



## Faculdade Estácio - Polo Barra

Curso : Desenvolvimento Full Stack

Disciplina : Vamos manter as informações

NÚMERO DA Turma : RPG0015 - Semestre Letivo: 3

**Integrante: Lucia Maria de Lima Martins**

Repositório:

<https://github.com/luciamartins/MissaoPraticaNivel2Mundo3.git>

### Objetivo da Prática :

1. Identificar os requisitos de um Sistema e transformá-los no modelo adequado.
2. Utilizar ferramentas de modelagem para bases de dados relacionais.
3. Explorar a sintaxe SQL na criação das estruturas do banco (DDL).
4. Explorar a sintaxe SQL na consulta e manipulação de banco (DML).
5. No final do exercício, o aluno terá vivenciado a experiência de modelar a base de dados para um Sistema simples, além de implementá-la, através da sintaxe SQL, na plataforma do SQL Serve.

Primeiro Procedimento :

```
USE Loja;  
GO
```

```
CREATE SEQUENCE orderPessoa  
as INT  
START WITH 1  
INCREMENT BY 1;
```

```
CREATE TABLE Pessoa (  
    idPessoa INTEGER NOT NULL,  
    nome VARCHAR(255),  
    logradouro VARCHAR(255),  
    cidade VARCHAR(255),  
    estado CHAR(2),  
    telefone VARCHAR(15),  
    email VARCHAR(255),  
    PRIMARY KEY(idPessoa));  
GO
```

```
CREATE TABLE PessoaFisica (  
    Pessoa_idPessoa INTEGER NOT NULL,  
    cpf VARCHAR(11) NOT NULL,  
    PRIMARY KEY(Pessoa_idPessoa),  
    FOREIGN KEY(Pessoa_idPessoa) REFERENCES Pessoa(idPessoa)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
);  
GO
```

```
CREATE TABLE PessoaJuridica (  
    Pessoa_idPessoa INTEGER NOT NULL,  
    cnpj VARCHAR(14) NOT NULL,  
    PRIMARY KEY(Pessoa_idPessoa),
```

```
FOREIGN KEY(Pessoa_idPessoa) REFERENCES Pessoa(idPessoa)
ON UPDATE CASCADE
ON DELETE CASCADE
);
GO
```

```
CREATE TABLE Usuario (
    idUsuario INTEGER NOT NULL IDENTITY,
    nome VARCHAR(20) NOT NULL,
    senha VARCHAR(20) NOT NULL,
    PRIMARY KEY(idUsuario));
GO
```

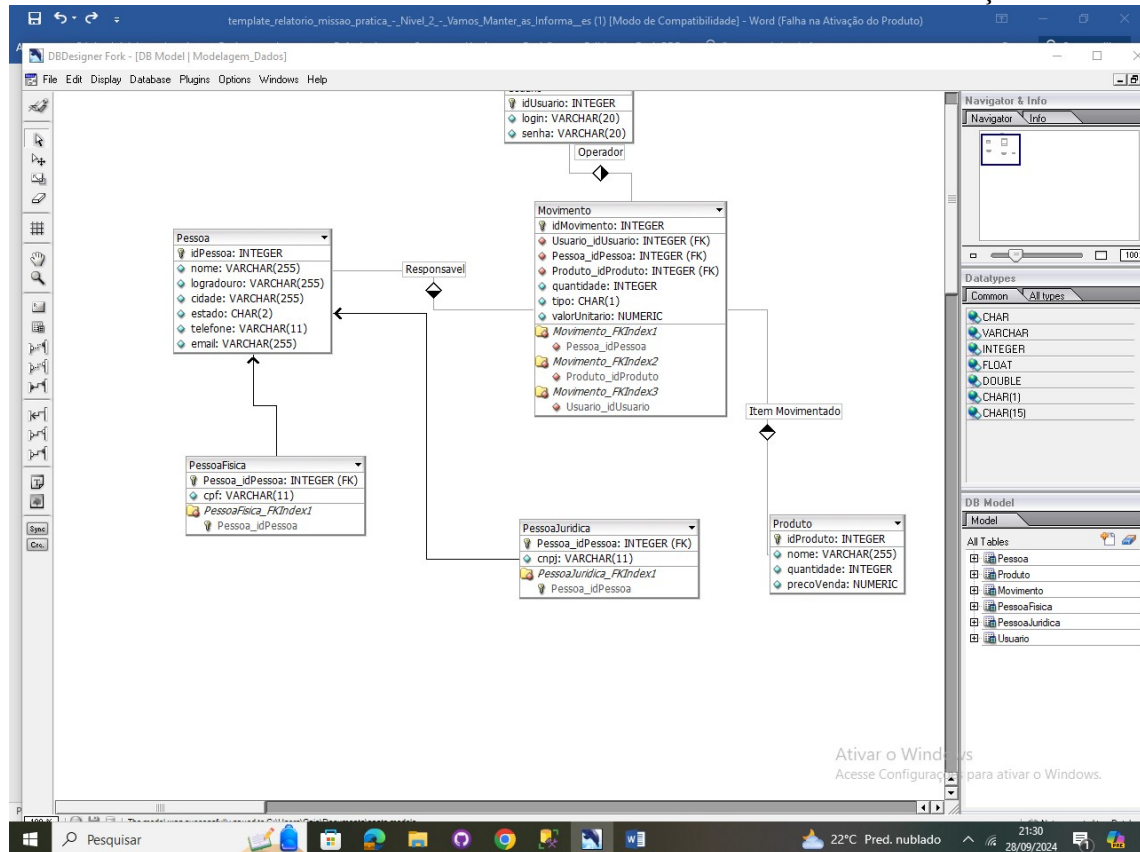
```
CREATE TABLE Produto (
    idProduto INTEGER NOT NULL IDENTITY,
    nome VARCHAR(255),
    quantidade INTEGER,
    precoVenda NUMERIC,
    PRIMARY KEY(idProduto));
GO
```

```
CREATE TABLE Movimento (
    idMovimento INTEGER NOT NULL IDENTITY,
    Pessoa_idPessoa INTEGER NOT NULL,
    Usuario_idUsuario INTEGER NOT NULL,
    Produto_idProduto INTEGER NOT NULL,
    quantidade INTEGER,
    tipo CHAR(1),
    valorUnitario NUMERIC,
    PRIMARY KEY(idMovimento),
    FOREIGN KEY(Produto_idProduto)
        REFERENCES Produto(idProduto)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    FOREIGN KEY(Usuario_idUsuario)
        REFERENCES Usuario(idUsuario)
    ON UPDATE CASCADE
    ON DELETE CASCADE,
    FOREIGN KEY(Pessoa_idPessoa)
        REFERENCES Pessoa(idPessoa)
    ON UPDATE CASCADE
    ON DELETE CASCADE);
GO
```

# RESULTADO

# DA

# EXECUÇÃO



Documents - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

loja

Executar

Pesquisador de Objetos

Conectar

Lucia\MSSQLSERVER01 (SQL Server 16.0.1125.1 - loja)

Lucia\MSSQLSERVER01 (SQL Server 16.0.1125.1 - sa)

Bancos de Dados

Bancos de Dados do Sistema

Instantâneos do Banco de Dados

AULA\_03

CriandoBancoDeDados

TutorialDB

loja

Diagramas de Banco de Dados

Tabelas

Tabelas do Sistema

FileTables

Tabelas Externas

Tabelas de Grafo

dbo.Movimento

dbo.Pessoa

dbo.PessoaFisica

dbo.PessoaJuridica

dbo.Produto

dbo.Usuario

Tabelas do Ráio Descartadas

Exibições

Recursos Externos

Sinônimos

Programação

Repositório de Consultas

Service Broker

Armazenamento

Segurança

Segurança

Objetos de Servidor

Replicação

Alta Disponibilidade Always On

Gerenciamento

Catálogos do Integration Services

SQL Server Agent

XEvent Profiler

Select\_script\03.sql...RVER01.loja (sa (67))

Create\_script.sql...RVER01.loja (sa (63))

```
USE Loja;
GO

CREATE SEQUENCE orderPessoa
AS INT
START WITH 1
INCREMENT BY 1;

CREATE TABLE Pessoa (
    idPessoa INTEGER NOT NULL,
    nome VARCHAR(255),
    logradouro VARCHAR(255),
    cidade VARCHAR(255),
    estado CHAR(2),
    telefone VARCHAR(15),
    email VARCHAR(255),
    PRIMARY KEY(idPessoa));
GO

CREATE TABLE PessoaFisica (
    Pessoa_idPessoa INTEGER NOT NULL,
    cpf VARCHAR(11) NOT NULL,
    PRIMARY KEY(Pessoa_idPessoa));
GO
```

Mensagens

Comandos concluídos com êxito.

Horário de conclusão: 2024-09-29T15:46:23.9961226-03:00

Consultas executadas com êxito.

Lucia\MSSQLSERVER01 (16.0 RTM) sa (63) loja | 00:00:04 | 0 linhas

## **Análise e Conclusão:**

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

**Resposta:** A cardinalidade 1x1 (um-para-um) em um banco de dados relacional se apresenta quando uma entidade se relaciona com apenas uma entrada de outra entidade. Por exemplo, em um hospital, cada médico (1) tem um (e somente um) crachá (1), e cada crachá pertence a um (e somente um) médico.

A cardinalidade 1xN (um-para-muitos) significa que uma entrada se relaciona com muitas ocorrências de outra tabela, mas que a recíproca não é verdadeira, por exemplo, um funcionário (1) possui vários dependentes (N), mas cada dependente pertence a apenas um funcionário.

Para a cardinalidade NxN (muitos-para-muitos) entende-se que podem existir múltiplas entradas associadas a múltiplas ocorrências em outra tabela. Por exemplo, um aluno (N) pode cursar várias disciplinas (N), e uma disciplina pode ter vários alunos.

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

**Resposta:** Em bancos de dados relacionais, representamos o uso de herança através da especialização/generalização.

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

**Resposta:** Ele permite rodar queries diretamente através de um console além de proporcionar funcionalidades como views, stored procedures, triggers, functions e índices.

## 2º Procedimento | Alimentando a Base

```
INSERT INTO Produto(nome, quantidade, precoVenda)
VALUES('Banana', 100, 5.00),
('Laranja', 500, 2.00),
('Manga', 800, 4.00);
```

```
INSERT INTO Pessoa(idPessoa,nome,logradouro,cidade,estado,telefone,email)
VALUES(NEXT VALUE FOR orderPessoa, 'Luiz Carlos','Rua das Flores, 123', 'São
Paulo', 'SP', '11987654321', 'luiz.carlos@example.com'),
(NEXT VALUE FOR orderPessoa,'Maria Luiza', 'Avenida Paulista, 456', 'São Paulo',
'SP', '11987654322', 'maria.luiza@example.com'),
(NEXT VALUE FOR orderPessoa,'Pedro Antonio', 'Praça da Sé, 789', 'Rio de Janeiro',
'RJ', '21987654323', 'pedro.antonio@example.com'),
(NEXT VALUE FOR orderPessoa,'Crala Mara', 'Rua XV de Novembro, 101', 'Curitiba',
'PR', '41987654324', 'carla.mara@example.com'),
(NEXT VALUE FOR orderPessoa,'Carlos Nascimento', 'Avenida Brasil, 202', 'Belo
Horizonte', 'MG', '31987654325', 'carlos.nascimento@example.com');
```

```
INSERT INTO PessoaFisica(Pessoa_idPessoa, cpf)
VALUES(1, '12345678900'),
(2, '23456789011'),
(3, '34567890122');
```

```
INSERT INTO PessoaJuridica(Pessoa_idPessoa,cnpj)
VALUES(4, '98765432000188'),
(5, '12345678000199');
```

```
INSERT INTO Movimento(Pessoa_idPessoa, Usuario_idUsuario, Produto_idProduto,
quantidade, tipo, valorUnitario)
VALUES(1,1,1,5,'S',4.00),
(3,2,3,8,'S',3.00),
(2,1,1,35,'E',4.00),
(5,2,2,25,'E',6.00);
```

---

```
--(a)
SELECT p.*, pf.cpf
FROM Pessoa p
INNER JOIN PessoaFisica pf ON p.idPessoa = pf.Pessoa_idPessoa;
```

```
--(b)
SELECT p.*, pj.cnpj
FROM Pessoa p
INNER JOIN PessoaJuridica pj ON p.idPessoa = pj.Pessoa_idPessoa;
```

```
--(c)
SELECT m.*, p.nome as fornecedor, pr.nome as Produto, m.quantidade,
m.valorUnitario, (m.quantidade * m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON p.idPessoa = m.Pessoa_idPessoa
INNER JOIN Produto pr ON pr.idProduto = m.Produto_idProduto
WHERE m.tipo = 'E';
```

```
--(d)
SELECT m.*, p.nome as comprador, pr.nome as Produto, m.quantidade,
m.valorUnitario, (m.quantidade * m.valorUnitario) as total
FROM Movimento m
INNER JOIN Pessoa p ON m.Pessoa_idPessoa = p.idPessoa
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'S';
```

```

--(e)
SELECT pr.nome, SUM(m.quantidade * m.valorUnitario) as valor_total
FROM Movimento m
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'E'
GROUP BY pr.nome;

--(f)
SELECT pr.nome, SUM(m.quantidade * m.valorUnitario) as valor_total
FROM Movimento m
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'S'
GROUP BY pr.nome;

--(g)
SELECT u.*
FROM Usuario u
LEFT JOIN Movimento m ON u.idUsuario = m.Usuario_idUsuario AND m.tipo = 'E'
WHERE m.idMovimento IS NULL;

--(h)
SELECT u.nome, SUM(m.quantidade * m.valorUnitario) as valor_total
FROM Movimento m
INNER JOIN Usuario u ON m.Usuario_idUsuario = u.idUsuario
WHERE m.tipo = 'E'
GROUP BY u.nome;

--(i)
SELECT u.nome, SUM(m.quantidade * m.valorUnitario) as valor_total
FROM Movimento m
INNER JOIN Usuario u ON m.Usuario_idUsuario = u.idUsuario
WHERE m.tipo = 'S'
GROUP BY u.nome;

--(j)
SELECT pr.nome, SUM(m.quantidade * m.valorUnitario) / SUM(m.quantidade) as media
FROM Movimento m
INNER JOIN Produto pr ON m.Produto_idProduto = pr.IdProduto
WHERE m.tipo = 'S'
GROUP BY pr.nome;

```

Microsoft SQL Server Management Studio interface showing the execution of a script named 'Create\_script01.sql' in the 'loja' database. The script defines a sequence 'orderPessoa' and two tables: 'Pessoa' and 'PessoaFisica'.

**Script Content:**

```
USE loja;
GO

CREATE SEQUENCE orderPessoa
AS INT
START WITH 1
INCREMENT BY 1;

CREATE TABLE Pessoa (
    idPessoa INTEGER NOT NULL,
    nome VARCHAR(255),
    logradouro VARCHAR(255),
    cidade VARCHAR(255),
    estado CHAR(2),
    telefone VARCHAR(15),
    email VARCHAR(255),
    PRIMARY KEY(idPessoa));
GO

CREATE TABLE PessoaFisica (
    Pessoa_idPessoa INTEGER NOT NULL,
    cpf VARCHAR(11) NOT NULL,
    PRIMARY KEY(Pessoa_idPessoa),
```

**Results:**

| idProduto | nome    | quantidade | precoVenda |
|-----------|---------|------------|------------|
| 1         | Banana  | 100        | 5          |
| 2         | Laranja | 500        | 2          |
| 3         | Manga   | 800        | 4          |

Consultas concluídas com erros. | LUCIA\MSSQLSERVER01 (16.0 RTM) | loja (55) | loja | 00:00:00 | 3 linhas

Microsoft SQL Server Management Studio interface showing the execution of a script named 'INSERT\_script02.sql' in the 'loja' database. The script inserts data into 'Usuario', 'Produto', 'Pessoa', 'PessoaFisica', and 'PessoaJuridica' tables.

**Script Content:**

```
INSERT INTO Usuario(nome, senha)
VALUES('op1','op1'),
('op2','op2');

INSERT INTO Produto(nome, quantidade, precoVenda)
VALUES('Banana', 100, 5.00),
('Laranja', 500, 2.00),
('Manga', 800, 4.00);

INSERT INTO Pessoa(idPessoa,nome,logradouro,cidade,estado,telefone,email)
VALUES(NEXT VALUE FOR orderPessoa, 'Luiz Carlos','Rua das Flores, 123', 'São P
(NEXT VALUE FOR orderPessoa,'Maria Luiza', 'Avenida Paulista, 456', 'São Paulo
(NEXT VALUE FOR orderPessoa,'Pedro Antonio', 'Praça da Sé, 789', 'Rio de Janei
(NEXT VALUE FOR orderPessoa,'Crala Mara', 'Rua XV de Novembro, 101', 'Curitiba
(NEXT VALUE FOR orderPessoa,'Carlos Nascimento', 'Avenida Brasil, 202', 'Beio

INSERT INTO PessoaFisica(Pessoa_idPessoa, cpf)
VALUES(1, '12345678900'),
(2, '23456789011'),
(3, '34567890122');

INSERT INTO PessoaJuridica(Pessoa_idPessoa, cnpj)
VALUES(4, '987654321001RR');
```

**Results:**

| idUsuario | nome | senha |
|-----------|------|-------|
| 1         | op1  | op1   |
| 2         | op2  | op2   |

Consultas executadas com êxito. | LUCIA\MSSQLSERVER01 (16.0 RTM) | loja (57) | loja | 00:00:00 | 2 linhas



select\_script03.sql - LUCIA\MSSQLSERVER01.loja (loja (58)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

loja

Pesquisador de Objetos

- Conectar
- LUCIA\MSSQLSERVER01 (SQL Server 16.0.1125.1 - loja)
  - Bancos de Dados
    - Bancos de Dados do Sistema
    - Instantâneos do Banco de Dados
    - AULA\_03
    - casa
    - CriandoBancoDeDados
    - loja
    - TutorialDB
  - Segurança
    - Objetos de Servidor
    - Replicação
    - Alta Disponibilidade Always On
    - Gerenciamento
    - Catálogos do Integration Services
    - XEvent Profiler

INSERT\_script02.sql...ER01.loja (loja (57))

```
--(a)
SELECT p.*, pf.cpf
FROM Pessoa p
INNER JOIN PessoaFisica pf ON p.idPessoa = pf.Pessoa_idPessoa;

--(b)
SELECT p.*, pj.cnpj
FROM Pessoa p
INNER JOIN PessoaJuridica pj ON p.idPessoa = pj.Pessoa_idPessoa;

--(c)
SELECT m.*, p.nome as fornecedor, pr.nome as Produto, m.quantidade, m.valorUnitario
FROM Movimento m
INNER JOIN Pessoa p ON p.idPessoa = m.Pessoa_idPessoa
INNER JOIN Produto pr ON pr.idProduto = m.Produto_idProduto
WHERE m.tipo = 'E';

--(d)
SELECT m.*, p.nome as comprador, pr.nome as Produto, m.quantidade, m.valorUnitario
FROM Movimento m
INNER JOIN Pessoa p ON m.Pessoa_idPessoa = p.idPessoa
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'S';
```

100 %

Resultados Mensagens

| idPessoa | nome        | logradouro            | cidade    | estado | telefone    | email          |
|----------|-------------|-----------------------|-----------|--------|-------------|----------------|
| 1        | Luiz Carlos | Rua das Flores, 123   | São Paulo | SP     | 11987654321 | luiz.carlos@ex |
| 2        | Maria Luiza | Avenida Paulista, 456 | São Paulo | SP     | 11987654322 | maria.luiza@ex |

| idPessoa | nome       | logradouro              | cidade   | estado | telefone    | email |
|----------|------------|-------------------------|----------|--------|-------------|-------|
| 4        | Crila Mara | Rua XV de Novembro, 101 | Curitiba | PR     | 41987654324 | carla |

| idMovimento | Pessoa_idPessoa | Usuario_idUsuario | Produto_idProduto | quantidade | tipo | valorUnitario |
|-------------|-----------------|-------------------|-------------------|------------|------|---------------|
| 3           | 2               | 1                 | 1                 | 35         | E    | 4             |

| idMovimento | Pessoa_idPessoa | Usuario_idUsuario | Produto_idProduto | quantidade | tipo | valorUnitario |
|-------------|-----------------|-------------------|-------------------|------------|------|---------------|
| 1           | 1               | 1                 | 1                 | 5          | S    | 4             |

| nome    | valor_total |
|---------|-------------|
| Banana  | 140         |
| Laranja | 150         |

Ativar o Windows

Consultas executadas com sucesso. LUCIA\MSSQLSERVER01 (16.0 RTM) loja (58) loja 00:00:00 24 linhas

Documents - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

loja

Pesquisador de Objetos

- Conectar
- LUCIA\MSSQLSERVER01 (SQL Server 16.0.1125.1 - loja)
- LUCIA\MSSQLSERVER01 (SQL Server 16.0.1125.1 - sa)
  - Bancos de Dados
    - Bancos de Dados do Sistema
    - Instantâneos do Banco de Dados
    - AULA\_03
    - CriandoBancoDeDados
    - TutorialDB
    - loja
      - Diagramas de Banco de Dados
      - Tabelas
        - Tabelas do Sistema
        - FileTables
        - Tabelas Externas
        - Tabelas de Grão
        - dbo.Movimento
        - dbo.Pessoa
        - dbo.PessoaFisica
        - dbo.PessoaJuridica
        - dbo.Produto
        - dbo.Usuario
      - Tabelas do Razoão Descartadas
      - Exibições
      - Recursos Externos
      - Sinônimos
      - Programação
      - Repositório de Consultas
      - Service Broker
      - Armazenamento
      - Segurança
    - Segurança
      - Objetos de Servidor
      - Replicação
      - Alta Disponibilidade Always On
      - Gerenciamento
      - Catálogos do Integration Services
      - SQL Server Agent
      - XEvent Profiler

Select\_script03.sql...RVER01.loja (sa (67))

```
--(f)
SELECT pr.nome, SUM(m.quantidade * m.valorUnitario) as valor_total
FROM Movimento m
INNER JOIN Produto pr ON m.Produto_idProduto = pr.idProduto
WHERE m.tipo = 'S'
GROUP BY pr.nome;

--(g)
SELECT u.*
FROM Usuario u
LEFT JOIN Movimento m ON u.idUsuario = m.Usuario_idUsuario AND m.tipo = 'E'
WHERE m.idMovimento IS NULL;

--(h)
SELECT u.nome, SUM(m.quantidade * m.valorUnitario) as valor_total
FROM Movimento m
INNER JOIN Usuario u ON m.Usuario_idUsuario = u.idUsuario
WHERE m.tipo = 'E'
GROUP BY u.nome;

--(i)
SELECT u.nome, SUM(m.quantidade * m.valorUnitario) as valor_total
```

100 %

Resultados Mensagens

| nome   | valor_total |
|--------|-------------|
| Banana | 20          |
| Manga  | 24          |

| idUsuario | nome | senha |
|-----------|------|-------|
| 1         | op1  | 140   |
| 2         | op2  | 150   |

| nome | valor_total |
|------|-------------|
| op1  | 20          |
| op2  | 24          |

| nome   | media    |
|--------|----------|
| Banana | 4.000000 |
| Manga  | 3.000000 |

Ativar o Windows

Consultas executadas com sucesso. LUCIA\MSSQLSERVER01 (16.0 RTM) sa (67) loja 00:00:04 19 linhas





SQLQuery1.sql - LUCIA\MSSQLSERVER01.loja (loja (61)) - Microsoft SQL Server Management Studio

Arquivo Editar Exibir Consulta Projeto Ferramentas Janela Ajuda

loja

Conectar -

## **Análise e Conclusão:**

### **a) Quais as diferenças no uso de sequence e identity?**

**Resposta:** A Identity é uma propriedade de uma coluna de uma tabela, portanto ela só pode ser usada na tabela em que foi definida, não podendo ser compartilhada por outras tabelas. A Sequence, por outro lado, é um objeto definido pelo usuário e não fica atrelado a nenhuma tabela específica, desta forma, podendo ser compartilhada por várias tabelas através do comando NEXT VALUE FOR.

### **b) Qual a importância das chaves estrangeiras para a consistência do banco?**

**Resposta:** As chaves estrangeiras são identificadores únicos que conectam duas ou mais tabelas, fazendo referência às chaves primárias ou campos únicos de outras tabelas, garantindo consistência dos dados relacionados.

### **c) Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no cálculo relacional?**

**Resposta:** Alguns dos operadores do SQL que pertencem à álgebra relacional são: seleção, projeção, união, interseção, diferença, através de SELECT, WHERE, UNION, JOIN, =, <, >, etc.

### **d) Como é feito o agrupamento em consultas, e qual requisito é obrigatório?**

**Resposta:** Os agrupamentos no SQL são feitos através do comando GROUP BY. É obrigatório uma função de agregação em uma coluna para produzir um resultado agrupado, além disso é necessário dizer a coluna responsável pela estratificação do resultado da função.

## **Conclusão:**

Nessa Missão Prática, foi possível vivenciar a experiência de modelar e implementar uma base de dados relacional para um sistema simples, além de explorar a sintaxe SQL na criação e manipulação das estruturas do banco de dados.

Além disso, foi possível praticar a utilização das ferramentas DBDesigner e SQL Server Management Studio, que facilitam o processo de desenvolvimento e gerenciamento das bases de dados.

Utilizar o DBDesigner permite uma fácil visualização de como os dados e tabelas se relacionam entre si através dos diagramas de entidade-relacionamento (DER), além disso ele também gera automaticamente os comandos SQL para a criação das tabelas, índices, chaves e

relacionamentos a partir do DER, exportando para vários tipos de banco de dados diferentes. Apesar dessa facilidade, no entanto, é importante adquirir o conhecimento necessário do SQL e revisar o código gerado, pois o programa pode apresentar erros ou ineficiências, principalmente quando há relacionamentos mais complexos.

SQL Server Management Studio permitiu a conexão e administração dos dados de forma centralizada, o que facilita executar os comandos SQL, visualizar e editar dados, criar e modificar tabelas, índices entre outras funcionalidades. Ele oferece várias ferramentas de testes, otimização, backup, restauração, importação e exportação, o que facilita o trabalho de um administrador de banco de dados. No entanto, a ferramenta pode ser complexa e confusa para iniciantes ou usuários menos experientes, pois possui muitas opções, menus, janelas e ferramentas, então requer prática e treinamento apropriado para extrair o máximo de suas funcionalidades.

No geral, houveram alguns desafios, como a necessidade de instalar e configurar as ferramentas, revisar os comandos SQL gerados e verificar a coerência dos modelos e diagramas, mas foi muito gratificante ver o resultado final e o aprendizado para utilização desses conceitos e ferramentas de uma forma mais prática.