



FACULDADE ESTACIO
CURSO TECNÓLOGO EM DESENVOLVIMENTO FULL STACK

MISSÃO PRÁTICA – NÍVEL 2 – MUNDO 3

Vamos manter as informações

MISSÃO PRÁTICA – NÍVEL 2 – MUNDO 3

Vamos manter as informações

Trabalho apresentado à disciplina Vamos manter as informações do Curso Tecnólogo em Desenvolvimento Full Stack, período 2024.4 Flex, como requisito parcial do relatório de acompanhamento.

OBJETIVOS

Este relatório apresenta a composição do trabalho proposto para o Nível 1: Vamos manter as informações, o qual está contido no semestre letivo no período 2024, apresentando todos os códigos solicitados, resultados da execução desses códigos e descrição de avaliação sobre o tema abordado, respondendo perguntas propostas pelo tutor.

Palavras-chave: SQL, server, DDL, DML, DBDDesigner, banco de dados, gerenciador SQL.

SUMÁRIO

1	CÓDIGOS	9
2	ANÁLISE	12

1 CÓDIGOS

O código abaixo apresenta a criação da estrutura do banco de dados e seus relacionamentos. Escrito e teve o modelo desenhado no programa DBDesign Fork, como mostra imagem abaixo.

```
USE model;
```

```
CREATE SEQUENCE sequencia
  START WITH 1 -- Inicia a sequência a partir de 1
  INCREMENT BY 1 -- Incrementa de 1 em 1
  MINVALUE 1 -- Valor mínimo permitido (1)
  NO CYCLE; -- Não reinicia a sequência quando atingir o valor máximo
```

```
CREATE TABLE Produtos (
  id_produto INTEGER NOT NULL ,
  nome TEXT ,
  quantidade TEXT ,
  preco_venda TEXT ,
  PRIMARY KEY(id_produto));
```

```
CREATE TABLE Usuarios (
  id_usuario INTEGER NOT NULL ,
  nome TEXT ,
  email TEXT ,
  senha TEXT ,
  data_cadastro DATE ,
  PRIMARY KEY(id_usuario));
```

```
CREATE TABLE Pessoa_Juridica (
  id_pessoa_juridica INTEGER NOT NULL ,
  nome VARCHAR ,
  cnpj INTEGER ,
  endereco VARCHAR ,
  telefone VARCHAR ,
  email VARCHAR ,
  PRIMARY KEY(id_pessoa_juridica));
```

```
CREATE TABLE Pessoa_Fisica (
  id_pessoa_fisica INTEGER NOT NULL ,
  nome VARCHAR ,
  cpf INTEGER ,
  endereco VARCHAR ,
  telefone VARCHAR ,
  email VARCHAR ,
  PRIMARY KEY(id_pessoa_fisica));
```

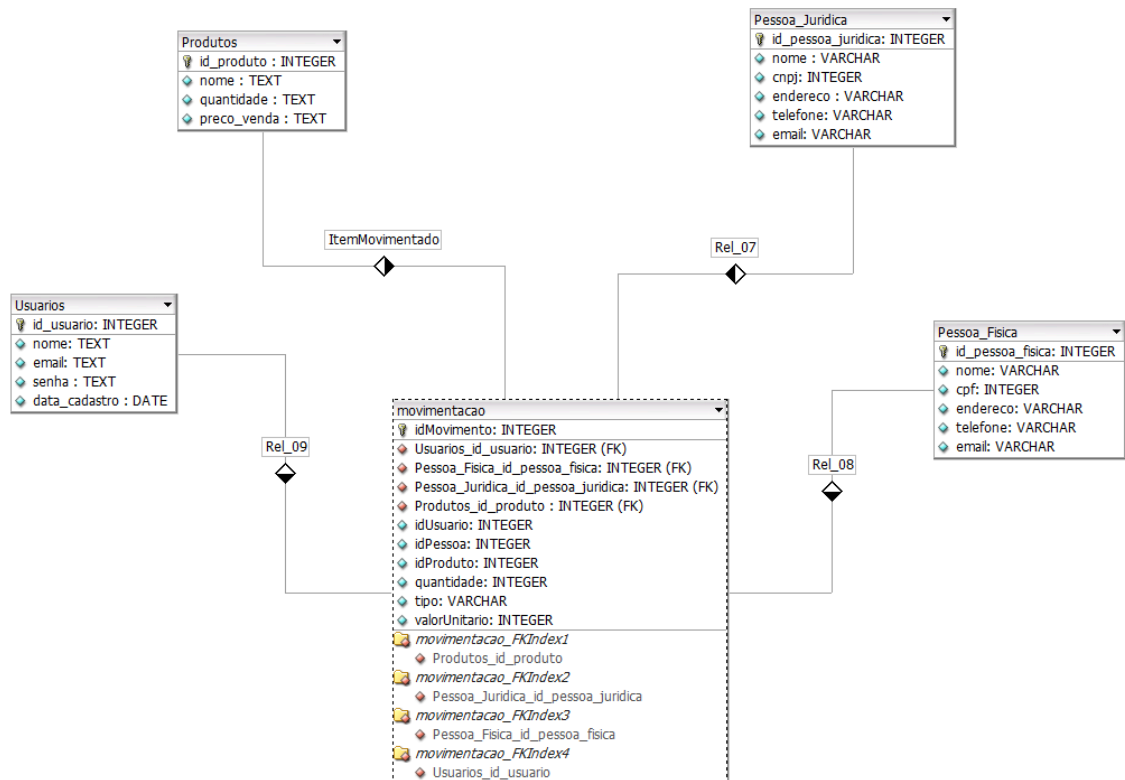
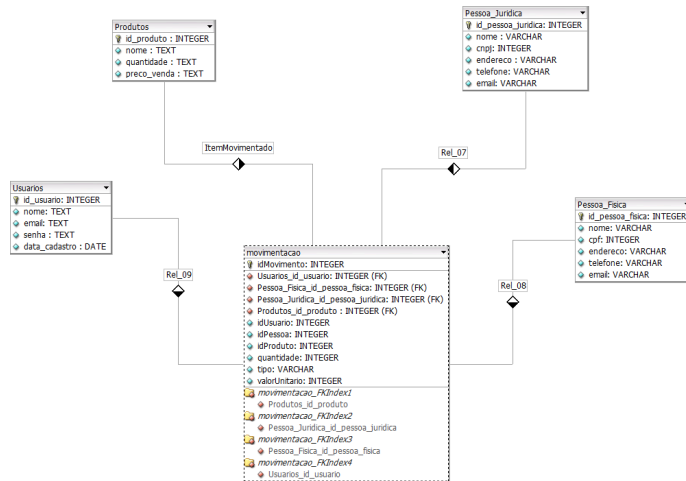
```

CREATE TABLE movimentacao (
  idMovimento INTEGER NOT NULL ,
  Usuarios_id_usuario INTEGER NOT NULL ,
  Pessoa_Fisica_id_pessoa_fisica INTEGER NOT NULL ,
  Pessoa_Juridica_id_pessoa_juridica INTEGER NOT NULL ,
  Produtos_id_produto INTEGER NOT NULL ,
  idUsuario INTEGER ,
  idPessoa INTEGER ,
  idProduto INTEGER ,
  quantidade INTEGER ,
  tipo VARCHAR ,
  valorUnitario INTEGER ,
PRIMARY KEY(idMovimento) ,
FOREIGN KEY(Produtos_id_produto )
  REFERENCES Produtos(id_produto ),
FOREIGN KEY(Pessoa_Juridica_id_pessoa_juridica)
  REFERENCES Pessoa_Juridica(id_pessoa_juridica),
FOREIGN KEY(Pessoa_Fisica_id_pessoa_fisica)
  REFERENCES Pessoa_Fisica(id_pessoa_fisica),
FOREIGN KEY(Usuarios_id_usuario)
  REFERENCES Usuarios(id_usuario));

CREATE INDEX movimentacao_FKIndex1 ON movimentacao (Produtos_id_produto );
CREATE INDEX movimentacao_FKIndex2 ON movimentacao
(Pessoa_Juridica_id_pessoa_juridica);
CREATE INDEX movimentacao_FKIndex3 ON movimentacao (Pessoa_Fisica_id_pessoa_fisica);
CREATE INDEX movimentacao_FKIndex4 ON movimentacao (Usuarios_id_usuario);

CREATE INDEX IFK_ItemMovimentado ON movimentacao (Produtos_id_produto );
CREATE INDEX IFK_Rel_07 ON movimentacao (Pessoa_Juridica_id_pessoa_juridica);
CREATE INDEX IFK_Rel_08 ON movimentacao (Pessoa_Fisica_id_pessoa_fisica);
CREATE INDEX IFK_Rel_09 ON movimentacao (Usuarios_id_usuario);

```



3 ANÁLISE

Como são implementadas as diferentes cardinalidades, basicamente 1x1, 1xN ou NxN, em um banco de dados relacional?

A implementação das cardinalidades em um banco de dados relacional é feita por meio de relacionamentos entre tabelas, usando chaves primárias (PK) e chaves estrangeiras (FK).

Um para um onde, para um onde cada registro de uma tabela está relacionada ao registro de outra (1x1). Um para muitos, onde o registro de uma tabela pode estar relacionado a muitos registros em outra tabela (1xN). Muitos para muitos, onde muitos registros de uma tabela podem estar relacionados a muitos registros de outra tabela (NxN).

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

Essa ferramenta centraliza diversas funcionalidades de gerenciamento, diagnóstico e automação em uma única ferramenta, reduzindo o tempo necessário para tarefas complexas e melhorando a eficiência. Sua combinação de interface gráfica, suporte a scripts e ferramentas avançadas é indispensável para quem trabalha com o banco de dados SQL.