



Curso: Desenvolvimento Full Stack 2024.1 3º Semestre

Aluno: Rogério Benedito Geraldo

Polo: Nova Ourinhos

Tutor: Alessandro

Disciplina: Vamos desenvolver as informações

Objetivos 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.

Link do Git: <https://github.com/Rogeriobg/vamos-Manter-As-Informacoes.git>

Códigos do 2º Procedimento

-- INSERINDO OS USUARIOS

INSERT INTO Usuario (UsuarioID, Login, Senha)

VALUES (1, 'op1', 'op2'),

(2, 'op2', 'op2'),

(3, 'op3', 'op3');

SELECT \* FROM Usuario;



--INSERÇÃO DOS DADOS COMUNS NA TABELA PESSOA

INSERT INTO Pessoa (PessoaID, Nome, Tipo, Logradouro, Cidade, Estado, Telefone, Email)

VALUES ('7', 'Joao', 'F', 'Rua 12,casa 3. Quitanda', 'Riacho do Sul', 'PA', '1111-1111','joao@riacho.com'),

('15', 'JJC', 'F', 'Rua 11,Centro', 'Riacho do Norte', 'PA', '1212-1212','jjc@riacho.com'),

('16', 'JJC2', 'F', 'Rua 11,Centro B', 'Riacho do Norte', 'PA', '1313-1313','jjc2@riacho.com'),

('17', 'JJC3', 'J', 'Rua 11,Centro c', 'Riacho do Norte', 'PA', '1414-1414','jjc4@riacho.com'),

('18', 'JJC4', 'J', 'Rua 11,Centro D', 'Riacho do Norte', 'PA', '1515-1515','jjc5@riacho.com');

SELECT \* FROM Pessoa;

--INSERÇÃO DOS DADOS DE CPF E CNPJ DE PESSOA FÍSICA E JURÍDICA

INSERT INTO PessoaFisica (PessoaID, Pessoa\_PessoaID, CPF)

VALUES (7, '7', '11111111111'),

(15, '15', '22222222222'),



```
(16, '16', '333333333333');
```

```
INSERT INTO PessoaJuridica (PessoaID, Pessoa_PessoaID, CNPJ)
```

```
VALUES ('17', '17', '4444444444444444'),
```

```
('18', '18', '5555555555555555');
```

```
SELECT * FROM PessoaJuridica
```

```
SELECT * FROM PessoaFisica;
```

```
-- INSERÇÃO DE PRODUTOS
```

```
SET IDENTITY_INSERT Produto ON;
```

```
INSERT INTO Produto (ProdutoID, Nome, Quantidade, PrecoVenda)
```

```
VALUES ('1', 'Banana', '100', '5.00'),
```

```
      ('3', 'Laranja', '500', '2.00'),
```

```
      ('4', 'Manga', '800', '4.00');
```

```
SELECT * FROM Produto;
```

```
--INSERÇÃO DOS DADOS DE MOVIMENTO DENOMINANDO S COMO SAÍDA E  
E COMO ENTRADA
```

```
INSERT INTO Movimento (MovimentoID, UsuarioID, Tipo, PessoaID, ProdutoID,  
Quantidade, PrecoUnitario)
```

```
VALUES
```

```
(1, 1, 'S', 7, 1, 20, 4.00),
```

```
(4, 1, 'S', 7, 3, 15, 2.00),
```



```
(5, 2, 'S', 7, 3, 10, 3.00),  
(7, 1, 'E', 15, 3, 15, 5.00),  
(8, 1, 'E', 15, 4, 20, 4.00);
```

```
SELECT * FROM Movimento;
```

```
--DADOS COMPLETOS DE PESSOA FÍSICA
```

```
SELECT
```

```
    p.PessoalD,  
    p.Nome,  
    p.Logradouro,  
    p.Cidade,  
    p.Estado,  
    p.Telefone,  
    p.Email,  
    pf.CPF
```

```
FROM
```

```
    Pessoa p
```

```
JOIN
```

```
    PessoaFisica pf ON p.PessoalD = pf.PessoalD;
```



--DADOS COMPLETOS DE PESSOA JURIDICA

SELECT

p.PessoalD,  
p.Nome,  
p.Logradouro,  
p.Cidade,  
p.Estado,  
p.Telefone,  
p.Email,  
pj.CNPJ

FROM

Pessoa p

JOIN

PessoaJuridica pj ON p.PessoalD = pj.PessoalD;

-- DADOS COMPLETOS DE PESSOA FÍSICA E JURÍDICA

SELECT

p.PessoalD,  
p.Nome,  
p.Logradouro,  
p.Cidade,  
p.Estado,



p.Telefone,

p.Email,

pf.CPF AS Identificador

FROM

Pessoa p

JOIN

PessoaFisica pf ON p.PessoalD = pf.PessoalD

UNION ALL

SELECT

p.PessoalD,

p.Nome,

p.Logradouro,

p.Cidade,

p.Estado,

p.Telefone,

p.Email,

pj.CNPJ AS Identificador

FROM

Pessoa p

JOIN

PessoaJuridica pj ON p.PessoalD = pj.PessoalD;



--MOVIMENTAÇÃO DE ENTRADA COM PRODUTO, FORNECEDOR,  
QUANTIDADE, PREÇO UNITÁRIO E VALOR TOTAL

SELECT

m.MovimentoID,

u.UsuarioID,

pr.Nome AS Produto,

p.Nome AS Fornecedor,

m.Quantidade,

m.PrecoUnitario,

m.Quantidade \* m.PrecoUnitario AS ValorTotal

FROM

Movimento m

JOIN

Produto pr ON m.ProdutoID = pr.ProdutoID

JOIN

Pessoa p ON m.PessoalID = p.PessoalID

JOIN

Usuario u ON m.UsuarioID = u.UsuarioID

WHERE

m.Tipo = 'E';



--MOVIMENTAÇÃO DE SAÍDA COM PRODUTO, COMPRADOR , QUANTIDADE,  
PREÇO UNITÁRIO E VALOR TOTAL

SELECT

m.MovimentoID,

u.UsuarioID,

pr.Nome AS Produto,

p.Nome AS Comprador,

m.Quantidade,

m.PrecoUnitario,

m.Quantidade \* m.PrecoUnitario AS ValorTotal

FROM

Movimento m

JOIN

Produto pr ON m.ProdutoID = pr.ProdutoID

JOIN

Pessoa p ON m.PessoalID = p.PessoalID

JOIN

Usuario u ON m.UsuarioID = u.UsuarioID

WHERE

m.Tipo = 'S';

--Valor total das entradas agrupadas por produto





SELECT

pr.Nome AS Produto,

SUM(m.Quantidade \* m.PrecoUnitario) AS ValorTotalEntradas

FROM

Movimento m

JOIN

Produto pr ON m.ProdutoID = pr.ProdutoID

WHERE

m.Tipo = 'E'

GROUP BY

pr.Nome;

--Valor total das saídas agrupadas por produto

SELECT

pr.Nome AS Produto,

SUM(m.Quantidade \* m.PrecoUnitario) AS ValorTotalSaidas

FROM

Movimento m

JOIN

Produto pr ON m.ProdutoID = pr.ProdutoID

WHERE

m.Tipo = 'S'

GROUP BY



pr.Nome;

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

SELECT

U.UsuarioID,

U.Login,

U.Senha

FROM

Usuario U

WHERE

U.UsuarioID NOT IN (

SELECT DISTINCT

M.UsuarioID

FROM

Movimento M

WHERE

M.Tipo = 'E'

);

--Valor total de entrada, agrupado por operador.

SELECT

M.UsuarioID,

U.Login,



```
SUM(M.Quantidade * M.PrecoUnitario) AS ValorTotalEntrada
FROM
    Movimento M
JOIN
    Usuario U ON M.UsuarioID = U.UsuarioID
WHERE
    M.Tipo = 'E'
GROUP BY
    M.UsuarioID,
    U.Login;
```

--Valor total de saída, agrupado por operador.

```
SELECT
    M.UsuarioID,
    U.Login,
    SUM(M.Quantidade * M.PrecoUnitario) AS ValorTotalSaida
FROM
    Movimento M
JOIN
    Usuario U ON M.UsuarioID = U.UsuarioID
WHERE
    M.Tipo = 'S'
GROUP BY
```



M.UsuarioID,

U.Login;

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

SELECT

    ProdutoID,

    AVG(PrecoUnitario) AS ValorMedioVenda

FROM

    Movimento

WHERE

    Tipo = 'S'

GROUP BY

    ProdutoID;

Conclusão e Análise:

[Quais as diferenças no uso de sequence e identity?](#)

Sequência (Sequence) é independente de tabela, permite controle flexível e pode ser compartilhada. Identity é específico da coluna, vinculado à tabela e limitado a uma por tabela.

[Qual a importância das chaves estrangeiras para a consistência do banco?](#)

As chaves estrangeiras garantem integridade referencial, mantendo consistência ao exigir que os valores em uma tabela de referência existam na tabela relacionada, evitando assim referências inválidas ou órfãs.



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

Na álgebra relacional, pertencem: SELEÇÃO ( $\sigma$ ), PROJEÇÃO ( $\pi$ ), UNIÃO ( $\cup$ ), INTERSEÇÃO ( $\cap$ ), DIFERENÇA ( $-$ ), PRODUTO CARTESIANO ( $\times$ ), e JUNÇÃO ( $\bowtie$ ). No cálculo relacional, são definidos os predicados, quantificadores (EXISTS, FOR ALL), e construtores lógicos como AND, OR, NOT.

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

O agrupamento em consultas é feito usando a cláusula GROUP BY, que agrupa linhas com base em valores semelhantes em uma ou mais colunas. Um requisito obrigatório é que todas as colunas listadas na instrução SELECT devem ser agregadas ou incluídas na cláusula GROUP BY.