

Nome: Michel Angelo da Silva Portugal .

Matrícula: 202208826174

Campus: Duque de Caxias

Disciplina: RPG0015 - Vamos manter as informações! / Sem 2/2023.

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)

Link do Repositório:

<https://github.com/miaspe/SQL>

Análise, Questionamentos e Conclusão:

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

R: Através das ditas chaves primárias e das chaves estrangeiras.

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

R: As chaves estrangeiras fazem uma correlação entre as tabelas juntamente com uma cardinalidade de 1:X, repare que é 1:X e não 1:N, onde x pode variar de 1 pra N.

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

R: Melhora a produtividade por permitir a manipulação (CRUD) de objetos direto no aplicativo sem o uso de outras IDEs e editores de texto tão comuns em outras linguagens.

- Inserir alguns produtos na base de dados, obtendo um conjunto como o que é apresentado a seguir:

Resultados		Mensagens		
	idProduto	nome	quantidade	precoVenda
1	1	Banana	100	5.00
2	3	Laranja	500	2.00
3	4	Manga	800	4.00

Nosso código e tabela abaixo :

```
INSERT INTO Produto (ProdutoID, Nome_Produto, Quantid_Prod, Preço_de_Venda)  
VALUES
```

```
(1, 'uva RED', '15Kg', 8.31),  
(2, 'Abacaxi', '30Unid', 6.22),  
(3, 'Cebolas', '30Kg', 4.55);
```

	ProdutoID	Nome_Produto	Quantid_Prod	Preço_de_Venda
1	1	uva RED	15Kg	8.31
2	2	Abacaxi	30Unid	6.22
3	3	Cebolas	30Kg	4.55

✓ Consulta executada com êxito.

- Efetuar as seguintes consultas sobre os dados inseridos:

Dados completos de pessoas físicas:

Alpha.sql - DESKTOP...R2G.loja (loja (53)) consulta.sql - DESKTOP...R2G.loja (loja (59))* criando a sequence.s...R2G.loja (loja (52))*

```

UPDATE PessoaFisica
SET
  Logradouro = 'Rua Castilho',
  Cidade = 'Cade Ocê',
  Estado = 'MG',
  Telefone = '111222333',
  Email = 'Bravo_cunha@email.com'
WHERE
  PessoaFID = 4;

SELECT *
FROM Pessoa
JOIN PessoaFisica ON Pessoa.PessoaID = PessoaFisica.PessoaFID;

```

100 %

Resultados Mensagens

	PessoaID	Nome	PessoaFID	CPF	Idade	logradouro	Cidade	Estado	Telefone	Email
1	1	Simão cunha	1	12345678901	30	Rua Aberta	Arimbá	SP	123456789	Simão_cunha@email.com
2	2	Mary Portland	2	23456789012	25	Rua do Sob	Bacará	RJ	987654321	Mary_Portland@email.com
3	3	Carlos Oakes	3	34567890123	40	NULL	NULL	NULL	NULL	NULL
4	4	Bravo cunha	4	45678901234	35	Rua Castilho	Cade Ocê	MG	111222333	Bravo_cunha@email.com
5	5	Ary Devito	5	56789012345	28	NULL	NULL	NULL	NULL	NULL

Dados completos de pessoas jurídicas:

Claro que poderíamos colocar uma tag/flag na forma de coluna a mais em pessoa alternando ou apontando o cadastro para jurídica ou física; E selecionar/apontar pra tag em nosso select PessoaJID, mas pra fins de demonstração, isso não foi implementado ...

```

SELECT *
FROM Pessoa
JOIN PessoaJuridica ON Pessoa.PessoaID = PessoaJuridica.PessoaJID;

```

100 %

Resultados Mensagens

	PessoaID	Nome	PessoaJID	CNPJ	Logradouro	Cidade	Estado	Telefone	Email	Nome
1	1	Simão cunha	1	12345678901234	Rua Castanha	Cadeira	MG	133222333	BBB@email.com	NULL
2	2	Mary Portland	2	23456789012345	NULL	NULL	NULL	NULL	NULL	NULL
3	3	Carlos Oakes	3	34567890123456	NULL	NULL	NULL	NULL	NULL	NULL
4	4	Bravo cunha	4	45678901234567	NULL	NULL	NULL	NULL	NULL	NULL
5	5	Ary Devito	5	56789012345678	NULL	NULL	NULL	NULL	NULL	NULL

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

```

use loja
SELECT * from dbo.Movimentacao

SELECT *
FROM Movimentacao
WHERE TipoMov_E_S = 'E';

SELECT MovimentID, Fornecedor, PreçoUnitario, Quantidade
FROM Movimentacao
WHERE TipoMov_E_S = 'S';

```

100 %

Resultados Mensagens

	MovimentID	TipoMov_E_S	Fornecedor	Comprador	PreçoUnitario	Quantidade	ValorTotal	ProdutoID	Nome_Produto	Operador_Usuario
1	1	S	fruits	mercadin-din	8.31	4	33.24	1	uva RED	2
2	2	S	fruits	mercado	8.31	2	16.62	1	uva RED	1
3	3	E	flora health	Extra mercadin	1.01	100	101.00	4	couve-flor	3
4	4	E	flora health	zé das couves	1.00	160	160.00	5	couve	3

	MovimentID	TipoMov_E_S	Fornecedor	Comprador	PreçoUnitario	Quantidade	ValorTotal	ProdutoID	Nome_Produto	Operador_Usuario
1	3	E	flora health	Extra mercadin	1.01	100	101.00	4	couve-flor	3
2	4	E	flora health	zé das couves	1.00	160	160.00	5	couve	3

fig1:Entradas

```
SELECT MovimentID, Nome_Produto, Fornecedor, Quantidade,PrecoUnitario,ValorTotal
FROM Movimentacao
WHERE TipoMov_E_S = 'E';
```

	MovimentID	Nome_Produto	Fornecedor	Quantidade	PrecoUnitario	ValorTotal
1	3	couve-flor	flora health	100	1.01	101.00
2	4	couve	flora health	160	1.00	160.00

fig2:Entradas filtradas

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

```
SELECT MovimentID, Nome_Produto, Comprador, Quantidade,PrecoUnitario,ValorTotal
FROM Movimentacao
WHERE TipoMov_E_S = 'S';
```

	MovimentID	Nome_Produto	Comprador	Quantidade	PrecoUnitario	ValorTotal
1	1	uva RED	mercadin-din	4	8.31	33.24
2	2	uva RED	mercado	2	8.31	16.62

fig3:Saidas filtradas

Valor total das entradas agrupadas por produto:

```
use loja
SELECT * from dbo.Movimentacao

SELECT
    ProdutoID,
    Nome_Produto,
    SUM(ValorTotal) AS ValorTotalEntradas
FROM
    Movimentacao
WHERE
    TipoMov_E_S = 'E'
GROUP BY
    ProdutoID, Nome_Produto;
```

	MovimentID	TipoMov_E_S	Fornecedor	Comprador	PrecoUnitario	Quantidade	ValorTotal	ProdutoID	Nome_Produto	Operador_Usuario
1	1	S	fruits	mercadin-din	8.31	4	33.24	1	uva RED	2
2	2	S	fruits	mercado	8.31	2	16.62	1	uva RED	1
3	3	E	flora health	Extra mercadin	1.01	100	101.00	4	couve-flor	3
4	4	E	flora health	zé das couves	1.00	160	160.00	5	couve	3
5	5	S	quintal do pai	Extra mercadin	6.22	10	62.20	2	Abacaxi	2
6	6	S	quintal do pai	mercado	6.22	11	68.42	2	Abacaxi	2

	ProdutoID	Nome_Produto	ValorTotalEntradas
1	5	couve	160.00
2	4	couve-flor	101.00

Consulta executada com êxito. DESKTOP-9LVDR2G (16.0 RTM) loja (66)

Usar - Group by SUM , faz a soma.

Valor total das saídas agrupadas por produto.

```

use loja
SELECT * from dbo.Movimentacao

SELECT
    ProdutoID,
    Nome_Produto,
    SUM(ValorTotal) AS ValorTotalSaidas
FROM
    Movimentacao
WHERE
    TipoMov_E_S = 'S'
GROUP BY
    ProdutoID, Nome_Produto;

```

MovimentID	TipoMov_E_S	Fornecedor	Comprador	PrecoUnitario	Quantidade	ValorTotal	ProdutoID	Nome_Produto	Operador_Usuario
1	S	fruits	mercadin-din	8.31	4	33.24	1	uva RED	2
2	S	fruits	mercado	8.31	2	16.62	1	uva RED	1
3	E	flora health	Extra mercadin	1.01	100	101.00	4	couve-flor	3
4	E	flora health	zé das couves	1.00	160	160.00	5	couve	3
5	S	quintal do pai	Extra mercadin	6.22	10	62.20	2	Abacaxi	2
6	S	quintal do pai	mercado	6.22	11	68.42	2	Abacaxi	2

ProdutoID	Nome_Produto	ValorTotalSaidas
2	Abacaxi	130.62
1	uva RED	49.86

Consulta executada com êxito. DESKTOP-9LVDR2G (16.0 RTM) loja (169)

Operadores que não efetuaram movimentações de entrada (compra):

```

SELECT DISTINCT
    Operador_Usuario
FROM
    Movimentacao
WHERE
    TipoMov_E_S IS NULL OR TipoMov_E_S != 'E';

```

Operador_Usuario
1
2

Aqui vemos/filtramos as matrículas de operadores/usuários que **Não (!=) efetuaram** entradas ou movimentações de compra.

Valor total de entrada (E), agrupado por operador :

```

select * from Movimentacao

SELECT
    Operador_Usuario,
    SUM(ValorTotal) AS ValorTotalEntradas
FROM
    Movimentacao
WHERE
    TipoMov_E_S = 'E'
GROUP BY
    Operador_Usuario;

```

MovimentID	TipoMov_E_S	Fornecedor	Comprador	PrecoUnitario	Quantidade	ValorTotal	ProdutoID	Nome_Produto	Operador_Usuario
1	S	fruits	mercadin-din	8.31	4	33.24	1	uva RED	2
2	S	fruits	mercado	8.31	2	16.62	1	uva RED	1
3	E	flora health	Extra mercadin	1.01	100	101.00	4	couve-flor	3
4	E	flora health	zé das couves	1.00	160	160.00	5	couve	3
5	S	quintal do pai	Extra mercadin	6.22	10	62.20	2	Abacaxi	2
6	S	quintal do pai	mercado	6.22	11	68.42	2	Abacaxi	2

Operador_Usuario	ValorTotalEntradas
3	261.00

Consulta executada com êxito. DESKTOP-9LVDR2G (16.0 RTM) loja (169)

Valor total de saída (S), agrupado por operador :

Aqui vemos o quanto o usuário (2) que poderia ser VENDEDOR, fez a mais que o operador (1).

```
select * from Movimentacao
SELECT
    Operador_Usuario,
    SUM(ValorTotal) AS ValorTotalEntradas
FROM
    Movimentacao
WHERE
    TipoMov_E_S = 'S'
GROUP BY
    Operador_Usuario;
```

	MovimentID	TipoMov_E_S	Fornecedor	Comprador	PrecoUnitario	Quantidade	ValorTotal	ProdutoID	Nome_Produto	Operador_Usuario
1	1	S	fruits	mercadin-din	8.31	4	33.24	1	uva RED	2
2	2	S	fruits	mercado	8.31	2	16.62	1	uva RED	1
3	3	E	flora health	Extra mercadin	1.01	100	101.00	4	couve-flor	3
4	4	E	flora health	zé das couves	1.00	160	160.00	5	couve	3
5	5	S	quintal do pai	Extra mercadin	6.22	10	62.20	2	Abacaxi	2
6	6	S	quintal do pai	mercado	6.22	11	68.42	2	Abacaxi	2

	Operador_Usuario	ValorTotalEntradas
1	1	16.62
2	2	163.86

Consulta executada com êxito. | DESKTOP-9LVDR2G (16.0 RTM) | loja (169) |

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

```
select * from Movimentacao
SELECT
    AVG(PrecoPonderado) AS MediaPonderada
FROM (
    SELECT
        ProdutoID,
        AVG(PrecoUnitario * Quantidade) AS PrecoPonderado
    FROM
        Movimentacao
    WHERE
        [TipoMov_E_S] = 'S'
    GROUP BY
        ProdutoID
) AS CalculosPorProduto;
```

	MovimentID	TipoMov_E_S	Fornecedor	Comprador	PrecoUnitario	Quantidade	ValorTotal	ProdutoID	Nome_Produto	Operador_Usuario
1	1	S	fruits	mercadin-din	8.31	4	33.24	1	uva RED	2
2	2	S	fruits	mercado	8.31	2	16.62	1	uva RED	1
3	3	E	flora health	Extra mercadin	1.01	100	101.00	4	couve-flor	3
4	4	E	flora health	zé das couves	1.00	160	160.00	5	couve	3
5	5	S	quintal do pai	Extra mercadin	6.22	10	62.20	2	Abacaxi	2
6	6	S	quintal do pai	mercado	6.22	11	68.42	2	Abacaxi	2

	MediaPonderada
1	45.120000

Consulta executada com êxito. | DESKTOP-9LVDR2G (16.0 RTM) | loja (169) | loj

Resultados da execução dos códigos também foram apresentados nas fotos;

Análise e Conclusão:

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

R: O recurso (sequence) é mais “novo” (2002) que o identity , que é mais antigo e menos versátil pra manipular/alterar.

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

R: Estabelecem **relação** entre as tabelas, com isso também temos mais segurança no tocante a informações sensíveis (confidenciais ou sigilosas)

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

R: Quando falamos de relação lembramos da teoria dos conjuntos (tuplas) que no caso formam as tabelas e por sua vez os objetos que serão manipulados. No tocante as relações podemos destacar a 1:1 onde na teoria dos conjuntos essa relação é não é só uma relação ela é uma função e pode ser expressa como tal, onde cada elemento em X corresponde a um único elemento em Y (no caso de tabelas x é uma tabela e y é outra) as outras relações são somente relações não chegando ao status de função porém temos também JOIN e INNER e todas as outras operações referentes a conjuntos como pra união, interseção(INTERSECT) e outras, podendo também fazermos operações com OR, AND e NOT e ainda > < e = claro que dependendo do conteúdo dos campos(colunas) em questão , que podem armazenar dados Alpha e dados Numéricos ou a junção dos dois.

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

R: Para isso existe a função (GROUP BY) e suas variantes agregatórias (SUM(), COUNT(), AVG(), MAX(), MIN(), etc.)

Link do Repositório:

<https://github.com/miaspe/SQL>