



## **Mundo 3**

### **Missão Prática Nível 2**

**429 Polo Centro – Porto Alegre – RS**

**Desenvolvimento Full Stack – 2023.1 – 3º Semestre Letivo**

<https://github.com/Jefferson-sandoval/missao-pratica-nivel-2-mundo-3>

#### **Objetivo 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.



## 1º Procedimento:

Modelo de Dados:

Tabela usuário

Usuario	
id_Usuario	integer
login	varchar
senha	varchar
email	varchar

Tabela Pessoa Física

Pessoa_Fisica	
id_PessoaFisica	integer
nome	varchar
logradouro	varchar
cidade	varchar
estado	varchar
telefone	varchar
email	varchar
cpf	varchar



Tabela Pessoa Jurídica

Pessoa_Juridica	
id_PessoaJuridica	integer
nome	varchar
logradouro	varchar
estado	varchar
cidade	varchar
telefone	varchar
email	varchar
cnpj	varchar

Tabela Produto

Produto	
id_Produto	integer
nome	varchar
quantidade	integer
precoVenda	decimal

Tabela Movimento de Venda

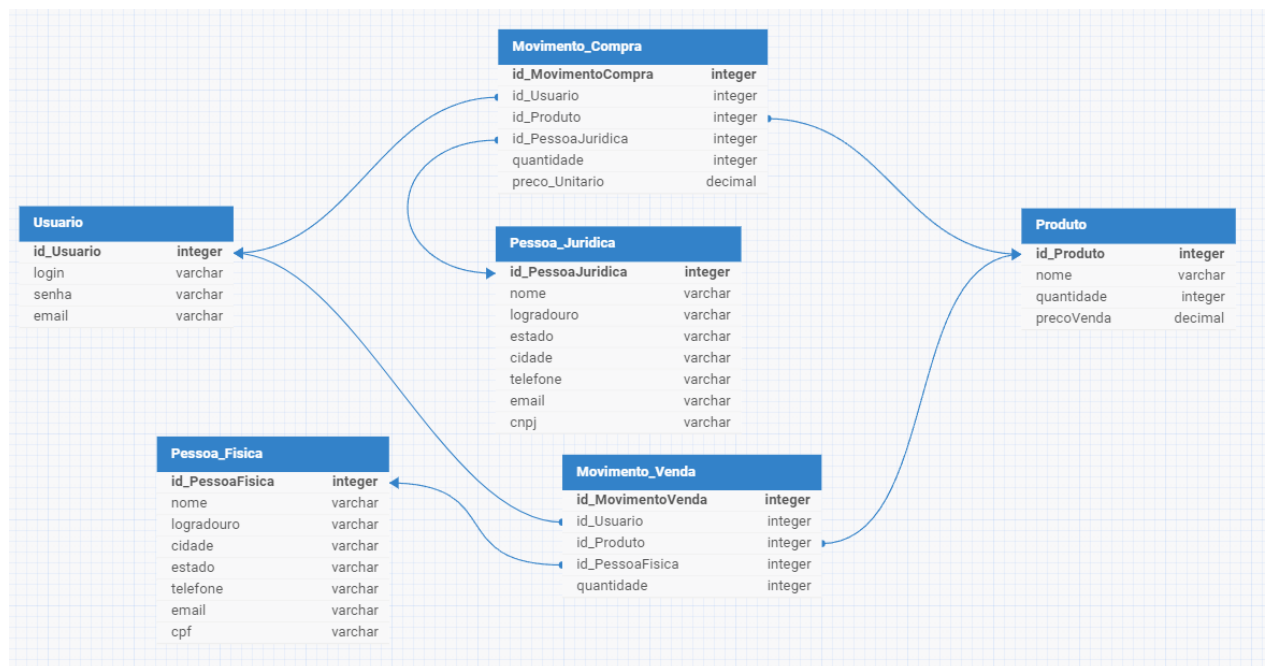
Movimento_Venda	
id_MovimentoVenda	integer
id_Usuario	integer
id_Produto	integer
id_PessoaFisica	integer
quantidade	integer



## Tabela Movimento de Compra

Movimento_Compra	
id_MovimentoCompra	integer
id_Usuario	integer
id_Produto	integer
id_PessoaJuridica	integer
quantidade	integer
preco_Unitario	decimal

## Modelagem Completa





Criando Base de Dados:

Logar no SQL Server Management Studio com o usuário **sa**.



# Estácio

Adicionar login loja com a senha loja, com as devidas permissões para criar tabelas e demais estruturas do banco de dados.

Logon - Novo

Selecionar uma página

- Geral
- Funções de Servidor
- Mapeamento de Usuário
- Protegíveis
- Status

Script ? Ajuda

Nome de logon: loja Pesquisar...

☐ Autenticação do Windows

☐ Autenticação do Microsoft Entra ID

☒ Autenticação do SQL Server

Senha: .....

Confirmar senha: .....

☐ Especificar senha antiga

Senha antiga:

☒ Impor política de senha

☐ Impor vencimento de senha

☐ O usuário deve alterar a senha no próximo logon

☐ Mapeado para certificado

☐ Mapeado para chave assimétrica

☒ Mapear para Credencial

Credenciais Mapeadas

Credencial

Adicionar

Remover

Banco de dados padrão: master

Idioma padrão: <padrão>

OK Cancelar

Conexão

Servidor: DESKTOP-U80A030

Conexão: sa

[Exibir propriedades da conexão](#)

Progresso

Pronto

Para criar o banco LojaDB.

```
create database LojaDB;
```



Criar usuário para login loja.

```
use LojaDB;  
create user loja for login loja;
```

Conceder permissões para o usuário loja.

```
GRANT CREATE TABLE TO loja;  
GRANT CREATE PROCEDURE TO loja;
```

Criando sequence.

```
SQLQuery4.sql - DES....LojaDB (loja (69))*  
CREATE SEQUENCE PessoaIdSeq  
START WITH 1  
INCREMENT BY 1;
```

Criando tabela Usuário.

```
SQLQuery4.sql - DES....LojaDB (loja (69))*  
CREATE TABLE Usuario (  
    id_Usuario INT PRIMARY KEY IDENTITY,  
    login VARCHAR(50),  
    senha VARCHAR(50),  
    email VARCHAR(100)  
);
```

Criando tabela Pessoa Física.

```
SQLQuery4.sql - DES....LojaDB (loja (69))*  
CREATE TABLE Pessoa_Fisica (  
    id_PessoaFisica INT PRIMARY KEY IDENTITY,  
    nome VARCHAR(100),  
    logradouro VARCHAR(255),  
    cidade VARCHAR(100),  
    estado VARCHAR(50),  
    telefone VARCHAR(20),  
    email VARCHAR(100),  
    cpf VARCHAR(11)  
);
```



Criando tabela Pessoa Jurídica.

```
SQLQuery4.sql - DES....LojaDB (loja (69))* X
CREATE TABLE Pessoa_Juridica (
    id_PessoaJuridica INT PRIMARY KEY IDENTITY,
    nome VARCHAR(100),
    logradouro VARCHAR(255),
    cidade VARCHAR(100),
    estado VARCHAR(50),
    telefone VARCHAR(20),
    email VARCHAR(100),
    cnpj VARCHAR(14)
);
```

Criando tabela Produto.

```
SQLQuery4.sql - DES....LojaDB (loja (69))* X
CREATE TABLE Produto (
    id_Produto INT PRIMARY KEY IDENTITY,
    nome VARCHAR(100),
    quantidade INT,
    precoVenda DECIMAL(10, 2)
);
```

Criando tabela Movimento de Compra.

```
SQLQuery4.sql - DES....LojaDB (loja (69))* X
CREATE TABLE Movimento_Compra (
    id_MovimentoCompra INT PRIMARY KEY IDENTITY,
    id_Usuario INT,
    id_Produto INT,
    id_PessoaJuridica INT,
    quantidade INT,
    preco_Unitario DECIMAL(10, 2),
    FOREIGN KEY (id_Usuario) REFERENCES Usuario(id_Usuario),
    FOREIGN KEY (id_Produto) REFERENCES Produto(id_Produto),
    FOREIGN KEY (id_PessoaJuridica) REFERENCES Pessoa_Juridica(id_PessoaJuridica)
);
```





Criando tabela Movimento de Venda.

```
SQLQuery4.sql - DES....LojaDB (loja (69))*  
CREATE TABLE Movimento_Venda (  
    id_MovimentoVenda INT PRIMARY KEY IDENTITY,  
    id_Usuario INT,  
    id_Produto INT,  
    id_PessoaFisica INT,  
    quantidade INT,  
    FOREIGN KEY (id_Usuario) REFERENCES Usuario(id_Usuario),  
    FOREIGN KEY (id_Produto) REFERENCES Produto(id_Produto),  
    FOREIGN KEY (id_PessoaFisica) REFERENCES Pessoa_Fisica(id_PessoaFisica)  
);
```

## 2º Procedimento

Inserindo dados na tabela Usuário.

```
SQLQuery15.sql - DE....LojaDB (loja (57))*  
INSERT INTO Usuario (login, senha, email)  
VALUES ('op1', 'op1', 'op1@gmail.com'),  
       ('op2', 'op2', 'op2@gmail.com');
```

Inserindo Produtos.

```
SQLQuery16.sql - DE....LojaDB (loja (69))*  
INSERT INTO Produto (nome, quantidade, precoVenda)  
VALUES ('banana', '100', '5.00'),  
       ('laranja', '500', '2.00'),  
       ('manga', '800', '4.00'),  
       ('morango', '300', '3.50');
```



Produtos inseridos.

	id_Produto	nome	quantidade	precoVenda
1	1	banana	100	5.00
2	2	laranja	500	2.00
3	3	manga	800	4.00
4	4	morango	300	3.50

Inserindo dados Pessoa Física.

```
SQLQuery17.sql - DE....LojaDB (loja (53))* - X
INSERT INTO Pessoa_Fisica (nome, logradouro, cidade, estado, telefone, email, cpf)
VALUES ('Joao', 'Rua 12, casa 3, Quitanda', 'Riacho do Sul', 'PA', '1111-1111', 'joao@riacho.com', '11111111111');
```

```
SQLQuery17.sql - DE....LojaDB (loja (53))* - X
INSERT INTO Pessoa_Fisica (nome, logradouro, cidade, estado, telefone, email, cpf)
VALUES ('Claudia', 'Rua 13, casa 146, Rio Verde', 'Curitiba', 'PA', '1111-1111', 'claudia@rioverde.com', '22255577795');
```

Dados Pessoa Física.

	id_PessoaFisica	nome	logradouro	cidade	estado	telefone	email	cpf
1	1	Joao	Rua 12, casa 3, Quitanda	Riacho do Sul	PA	1111-1111	joao@riacho.com	11111111111
2	2	Claudia	Rua 13, casa 146, Rio Verde	Curitiba	PA	1111-1111	claudia@rioverde.com	22255577795

Inserindo dados Pessoa Jurídica.

```
SQLQuery17.sql - DE....LojaDB (loja (53))* - X
INSERT INTO Pessoa_Juridica (nome, logradouro, cidade, estado, telefone, email, cnpj)
VALUES ('JJC', 'Rua 11, Centro', 'Riacho Norte', 'PA', '1212-1212', 'jjc@gmail.com', '22222222222222'),
('LBC', 'Rua A, Ipanema', 'Rio de Janeiro', 'RJ', '3555-3553', 'lbc@gmail.com', '12345877298595');
```

Dados Pessoa Jurídica.

	id_PessoaJuridica	nome	logradouro	cidade	estado	telefone	email	cnpj
1	1	JJC	Rua 11, Centro	Riacho Norte	PA	1212-1212	jjc@gmail.com	22222222222222
2	2	LBC	Rua A, Ipanema	Rio de Janeiro	RJ	3555-3553	lbc@gmail.com	12345877298595



Movimento de Entrada.

```
SQLQuery30.sql - DE....LojaDB (loja (53))* X
INSERT INTO Movimento_Compra (id_Usuario, id_Produto, id_PessoaJuridica, quantidade, preco_Unitario)
VALUES (1, 1, 1, 100, 5.00)
(2, 2, 4, 100, 3.50);
```

Dados do Movimento de Entrada.

	id_MovimentoCompra	id_Usuario	id_Produto	id_PessoaJuridica	quantidade	preco_Unitario
1	1	1	1	1	100	5.00
2	2	2	4	2	100	3.50

Movimento de Saída.

```
SQLQuery31.sql - DE....LojaDB (loja (54))* X
INSERT INTO Movimento_Venda (id_Usuario, id_Produto, id_PessoaFisica, quantidade)
VALUES (2, 2, 1, 50)
(1, 3, 2, 150);
```

Dados do Movimento de Saída.

	id_MovimentoVenda	id_Usuario	id_Produto	id_PessoaFisica	quantidade
1	1	2	2	1	50
2	2	1	3	2	150

Selecionando dados completos de Pessoas Físicas.

```
SQLQuery1.sql - DES....LojaDB (loja (64))* X
SELECT *
FROM Pessoa_Fisica;
```



Dados completos da Tabela Pessoas Físicas.

	id_PessoaFisica	nome	logradouro	cidade	estado	telefone	email	cpf
1	1	Joao	Rua 12, casa 3, Quitanda	Riacho do Sul	PA	1111-1111	joao@riacho.com	11111111111
2	2	Claudia	Rua 13, casa 146, Rio Verde	Curitiba	PA	1111-1111	claudia@rioverde.com	22255577795

Selecionando dados completos Pessoas Jurídicas.

```
SQLQuery2.sql - DES....LojaDB (loja (52))*  
SELECT *  
FROM Pessoa_Juridica;
```

Dados completos da Tabela Pessoas Jurídicas.

	id_PessoaJuridica	nome	logradouro	cidade	estado	telefone	email	cnpj
1	1	JJC	Rua 11, Centro	Riacho Norte	PA	1212-1212	jjc@gmail.com	22222222222222
2	2	LBC	Rua A, Ipanema	Rio de Janeiro	RJ	3555-3553	lbc@gmail.com	12345677298595

Movimentações de Entrada.

```
SQLQuery3.sql - DES....LojaDB (loja (63))* SQLQuery2.sql - DES....LojaDB (loja (52))*  
SELECT MC.id_MovimentoCompra, P.nome AS Produto, PJ.nome AS Fornecedor, MC.quantidade, MC.preco_Unitario, (MC.quantidade * MC.preco_Unitario) AS Valor_Total  
FROM Movimento_Compra MC  
INNER JOIN Produto P ON MC.id_Produto = P.id_Produto  
INNER JOIN Pessoa_Juridica PJ ON MC.id_PessoaJuridica = PJ.id_PessoaJuridica;
```

	id_MovimentoCompra	Produto	Fornecedor	quantidade	preco_Unitario	Valor_Total
1	1	banana	JJC	100	5.00	500.00
2	2	morango	LBC	100	3.50	350.00



## Movimentações de Saída.

SQLQuery3.sql - DES....LojaDB (loja (63))\* SQLQuery2.sql - DES....LojaDB (loja (52))\*

```
SELECT MV.id_MovimentoVenda, P.nome AS Produto, PF.nome AS Comprador, MV.quantidade, P.precoVenda AS Preco_Unitario, (MV.quantidade * P.precoVenda) AS Valor_Total
FROM Movimento_Venda MV
INNER JOIN Produto P ON MV.id_Produto = P.id_Produto
INNER JOIN Pessoa_Fisica PF ON MV.id_PessoaFisica = PF.id_PessoaFisica;
```

Resultados Mensagens

	id_MovimentoVenda	Produto	Comprador	quantidade	Preco_Unitario	Valor_Total
1	1	laranja	Joao	50	2.00	100.00
2	2	manga	Claudia	150	4.00	600.00

## Valor total das entradas agrupadas por Produto.

SQLQuery4.sql - DES....LojaDB (loja (52))\*

```
SELECT P.nome AS Produto, SUM(MC.quantidade * MC.preco_Unitario) AS Valor_Total_Entradas
FROM Movimento_Compra MC
INNER JOIN Produto P ON MC.id_Produto = P.id_Produto
GROUP BY P.nome;
```

Resultados Mensagens

	Produto	Valor_Total_Entradas
1	banana	500.00
2	morango	350.00

## Valor total das saídas agrupadas por Produto.

SQLQuery4.sql - DES....LojaDB (loja (52))\*

```
SELECT P.nome AS Produto, SUM(MV.quantidade * P.precoVenda) AS Valor_Total_Saidas
FROM Movimento_Venda MV
INNER JOIN Produto P ON MV.id_Produto = P.id_Produto
GROUP BY P.nome;
```

Resultados Mensagens

	Produto	Valor_Total_Saidas
1	laranja	100.00
2	manga	600.00



Operadores que não Efetuaram Movimentações de Entrada.

SQLQuery5.sql - DES....LojaDB (loja (61))*	
<pre>SELECT U.login AS Operador FROM Usuario U LEFT JOIN Movimento_Compra MC ON U.id_Usuario = MC.id_Usuario WHERE MC.id_MovimentoCompra IS NULL;</pre>	
Resultados	Mensagens
Operador	
1 op3	
2 op4	

Valor total de entrada agrupado por Operador.

SQLQuery5.sql - DES....LojaDB (loja (61))*	
<pre>SELECT U.login AS Operador, SUM(MC.quantidade * MC.preco_Unitario) AS Valor_Total_Entrada FROM Movimento_Compra MC INNER JOIN Usuario U ON MC.id_Usuario = U.id_Usuario GROUP BY U.login;</pre>	
Resultados	Mensagens
Operador	Valor_Total_Entrada
1 op1	500.00
2 op2	350.00

Valor total de saída agrupado por Operador.

SQLQuery5.sql - DES....LojaDB (loja (61))*	
<pre>SELECT U.login AS Operador, SUM(MV.quantidade * P.precoVenda) AS Valor_Total_Saida FROM Movimento_Venda MV INNER JOIN Usuario U ON MV.id_Usuario = U.id_Usuario INNER JOIN Produto P ON MV.id_Produto = P.id_Produto GROUP BY U.login;</pre>	
Resultados	Mensagens
Operador	Valor_Total_Saida
1 op1	600.00
2 op2	100.00



Valor médio de venda por produto, com média ponderada.

SQLQuery5.sql - DES....LojaDB (loja (61))\*

```
SELECT P.nome AS Produto, SUM(MV.quantidade * P.precoVenda) / SUM(MV.quantidade) AS Valor_Medio_Venda
FROM Movimento_Venda MV
INNER JOIN Produto P ON MV.id_Produto = P.id_Produto
GROUP BY P.nome;
```

Resultados

	Produto	Valor_Medio_Venda
1	laranja	2.000000
2	manga	4.000000

### Diferença entre Sequence e Identity?

A diferença está relacionada a forma que se é implementada para gerar valores automáticas para colunas em tabelas.

Sequence gera uma sequência de números únicos em ordem a definir, utilizando uma ou mais colunas, podendo ser compartilhada por várias tabelas.

Identity gera automaticamente valores numéricos únicos à medida que novas linhas são inseridas na tabela, porém cada tabela pode ter somente uma coluna identity.

### Qual a importância das chaves estrangeiras para a consistência do banco?

As chaves estrangeiras são importantes para garantir a consistência dos dados, prevenindo inconsistências, mantendo a integridade e facilitando na modelagem do banco de dados, assim assegurando a qualidade e confiabilidade dos dados armazenados.



**Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no cálculo relacional?**

Operadores são Where, Select, Union, Intersect, Except, Join.

**Como é feito o agrupamento em consultas, e qual requisito é obrigatório?**

O agrupamento é feito usando GROUP BY, agrupando linhas de dados de valores comuns em um ou mais colunas, o requisito obrigatório é que todas as colunas selecionadas devem estar listadas no GROUP BY, ou em SUM, COUNT e AVG.