

Disciplina: Nível 2: Vamos Manter as Informações? Turma: 22.3 Semestre: 3º
 Aluno: André Luiz Ferreira de Souza
 Github: <https://github.com/adventureandre/CadastroPOO-Estacio>

2º Procedimento | Alimentando a Base

Habilidades básicas para a modelagem da base de dados em um sistema, além do uso da sintaxe SQL para a criação das estruturas necessárias.

Dados.sql

```
-- inserindo pessoa física
INSERT INTO pessoas (nome, endereco, cidade, telefone, email)
VALUES ('André', 'Rua prof julita', 'Leopoldo de Bulhões', '62996106990',
'adventureandre@hotmail.com');

INSERT INTO pessoa_física (id_pessoa, cpf)
VALUES (SCOPE_IDENTITY(), '15687456947');

-- inserindo pessoa jurídica
INSERT INTO pessoas (nome, endereco, cidade, telefone, email)
VALUES ('ALFA', 'Rua prof julita', 'Leopoldo de Bulhões', '33371570',
'admin@funilariaepintura.com.br');

INSERT INTO pessoa_jurídica (id_pessoa, cnpj)
VALUES (SCOPE_IDENTITY(), '4545540001-78');

-- inserindo produto
INSERT INTO produto
(nome, quantidade, precoVenda)
VALUES('Banana', 100, 5.00);

-- inserindo produto
INSERT INTO produto
(nome, quantidade, precoVenda)
VALUES('laranja', 500, 4.50);

-- inserindo produto
INSERT INTO produto
(nome, quantidade, precoVenda)
VALUES('uva', 20, 50.00);

-- inserindo produto
INSERT INTO produto
(nome, quantidade, precoVenda)
VALUES('ameixa', 40, 18.25);

-- inserindo usuário
INSERT INTO usuário
([login], senha)
VALUES('op1', 'op1');

-- inserindo usuário
INSERT INTO usuário
([login], senha)
VALUES('op2', 'op2');

-- Inserir uma entrada (E)
INSERT INTO movimento (id_usuario, id_pessoa, id_produto, quantidade, tipo,
valorUnitario)
VALUES (0, 4, 15, 'E', 10.00);

-- Inserir uma saída (S)
INSERT INTO movimento (id_usuario, id_pessoa, id_produto, quantidade, tipo,
valorUnitario)
VALUES (0, 2, 2, 'S', 15.00);

-- Inserir uma entrada (E)
INSERT INTO movimento (id_usuario, id_pessoa, id_produto, quantidade, tipo,
valorUnitario)
VALUES (1, 1, 3, 9, 'E', 20.00);

-- Inserir uma saída (S)
INSERT INTO movimento (id_usuario, id_pessoa, id_produto, quantidade, tipo,
valorUnitario)
VALUES (1, 3, 1, 2, 'S', 10.00);
```

Resultado

```
-- Consultas
-- Dados completo de pessoas físicas
SELECT p.*, pf.cpf
FROM pessoas p
INNER JOIN pessoa_física pf ON p.id_pessoa = pf.id_pessoa;
```

id_pessoa	nome	endereco	cidade	telefone	email	cpf
0	André	Rua prof julita	Leopoldo de Bulhões	62996106990	adventureandre@hotmail.com	15687456947
1	milena	Rua prof julita	Leopoldo de Bulhões	62554215545	milena@hotmail.com	54542215454
2	marco	Rua da feira	Goiânia	88888888	marco@hotmail.com	1111111111
5	Eunice	Rua prof julita	Leopoldo de Bulhões	6245454545	eunice@hotmail.com	7777777777


```
-- Consultas
-- Dados completo de pessoas jurídica
SELECT p.* , pj.cnpj
FROM pessoas p
INNER JOIN pessoa_jurídica pj ON p.id_pessoa = pj.id_pessoa;
```

id_pessoa	nome	endereco	cidade	telefone	email	cnpj
3	ALFA	Rua prof julita	Leopoldo de Bulhões	33371570	admin@funilariaepintura.com.br	4545540001-78
4	JJC	Rua pedro loduvico	Anápolis	33370000	admin@jjc.com.br	77885550001-78


```
-- Consultas
-- Dados da movimentação e junta eles criando um total
-- Consulta para entradas (tipo 'E')
SELECT
    m.id_movimento,
    pr.nome AS nome_produto,
    p.nome AS nome_fornecedor,
    m.quantidade,
    m.valorUnitario,
    (m.quantidade * m.valorUnitario) AS valor_total
FROM movimento m
INNER JOIN produto pr ON m.id_produto = pr.id_produto
INNER JOIN pessoa_física p ON m.id_pessoa = p.id_pessoa
WHERE m.tipo = 'E';
```

id_movimento	nome_produto	nome_fornecedor	quantidade	valorUnitario	valor_total
0	Banana	JJC	15	10	150
2	ameixa	ALFA	9	20	180


```
-- Consulta para saídas (tipo 'S')
SELECT
    m.id_movimento,
    pr.nome AS nome_produto,
    p.nome AS nome_fornecedor,
    m.quantidade,
    m.valorUnitario,
    (m.quantidade * m.valorUnitario) AS valor_total
FROM movimento m
INNER JOIN produto pr ON m.id_produto = pr.id_produto
INNER JOIN pessoa_física p ON m.id_pessoa = p.id_pessoa
WHERE m.tipo = 'S';
```

id_movimento	nome_produto	nome_fornecedor	quantidade	valorUnitario	valor_total
3	laranja	milena	2	10	20
1	uva	marco	2	15	30


```
-- Valor total das entradas agrupadas por produto
SELECT
    m.id_produto,
    pr.nome AS nome_produto,
    SUM(m.quantidade * m.valorUnitario) AS valor_total_entradas
FROM movimento m
INNER JOIN produto pr ON m.id_produto = pr.id_produto
WHERE m.tipo = 'E'
GROUP BY m.id_produto, pr.nome;
```

id_produto	nome_produto	valor_total_entradas
0	Banana	150
3	ameixa	180


```
-- Valor total das saídas agrupadas por produto
SELECT
    m.id_produto,
    pr.nome AS nome_produto,
    SUM(m.quantidade * m.valorUnitario) AS valor_total_saídas
FROM movimento m
INNER JOIN produto pr ON m.id_produto = pr.id_produto
WHERE m.tipo = 'S'
GROUP BY m.id_produto, pr.nome;
```

id_produto	nome_produto	valor_total_saídas
1	laranja	20
2	uva	30


```
-- Operadores que não efetuaram movimentações de entrada (compra)
SELECT u.id_usuario, u.login
FROM usuário u
LEFT JOIN movimento m ON u.id_usuario = m.id_usuario AND m.tipo = 'E'
WHERE m.id_movimento IS NULL;
```

id_usuario	login
0	op1
1	op2


```
-- Valor total de entrada, agrupado por operador
SELECT
    u.id_usuario,
    u.login AS nome_operador,
    SUM(m.quantidade * m.valorUnitario) AS valor_total_entrada
FROM usuário u
LEFT JOIN movimento m ON u.id_usuario = m.id_usuario AND m.tipo = 'E'
GROUP BY u.id_usuario, u.login;
```

id_usuario	nome_operador	valor_total_entrada
0	op1	150
1	op2	180


```
-- Valor total de saída, agrupado por operador
SELECT
    u.id_usuario,
    u.login AS nome_operador,
    SUM(m.quantidade * m.valorUnitario) AS valor_total_saida
FROM usuário u
LEFT JOIN movimento m ON u.id_usuario = m.id_usuario AND m.tipo = 'S'
GROUP BY u.id_usuario, u.login;
```

id_usuario	nome_operador	valor_total_saida
0	op1	30
1	op2	20


```
-- Valor médio de venda por produto (média ponderada)
SELECT
    p.id_produto,
    p.nome AS nome_produto,
    SUM(m.quantidade * m.valorUnitario) / SUM(m.quantidade) AS valor_médio_venda
FROM movimento m
INNER JOIN produto p ON m.id_produto = p.id_produto
WHERE m.tipo = 'S'
GROUP BY p.id_produto, p.nome;
```

id_produto	nome_produto	valor_médio_venda
1	laranja	10
2	uva	15

Análise e Conclusão

Sequence vs. Identity:

- "Sequence" e "Identity" são formas de criar números únicos em bancos de dados. "Sequence" é mais flexível e funciona em vários sistemas de banco de dados, enquanto "Identity" é específico para alguns sistemas, como SQL Server e MySQL.

Chaves Estrangeiras:

- Chaves estrangeiras são como conexões entre tabelas em bancos de dados. Elas garantem relacionamentos consistentes e corretos entre dados, além de facilitarem a manutenção e consulta dos dados.

Operadores SQL:

- Operadores SQL são comandos usados para fazer perguntas e manipular dados em bancos de dados. Exemplos incluem SELECT (para obter dados), JOIN (para combinar tabelas) e GROUP BY (para agrupar dados).

Agrupamento em Consultas:

- O agrupamento em consultas SQL é usado para resumir dados. Um requisito importante é que todas as colunas na consulta devem estar agrupadas ou usadas em funções de resumo, como SOMA ou CONTAGEM.