

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,

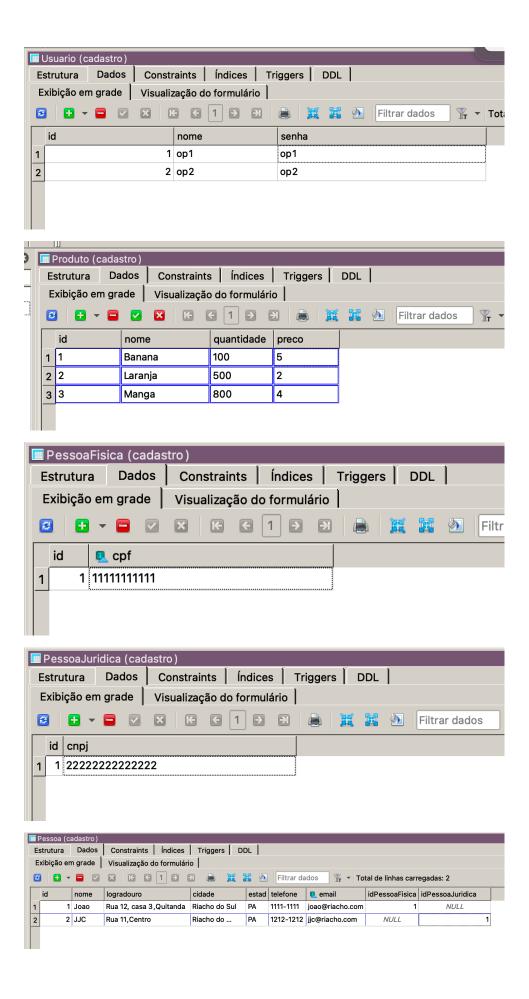
PRIMARY KEY (id AUTOINCREMENT));
```

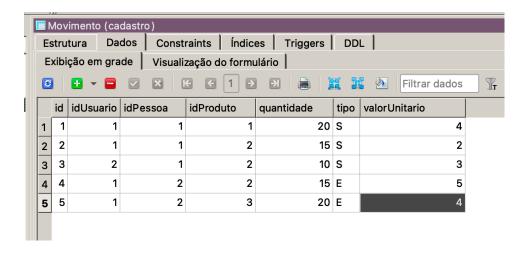
```
CREATE TABLE PessoaFisica (
       id INTEGER NOT NULL,
       cpf TEXT NOT NULL UNIQUE,
       PRIMARY KEY (id AUTOINCREMENT));
CREATE TABLE Pessoa (id INTEGER NOT NULL UNIQUE,
       nome TEXT NOT NULL,
       logradouro TEXT NOT NULL,
       cidade TEXT NOT NULL,
       estado TEXT NOT NULL,
       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.

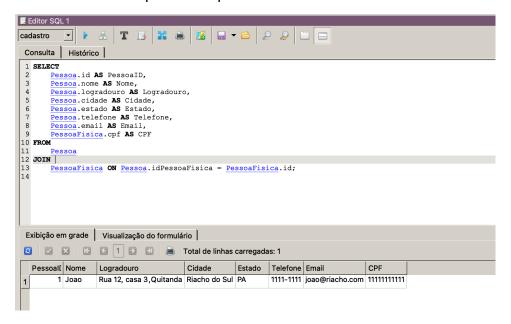
#### Inserção dos dados no BD:



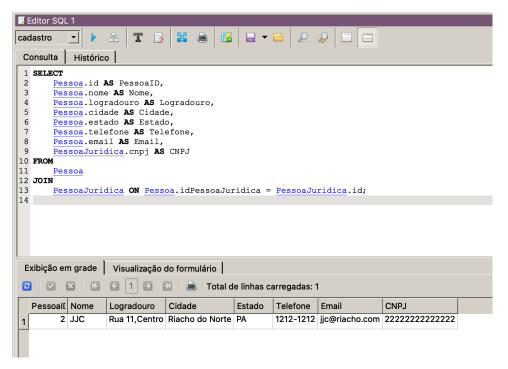


### **Consultas SQL:**

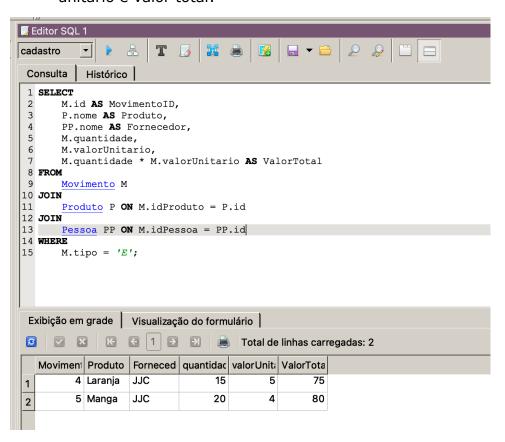
1. Dados completos de pessoas físicas:



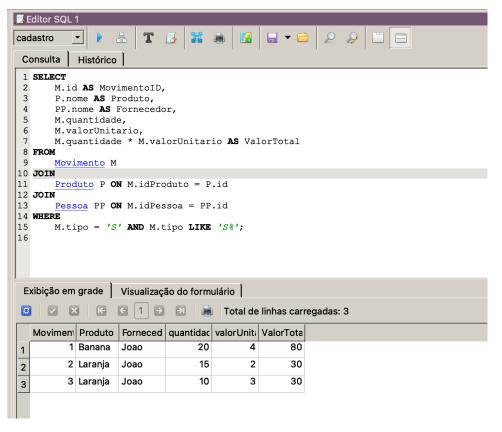
2. Dados completos de pessoas jurídicas:



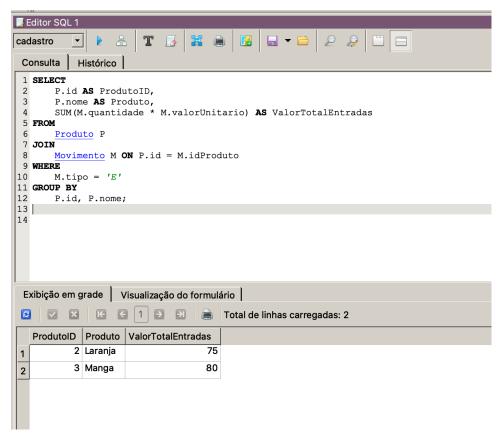
3. Movimentações de entrada, com produto, fornecedor, quantidade, preço unitário e valor total:



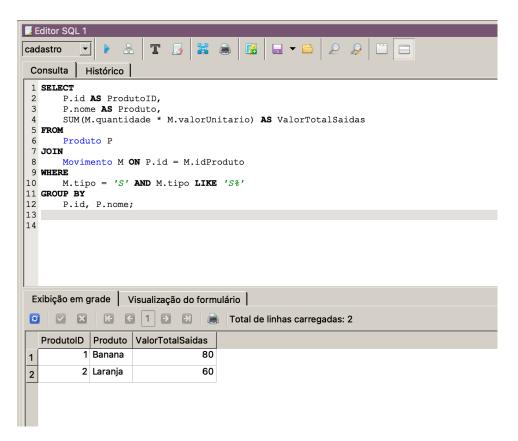
4. Movimentações de saída, com produto, comprador, quantidade, preço unitário e valor total:



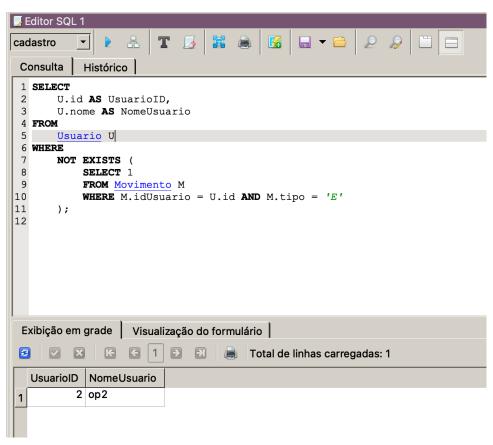
5. Valor total das entradas agrupadas por produto:



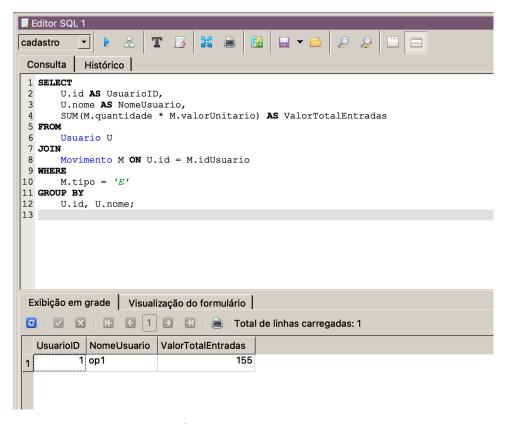
6. Valor total das saídas agrupadas por produto:



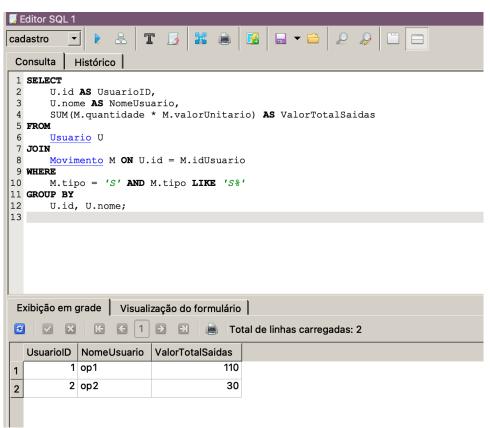
7. Operadores que não efetuaram movimentações de entrada (compra):



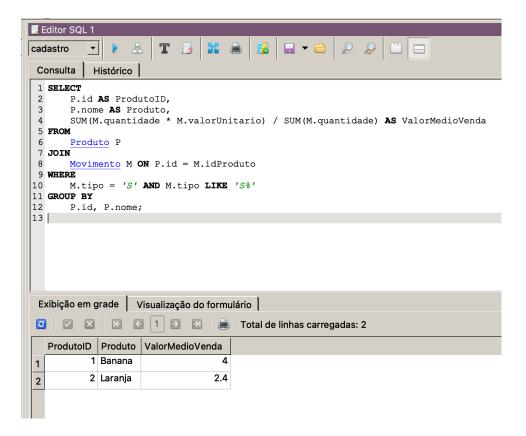
8. Valor total de entrada, agrupado por operador:



9. Valor total de saída, agrupado por operador:



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



#### **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 estrangerias 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?