

UNIVERSIDADE XYZ - CAMPUS CENTRAL

Curso: Desenvolvimento Full Stack

Disciplina: Banco de Dados

Turma: 3º Período - Semestre 2025/1

Título da Prática

Modelagem e Criação de Banco de Dados com SQL Server Management Studio.

Objetivo da Prática

Demonstrar a habilidade de modelagem e criação de banco de dados relacional com uso de herança, relacionamentos e consultas SQL.

Código: Criação do Banco de Dados

```
-- Criação do banco e login
CREATE LOGIN loja WITH PASSWORD = 'loja';
CREATE DATABASE LojaDB;
GO

USE LojaDB;
GO

CREATE USER loja FOR LOGIN loja;
ALTER ROLE db_owner ADD MEMBER loja;
GO

-- Sequence para ID de Pessoa
CREATE SEQUENCE seq_pessoa_id
    START WITH 1
    INCREMENT BY 1;
GO

-- Tabelas
CREATE TABLE Usuario (
    id INT IDENTITY(1,1) PRIMARY KEY,
    nome_usuario VARCHAR(50) NOT NULL,
    senha VARCHAR(50) NOT NULL
```

UNIVERSIDADE XYZ - CAMPUS CENTRAL

Curso: Desenvolvimento Full Stack

Disciplina: Banco de Dados

Turma: 3º Período - Semestre 2025/1

```
);
```

```
CREATE TABLE Pessoa (  
    id INT PRIMARY KEY,  
    nome VARCHAR(100),  
    endereco VARCHAR(150),  
    telefone VARCHAR(20),  
    email VARCHAR(100)  
);
```

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

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

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

```
CREATE TABLE Movimentacao (  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    tipo CHAR(1) CHECK (tipo IN ('E', 'S')),  
    id_produto INT,  
    id_operador INT,  
    id_pessoa INT,  
    quantidade INT,  
    preco_unitario DECIMAL(10,2),  
    FOREIGN KEY (id_produto) REFERENCES Produto(id),  
    FOREIGN KEY (id_operador) REFERENCES Usuario(id),  
    FOREIGN KEY (id_pessoa) REFERENCES Pessoa(id)  
);
```

UNIVERSIDADE XYZ - CAMPUS CENTRAL

Curso: Desenvolvimento Full Stack

Disciplina: Banco de Dados

Turma: 3º Período - Semestre 2025/1

Código: Inserção de Dados

```
-- Inserir usuários
INSERT INTO Usuario (nome_usuario, senha) VALUES ('op1', 'op1');
INSERT INTO Usuario (nome_usuario, senha) VALUES ('op2', 'op2');

-- Inserir produtos
INSERT INTO Produto (nome, quantidade, preco_venda) VALUES ('Mouse', 100, 50.00);
INSERT INTO Produto (nome, quantidade, preco_venda) VALUES ('Teclado', 150, 80.00);
INSERT INTO Produto (nome, quantidade, preco_venda) VALUES ('Monitor', 50, 600.00);

-- Inserir pessoas físicas
DECLARE @id1 INT = NEXT VALUE FOR seq_pessoa_id;
INSERT INTO Pessoa (id, nome, endereco, telefone, email)
VALUES (@id1, 'João Silva', 'Rua A, 123', '(11) 99999-0001', 'joao@email.com');
INSERT INTO PessoaFisica (id, cpf) VALUES (@id1, '123.456.789-00');

DECLARE @id2 INT = NEXT VALUE FOR seq_pessoa_id;
INSERT INTO Pessoa (id, nome, endereco, telefone, email)
VALUES (@id2, 'Maria Oliveira', 'Rua B, 456', '(11) 99999-0002', 'maria@email.com');
INSERT INTO PessoaFisica (id, cpf) VALUES (@id2, '987.654.321-00');

-- Inserir pessoas jurídicas
DECLARE @id3 INT = NEXT VALUE FOR seq_pessoa_id;
INSERT INTO Pessoa (id, nome, endereco, telefone, email)
VALUES (@id3, 'Tech Distribuidora', 'Av Central, 1000', '(11) 4002-8922', 'contato@tech.com');
INSERT INTO PessoaJuridica (id, cnpj) VALUES (@id3, '12.345.678/0001-99');

DECLARE @id4 INT = NEXT VALUE FOR seq_pessoa_id;
INSERT INTO Pessoa (id, nome, endereco, telefone, email)
VALUES (@id4, 'Mega Fornecedores', 'Av Norte, 200', '(11) 3003-1313', 'vendas@mega.com');
INSERT INTO PessoaJuridica (id, cnpj) VALUES (@id4, '98.765.432/0001-11');

-- Inserir movimentações
INSERT INTO Movimentacao (tipo, id_produto, id_operador, id_pessoa, quantidade, preco_unitario)
VALUES ('E', 1, 1, 3, 10, 45.00);

INSERT INTO Movimentacao (tipo, id_produto, id_operador, id_pessoa, quantidade, preco_unitario)
VALUES ('E', 3, 2, 4, 5, 550.00);

INSERT INTO Movimentacao (tipo, id_produto, id_operador, id_pessoa, quantidade, preco_unitario)
```

UNIVERSIDADE XYZ - CAMPUS CENTRAL

Curso: Desenvolvimento Full Stack

Disciplina: Banco de Dados

Turma: 3º Período - Semestre 2025/1

```
VALUES ('S', 1, 1, 1, 2, 50.00);
```

```
INSERT INTO Movimentacao (tipo, id_produto, id_operador, id_pessoa, quantidade, preco_unitario)
VALUES ('S', 2, 2, 2, 1, 80.00);
```

Código: Consultas SQL

```
-- 1. Pessoas físicas
```

```
SELECT p.id, p.nome, p.endereco, p.telefone, p.email, pf.cpf
FROM Pessoa p JOIN PessoaFisica pf ON p.id = pf.id;
```

```
-- 2. Pessoas jurídicas
```

```
SELECT p.id, p.nome, p.endereco, p.telefone, p.email, pj.cnpj
FROM Pessoa p JOIN PessoaJuridica pj ON p.id = pj.id;
```

```
-- 3. Movimentações de entrada
```

```
SELECT m.id, pr.nome AS produto, p.nome AS fornecedor, m.quantidade, m.preco_unitario,
       (m.quantidade * m.preco_unitario) AS valor_total
FROM Movimentacao m
JOIN Produto pr ON pr.id = m.id_produto
JOIN Pessoa p ON p.id = m.id_pessoa
WHERE m.tipo = 'E';
```

```
-- 4. Movimentações de saída
```

```
SELECT m.id, pr.nome AS produto, p.nome AS comprador, m.quantidade, m.preco_unitario,
       (m.quantidade * m.preco_unitario) AS valor_total
FROM Movimentacao m
JOIN Produto pr ON pr.id = m.id_produto
JOIN Pessoa p ON p.id = m.id_pessoa
WHERE m.tipo = 'S';
```

```
-- 5. Total de entradas por produto
```

```
SELECT pr.nome AS produto, SUM(m.quantidade * m.preco_unitario) AS total_entrada
FROM Movimentacao m
JOIN Produto pr ON pr.id = m.id_produto
WHERE m.tipo = 'E'
GROUP BY pr.nome;
```

```
-- 6. Total de saídas por produto
```

```
SELECT pr.nome AS produto, SUM(m.quantidade * m.preco_unitario) AS total_saida
FROM Movimentacao m
```

UNIVERSIDADE XYZ - CAMPUS CENTRAL

Curso: Desenvolvimento Full Stack

Disciplina: Banco de Dados

Turma: 3º Período - Semestre 2025/1

```
JOIN Produto pr ON pr.id = m.id_produto
WHERE m.tipo = 'S'
GROUP BY pr.nome;

-- 7. Operadores que não compraram
SELECT u.id, u.nome_usuario
FROM Usuario u
WHERE u.id NOT IN (
    SELECT DISTINCT id_operador FROM Movimentacao WHERE tipo = 'E'
);

-- 8. Total de entrada por operador
SELECT u.nome_usuario, SUM(m.quantidade * m.preco_unitario) AS total_entrada
FROM Movimentacao m
JOIN Usuario u ON u.id = m.id_operador
WHERE m.tipo = 'E'
GROUP BY u.nome_usuario;

-- 9. Total de saída por operador
SELECT u.nome_usuario, SUM(m.quantidade * m.preco_unitario) AS total_saida
FROM Movimentacao m
JOIN Usuario u ON u.id = m.id_operador
WHERE m.tipo = 'S'
GROUP BY u.nome_usuario;

-- 10. Média ponderada de venda por produto
SELECT p.nome AS produto,
    SUM(m.quantidade * m.preco_unitario) * 1.0 / SUM(m.quantidade) AS media_ponderada
FROM Movimentacao m
JOIN Produto p ON p.id = m.id_produto
WHERE m.tipo = 'S'
GROUP BY p.nome;
```

Análise e Conclusão

As cardinalidades 1x1, 1xN e NxN são implementadas com o uso de chaves primárias e estrangeiras.

A herança entre entidades, como Pessoa Física e Jurídica, foi representada com relacionamento 1x1 usando

UNIVERSIDADE XYZ - CAMPUS CENTRAL

Curso: Desenvolvimento Full Stack

Disciplina: Banco de Dados

Turma: 3º Período - Semestre 2025/1

a tabela Pessoa como base.

O uso de sequence permite controle manual e mais flexível da geração de IDs, enquanto identity é automatizado.

As chaves estrangeiras garantem integridade referencial. Operadores como SELECT, JOIN, GROUP BY, SUM, AVG fazem parte da álgebra e cálculo relacional.

O SQL Server Management Studio melhora a produtividade com funcionalidades como auto completar, geração de diagramas e execução assistida de scripts.