Modelagem e implementação de um banco de dados simples, utilizando como base o SQL Server.

1º Procedimento | Criando o Banco de Dados

Aluno: Caio Viana Castelo Branco

Matrícula: 202307465925

Disciplina: RPG0015 - Vamos manter as informações!

Curso: Desenvolvimento Full Stack

Turma: 9001

Semestre Letivo: 2025.1

Campus: Parangaba

1. Objetivos da prática

- Identificar os requisitos de um sistema e transformá-los no modelo adequado;
- Utilizar ferramentas de modelagem para bases de dados relacionais;
- Explorar a sintaxe SQL na criação das estruturas do banco (DDL);
- Explorar a sintaxe SQL na consulta e manipulação de dados (DML);
- No final do exercício, o aluno terá vivenciado a experiência de modelar a base de dados para um sistema simples, além de implementá-la, através da sintaxe SQL, na plataforma do SQL Server.

2. Códigos utilizados na prática

- Criação das tabelas e relacionamentos:

```
CREATE TABLE Pessoa (
 id_pessoa INT PRIMARY KEY,
 tipo CHAR(1) NOT NULL CHECK (tipo IN ('F', 'J')),
 nome VARCHAR(100) NOT NULL,
 endereco VARCHAR(200),
 telefone VARCHAR(20)
);


CREATE SEQUENCE seq_pessoa_id START WITH 1 INCREMENT BY 1;


CREATE TABLE PessoaFisica (
 id_pessoa INT PRIMARY KEY,
 cpf VARCHAR(14) UNIQUE NOT NULL,
 FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id_pessoa)
);


CREATE TABLE PessoaJuridica (
id_pessoa INT PRIMARY KEY,
cnpj VARCHAR(18) UNIQUE NOT NULL,
FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id_pessoa)
);


CREATE TABLE Usuario (
 id_usuario INT PRIMARY KEY IDENTITY(1,1),
 login VARCHAR(50) UNIQUE NOT NULL,
 senha VARCHAR(100) NOT NULL,
 nome VARCHAR(100) NOT NULL
);
```

```
CREATE TABLE Produto (
 id_produto INT PRIMARY KEY IDENTITY(1,1),
 nome VARCHAR(100) NOT NULL, quantidade INT DEFAULT 0,
 preco_venda DECIMAL(10,2) NOT NULL
);


CREATE TABLE Movimento (
 id_movimento INT PRIMARY KEY IDENTITY(1,1),
 tipo CHAR(1) CHECK (tipo IN ('C', 'V')),
 id_usuario INT NOT NULL,
 id_produto INT NOT NULL,
 id_pessoa INT NOT NULL,
 quantidade INT NOT NULL,
 preco_unitario DECIMAL(10,2) NOT NULL,
 data_hora DATETIME DEFAULT GETDATE(),
 FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
 FOREIGN KEY (id_produto) REFERENCES Produto(id_produto),
 FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id_pessoa)
);
```

- Inserção de dados para testes

```
INSERT INTO Pessoa (id_pessoa, tipo, nome, endereco, telefone)
VALUES (NEXT VALUE FOR seq_pessoa_id, 'F', 'João Silva', 'Rua A, 123', '(85) 9999-8888');

INSERT INTO PessoaFisica (id_pessoa, cpf)
VALUES (IDENT_CURRENT('Pessoa'), '123.456.789-00');


INSERT INTO Produto (nome, quantidade, preco_venda)
VALUES ('Notebook Dell', 10, 4500.00);


INSERT INTO Usuario (login, senha, nome)
VALUES ('admin', '123senha123', 'Administrador');
```

3. Análise e Conclusão

a) Como são implementadas as diferentes cardinalidades, basicamente 1X1, 1XN ou NxN, em um banco de dados relacional?

Em bancos relacionais, as cardinalidades são implementadas usando chaves primárias (PK) e chaves estrangeiras (FK):

- 1:1 (Um para Um): Uma tabela referencia a outra usando uma FK que também é PK ou tem uma restrição UNIQUE.
- 1:N (Um para Muitos): A tabela do lado "N" referencia a tabela do lado "1" via FK.
- N:N (Muitos para Muitos): Usa-se uma tabela de junção (associação) com FKs para as duas tabelas relacionadas.

b) Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais?

Existem três abordagens comuns para mapear herança (herança de classes em POO para o modelo relacional):

- Tabela única (Single Table): Todas as subclasses são armazenadas em uma única tabela com campos opcionais (nullable) e um discriminador.
- Tabela por classe (Class Table): Cada classe (pai e filhas) vira uma tabela, com FKs ligando-as.
- Tabela por concreto (Concrete Table): Cada subclasse vira uma tabela independente, repetindo os campos da superclasse (sem FK).

c) Como o SQL Server Management Studio permite a melhoria da produtividade nas tarefas relacionadas ao gerenciamento do banco de dados?

O SQL Server Management Studio aumenta a produtividade oferecendo uma interface gráfica que simplifica a criação e gestão de bancos de dados, dispensando a necessidade de digitar SQL manualmente.

Ele inclui ferramentas como Query Builder para consultas visuais, depurador para procedures, análise de desempenho de queries e automação de scripts. Além disso, integra-se com Azure e outros serviços da Microsoft, o que agiliza tarefas de administração e reduz erros, especialmente para quem tem menos experiência com SQL “puro”.