporalidade.

Dicionário de Dados:

- EMPRESAS_FILIAIS: Gerencia filiais e empresas (ID, Nome, Endereço, Tipo).
- PRODUTOS: Informações sobre produtos (ID, Descrição, Categoria, Fornecedor).
- FORNECEDORES: Mantém dados de fornecedores (ID, Nome, Contato, Endereco).

- PRODUTOS_FORNECEDORE: Relaciona produtos a fornecedores (Produto_ID, Fornecedor_ID).
- ESTOQUE: Quantidade de produtos armazenados (Produto_ID, Localizacao_ID, Quantidade, Fornecedor_ID, Numero_Lote, Data_Fabricacao, Data_Validade).
- MOVIMENTACOES: Histórico de transferências (Produto_ID, Origem_ID, Destino_ID, Quantidade, Data_Movimentacao, Tipo).

5. Projeto de Banco de Dados

Diagrama Entidade-Relacionamento

Inclui as entidades mencionadas e seus relacionamentos.

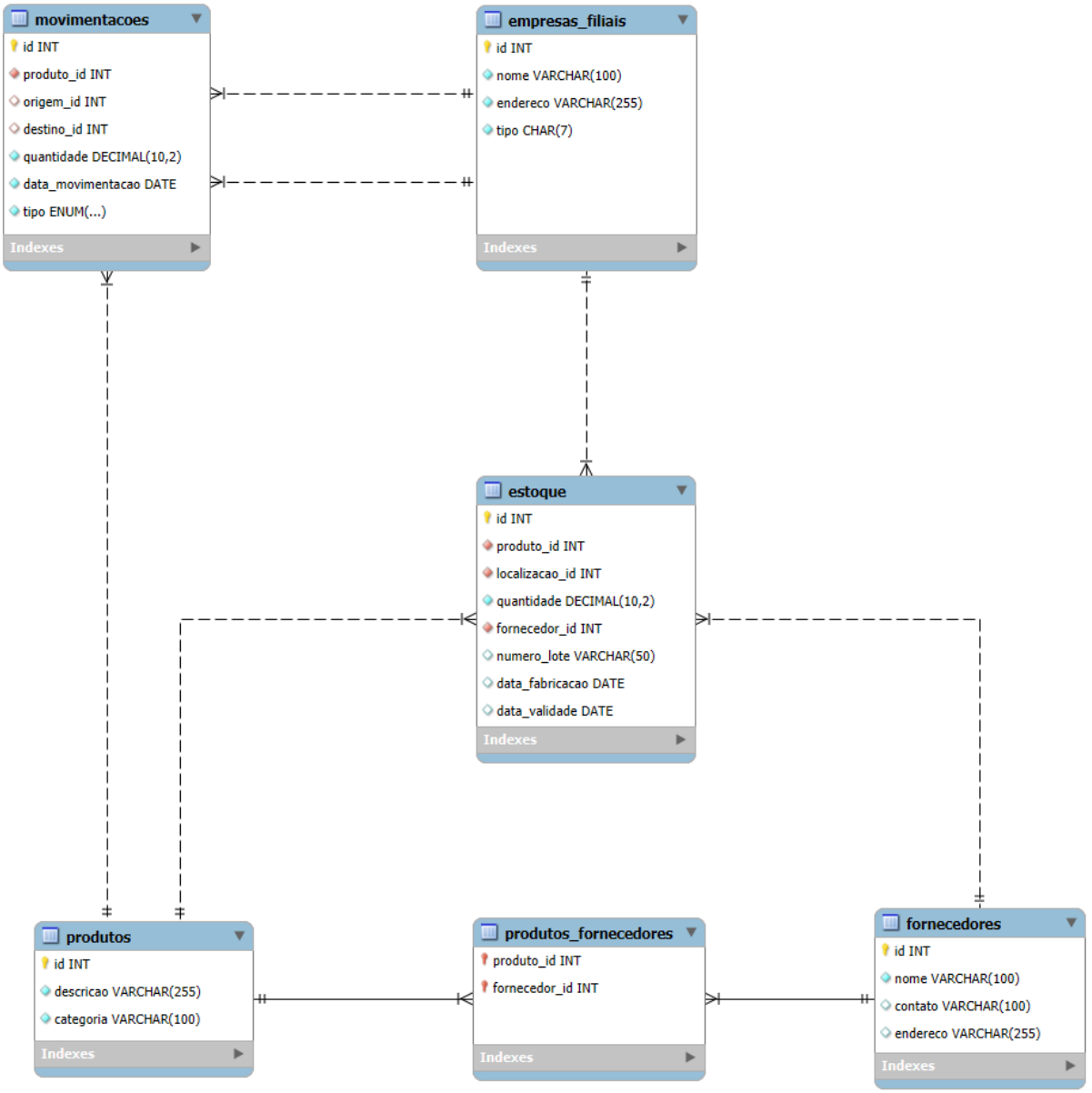

Figura 1 - Modelo Entidade Relacionamento

Estrutura do Banco de Dados

O sistema foi implementado em MySQL contendo as tabelas descritas no item 5.

Script SQL

```sql
-- Criação do banco de dados
CREATE DATABASE ESTOQUE_CENTRAL;

-- Criação da tabela empresas_filiais
CREATE TABLE empresas_filiais (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    endereco VARCHAR(255) NOT NULL,
    tipo CHAR(7) NOT NULL CHECK (tipo IN ('empresa', 'filial'))
);

-- Criação da tabela produtos
CREATE TABLE produtos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    descricao VARCHAR(255) NOT NULL,
    categoria VARCHAR(100) NOT NULL,
    fornecedor VARCHAR(100) NOT NULL
);

-- Tabela de fornecedores
CREATE TABLE fornecedores (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    contato VARCHAR(100),
    endereco VARCHAR(255)
);

-- Tabela intermediária para produtos e fornecedores (muitos-para-muitos)
CREATE TABLE produtos_fornecedores (
    produto_id INT NOT NULL,
    fornecedor_id INT NOT NULL,
    PRIMARY KEY (produto_id, fornecedor_id),
    FOREIGN KEY (produto_id) REFERENCES produtos (id),
    FOREIGN KEY (fornecedor_id) REFERENCES fornecedores (id)
);

-- Atualização da tabela de estoque para incluir novos campos
CREATE TABLE estoque (
    id INT AUTO_INCREMENT PRIMARY KEY,
    produto_id INT NOT NULL,
    localizacao_id INT NOT NULL,
    quantidade DECIMAL(10, 2) NOT NULL,
```

```
    numero_fornecedor VARCHAR(50),
    numero_lote VARCHAR(50),
    data_fabricacao DATE,
    data_validade DATE,
    FOREIGN KEY (produto_id) REFERENCES produtos (id),
    FOREIGN KEY (localizacao_id) REFERENCES empresas_filiais (id)
);

CREATE TABLE movimentacoes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    produto_id INT NOT NULL,
    origem_id INT,
    destino_id INT,
    quantidade DECIMAL(10, 2) NOT NULL,
    data_movimentacao DATE NOT NULL,
    tipo ENUM('interna', 'entrada', 'saida', 'perda') NOT NULL,
    FOREIGN KEY (produto_id) REFERENCES produtos (id),
    FOREIGN KEY (origem_id) REFERENCES empresas_filiais (id),
    FOREIGN KEY (destino_id) REFERENCES empresas_filiais (id)
);
```

6.A Aplicação

A aplicação a ser desenvolvida deve permitir executar comandos SQL relacionadas aos requisitos funcionais.

```
-- Registrar produtos no sistema (RF01)
INSERT INTO produtos (descricao, categoria, fornecedor)
VALUES ('Notebook Dell Inspiron', 'Eletrônicos', 'Dell Computadores');

-- Consultar estoque atual por filial (RF02)
SELECT p.descricao, e.nome AS localizacao, s.quantidade
FROM estoque s
JOIN produtos p ON s.produto_id = p.id
JOIN empresas_filiais e ON s.localizacao_id = e.id;

-- Registrar movimentação de produto (RF03)
INSERT INTO movimentacoes (produto_id, origem_id, destino_id, quantidade,
data_movimentacao)
VALUES (1, 2, 3, 50, '2024-11-24');

-- Gerar relatório de movimentações por período (RF04)
SELECT p.descricao, m.origem_id, m.destino_id, m.quantidade,
m.data_movimentacao
FROM movimentacoes m
JOIN produtos p ON m.produto_id = p.id
WHERE m.data_movimentacao BETWEEN '2024-11-01' AND '2024-11-30';
```

```sql
-- Relatório por categorias de produtos mais movimentados (RF05)
SELECT p.categoria, MONTH(m.data_movimentacao) AS mes, SUM(m.quantidade) AS
total_movimentado
FROM movimentacoes m
JOIN produtos p ON m.produto_id = p.id
WHERE YEAR(m.data_movimentacao) = 2024
GROUP BY p.categoria, mes
ORDER BY total_movimentado DESC;

-- Filial com mais movimentações de saída de produtos em um período (RF06):
-- Para operações interna, entrada ou perdas, adequar a cláusula WHERE
SELECT e.nome AS filial, SUM(m.quantidade) AS total_movimentado
FROM movimentacoes m
JOIN empresas_filiais e ON m.origem_id = e.id
WHERE m.tipo = 'saida'
  AND m.data_movimentacao BETWEEN '2024-01-01' AND '2024-12-31'
GROUP BY e.nome
ORDER BY total_movimentado DESC
LIMIT 1;

-- Produtos disponíveis apenas no estoque central (RF07)
SELECT p.descricao
FROM produtos p
WHERE p.id NOT IN (
    SELECT produto_id
    FROM estoque
    WHERE localizacao_id != 1 -- ID do estoque central
);

-- Registrar e gerenciar fornecedores (RF08):
INSERT INTO fornecedores (nome, contato, endereco)
VALUES ('Fornecedor A', 'contato@fornecedora.com', 'Rua Principal, 123');

-- Relacionar produtos com fornecedores (RF08)
INSERT INTO produtos_fornecedores (produto_id, fornecedor_id)
VALUES (1, 1); -- Relaciona produto com ID 1 ao fornecedor com ID 1

-- Consultar produtos por fornecedor (RF09)
SELECT f.nome AS fornecedor, p.descricao AS produto
FROM produtos_fornecedores pf
JOIN fornecedores f ON pf.fornecedor_id = f.id
JOIN produtos p ON pf.produto_id = p.id
ORDER BY f.nome, p.descricao;

-- Relatório de lotes próximos da data de validade (RF10) (ex.: 30 dias)
SELECT e.localizacao_id, p.descricao AS produto, e.numero_lote,
e.data_validade, e.quantidade, f.nome AS fornecedor
FROM estoque e
```

```sql
JOIN produtos p ON e.produto_id = p.id
JOIN produtos_fornecedores pf ON p.id = pf.produto_id
JOIN fornecedores f ON pf.fornecedor_id = f.id
WHERE e.data_validade BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 30
DAY)
ORDER BY e.data_validade;

-- Movimentações de Entrada Agrupadas por Empresa/Filial (RF11):
SELECT
    ef.nome AS empresa_filial,
    SUM(m.quantidade) AS quantidade_total_entrada
FROM movimentacoes m
JOIN empresas_filiais ef ON m.destino_id = ef.id
WHERE m.tipo = 'entrada'
GROUP BY ef.nome
ORDER BY quantidade_total_entrada DESC;

-- Movimentações de Saída Agrupadas por Empresa/Filial (RF12):
SELECT
    ef.nome AS empresa_filial,
    SUM(m.quantidade) AS quantidade_total_saida
FROM movimentacoes m
JOIN empresas_filiais ef ON m.origem_id = ef.id
WHERE m.tipo = 'saida'
GROUP BY ef.nome
ORDER BY quantidade_total_saida DESC;

-- Movimentações de Perda Agrupadas por Empresa/Filial (RF13):
SELECT
    ef.nome AS empresa_filial,
    SUM(m.quantidade) AS quantidade_total_perda
FROM movimentacoes m
JOIN empresas_filiais ef ON m.origem_id = ef.id
WHERE m.tipo = 'perda'
GROUP BY ef.nome
ORDER BY quantidade_total_perda DESC;
```

7. Considerações finais

O sistema proposto garante controle detalhado de produtos, fornecedores e estoques, promovendo eficiência e rastreabilidade. Ele pode ser expandido para atender a novas filiais ou requisitos específicos, oferecendo uma solução escalável e robusta para o gerenciamento logístico.

Os itens abaixo foram identificados como pontos de melhoria a serem implantados considerando um sistema real:

- **Cadastro de Usuários e Permissões:**

    - Implementar um módulo para gerenciar usuários do sistema, definindo níveis de acesso e responsabilidades.
    - Incluir o ID do usuário na tabela de movimentações para registrar quem realizou cada ação.
- **Cadastro de Notas Fiscais de Transporte:**
  Criar uma tabela para armazenar informações sobre notas fiscais associadas a movimentações de produtos, garantindo rastreabilidade fiscal e logística.
- **Funções para Disparar Requisições de Compra:**
  Desenvolver funções automáticas que monitorem produtos com quantidade abaixo do limite mínimo e enviem solicitações de compra para o setor responsável, reduzindo o risco de desabastecimento.
- **Integração com Sistemas de Vendas:**
  Conectar o sistema ao módulo de vendas de lojas físicas e online para sincronizar dados de estoque em tempo real, permitindo atualização dinâmica de disponibilidade de produtos.
- **Listagem de Itens Armazenados por Longo Período:**
  Implementar funções que identifiquem produtos armazenados por mais de "X" dias e sugiram promoções para liberar espaço físico, priorizando produtos com maior rotatividade.
- **Atualização da Tabela Estoque via triggers ou stored procedures:**
  A atualização da tabela estoque será gerenciada pela aplicação externa, que garantirá a consistência dos dados ao registrar movimentações e refletir as alterações no estoque. Essa abordagem é adequada para o contexto acadêmico do projeto, facilitando o aprendizado e a centralização da lógica de negócios. No entanto, para sistemas em produção com alto volume de dados e maior complexidade, recomenda-se utilizar triggers ou stored procedures no banco de dados, assegurando maior desempenho e integridade nas operações críticas.

Essas funcionalidades complementariam o sistema básico, tornando-o mais robusto, automatizado e alinhado às necessidades de uma organização moderna. Sua implantação gradual, alinhada às demandas específicas da empresa, permitiria uma integração completa entre os setores de logística, vendas e compras, otimizando os processos internos e a experiência do cliente.

8. Referências

- Sistemas de Banco de Dados - Notas de Aula. Rodrigo Baroni. PUC Minas, 2024/02.
- Sistema de Banco de Dados, Abraham SILBERSCHATZ (Autor), S. KORTH Henry F. SUDARSHAN, GEN LTC; 7ª edição (18 setembro 2020).

- MySQL 8.0 Reference Manual, https://dev.mysql.com/doc/refman/8.0/en/, consultado em 24/11/2024.