



Campus: Polo Ingleses

Curso: Desenvolvimento Full Stack

Disciplina: Nível 2 - Vamos Manter as Informações?

Turma: 9001

Semestre: 23.3

Aluno: Alexandre Henrique Fernandes Nolla

1º Procedimento | Criando o Banco de Dados

Objetivo da prática: Realizar o modelo relacional físico do banco de dados chamado 'Loja.db', e após desenhado o modelo relacional físico, criar o banco de dados propriamente dito.

Comandos SQL utilizados para a criação do banco de dados:

```
CREATE TABLE Usuario (  
    id INTEGER NOT NULL UNIQUE,  
    nome TEXT NOT NULL UNIQUE,  
    senha TEXT NOT NULL,  
    PRIMARY KEY (id AUTOINCREMENT));
```

```
CREATE TABLE Produto (  
    id INTEGER NOT NULL UNIQUE,  
    nome TEXT NOT NULL UNIQUE,  
    quantidade INTEGER NOT NULL,  
    preco REAL NOT NULL,  
    PRIMARY KEY (id AUTOINCREMENT));
```

```
CREATE TABLE PessoaJuridica (  
    id INTEGER NOT NULL,  
    cnpj TEXT NOT NULL UNIQUE,  
    PRIMARY KEY (id AUTOINCREMENT));
```

```

CREATE TABLE PessoaFisica (
    id INTEGER NOT NULL,
    cpf TEXT NOT NULL UNIQUE,
    PRIMARY KEY (id AUTOINCREMENT));

CREATE TABLE Pessoa (id INTEGER NOT NULL UNIQUE,
    nome TEXT NOT NULL,
    logradouro TEXT NOT NULL,
    cidade TEXT NOT NULL,
    estado TEXT NOT NULL,
    telefone TEXT NOT NULL,
    email TEXT NOT NULL,
    idPessoaFisica INTEGER,
    idPessoaJuridica INTEGER,
    PRIMARY KEY (id AUTOINCREMENT),
    FOREIGN KEY (idPessoaJuridica) REFERENCES PessoaJuridica (id),
    FOREIGN KEY (idPessoaFisica) REFERENCES PessoaFisica (id));

CREATE TABLE Movimento (
    id INTEGER NOT NULL UNIQUE,
    idUsuario INTEGER NOT NULL,
    idPessoa INTEGER NOT NULL,
    idProduto INTEGER NOT NULL,
    quantidade INTEGER NOT NULL,
    tipo TEXT NOT NULL,
    valorUnitario REAL NOT NULL,
    PRIMARY KEY (id AUTOINCREMENT),
    FOREIGN KEY (idProduto) REFERENCES Produto (id),
    FOREIGN KEY (idUsuario) REFERENCES Usuario (id),
    FOREIGN KEY (idPessoa) REFERENCES Pessoa (id));

```

2º Procedimento | Alimentando a Base

Objetivo da prática: Inserir dados no banco de dados criado no '1º Procedimento', e realizar consultas através de comandos SQL.

Inserção dos dados no BD:

Usuário (cadastro)			
Estrutura			
Dados			
Constraints			
Índices			
Triggers			
DDL			
Exibição em grade			
Visualização do formulário			
Filtrar dados			
Total de linhas carregadas: 2			
id	nome	senha	
1	op1	op1	
2	op2	op2	

Produto (cadastro)				
Estrutura				
Dados				
Constraints				
Índices				
Triggers				
DDL				
Exibição em grade				
Visualização do formulário				
Filtrar dados				
Total de linhas carregadas: 3				
id	nome	quantidade	preco	
1	Banana	100	5	
2	Laranja	500	2	
3	Manga	800	4	

PessoaFisica (cadastro)		
Estrutura		
Dados		
Constraints		
Índices		
Triggers		
DDL		
Exibição em grade		
Visualização do formulário		
Filtrar dados		
Total de linhas carregadas: 1		
id	cpf	
1	11111111111	

PessoaJuridica (cadastro)	
Estrutura	
Dados	
Constraints	
Índices	
Triggers	
DDL	
Exibição em grade	
Visualização do formulário	
Filtrar dados	
Total de linhas carregadas: 1	
id	cnpj
1	22222222222222

Pessoa (cadastro)								
Estrutura								
Dados								
Constraints								
Índices								
Triggers								
DDL								
Exibição em grade								
Visualização do formulário								
Filtrar dados								
Total de linhas carregadas: 2								
id	nome	logradouro	cidade	estad	telefone	email	idPessoaFisica	idPessoaJuridica
1	Joao	Rua 12, casa 3, Quitanda	Riacho do Sul	PA	1111-1111	joao@riacho.com	1	NULL
2	JJC	Rua 11, Centro	Riacho do ...	PA	1212-1212	jic@riacho.com	NULL	1

Movimento (cadastro)

Estrutura

Dados

Constraints

Índices

Triggers

DDL

Exibição em grade

Visualização do formulário

Consultas SQL:

1. Dados completos de pessoas físicas:

Editor SQL 1

cadastro

<

2. Dados completos de pessoas jurídicas:

Editor SQL 1

cadastro

Consulta | Histórico

```

1 SELECT
2   Pessoa.id AS PessoaID,
3   Pessoa.nome AS Nome,
4   Pessoa.logradouro AS Logradouro,
5   Pessoa.cidade AS Cidade,
6   Pessoa.estado AS Estado,
7   Pessoa.telefone AS Telefone,
8   Pessoa.email AS Email,
9   PessoaJuridica.cnpj AS CNPJ
10 FROM
11   Pessoa
12 JOIN
13   PessoaJuridica ON Pessoa.idPessoaJuridica = PessoaJuridica.id;
14

```

Exibição em grade | Visualização do formulário

Total de linhas carregadas: 1

	PessoaID	Nome	Logradouro	Cidade	Estado	Telefone	Email	CNPJ
1	2	JJC	Rua 11, Centro	Riacho do Norte	PA	1212-1212	jjc@riacho.com	22222222222222

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

Editor SQL 1

cadastro

Consulta | Histórico

```

1 SELECT
2   M.id AS MovimentoID,
3   P.nome AS Produto,
4   PP.nome AS Fornecedor,
5   M.quantidade,
6   M.valorUnitario,
7   M.quantidade * M.valorUnitario AS ValorTotal
8 FROM
9   Movimento M
10 JOIN
11   Produto P ON M.idProduto = P.id
12 JOIN
13   Pessoa PP ON M.idPessoa = PP.id
14 WHERE
15   M.tipo = 'E';

```

Exibição em grade | Visualização do formulário

Total de linhas carregadas: 2

	Movimen	Produto	Forneced	quantidac	valorUnit	ValorTota
1	4	Laranja	JJC	15	5	75
2	5	Manga	JJC	20	4	80

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

Editor SQL 1

cadastro

Consulta | Histórico

```

1 SELECT
2     M.id AS MovimentoID,
3     P.nome AS Produto,
4     PP.nome AS Fornecedor,
5     M.quantidade,
6     M.valorUnitario,
7     M.quantidade * M.valorUnitario AS ValorTotal
8 FROM
9     Movimento M
10 JOIN
11     Produto P ON M.idProduto = P.id
12 JOIN
13     Pessoa PP ON M.idPessoa = PP.id
14 WHERE
15     M.tipo = 'S' AND M.tipo LIKE 'S%';
16

```

Exibição em grade | Visualização do formulário

Total de linhas carregadas: 3

	Movimen	Produto	Forneced	quantidac	valorUnit	ValorTota
1	1	Banana	Joao	20	4	80
2	2	Laranja	Joao	15	2	30
3	3	Laranja	Joao	10	3	30

5. Valor total das entradas agrupadas por produto:

Editor SQL 1

cadastro

Consulta | Histórico

```

1 SELECT
2     P.id AS ProdutoID,
3     P.nome AS Produto,
4     SUM(M.quantidade * M.valorUnitario) AS ValorTotalEntradas
5 FROM
6     Produto P
7 JOIN
8     Movimento M ON P.id = M.idProduto
9 WHERE
10    M.tipo = 'E'
11 GROUP BY
12     P.id, P.nome;
13
14

```

Exibição em grade | Visualização do formulário

Total de linhas carregadas: 2

	ProdutoID	Produto	ValorTotalEntradas
1	2	Laranja	75
2	3	Manga	80

6. Valor total das saídas agrupadas por produto:

Editor SQL 1

cadastro

Consulta | Histórico

```

1 SELECT
2     P.id AS ProdutoID,
3     P.nome AS Produto,
4     SUM(M.quantidade * M.valorUnitario) AS ValorTotalSaidas
5 FROM
6     Produto P
7 JOIN
8     Movimento M ON P.id = M.idProduto
9 WHERE
10    M.tipo = 'S' AND M.tipo LIKE 'S%'
11 GROUP BY
12     P.id, P.nome;
13
14

```

Exibição em grade | Visualização do formulário

Total de linhas carregadas: 2

	ProdutoID	Produto	ValorTotalSaidas
1	1	Banana	80
2	2	Laranja	60

7. Operadores que não efetuaram movimentações de entrada (compra):

Editor SQL 1

cadastro

Consulta | Histórico

```

1 SELECT
2     U.id AS UsuarioID,
3     U.nome AS NomeUsuario
4 FROM
5     Usuario U
6 WHERE
7     NOT EXISTS (
8         SELECT 1
9         FROM Movimento M
10        WHERE M.idUsuario = U.id AND M.tipo = 'E'
11    );
12

```

Exibição em grade | Visualização do formulário

Total de linhas carregadas: 1

	UsuarioID	NomeUsuario
1	2	op2

8. Valor total de entrada, agrupado por operador:

Editor SQL 1

cadastro

Consulta | Histórico

```

1 SELECT
2     U.id AS UsuarioID,
3     U.nome AS NomeUsuario,
4     SUM(M.quantidade * M.valorUnitario) AS ValorTotalEntradas
5 FROM
6     Usuario U
7 JOIN
8     Movimento M ON U.id = M.idUsuario
9 WHERE
10    M.tipo = 'E'
11 GROUP BY
12     U.id, U.nome;
13

```

Exibição em grade | Visualização do formulário

Total de linhas carregadas: 1

	UsuarioID	NomeUsuario	ValorTotalEntradas
1	1	op1	155

9. Valor total de saída, agrupado por operador:

Editor SQL 1

cadastro

Consulta | Histórico

```

1 SELECT
2     U.id AS UsuarioID,
3     U.nome AS NomeUsuario,
4     SUM(M.quantidade * M.valorUnitario) AS ValorTotalSaidas
5 FROM
6     Usuario U
7 JOIN
8     Movimento M ON U.id = M.idUsuario
9 WHERE
10    M.tipo = 'S' AND M.tipo LIKE 'S%'
11 GROUP BY
12     U.id, U.nome;
13

```

Exibição em grade | Visualização do formulário

Total de linhas carregadas: 2

	UsuarioID	NomeUsuario	ValorTotalSaidas
1	1	op1	110
2	2	op2	30

10. Valor médio de venda por produto, utilizando média ponderada:

Editor SQL 1

cadastro

Consulta | Histórico

```

1 SELECT
2     P.id AS ProdutoID,
3     P.nome AS Produto,
4     SUM(M.quantidade * M.valorUnitario) / SUM(M.quantidade) AS ValorMedioVenda
5 FROM
6     Produto P
7 JOIN
8     Movimento M ON P.id = M.idProduto
9 WHERE
10    M.tipo = 'S' AND M.tipo LIKE 'S%'
11 GROUP BY
12     P.id, P.nome;
13

```

Exibição em grade | Visualização do formulário

Total de linhas carregadas: 2

	ProdutoID	Produto	ValorMedioVenda
1	1	Banana	4
2	2	Laranja	2.4

Análise e Conclusão:

1. Quais as diferenças no uso de sequence e identity?

Diferenças entre sequence e identity:

Sequence e identity são duas técnicas para gerar números sequenciais em bancos de dados relacionais. O principal ponto de diferença entre as duas técnicas é que a sequence é um objeto de banco de dados, enquanto o identity é uma propriedade de uma coluna.

Sequence:

Uma sequence é um objeto de banco de dados que pode ser usado para gerar números sequenciais. As sequences são independentes de tabelas, o que significa que podem ser usadas por várias tabelas.

Identity:

O identity é uma propriedade de uma coluna que permite que o banco de dados gere números sequenciais automaticamente. O identity está vinculado a uma tabela específica, o que significa que só pode ser usado por essa tabela.

2. Qual a importância das chaves estrangeiras para a consistência do banco?

As chaves estrangeiras também são importantes para a integridade referencial. A integridade referencial é uma propriedade de um banco de dados que garante que as chaves estrangeiras sejam válidas.

Na imagem, a tabela Pedidos tem uma chave estrangeira para a tabela Clientes. Isso significa que o valor da coluna `id_cliente` na tabela Pedidos deve corresponder a um valor da coluna `id` na tabela Clientes.

Se um registro for inserido na tabela Pedidos com um valor inválido para a coluna `id_cliente`, o banco de dados rejeitará a operação. Isso evita que dados inconsistentes sejam inseridos no banco de dados.

Em resumo, as chaves estrangeiras são importantes para a consistência e a integridade referencial do banco de dados. Elas ajudam a garantir que os dados nas tabelas relacionadas sejam consistentes e que as chaves estrangeiras sejam válidas.

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

A álgebra relacional é um conjunto de operações que podem ser usadas para manipular dados em um banco de dados relacional. O SQL é uma linguagem de consulta que pode ser usada para executar operações de álgebra relacional.

Os operadores de seleção, projeção e produto cartesiano são operações básicas da álgebra relacional. Eles são usados para manipular dados dentro de uma única relação.

Os operadores de união, diferença e intersecção são operações comparativas da álgebra relacional. Eles são usados para combinar dados de duas relações.

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