

Documento de Levantamento de Requisitos

Nome: Cahilo António Sacunja

Empresa: AntonioShop

O Problema da Empresa “AntonioShop”

A “AntonioShop”, é uma empresa que comercializa diversos produtos, enfrenta dificuldades significativas no gerenciamento de suas operações de vendas devido à ausência de um sistema estruturado e informatizado. Atualmente, os registros de clientes, vendedores, produtos e fornecedores são mantidos em planilhas e documentos físicos, o que gera os seguintes problemas:

- **Falta de controle de estoque:** Não há uma maneira eficiente de acompanhar a quantidade de produtos disponíveis, resultando em vendas de itens indisponíveis ou excesso de estoque parado.
- **Dificuldade em rastrear vendas:** As informações sobre quem vendeu, para quem e quais produtos foram vendidos não são centralizadas, dificultando a geração de relatórios e a análise de desempenho dos vendedores.
- **Dados duplicados ou inconsistentes:** A falta de um banco de dados relacional leva à redundância de informações, como endereços de clientes ou códigos de produtos registrados mais de uma vez.
- **Processos manuais demorados:** O cadastro de novos clientes, produtos ou fornecedores é lento e suscetível a erros humanos.
- **Ineficiência na gestão de fornecedores:** Não há um controle claro sobre quais fornecedores estão associados a quais produtos, dificultando a reposição de estoque.

Esses problemas impactam diretamente a eficiência operacional, a satisfação dos clientes e a lucratividade da AntonioShop. A implementação de um sistema de vendas com banco de dados relacional é essencial para resolver essas questões e permitir o crescimento sustentável da empresa.

Objetivo do Sistema

Desenvolver um sistema de vendas que permita o cadastro, gerenciamento e consulta de dados relacionados a clientes, vendedores, produtos, fornecedores e vendas realizadas, utilizando um banco de dados relacional (PostgreSQL) ou MySQL no Workbench.

Escopo do Sistema

O sistema deverá:

- Armazenar informações de clientes (ex: nome, telefone, endereço).
- Armazenar informações de vendedores (ex: nome, telefone).
- Armazenar informações de produtos (ex: nome, código, preço, estoque, fornecedor).
- Armazenar informações de fornecedores (ex.: nome, contato).
- Gerenciar vendas, associando cliente, vendedor e produtos vendidos.
- Permitir consultas e relatórios (ex: vendas por vendedor, estoque disponível).

Requisitos Funcionais

RF01: Cadastro de clientes com dados pessoais.

RF02: Cadastro de vendedores com identificação única.

RF03: Cadastro de fornecedores com informações de contato.

RF04: Cadastro de produtos com preço e estoque.

RF05: Registro de vendas com data, cliente, vendedor e produtos.

RF06: Consulta de estoque de produtos.

RF07: Relatório de vendas por vendedor ou período.

Requisitos Não Funcionais

RNF01: O sistema deve ser implementado em um banco relacional (PostgreSQL ou MySQL).

RNF02: O banco deve suportar até 10.000 vendas iniciais, com escalabilidade.

RNF03: Interface de gerenciamento via Workbench.

RNF04: Garantir integridade e consistência dos dados (ex.: evitar duplicação de dados).

Entidades e Atributos

Baseado no escopo, as entidades e seus atributos iniciais são:

Cliente:

- ID_Cliente (chave primária)
- Nome
- Telefone
- Endereço

Vendedor:

- ID_Vendedor (chave primária)
- Nome
- Telefone

Fornecedor:

- ID_Fornecedor (chave primária)
- Nome
- Contato

Produto:

- ID_Produto (chave primária)
- Nome
- Código (único)
- Preço
- Estoque
- ID_Fornecedor (chave estrangeira)

Venda:

- ID_Venda (chave primária)
- Data
- ID_Cliente (chave estrangeira)
- ID_Vendedor (chave estrangeira)
- Total

Itens_Venda (relacionamento muitos-para-muitos entre Venda e Produto):

- ID_Venda (chave estrangeira)
- ID_Produto (chave estrangeira)
- Quantidade
- Preço_Unitário

Regras de Negócio

RN01: Um cliente pode realizar várias vendas, mas uma venda pertence a um único cliente.

RN02: Um vendedor pode realizar várias vendas, mas uma venda é associada a um único vendedor.

RN03: Um fornecedor pode fornecer vários produtos, mas um produto é fornecido por apenas um fornecedor.

RN04: Uma venda pode conter vários produtos, e um produto pode estar em várias vendas.

RN05: O estoque de um produto deve ser atualizado após cada venda.

Diagrama Entidade-Relacionamento (DER)

Descrição textual :

Cliente (1,N) ----- realiza ----- (0,N) Venda

Vendedor (1,N) ----- realiza ----- (0,N) Venda

Fornecedor(1,N) ----- fornece ----- (0,N) Produto

Venda(1,N) ----- contém ----- (0,N) Itens_Venda

Produto (1,N) ----- contido em --- (0,N) Itens_Venda

Normalização até a Terceira Forma Normal (3NF)

Passo 1: Forma Não Normalizada (FNN)

Tabela Vendas_Nao_Normalizada

ID_Venda, Data, ID_Cliente, Nome_Cliente, Telefone_Cliente, Endereço_Cliente, ID_Vendedor, Nome_Vendedor, Telefone_Vendedor, ID_Produto, Nome_Produto, Código_Produto, Preço_Produto, Estoque_Produto, Quantidade_Vendida, ID_Fornecedor, Nome_Fornecedor, Contato_Fornecedor, Total_Venda

Problemas:

- Repetição de dados (ex.: Nome_Cliente repetido em várias vendas).
- Dependências funcionais não tratadas.

Passo 2: Primeira Forma Normal (1NF)

- Eliminar repetições e garantir que cada coluna tenha valores atômicos.
- Separar os itens de venda em uma tabela própria (relação muitos-para-muitos).

Tabelas:

1. Cliente: ID_Cliente, Telefone, Endereço
2. Vendedor: ID_Vendedor, Nome, Telefone
3. Fornecedor: ID_Fornecedor, Nome, Contato
4. Produto: ID_Produto, Nome, Código, Preço, Estoque, ID_Fornecedor
5. Venda: ID_Venda, Data, ID_Cliente, ID_Vendedor, Total
6. Itens_Venda: ID_Venda, ID_Produto, Quantidade, Preço_Unitário

Passo 3: Segunda Forma Normal (2NF)

- Garantir que todos os atributos não-chave dependam completamente da chave primária.
- Produto: Todos os atributos (Nome, Código, Preço, Estoque) dependem de ID_Produto.
- Venda: Data, ID_Cliente, ID_Vendedor e Total dependem de ID_Venda.
- Itens_Venda: Quantidade e Preço_Unitário dependem da combinação ID_Venda e ID_Produto (chave composta).

Até aqui, Todas as tabelas já estão na 2NF

Passo 4: Terceira Forma Normal (3NF)

- Eliminar dependências transitivas (atributos que dependem de outros atributos não-chave).
- Verificar cada tabela:
 - Cliente: Nenhum atributo depende de outro que não seja ID_Cliente.
 - Vendedor: OK.
 - Fornecedor: OK.
 - Produto: ID_Fornecedor é uma chave estrangeira, mas todos os outros atributos dependem apenas de ID_Produto. OK.
 - Venda: Total depende de ID_Venda (pode ser calculado com base em Itens_Venda, mas é aceitável mantê-lo como atributo). OK.
 - Itens_Venda: Preço_Unitário pode ser transitivo (depende do preço do produto no momento da venda), mas como é fixado na venda, é OK mantê-lo.

Resultado final em 3NF:

1. Cliente: (ID_Cliente PK, Nome, CPF UNIQUE, Telefone, Endereço)
2. Vendedor: (ID_Vendedor PK, Nome, Matrícula UNIQUE, Telefone)
3. Fornecedor: (ID_Fornecedor PK, Nome, CNPJ UNIQUE, Contato)
4. Produto: (ID_Produto PK, Nome, Código UNIQUE, Preço, Estoque, ID_Fornecedor FK)
5. Venda: (ID_Venda PK, Data, ID_Cliente FK, ID_Vendedor FK, Total)
6. Itens_Venda: (ID_Venda FK, ID_Produto FK, Quantidade, Preço_Unitário) - Chave primária composta: (ID_Venda, ID_Produto)

Thankyou