



Campus Boa Viagem

Curso: Desenvolvimento Full Stack

Disciplina: Vamos Manter as Informações?

Turma: 2025.1 - 3º semestre letivo

Integrante: Anderson Felipe Felix da Silva

Matrícula: 202401185711

Relatório de Prática – Projeto SQL

Objetivo da Prática

O objetivo desta prática foi desenvolver a estrutura de um banco de dados relacional utilizando SQL, com a criação de tabelas, definição de chaves primárias e estrangeiras, e aplicação de comandos DDL (Data Definition Language) e DML (Data Manipulation Language).

Códigos Desenvolvidos

```
CREATE TABLE Pessoa (  
    idPessoa INT NOT NULL IDENTITY,  
    nome VARCHAR(100) NOT NULL,  
    endereco VARCHAR(255) NOT NULL,  
    telefone VARCHAR(255),  
    email VARCHAR(255),  
    PRIMARY KEY(idPessoa)  
);
```

```
CREATE TABLE PessoaFisica (  
    idPessoa INT NOT NULL,  
    cpf CHAR(11),  
    PRIMARY KEY(idPessoa),  
    FOREIGN KEY(idPessoa) REFERENCES Pessoa(idPessoa)  
);
```

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

```
CREATE TABLE Produto (  
    idProduto INT NOT NULL IDENTITY,  
    nome VARCHAR(255),  
    quantidade INT,  
    precoVenda DECIMAL(10, 2),  
    PRIMARY KEY(idProduto)  
);
```

```
CREATE TABLE Compra (  
    idCompra INT NOT NULL IDENTITY,
```

```
idCompra INT NOT NULL IDENTITY,  
idUsuario INT,  
idPessoa INT,  
idProduto INT,  
quantidade INT,  
preco_unitario DECIMAL(10, 2),  
data_compra DATETIME,  
PRIMARY KEY(idCompra),  
FOREIGN KEY(idUsuario) REFERENCES Usuario(idUsuario),  
FOREIGN KEY(idPessoa) REFERENCES Pessoa(idPessoa),  
FOREIGN KEY(idProduto) REFERENCES Produto(idProduto)  
);
```

```
CREATE TABLE Movimento (  
idMovimento INT NOT NULL IDENTITY,  
idCompra INT,  
idUsuario INT,  
PRIMARY KEY(idMovimento),  
FOREIGN KEY(idCompra) REFERENCES Compra(idCompra),  
FOREIGN KEY(idUsuario) REFERENCES Usuario(idUsuario)  
);
```

Resultados da Execução

As tabelas foram criadas com sucesso no banco de dados, utilizando o SGBD compatível com SQL Server. Cada tabela possui suas chaves primárias e relacionamentos definidos com FOREIGN KEY, garantindo a integridade dos dados. O campo IDENTITY foi utilizado para gerar identificadores únicos automaticamente.

Análise e Conclusão

1. Diferenças entre SEQUENCE e IDENTITY:

- IDENTITY é utilizado diretamente na definição de uma coluna para gerar valores incrementais automaticamente. Ele pertence à tabela.
- SEQUENCE é um objeto separado no banco de dados que pode ser reutilizado por várias tabelas. É mais flexível.

2. Importância das chaves estrangeiras:

As chaves estrangeiras garantem a integridade referencial entre tabelas, evitando dados inconsistentes.

3. Operadores da Álgebra Relacional vs. Cálculo Relacional:

- Álgebra Relacional: SELECT, PROJECT, UNION, DIFFERENCE, JOIN.

- Cálculo Relacional: Expressões lógicas. SQL combina os dois, mas se aproxima mais da álgebra.

4. Agrupamento em Consultas:

Feito com GROUP BY. Toda coluna no SELECT deve estar no GROUP BY ou ser usada com funções de agregação (ex: SUM, COUNT).

Exemplo:

```
SELECT idUsuario, COUNT(*) AS total_compras  
FROM Compra  
GROUP BY idUsuario;
```