

# Missão prática | Nível 1 | Mundo 3



# Estácio

CAMPUS: POLO DISTRITO JK - ANÁPOLIS - GO

CURSO: DESENVOLVIMENTO FULL STACK

NÚMERO DA TURMA: 2024.3

SEMESTRE LETIVO: Segundo semestre de 2024

ALUNO: Henrique Rodrigues Rabello Vieira

MATRÍCULA: 202301230527

## RPG0015 – Vamos manter as informações?

### Objetivos:

- 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)

## Segundo procedimento:

### Códigos solicitados:

```
USE LojaEstacio;

INSERT INTO Usuario (login, senha)
VALUES ('op1', 'op1'),
('op2', 'op2'),
('op3', 'op3'),
('op4', 'op4');

INSERT INTO Produto (idProduto, nome, quantidade, precoVenda)
VALUES ('1', 'Banana', '100', '5.00'),
('3', 'Laranja', '500', '2.00'),
('4', 'Manga', '800', '4.00');

INSERT INTO Pessoa (idPessoa, nome, logradouro, cidade, estado, telefone, email)
VALUES
(NEXT VALUE FOR ordemPessoaId, 'Henrique', 'Rua 1, 1', 'Goiás', 'GO', '1111-1111', 'henrique@gmail.com'),
(NEXT VALUE FOR ordemPessoaId, 'Breno', 'Rua 2, 2', 'Rio de Janeiro', 'RJ', '2222-2222', 'breno@gmail.com'),
(NEXT VALUE FOR ordemPessoaId, 'Pedro', 'Rua 3, 3', 'Sao Paulo', 'SP', '3333-3333', 'pedro@gmail.com'),
(NEXT VALUE FOR ordemPessoaId, 'Distribuidora 1', 'Rua 4, 4', 'Curitiba', 'PR', '4444-4444', 'd1@gmail.com'),
(NEXT VALUE FOR ordemPessoaId, 'Distribuidora 2', 'Rua 5, 5', 'Maranhão', 'MA', '5555-5555', 'd2@gmail.com');

INSERT INTO PessoaFisica (idPessoa, cpf)
VALUES (1, '11111111111'),
(2, '22222222222'),
(3, '33333333333');

INSERT INTO PessoaJuridica (idPessoa, cnpj)
VALUES (4, '44444444444444'),
(5, '55555555555555');

INSERT INTO Movimento (idMovimento, idUsuario, idPessoa, idProduto, quantidade, tipo, valorUnitario)
VALUES (1, 1, 5, 1, 40, 'E', 5.00),
(3, 2, 3, 3, 20, 'S', 2.00),
(5, 1, 4, 4, 60, 'E', 4.00),
(6, 2, 1, 1, 15, 'S', 5.00),
(7, 4, 2, 4, 25, 'S', 4.00),
(9, 3, 5, 3, 50, 'E', 2.00);
```

```
SELECT *
FROM PessoaFisica
INNER JOIN Pessoa ON PessoaFisica.idPessoa = Pessoa.idPessoa

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

SELECT
Produto.nome AS 'produto', Pessoa.nome AS 'fornecedor',
Movimento.quantidade, Movimento.valorUnitario,
Movimento.quantidade * Movimento.valorUnitario AS 'valorTotal'
FROM Movimento
INNER JOIN Produto ON Movimento.idProduto = Produto.idProduto
INNER JOIN Pessoa ON Movimento.idPessoa = Pessoa.idPessoa
WHERE Movimento.tipo = 'E';

SELECT
Produto.nome AS 'produto', Pessoa.nome AS 'comprador',
Movimento.quantidade, Movimento.valorUnitario,
Movimento.quantidade * Movimento.valorUnitario AS 'valorTotal'
FROM Movimento
INNER JOIN Produto ON Movimento.idProduto = Produto.idProduto
INNER JOIN Pessoa ON Movimento.idPessoa = Pessoa.idPessoa
WHERE Movimento.tipo = 'S';

SELECT
Produto.nome AS 'produto',
SUM(Movimento.quantidade * Movimento.valorUnitario) AS 'valorTotalEntrada'
FROM Movimento
JOIN Produto ON Movimento.idProduto = Produto.idProduto
WHERE Movimento.tipo = 'E'
GROUP BY Produto.nome;

SELECT
Produto.nome AS 'produto',
SUM(Movimento.quantidade * Movimento.valorUnitario) AS 'valorTotalSaida'
FROM Movimento
JOIN Produto ON Movimento.idProduto = Produto.idProduto
WHERE Movimento.tipo = 'S'
GROUP BY Produto.nome;

SELECT DISTINCT
Usuario.idUsuario, Usuario.login, Movimento.idMovimento
FROM Usuario
LEFT JOIN Movimento ON Usuario.idUsuario = Movimento.idUsuario AND Movimento.tipo = 'E'
WHERE idMovimento IS NULL;
```

```

SELECT
  Usuario.login AS 'operador',
  SUM(Movimento.quantidade * Movimento.valorUnitario) AS 'valorTotalEntrada'
FROM Movimento
JOIN Usuario ON Movimento.idUsuario = Usuario.idUsuario
WHERE Movimento.tipo = 'E'
GROUP BY Usuario.login;

SELECT
  Usuario.login AS 'operador',
  SUM(Movimento.quantidade * Movimento.valorUnitario) AS 'valorTotalSaida'
FROM Movimento
JOIN Usuario ON Movimento.idUsuario = Usuario.idUsuario
WHERE Movimento.tipo = 'S'
GROUP BY Usuario.login;

SELECT
  SUM(Movimento.valorUnitario * Movimento.quantidade) / SUM(Movimento.quantidade) AS 'médiaPonderada'
FROM Movimento
WHERE Movimento.tipo = 'S';

```

## Resultados:

Resultados

Mensagens

	idPessoa	cpf	idPessoa	nome	logradouro	cidade	estado	telefone	email
1	1	11111111111	1	Henrique	Rua 1, 1	Goiás	GO	1111-1111	henrique@gmail.com
2	2	22222222222	2	Breno	Rua 2, 2	Rio de Janeiro	RJ	2222-2222	breno@gmail.com
3	3	33333333333	3	Pedro	Rua 3, 3	Sao Paulo	SP	3333-3333	pedro@gmail.com

	idPessoa	cnpj	idPessoa	nome	logradouro	cidade	estado	telefone	email
1	4	44444444444444	4	Distribuidora 1	Rua 4, 4	Curitiba	PR	4444-4444	d1@gmail.com
2	5	55555555555555	5	Distribuidora 2	Rua 5, 5	Maranhao	MA	5555-5555	d2@gmail.com

	produto	fornecedor	quantidade	valorUnitario	valorTotal
1	Banana	Distribuidora 2	40	5.00	200.00
2	Manga	Distribuidora 1	60	4.00	240.00
3	Laranja	Distribuidora 2	50	2.00	100.00

	produto	comprador	quantidade	valorUnitario	valorTotal
1	Laranja	Pedro	20	2.00	40.00
2	Banana	Henrique	15	5.00	75.00
3	Manga	Breno	25	4.00	100.00

	produto	valorTotalEntrada
1	Banana	200.00
2	Laranja	100.00
3	Manga	240.00

	operador	valorTotalEntrada
1	op1	440.00
2	op3	100.00

	operador	valorTotalSaida
1	op2	115.00
2	op4	100.00

	médiaPonderada
1	3.583333

## Análise e conclusão:

### Quais as diferenças no uso de sequence e identity?

Sequence gera valores sequenciais únicos mesmo que seja utilizado por várias conexões ao mesmo tempo, onde os valores são armazenados em um objeto separado, já o identity gera um sequencial único dentro de apenas 1 conexão, armazenando os valores na própria tabela.

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

Impede inserções inválidas de dados, impedem atualizações/exclusões inválidas, como por exemplo excluir dados em uma tabela pai que tenham referências a tabelas filhas e permitem exclusões em cascata.

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

Álgebra relacional: UNION, INTERSECT, EXCEPT, CROSS JOIN, JOIN e SELECT

Cálculo relacional: WHERE, GROUP BY e ORDER BY

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

É feito utilizando GROUP BY. A cláusula GROUP BY deve especificar todas as colunas não agregadas na cláusula SELECT.