



UNIVERSIDADE ESTÁCIO DE SÁ
DESENVOLVIMENTO FULLSTACK

Mundo 03 - Nível 02

RPG0015 - Vamos manter as informações!

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

Ruan Hernandez Finamor Correia

Matrícula 202208175252

Gravataí – RS

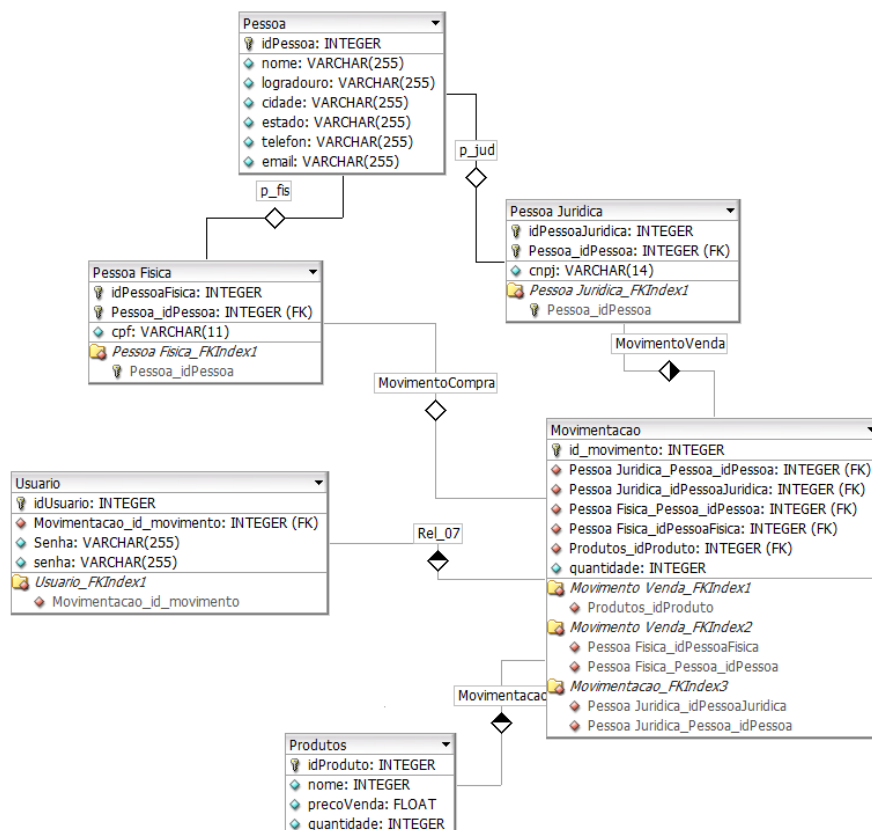
2023

Objetivo da Prática

1. Identificar os requisitos de um sistema e transformá-los no modelo adequado.
2. Utilizar ferramentas de modelagem para bases de dados relacionais.
3. Explorar a sintaxe SQL na criação das estruturas do banco (DDL).
4. Explorar a sintaxe SQL na consulta e manipulação de dados (DML)
5. 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

Códigos solicitados no roteiro desta aula



```

/*Criando a tabela usuário*/
CREATE TABLE Usuario (
  id_usuario INT IDENTITY (1,1) PRIMARY KEY,
  Login VARCHAR(255) NOT NULL,
  Senha VARCHAR(255) NOT NULL,
);
/*Criando a tabela Pessoa*/
CREATE TABLE Pessoa (
  idPessoa INT PRIMARY KEY,
  nome VARCHAR(255) NOT NULL,
  logradouro VARCHAR(255) NOT NULL,
  cidade VARCHAR(255) NOT NULL,
  estado VARCHAR(255) NOT NULL,
  telefone VARCHAR(255),
  email VARCHAR(255) NOT NULL,
);

/*Criando a tabela PessoaFisica*/
CREATE TABLE PessoaFisica (
  idPessoaFisica INT PRIMARY KEY,
  cpf VARCHAR(11) NOT NULL,
  idPessoa INT FOREIGN KEY REFERENCES Pessoa(idPessoa)
);

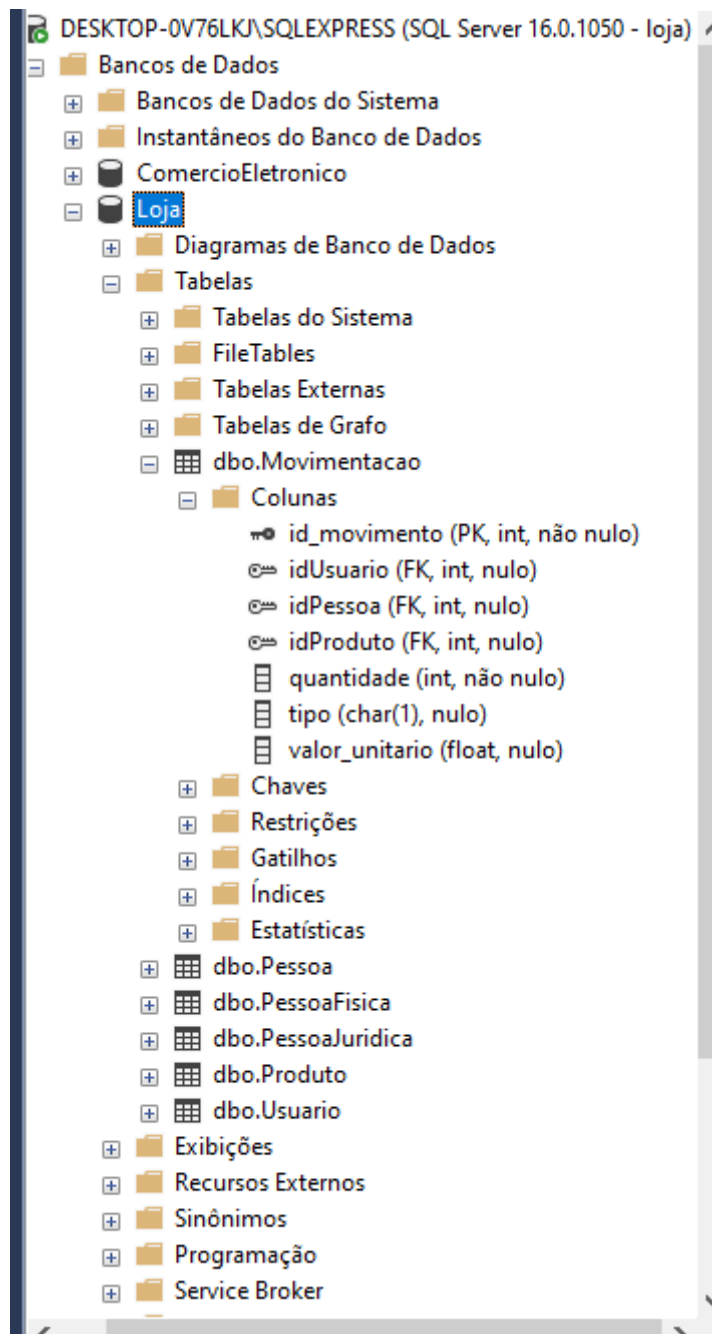
/*Criando a tabela PessoaJuridica*/
CREATE TABLE PessoaJuridica (
  idPessoaJuridica INT PRIMARY KEY,
  cnpj VARCHAR(14) NOT NULL,
  idPessoa INT FOREIGN KEY REFERENCES Pessoa(idPessoa)
);

/*Criando a tabela Produto*/
CREATE TABLE Produto (
  idProduto INT PRIMARY KEY,
  nome VARCHAR(255) NOT NULL,
  quantidade INT NOT NULL,
  precoVenda NUMERIC(20, 2) NOT NULL,
);

/*Criando a tabela Movimentação*/
CREATE TABLE Movimentacao(
  id_movimento INT PRIMARY KEY,
  valor_unitario NUMERIC(20, 2) NOT NULL,
  quantidade INT NOT NULL,
  tipo CHAR NOT NULL,
  idPessoa INT FOREIGN KEY REFERENCES Pessoa(idPessoa),
  idProduto INT FOREIGN KEY REFERENCES Produto(idProduto),
  idUsuario INT FOREIGN KEY REFERENCES Usuario(idUsuario),
);

```

Resultado da execução dos códigos



Análise e conclusão:

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

R – Na cardinalidade 1 X 1 cada linha de uma tabela será referenciada em uma linha na outra tabela e vice-versa. Na cardinalidade 1xN cada linha em uma tabela pode se referenciar a várias linhas em outra tabela, que por sua vez, só pode ter uma linha referenciada naquela. Já a N X N significa que cada linha pode ser referenciadas a varias em outra tabela, e vice-versa.

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

R – Herança por tabelas. Deve-se criar uma tabela de base com atributos comuns a todas as subclasses e tabelas, com chave estrangeira para referenciá-las.

P - 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 possui uma interface amigável, editor avançado, além de um assistente de importação, que auxiliam muito no trabalho do desenvolvedor

2º Procedimento

Códigos solicitados no roteiro desta aula

```
/*Inserindo os usuários */
INSERT INTO Usuario (Login, Senha) VALUES
('op1', 'op1'),
('op2', 'op2');

/*Inserindo os Produtos */
INSERT INTO Produto (nome, quantidade, precoVenda) VALUES
('Banana', '100', '5.01'),
('Laranja', '500', '2.02'),
('Manga', '800', '4.03'),
('Pera tomy', '40', '300');

/*Inserindo as Pessoas */
INSERT INTO Pessoa ( nome, logradouro, cidade, estado, telefone, email) VALUES
( 'João', 'Rua 17', 'Albuquerque', 'PA', '9999999998', 'email@joao.com'),
( 'Jose', 'Rua 7', 'Albuquerque', 'PA', '9999999999', 'email@jose.com'),
( 'Ruan', 'rua das couves', 'londres', 'rj', '1212112', 'hhiahsias');
```

```

/*Inserindo as Pessoas Fisicas */
INSERT INTO PessoaFisica (idPessoa, nome, logradouro, cidade, estado, telefone, email,
cpf) VALUES
( 10, 'Jose', 'Rua 7', 'Albuquerque', 'PA', '999999999', 'email@jose.com',
'12345678900'),
( 5, 'Jose', 'Rua 7', 'Albuquerque', 'PA', '999999999', 'email@jose.com',
'1212121212');

/*Inserindo as Pessoas Juridicas */
INSERT INTO PessoaJuridica (idPessoa, nome, logradouro, cidade, estado, telefone,
email, cnpj) VALUES
( 13, 'João', 'Rua 17', 'Albuquerque', 'PA', '9999999998', 'email@joao.com',
'1212121213');

/*Inserindo as Movimentações */
INSERT INTO Movimentacao (idUserio, idPessoa, idProduto, quantidade, tipo,
valor_unitario) VALUES
(1, 5, 1, '10', 'S', '4.00'),
(1, 10, 2, '100', 'E', '41.10'),
(2, 5, 3, '56', 'S', '200.01'),
(2, 13, 4, '12', 'E', '11.11');

/*Inserindo os usuários */
SELECT * FROM PessoaFisica;

/*Inserindo os usuários */
SELECT * FROM PessoaJuridica;

/*Inserindo os usuários */
SELECT
    Movimentacao.tipo,
    Pessoa.nome AS Fornecedor,
    Produto.nome,
    Movimentacao.quantidade,
    Produto.precoVenda AS Preco_Unitario,
    (Movimentacao.quantidade * Produto.precoVenda) AS ValorTotal
FROM
    Movimentacao
JOIN
    Pessoa ON Movimentacao.idPessoa = Pessoa.idPessoa
JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'E';

/*Inserindo os usuários */
SELECT
    Movimentacao.tipo,
    Pessoa.nome AS Comprador,
    Produto.nome,
    Movimentacao.quantidade,
    Produto.precoVenda AS Preco_Unitario,
    (Movimentacao.quantidade * Produto.precoVenda) AS ValorTotal
FROM
    Movimentacao
JOIN
    Pessoa ON Movimentacao.idPessoa = Pessoa.idPessoa
JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'S';

```

```

/*Inserindo os usuários */
SELECT
    Produto.nome AS Produto,
    SUM(Movimentacao.quantidade * Produto.precoVenda) AS ValorTotalEntradas
FROM
    Movimentacao
JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'E'
GROUP BY
    Produto.nome;

/*Inserindo os usuários */
SELECT
    Produto.nome AS Produto,
    SUM(Movimentacao.quantidade * Produto.precoVenda) AS ValorTotalSaidas
FROM
    Movimentacao
JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'S'
GROUP BY
    Produto.nome;

/*Inserindo os usuários */
SELECT
    Pessoa.nome AS OperadorSemEntrada
FROM
    Pessoa
WHERE
    idPessoa NOT IN (
        SELECT DISTINCT Movimentacao.idPessoa
        FROM Movimentacao
        WHERE Movimentacao.tipo = 'E'
    );

/*Inserindo os usuários */
SELECT
    Pessoa.nome AS Operador,
    SUM(Movimentacao.quantidade * Produto.precoVenda) AS ValorTotalEntrada
FROM
    Pessoa
LEFT JOIN
    Movimentacao ON Pessoa.idPessoa = Movimentacao.idPessoa
LEFT JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'E'
GROUP BY
    Pessoa.nome;

/*Inserindo os usuários */
SELECT
    Pessoa.nome AS Operador,
    SUM(Movimentacao.quantidade * Produto.precoVenda) AS ValorTotalSaida
FROM
    Pessoa
LEFT JOIN

```

```

Movimentacao ON Pessoa.idPessoa = Movimentacao.idPessoa
LEFT JOIN
Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
Movimentacao.tipo = 'S'
GROUP BY
Pessoa.nome;

/*Inserindo os usuários */
SELECT
    Produto.nome AS Produto,
    SUM(Movimentacao.quantidade * Movimentacao.valor_unitario) /
SUM(Movimentacao.quantidade) AS ValorMedioVenda
FROM
    Produto
LEFT JOIN
    Movimentacao ON Produto.idProduto = Movimentacao.idProduto
WHERE
    Movimentacao.tipo = 'S'
GROUP BY
    Produto.nome;

```

Resultado da execução dos códigos

4. Efetuar as seguintes consultas sobre os dados inseridos:

- Dados completos de pessoas físicas.

```

/*Dados completos de pessoas físicas. */
SELECT * FROM PessoaFisica;

```

90 %

	idPessoaFisica	idPessoa	nome	logradouro	cidade	estado	telefone	email	CPF
1	3	5	Ruan	rua das couves	londres	rj	1212112	hhiahsias	1212121212
2	9	10	Jose	Rua 7	Albuquerque	PA	9999999999	email@jose.com	12345678900

- Dados completos de pessoas jurídicas.

```

/*Dados completos de pessoas jurídicas. */
SELECT * FROM PessoaJuridica;

```

0 %

	idPessoaJuridica	idPessoa	nome	logradouro	cidade	estado	telefone	email	CNPJ
1	4	13	João	Rua 17	Albuquerque	PA	9999999998	email@joao.com	1212121213

- Movimentações de entrada, com produto, fornecedor, quantidade, preço unitário e valor total.


```

/*Movimentações de entrada, com produto, fornecedor, quantidade, preço unitário e valor total.*/
SELECT
    Movimentacao.tipo,
    Pessoa.nome AS Fornecedor,
    Produto.nome,
    Movimentacao.quantidade,
    Produto.precoVenda AS Preco_Unitario,
    (Movimentacao.quantidade * Produto.precoVenda) AS ValorTotal
FROM
    Movimentacao
JOIN
    Pessoa ON Movimentacao.idPessoa = Pessoa.idPessoa
JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'E';

```

```

/*Movimentações de saída, com produto, comprador, quantidade, preço unitário e valor total.*/
SELECT

```

	tipo	Fornecedor	nome	quantidade	Preco_Unitario	ValorTotal
1	E	Jose	Laranja	100	2,02	202
2	E	João	Pera tomy	12	300	3600

- Movimentações de saída, com produto, fornecedor, quantidade, preço unitário e valor total.

```

/*Movimentações de saída, com produto, comprador, quantidade, preço unitário e valor total.*/
SELECT
    Movimentacao.tipo,
    Pessoa.nome AS Comprador,
    Produto.nome,
    Movimentacao.quantidade,
    Produto.precoVenda AS Preco_Unitario,
    (Movimentacao.quantidade * Produto.precoVenda) AS ValorTotal
FROM
    Movimentacao
JOIN
    Pessoa ON Movimentacao.idPessoa = Pessoa.idPessoa
JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'S';

```

```

/*Valor total das entradas agrupadas por produto.*/

```

```

SELECT
    Produto.nome AS Produto,

```

	tipo	Comprador	nome	quantidade	Preco_Unitario	ValorTotal
1	S	Ruan	Banana	10	5,01	50,1
2	S	Ruan	Manga	56	4,03	225,68

- Valor total das entradas agrupadas por produto.

```

/*Valor total das entradas agrupadas por produto. */
SELECT
    Produto.nome AS Produto,
    SUM(Movimentacao.quantidade * Produto.precoVenda) AS ValorTotalEntradas
FROM
    Movimentacao
JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'E'
GROUP BY
    Produto.nome;

/*Valor total das saídas agrupadas por produto. */
SELECT

```

90 %

Resultados Mensagens

	Produto	ValorTotalEntradas
1	Laranja	202
2	Pera tomy	3600

- Valor total das saídas agrupadas por produto.

```

/*Valor total das saídas agrupadas por produto. */
SELECT
    Produto.nome AS Produto,
    SUM(Movimentacao.quantidade * Produto.precoVenda) AS ValorTotalSaidas
FROM
    Movimentacao
JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'S'
GROUP BY
    Produto.nome;

/*Operadores que não efetuaram movimentações de entrada (compra). */

```

0 %

Resultados Mensagens

	Produto	ValorTotalSaidas
1	Banana	50,1
2	Manga	225,68

- Operadores que não efetuaram movimentações de entrada (compra).

```

/*Operadores que não efetuaram movimentações de entrada (compra). */
SELECT
    Pessoa.nome AS OperadorSemEntrada
FROM
    Pessoa
WHERE
    idPessoa NOT IN (
        SELECT DISTINCT Movimentacao.idPessoa
        FROM Movimentacao
        WHERE Movimentacao.tipo = 'E'
    );

/*Valor total de entrada, agrupado por operador. */

```

90 %

Resultados Mensagens

	OperadorSemEntrada
1	Ruan

- Valor total de entrada, agrupado por operador.

```

/*Valor total de entrada, agrupado por operador. */
SELECT
    Pessoa.nome AS Operador,
    SUM(Movimentacao.quantidade * Produto.precoVenda) AS ValorTotalEntrada
FROM
    Pessoa
LEFT JOIN
    Movimentacao ON Pessoa.idPessoa = Movimentacao.idPessoa
LEFT JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'E'
GROUP BY
    Pessoa.nome;

/*Valor total de saída, agrupado por operador. */

```

90 %

Resultados Mensagens

	Operador	ValorTotalEntrada
1	João	3600
2	Jose	202

- Valor total de saída, agrupado por operador.

```

/*Valor total de saída, agrupado por operador. */
SELECT
    Pessoa.nome AS Operador,
    SUM(Movimentacao.quantidade * Produto.precoVenda) AS ValorTotalSaida
FROM
    Pessoa
LEFT JOIN
    Movimentacao ON Pessoa.idPessoa = Movimentacao.idPessoa
LEFT JOIN
    Produto ON Movimentacao.idProduto = Produto.idProduto
WHERE
    Movimentacao.tipo = 'S'
GROUP BY
    Pessoa.nome;

/*Valor médio de venda por produto, utilizando média ponderada. */

```

90 %

Resultados Mensagens

	Operador	ValorTotalSaida
1	Ruan	275,78

- Valor médio de venda por produto, utilizando média ponderada.

```

/*Valor médio de venda por produto, utilizando média ponderada. */
SELECT
    Produto.nome AS Produto,
    SUM(Movimentacao.quantidade * Movimentacao.valor_unitario) / SUM(Movimentacao.quantidade) AS ValorMedioVenda
FROM
    Produto
LEFT JOIN
    Movimentacao ON Produto.idProduto = Movimentacao.idProduto
WHERE
    Movimentacao.tipo = 'S'
GROUP BY
    Produto.nome;

```

90 %

Resultados Mensagens

	Produto	ValorMedioVenda
1	Banana	4
2	Manga	200,01

Análise e conclusão:

P – Quais as diferenças no uso de sequence e identity?

R – O Sequence é mais indicado quando é necessário mais controle sobre a sequência de números ou portabilidade entre diferentes sistemas, já o Identity é preferível em uma abordagem mais simples e direta.

P – Qual a importância das chaves estrangeiras para a consistência do banco?

R – Elas garantem consistência de dados, precisão das referências entre as tabelas e mais facilidade na hora de dar manutenção.

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

R – Os operadores da Álgebra Relacional são: Seleção, Projeção, Produto cartesiano, Diferença de conjunto e União. Já no Cálculo Relacional podemos usar Tupla(TRC ou Domínio(DRC)

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

R – O agrupamento é feito através do GROUP BY, e o seu requisito obrigatório é que é necessário dizer a coluna responsável pela estratificação do resultado da função.