

Campos : Polo Barbosa II - Marília - SP

Curso: Desenvolvedor Full Stack

Turma: 2023.2

Integrante : Rafael Leal Altero

1º Procedimento:

Criando um banco de dados para um sistema de comércio eletrônico

Objetivo da Prática:

- 1 Criar um modelo de dados para um sistema de comércio eletrônico Utilizando o DER (Diagrama Entidade-Relacionamento);
- 2 Criar um banco de dados para o sistema de comércio eletrônico usando o SQL Server Management Studio;
- 3 Definir uma SEQUENCE para Pessoa
- 4 Definir um IDENTITY para geração dos identificadores para todas as tabelas que tem o id_* PRIMARY KEY.

segue o LINK Codigos

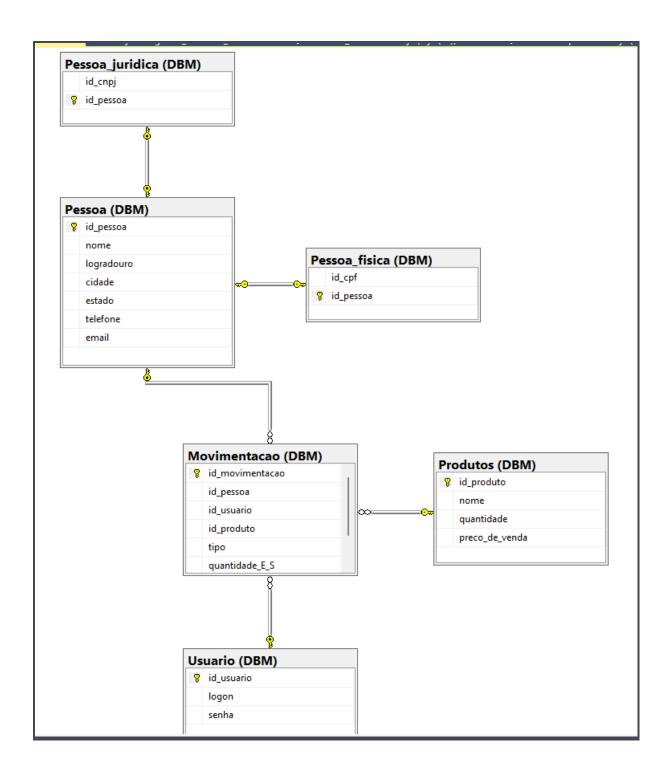
Códigos solicitados no roteiro de aula:

Link GetHub:

https://github.com/Rafa1a/DB_sql/tree/main/SQL

Resultados da execução dos códigos:

DER:



CREATE TABLES:

```
- 1 ×
                                                                              SQLcreattable_nv2.s...AEL.Loja (loja (67))
Pesquisador de Objetos
                                                rafael.Loja - Facu_nv2_Diagrama*
                                                     USE Loja
Conectar ▼ ¥ ¥ ■ ▼ 🖒 🦇
                                                     go

☐ RAFAEL (SQL Server 16.0.1000.6 - Ioja)

                                                     CREATE SCHEMA DBM

☐ ■ Bancos de Dados

     -sequencia
     🔢 📕 Instantâneos do Banco de Dados
                                                     CREATE SEQUENCE sequencia

■ AULA_01

⊕ CRM_BD

                                                     INCREMENT BY 1
     🖃 🗑 Loja
                                                     MAXVALUE 100
                                                     MINVALUE 1:
       🖃 📕 Diagramas de Banco de Dados
                                                     DROP SEQUENCE sequencia;
            탭 loja.Facu_nv2_Diagrama
                                                     --Pessoa
       Tabelas
                                                     CREATE TABLE DBM.Pessoa
         🖽 📕 Tabelas do Sistema
                                                     (id_pessoa int PRIMARY KEY,
          nome varchar(50) not null,
                                                     logradouro varchar(50) not null,
         cidade varchar(20) not null,
          표 📕 Tabelas de Grafo
                                                     estado varchar(3) not null,
          telefone varchar(11),
          email varchar(35)

    ⊞ DBM.Pessoa_juridica

          --Pessoa Fisica e Juridica
          □ CREATE TABLE DBM.Pessoa_fisica
         Tabelas do Razão Descartadas
                                                     (id_cpf int not null,
       Exibições
                                                     id_pessoa int not null
       CONSTRAINT PK_Pessoa_fisica PRIMARY KEY (id_pessoa),
                                                                           FOREIGN KEY (id_pessoa)
       Sinônimos
                                                     CONSTRAINT FK_Pessoa_f
                                                     REFERENCES DBM.Pessoa
       Programação
       Repositório de Consultas
       □CREATE TABLE DBM.Pessoa_juridica
       Armazenamento
                                                     (id_cnpj int not null,
        🔳 Segurança
                                                     id pessoa int not null.
                                                     CONSTRAINT PK_Pessoa_juridica PRIMARY KEY (id_pessoa),
     # estagenda
                                                     CONSTRAINT FK_Pessoa_j
                                                                           FOREIGN KEY (id pessoa)
  Segurança
                                                     REFERENCES DBM. Pessoa
  Objetos de Servidor

    Alta Disponibilidade Always On

  --Produtos e Usuarios
                                                    □ CREATE TABLE DBM. Produtos
  (id_produto int IDENTITY(1,1) PRIMARY KEY,
  nome varchar(50) not null,
                                                     quantidade int not null
                                                     preco_de_venda decimal(18,2) not null
                                                   □CREATE TABLE DBM.Usuario
                                                    (id usuacio int TDENTTTV/1 1) DOTMADV VEV

    Mensagens
```

Análise e Conclusão:

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

As diferentes cardinalidades são implementadas em um banco de dados relacional usando chaves estrangeiras (FOREIGN). Uma chave estrangeira é uma coluna ou um conjunto de colunas em uma tabela que faz referência a uma chave primária em outra tabela.

b)Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais

Pelo o'que foi estudado e o que entendi o mais próximo de herança seria as foreign ou seja chave estrangeiras passadas para outras tabelas "filhas" de suas tabelas "pais"

Um exemplo seria: A tabela "Pessoa" é a tabela base que contém os atributos comuns a todas as pessoas. As tabelas "Pessoa Física" e "Pessoa Jurídica" são tabelas especializadas que herdam o atributo "id_pessoa" da tabela base. Através dessa herança, podemos estabelecer relacionamentos com outras tabelas e representar diferentes tipos de pessoas no banco de dados.

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

Bom existem vários fatores mas os principais que mais impactaram na minha opinião foram a :

- Uma interface gráfica do usuário (GUI) intuitiva que facilita a navegação pelos bancos de dados e as suas tabelas.
- DER com a possibilidade de criar e gerenciar os diagramas através das próprias tabelas podendo ainda criar tabelas utilizando os diagramas.
- Administrar bancos de dados, como a criação e gerenciamento de usuários, o gerenciamento de backups, gerenciamento de estatísticas e o gerenciamento e criação de índices.
- Editor SQL Avançado: O SSMS inclui um editor de código SQL.