



# Estácio

POLO PRINC IZABEL -  
CACHOEIRINHA - RS

DESENVOLVIMENTO FULL STACK  
RPG0015 -Vamos Manter as Informações?

Turma 9001

Primeiro semestre de 2024

Integrantes: Fagner Luiz Gimenez Mendes.

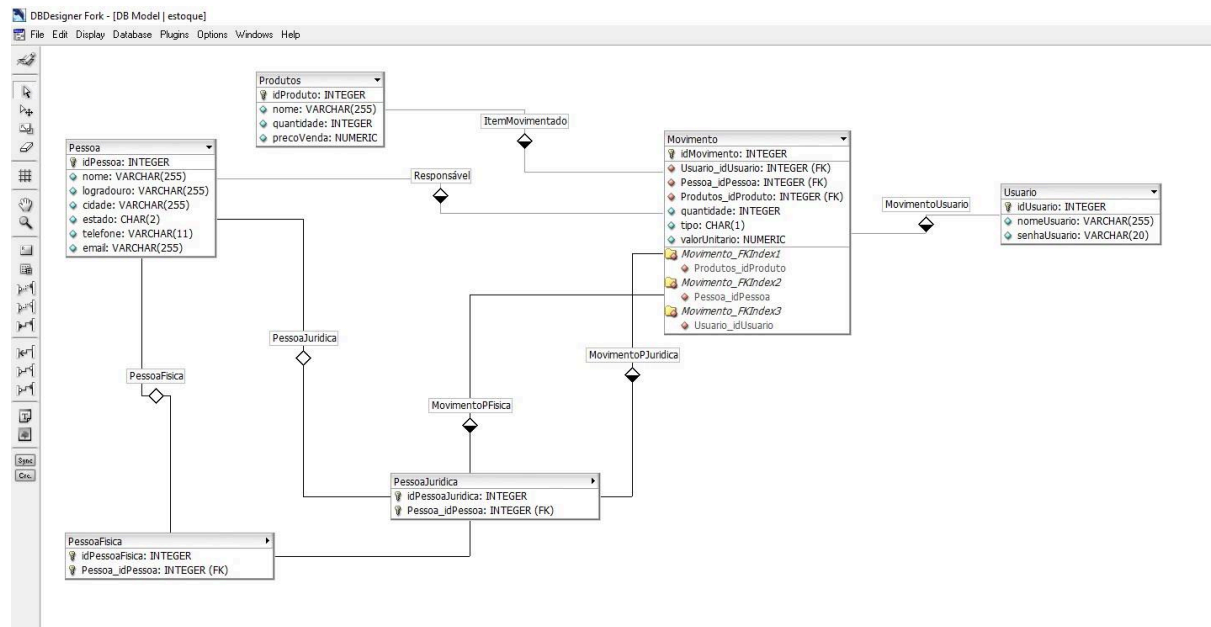
Repositório:

<https://github.com/GafnerMendes/Miss-o-Pr-tica-2-M2>.

git

## 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).



```
CREATE TABLE Movimento (
  idMovimento INTEGER NOT NULL IDENTITY ,
  Usuario_idUsuario INTEGER NOT NULL ,
  Pessoa_idPessoa INTEGER NOT NULL ,
  Produtos_idProduto INTEGER NOT NULL ,
  quantidade INTEGER ,
  tipo CHAR(1) NOT NULL ,
  valorUnitario NUMERIC ,
  PRIMARY KEY(idMovimento));
GO
```

```
CREATE TABLE Pessoa (
  idPessoa INTEGER NOT NULL IDENTITY ,
  nome VARCHAR(255) ,
  logradouro VARCHAR(255) ,
  cidade VARCHAR(255) ,
  estado CHAR(2) ,
  telefone VARCHAR(11) ,
  email VARCHAR(255) ,
  PRIMARY KEY(idPessoa));
GO
```

```
CREATE TABLE PessoaFisica (
  idPessoaFisica INTEGER NOT NULL IDENTITY ,
```

```
Pessoa_idPessoa INTEGER NOT NULL ,
Movimento_idMovimento INTEGER NOT NULL ,
CPF INTEGER ,
PRIMARY KEY(idPessoaFisica, Pessoa_idPessoa, Movimento_idMovimento));
GO
```

```
CREATE TABLE PessoaJuridica (
idPessoaJuridica INTEGER NOT NULL IDENTITY ,
Pessoa_idPessoa INTEGER NOT NULL ,
Movimento_idMovimento INTEGER NOT NULL ,
CNPJ INTEGER ,
PRIMARY KEY(idPessoaJuridica, Pessoa_idPessoa, Movimento_idMovimento));
GO
```

```
CREATE TABLE Produtos (
idProduto INTEGER NOT NULL IDENTITY ,
nome VARCHAR(255) ,
quantidade INTEGER ,
precoVenda NUMERIC ,
```

```
PRIMARY KEY(idProduto));
GO
```

```
CREATE TABLE Usuario (
idUserario INTEGER NOT NULL IDENTITY ,
nomeUsuario VARCHAR(255) ,
senhaUsuario VARCHAR(20) ,
PRIMARY KEY(idUsuario));
```

Sequence.sql:

```
CREATE SEQUENCE Seq_Pessoa
START WITH 1
INCREMENT BY 1;
```

```
Script.sql
USE [master]
GO
```

```
CREATE DATABASE [Loja]
CONTAINMENT = NONE
ON PRIMARY
```

```
( NAME = N'Loja', FILENAME = N'D:\Programas\Microsoft SQL
Server\MSSQL16.SQLEXPRESS\MSSQL\DATA\Loja.mdf' , SIZE = 8192KB , MAXSIZE
= UNLIMITED, FILEGROWTH = 65536KB )
LOG ON
```

```
( NAME = N'Loja_log', FILENAME = N'D:\Programas\Microsoft SQL
Server\MSSQL16.SQLEXPRESS\MSSQL\DATA\Loja_log.ldf' , SIZE = 8192KB ,
MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
WITH CATALOG_COLLATION = DATABASE_DEFAULT, LEDGER = OFF
GO
ALTER DATABASE [Loja] SET COMPATIBILITY_LEVEL = 160
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
```

```
begin
EXEC [Loja].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [Loja] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [Loja] SET ANSI_NULLS OFF
GO
```

```
ALTER DATABASE [Loja] SET ANSI_PADDING OFF
GO
ALTER DATABASE [Loja] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [Loja] SET ARITHABORT OFF
GO
ALTER DATABASE [Loja] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [Loja] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [Loja] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [Loja] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [Loja] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [Loja] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [Loja] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [Loja] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [Loja] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [Loja] SET DISABLE_BROKER
GO
ALTER DATABASE [Loja] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [Loja] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
```

```
ALTER DATABASE [Loja] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [Loja] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [Loja] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [Loja] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [Loja] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [Loja] SET RECOVERY SIMPLE
GO
ALTER DATABASE [Loja] SET MULTI_USER
GO
ALTER DATABASE [Loja] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [Loja] SET DB_CHAINING OFF
GO
ALTER DATABASE [Loja] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [Loja] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [Loja] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [Loja] SET ACCELERATED_DATABASE_RECOVERY = OFF
GO
ALTER DATABASE [Loja] SET QUERY_STORE = ON
GO
```

```
ALTER DATABASE [Loja] SET QUERY_STORE (OPERATION_MODE = READ_WRITE,
CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 30),
DATA_FLUSH_INTERVAL_SECONDS = 900, INTERVAL_LENGTH_MINUTES = 60,
MAX_STORAGE_SIZE_MB = 1000, QUERY_CAPTURE_MODE = AUTO,
SIZE_BASED_CLEANUP_MODE = AUTO, MAX_PLANS_PER_QUERY = 200,
WAIT_STATS_CAPTURE_MODE = ON)
GO
USE [Loja]
GO
USE [Loja]
GO
```

```
CREATE SEQUENCE [dbo].[identificadores]
AS [int]
START WITH 1
INCREMENT BY 1
MINVALUE -2147483648
MAXVALUE 2147483647
```

```
CACHE  
GO  
USE [Loja]  
GO
```

```
CREATE SEQUENCE [dbo].[MinhaSequencia]  
AS [bigint]  
START WITH 1  
INCREMENT BY 1  
MINVALUE -9223372036854775808  
MAXVALUE 9223372036854775807  
CACHE  
GO
```

```
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON
```

```
O  
CREATE TABLE [dbo].[movimento](  
[idMovimento] [int] NOT NULL,  
[quantidade] [int] NOT NULL,  
[tipo] [char](1) NOT NULL,  
[valorUnitario] [numeric](18, 0) NOT NULL  
) ON [PRIMARY]  
GO
```

```
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE TABLE [dbo].[pessoa](  
[idPessoa] [int] NOT NULL,  
[nome] [varchar](255) NOT NULL,  
[logradouro] [varchar](255) NOT NULL,  
[cidade] [varchar](255) NOT NULL,  
[estado] [char](2) NOT NULL,  
[telefone] [varchar](11) NULL,  
[email] [varchar](255) NULL  
) ON [PRIMARY]  
GO
```

```
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO
```

```
CREATE TABLE [dbo].[pessoa_fisica](
[idPessoasFisicas] [int] NOT NULL,
[CPF] [int] NULL
) ON [PRIMARY]
GO
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[pessoa_juridica](
[idPessoasJuridicas] [int] NOT NULL,
[CNPJ] [int] NULL
) ON [PRIMARY]
GO
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[produtos](
[idProduto] [int] NOT NULL,
[nome] [varchar](255) NOT NULL,
[quantidade] [int] NOT NULL,
[precoVenda] [numeric](18, 0) NOT NULL
) ON [PRIMARY]
GO
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[usuario](
[idUsuario] [int] NOT NULL,
[nomeUsuario] [varchar](255) NOT NULL,
[senhaUsuario] [varchar](20) NOT NULL
) ON [PRIMARY]
GO
```

```
USE [master]
```

```
GO
```

```
ALTER DATABASE [Loja] SET READ_WRITE
GO
```

## Análise e Conclusão:

1. Como são implementadas as diferentes cardinalidades, basicamente 1x1, 1xN ou NxN, em um banco de dados relacional?

R: As diferentes cardinalidades são implementadas usando chaves estrangeiras. Para um relacionamento 1X1, você pode usar uma chave estrangeira em ambas as tabelas, apontando para a chave primária da outra. Para um relacionamento 1XN, você usa uma chave estrangeira em uma tabela que se relaciona com a chave primária da outra. Para NxN, você cria uma tabela de associação que contém chaves estrangeiras para as duas entidades envolvidas.

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

R: Para representar herança em bancos de dados relacionais, é geralmente usado um relacionamento de tabela única, onde todas as classes filhas compartilham uma tabela comum, e uma coluna identificadora é usada para distinguir o tipo de objeto.

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

R: O SQL Server Management Studio oferece uma interface gráfica amigável para criar, modificar e consultar bancos de dados. Ele fornece ferramentas de design de tabelas, consultas SQL interativas, visualização de planos de execução, tarefas de administração simplificadas e muito mais, facilitando a administração e o desenvolvimento de bancos de dados SQL Server.

## 2º Procedimento | Alimentando a base

```
DECLARE @ProximoValor INT;
SET @ProximoValor = NEXT VALUE FOR Seq_Pessoa;
DECLARE @Pessoaid INT;
SELECT @Pessoaid = MAX(idPessoa)
FROM Pessoa;
INSERT INTO Pessoa (idPessoa, nome, logradouro, cidade, estado, telefone,
email)
VALUES (@ProximoValor, 'Marta', 'Rua das Flores, Centro', 'Montanha do
Norte', 'PA', '5555-5555', 'marta@montanha.com');
--INSERT INTO PessoaJuridica(Pessoa_idPessoa, CNPJ)
--VALUES ('28', '12345698745632');
SELECT * FROM Pessoa
SELECT * FROM PessoaJuridica
```

```
DECLARE @ProximoValor INT;
SET @ProximoValor = NEXT VALUE FOR Seq_Pessoa;
INSERT INTO Pessoa (idPessoa, nome, logradouro, cidade, estado, telefone,
email)
VALUES (@ProximoValor, 'Joao', 'Rua 12, casa 3, Quitanda', 'Riacho do
Sul', 'PA', '1111-1111', 'joao@riacho.com');
```

--Dados completos de pessoas fisicas



```

SELECT * FROM Pessoa INNER JOIN PessoaFisica
ON Pessoa.idPessoa = PessoaFisica.Pessoa_idPessoa
--Dados completos de pessoas juridicas
SELECT * FROM Pessoa INNER JOIN PessoaJuridica
ON Pessoa.idPessoa = PessoaJuridica.Pessoa_idPessoa
--Movimentaões de entrada, com produto, fornecedor, quantidade, preço
unitário e valor total
SELECT *,
(CAST(REPLACE(valorUnitario, ',', '.') AS DECIMAL(10, 2)) *
quantidade) AS Total
FROM Movimento
WHERE Tipo = 'S';
--Movimentaões de saída, com produto, comprador, quantidade, preço
unitário e valor total

```

```

SELECT *,
(CAST(REPLACE(valorUnitario, ',', '.') AS DECIMAL(10, 2)) *
quantidade) AS Total
FROM Movimento
WHERE Tipo = 'E';
--Valor total das entradas agrupadas por produto
SELECT Tipo, Produtos_idProduto,
SUM(CAST(REPLACE(valorUnitario, ',', '.') AS DECIMAL(10, 2)) *
quantidade) AS TotalEntradas
FROM Movimento
WHERE Tipo = 'E'
GROUP BY Tipo, Produtos_idProduto;
--Valor total das saídas agrupadas por produto
SELECT Tipo, Produtos_idProduto,
SUM(CAST(REPLACE(valorUnitario, ',', '.') AS DECIMAL(10, 2)) *
quantidade) AS TotalSaidas
FROM Movimento
WHERE Tipo = 'S'
GROUP BY Tipo, Produtos_idProduto;
--Operadores que não efetuaram movimentações de entrada (compra)
SELECT DISTINCT m.Usuario_idUsuario
FROM Movimento m
LEFT JOIN (
SELECT DISTINCT Usuario_idUsuario
FROM Movimento
WHERE Tipo = 'E'
) e ON m.Usuario_idUsuario = e.Usuario_idUsuario
WHERE e.Usuario_idUsuario IS NULL;
--Valor total de entrada, agrupado por operador
SELECT Tipo, Usuario_idUsuario,
SUM(CAST(REPLACE(valorUnitario, ',', '.') AS DECIMAL(10, 2)) *
quantidade) AS TotalEntradas
FROM Movimento

```

```

WHERE Tipo = 'E'
GROUP BY Tipo, Usuario_idUsuario;
--Valor total de saída, agrupado por operador
SELECT Tipo, Usuario_idUsuario,
SUM(CAST(REPLACE(valorUnitario, ',', '.') AS DECIMAL(10, 2)) *
quantidade) AS TotalEntradas
FROM Movimento
WHERE Tipo = 'S'
GROUP BY Tipo, Usuario_idUsuario;

```

```

--Valor médio de venda por produto, utilizando média ponderada
SELECT Produtos_idProduto,
SUM(quantidade * CAST(REPLACE(valorUnitario, ',', '.') AS
DECIMAL(10, 2))) / SUM(quantidade) AS ValorMedioVenda
FROM Movimento
GROUP BY Produtos_idProduto;

```

SQLQuery8.sql - DES...ESS.Loja (loja (74))\*
SQLQuery3.sql - DES...ESS.Loja (loja (71))
SQLQuery2.sql - DES...SS.Loja (loja (109))\*

```

INSERT INTO Movimento (Usuario_idUsuario, Pessoa_idPessoa, Produtos_idProduto, quantidade, tipo, valorUnitario)
VALUES ('1', '16', '2', '20', 'E', '4,00')

SELECT * FROM Usuario

SELECT * FROM Movimento

```

100 %

Results

Messages

	idUsuario	login	senha
1	1	op1	op1
2	2	op2	op2

	idMovimento	Usuario_idUsuario	Pessoa_idPessoa	Produtos_idProduto	quantidade	tipo	valorUnitario
1	2	1	2	1	20	S	4,00
2	3	1	16	3	15	S	2,00
3	4	2	42	3	10	S	3,00
4	5	1	16	3	15	E	5,00
5	6	1	16	2	20	E	4,00

Query executed successfully.

Results		Messages		
	idProduto	nome	quantidade	precoVenda
1	1	Banana	100	5
2	2	Laranja	500	2
3	3	Manga	800	4

SQLQuery3.sql - DES...ESS.Loja (loja (71))

SQLQuery2.sql - DES...SS.Loja (loja (109))

SELECT \* FROM Pessoa

SELECT \* FROM PessoaFisica

SELECT \* FROM PessoaJuridica

SELECT \* FROM Pessoa INNER JOIN PessoaFisica ON Pessoa.idPessoa = PessoaFisica.Pessoa\_idPessoa

SELECT \* FROM Pessoa INNER JOIN PessoaJuridica ON Pessoa.idPessoa = PessoaJuridica.Pessoa\_idPessoa

100 %

Results Messages

	idPessoa	nome	logradouro	cidade	estado	telefone	email
1	2	Joao	Rua 12, casa 3, Quitanda	Riacho do Sul	PA	1111-1111	joao@riacho.com
2	15	Maria	Rua 0, casa 0, Mercaria	Riacho do Norte	SC	2222-2222	maria@riachon.com
3	16	Joana	Rua dos bobos, casa 20, Leopoldina	Montanha do leste	PO	3333-3333	joana@montanha.com
4	24	JJC	Rua 11, Centro	Riacho do Norte	PA	1212-1212	jjc@riacho.com
5	28	CCJ	Rua 22, Centro	Riacho do Leste	PA	3434-3434	ccj@riacho.com
6	42	Marta	Rua das Flores, Centro	Montanha do Norte	PA	5555-5555	marta@montanha.com

	idPessoaFisica	Pessoa_idPessoa	Movimento_idMovimento	CPF
1	2	2	NULL	11111111111
2	17	15	NULL	22222222222
3	22	16	NULL	33333333333
4	25	42	NULL	35264879532

	idPessoaJuridica	Pessoa_idPessoa	Movimento_idMovimento	CNPJ
1	24	24	NULL	98745632178965
2	28	28	NULL	12345698745632

	idPessoa	nome	logradouro	cidade	estado	telefone	email	idPessoaFisica	Pessoa_idPessoa	Movimento_idMovimento	CPF
1	2	Joao	Rua 12, casa 3, Quitanda	Riacho do Sul	PA	1111-1111	joao@riacho.com	2	2	NULL	11111111111
2	15	Maria	Rua 0, casa 0, Mercaria	Riacho do ...	SC	2222-2222	maria@riachon...	17	15	NULL	22222222222
3	16	Joa...	Rua dos bobos, casa 2...	Montanha d...	PO	3333-3333	joana@montan...	22	16	NULL	33333333333
4	42	Marta	Rua das Flores, Centro	Montanha d...	PA	5555-5555	marta@montan...	25	42	NULL	35264879532

	idPessoa	nome	logradouro	cidade	estado	telefone	email	idPessoaJuridica	Pessoa_idPessoa	Movimento_idMovimento	CNPJ
1	24	JJC	Rua 11, Centro	Riacho do Norte	PA	1212-1212	jjc@riacho.com	24	24	NULL	98745632178965
2	28	CCJ	Rua 22, Centro	Riacho do Leste	PA	3434-3434	ccj@riacho.c...	28	28	NULL	12345698745632

Query executed successfully.

SQLQuery2.sql - DES...SS.Loja (loja (109))

--Dados completos de pessoas fisicas  
SELECT \* FROM Pessoa INNER JOIN PessoaFisica  
ON Pessoa.idPessoa = PessoaFisica.Pessoa\_idPessoa  
  
--Dados completos de pessoas juridicas  
SELECT \* FROM Pessoa INNER JOIN PessoaJuridica

100 %

Results Messages

	idPessoa	nome	logradouro	cidade	estado	telefone	email	idPessoaFisica	Pessoa_idPessoa	CPF
1	2	Joao	Rua 12, casa 3, Quitanda	Riacho do Sul	PA	1111-1111	joao@riacho.com	2	2	11111111111
2	15	Maria	Rua 0, casa 0, Mercaria	Riacho do Norte	SC	2222-2222	maria@riachon.com	17	15	22222222222
3	16	Joana	Rua dos bobos, casa 20, Leopoldina	Montanha do leste	PO	3333-3333	joana@montanha.com	22	16	33333333333
4	42	Marta	Rua das Flores, Centro	Montanha do Norte	PA	5555-5555	marta@montanha.com	25	42	35264879532

	idPessoa	nome	logradouro	cidade	estado	telefone	email	idPessoaJuridica	Pessoa_idPessoa	CNPJ
1	24	JJC	Rua 11, Centro	Riacho do Norte	PA	1212-1212	jic@riacho.com	24	24	98745632178965
2	28	CCJ	Rua 22, Centro	Riacho do Leste	PA	3434-3434	ccj@riacho.com	28	28	12345698745632

	idMovimento	Usuario_idUsuario	Pessoa_idPessoa	Produtos_idProduto	quantidade	tipo	valorUnitario	Total
1	2	1	2	1	20	S	4,00	80.00
2	3	1	16	3	15	S	2,00	30.00
3	4	2	42	3	10	S	3,00	30.00

	idMovimento	Usuario_idUsuario	Pessoa_idPessoa	Produtos_idProduto	quantidade	tipo	valorUnitario	Total
1	5	1	16	3	15	E	5,00	75.00
2	6	1	16	2	20	E	4,00	80.00

	Tipo	Produtos_idProduto	TotalEntradas
1	E	2	80.00
2	E	3	75.00

	Tipo	Produtos_idProduto	TotalSaídas
1	S	1	80.00
2	S	3	60.00

	Usuario_idUsuario
1	2

	Tipo	Usuario_idUsuario	TotalEntradas
1	E	1	155.00

	Tipo	Usuario_idUsuario	TotalEntradas
1	S	1	110.00
2	S	2	30.00

	Produtos_idProduto	ValorMedioVenda
1	1	4.000000
2	2	4.000000
3	3	3.375000

Query executed successfully.

## Análise e Conclusão:

1.Quais as diferenças no uso de sequence e identity?

R: A diferença entre sequence e identity está relacionada a sistemas de gerenciamento de bancos de dados específicos. Identity é mais comum no SQL Server, enquanto sequence é usado no Oracle. Ambos são usados para gerar valores únicos automaticamente, mas a sintaxe e a funcionalidade podem variar entre eles.

2.Qual a importância das chaves estrangeiras para a consistência do banco?

R: As chaves estrangeiras são essenciais para a consistência do banco de dados, pois garantem que as relações entre tabelas sejam mantidas. Elas impõem integridade referencial, evitando que sejam inseridos dados inconsistentes que não correspondam a registros na tabela relacionada.

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

R: Na álgebra relacional, os operadores incluem projeção, seleção, união, interseção, diferença, junção, entre outros. No cálculo relacional, as operações são representadas de forma mais declarativa, onde você especifica o que deseja obter sem se preocupar com a implementação específica.

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

R: O agrupamento em consultas SQL é feito usando a cláusula `GROUP BY`, e o requisito obrigatório é que você deve incluir colunas de agregação (como `SUM`, `COUNT`, `AVG`, etc.) ou colunas não agregadas nas quais está agrupando. Isso ajuda a resumir os dados de acordo com critérios específicos e obter informações resumidas.