**SENAI NAMI JAFET**

**DESENVOLVIMENTO DE SISTEMA S**

**IGOR TEIXEIRA SOLSA**

**LEONARDO PORFIRIO CORREA**

**LUCAS DO ESPIRITO SANTOS GOMES**

**OTÁVIO FLAUZINO DE SOLSA**

**DOCES S/A**

**MOGI DAS CRUZES**

**2024**

**VISÃO GERAL DO BANCO DE DADOS**

O banco de dados DB\_DocesSA foi criado para gerenciar um sistema de pedidos de produtos, com informações sobre clientes, consultores, pedidos, estoque, e lista de pedidos. Ele possui as seguintes tabelas:

1. Cliente
2. Consultor
3. Pedido
4. Produto
5. Estoque
6. Endereco\_Estoque
7. Lista\_Pedido

**ESTRUTURA DAS TABELAS**

Cliente

Descrição: Armazena informações dos clientes.

Campos:

id\_cliente (PK): Identificador único do cliente.

nome: Nome do cliente.

cnpj: CNPJ do cliente.

endereco: Endereço do cliente.

cidade: Cidade do cliente.

Consultor

Descrição: Armazena informações dos consultores responsáveis pelos pedidos.

Campos:

id\_consultor (PK): Identificador único do consultor.

nome: Nome do consultor.

sexo: Sexo do consultor (M/F).

cidade: Cidade do consultor.

cpf: CPF do consultor.

email: E-mail do consultor.

Pedido

Descrição: Armazena os pedidos realizados pelos clientes.

Campos:

id\_pedido (PK): Identificador único do pedido.

id\_consultor (FK): Identificador do consultor que fez o pedido (relacionado à tabela Consultor).

id\_cliente (FK): Identificador do cliente que fez o pedido (relacionado à tabela Cliente).

id\_lista (FK): Identificador da lista de produtos incluídos no pedido (relacionado à tabela Lista\_Pedido).

data\_pedido: Data do pedido.

Produto

Descrição: Armazena informações sobre os produtos disponíveis para pedidos.

Campos:

id\_produto (PK): Identificador único do produto.

nome: Nome do produto.

valor: Preço do produto.

validade: Validade do produto (em meses).

Estoque

Descrição: Armazena informações sobre o estoque de produtos.

Campos:

id\_estoque: Identificador único do estoque.

data\_compra: Data da compra do produto.

lote: Lote do produto.

quantidade: Quantidade do produto no estoque.

id\_produto (FK): Identificador do produto (relacionado à tabela Produto).

Endereco\_Estoque

Descrição: Armazena o endereço do estoque de cada produto.

Campos:

id\_estoque (FK): Identificador do estoque (relacionado à tabela Estoque).

endereco: Endereço do estoque.

cidade: Cidade do estoque.

Lista\_Pedido

Descrição: Armazena os produtos incluídos em cada pedido, com a quantidade de cada um.

Campos:

id\_lista (PK): Identificador único da lista de produtos.

id\_produto (FK): Identificador do produto (relacionado à tabela Produto).

quantidade: Quantidade do produto no pedido.

**RELACIONAMENTOS ENTRE AS TABELAS**

Pedido ↔ Consultor: Um pedido está relacionado a um consultor através da chave estrangeira id\_consultor.

Pedido ↔ Cliente: Um pedido está relacionado a um cliente através da chave estrangeira id\_cliente.

Pedido ↔ Lista\_Pedido: Um pedido possui uma lista de produtos (id\_lista), que é registrada na tabela Lista\_Pedido.

Lista\_Pedido ↔ Produto: Cada item na lista de pedidos é relacionado a um produto através da chave estrangeira id\_produto.

Estoque ↔ Produto: Cada item de estoque está relacionado a um produto através da chave estrangeira id\_produto.

Endereco\_Estoque ↔ Estoque: Cada endereço de estoque é vinculado ao estoque através da chave estrangeira id\_estoque.

**OPERAÇÕES CRUD** (Create, Read, Update, Delete)

Cliente

Criar (Insert):

sql

**INSERT INTO Cliente (nome, cnpj, endereco, cidade)**

**VALUES ('Nome Cliente', 'XX.XXX.XXX/0001-XX', 'Rua Exemplo, 123', 'Cidade Exemplo');**

**Ler (Select):**

**SELECT \* FROM Cliente WHERE id\_cliente = 1;**

**Atualizar (Update):**

**UPDATE Cliente**

**SET nome = 'Novo Nome', endereco = 'Novo Endereco'**

**WHERE id\_cliente = 1;**

**Deletar (Delete):**

**DELETE FROM Cliente WHERE id\_cliente = 1;**

**Criar (Insert):**

**INSERT INTO Consultor (nome, sexo, cidade, cpf, email)**

**VALUES ('Nome Consultor', 'M', 'Cidade Consultor', '123.456.789-00', 'email@dominio.com');**

**Ler (Select):**

**SELECT \* FROM Consultor WHERE id\_consultor = 1;**

**Atualizar (Update):**

**UPDATE Consultor**

**SET nome = 'Novo Nome', cidade = 'Nova Cidade'**

**WHERE id\_consultor = 1;**

**Deletar (Delete):**

**DELETE FROM Consultor WHERE id\_consultor = 1;**

**Pedido**

**Criar (Insert):**

**INSERT INTO Pedido (id\_consultor, id\_cliente, id\_lista, data\_pedido)**

**VALUES (1, 1, 2, '2024-11-21');**

**Ler (Select):**

**SELECT \* FROM Pedido WHERE id\_pedido = 1;**

**Atualizar (Update):**

**UPDATE Pedido**

**SET data\_pedido = '2024-11-22'**

**WHERE id\_pedido = 1;**

**Deletar (Delete):**

**DELETE FROM Pedido WHERE id\_pedido = 1;**

**Produto**

**Criar (Insert):**

**INSERT INTO Produto (nome, valor, validade)**

**VALUES ('Produto Exemplo', 50.0, 12);**

**Ler (Select):**

**SELECT \* FROM Produto WHERE id\_produto = 1;**

**Atualizar (Update):**

**UPDATE Produto**

**SET valor = 55.0**

**WHERE id\_produto = 1;**

**Deletar (Delete):**

**DELETE FROM Produto WHERE id\_produto = 1;**

**Estoque**

**Criar (Insert):**

**INSERT INTO Estoque (data\_compra, lote, quantidade, id\_produto)**

**VALUES ('2024-11-20', 123, 100, 1);**

**Ler (Select):**

**SELECT \* FROM Estoque WHERE id\_produto = 1;**

**Atualizar (Update):**

UPDATE Estoque

**SET quantidade = 120**

**WHERE id\_produto = 1;**

**Deletar (Delete):**

**DELETE FROM Estoque WHERE id\_produto = 1;**

**Lista\_Pedido**

**Criar (Insert):**

**INSERT INTO Lista\_Pedido (id\_produto, quantidade)**

**VALUES (1, 2);**

**Ler (Select):**

**SELECT \* FROM Lista\_Pedido WHERE id\_lista = 1;**

**Atualizar (Update):**

**UPDATE Lista\_Pedido**

**SET quantidade = 3**

**WHERE id\_lista = 1 AND id\_produto = 1;**

**Deletar (Delete):**

**DELETE FROM Lista\_Pedido WHERE id\_lista = 1 AND id\_produto = 1;**

**CONSIDERAÇÕES PARA A API**

A API deve garantir que os dados sejam validados antes de serem inseridos no banco de dados (por exemplo, garantir que os campos de CPF, CNPJ, e-mail estejam no formato correto).

Utilizar transações para garantir que a inserção de dados em múltiplas tabelas (como em um pedido e sua lista de produtos) seja realizada de forma atômica.