



Vamos manter as informações!

João Pedro da Silva Zanirati Nunes – 202209081405

Polo Azenha

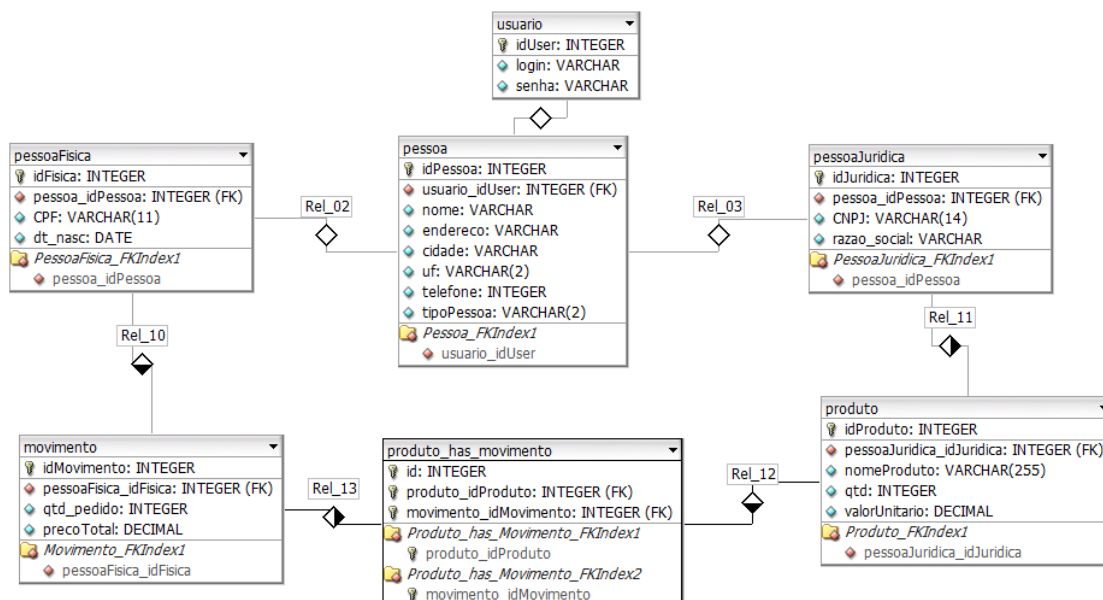
Nível 2: Vamos Manter as Informações? – 9003 – Mundo 3

Objetivo da Prática

O objetivo da prática é estruturar um banco de dados de forma eficiente e organizada, permitindo o gerenciamento de diferentes tipos de entidades relacionadas. Auxilia em consultas, melhora a integridade dos dados, facilita a manutenção e possibilita a escalabilidade do banco.

1º Procedimento – Criando o Banco de Dados

Inserir neste campo, **de forma organizada**, todos os códigos do roteiro do 1º Procedimento da Atividade Prática, os resultados da execução do código e a Análise e Conclusão:





```
create database loja
```

```
use LOJA
```

```
create sequence idSequence as int
```

```
    start with 1
```

```
    increment by 1
```

```
create table usuario(
```

```
    idUser int default next value for idSequence primary key,
```

```
    login varchar (max) not null,
```

```
    senha varchar (max) not null,
```

```
)
```

```
create table pessoa(
```

```
    idPessoa int default next value for idSequence primary key,
```

```
    nome varchar (max) not null,
```

```
    cidade varchar (max) not null,
```

```
    endereco varchar (max) not null,
```

```
    uf varchar (2) not null,
```

```
    telefone int,
```

```
    tipoPessoa varchar (2) not null check (tipoPessoa in('PF','PJ')),
```

```
    idUser int,
```

```
)
```

```
alter table pessoa
```

```
add constraint fk_idUser foreign key (idUser) references usuario (idUser)
```

```
create table pessoa_fisica(
```

```
    idFisica int default next value for idSequence primary key,
```

```
    cpf varchar (11) not null,
```



```
        dt_nasc date not null,

        idPessoa int,

    )

alter table pessoa_fisica

add constraint fk_idPessoa foreign key (idPessoa) references pessoa (idPessoa)

create table pessoa_juridica(

    idJuridica int default next value for idSequence primary key,

    cnpj varchar (14) not null,

    razaoSocial varchar (max) not null,

    idPessoa int,

)

alter table pessoa_juridica

add constraint fk_idPessoaJuridica foreign key (idPessoa) references pessoa (idPessoa)


create table movimento(

    idMovimento int default next value for idSequence primary key,

    qtd_pedido int not null,

    precoTotal decimal not null,

    idFisica int not null,

)

alter table movimento

add constraint fk_idFisica foreign key (idFisica) references pessoa_fisica (idFisica)


create table produto(

    idProduto int default next value for idSequence primary key,
```



```
nomeProduto varchar (max) not null,  
  
qtd int not null,  
  
valorUnitario decimal not null,  
  
tipo varchar (1) not null check (tipo in('E','S'))  
  
idJuridica int not null,  
  
)
```

```
alter table produto
```

```
add constraint fk_idJuridica foreign key (idJuridica) references pessoa_juridica  
(idJuridica)
```

```
create table produto_movimento(  
  
    idPM int default next value for idSequence primary key,  
  
    idProduto int not null,  
  
    idMovimento int not null,  
  
)
```

```
alter table produto_movimento
```

```
add constraint fk_idProduto foreign key (idProduto) references produto (idProduto)
```

```
alter table produto_movimento
```

```
add constraint fk_idMovimento foreign key (idMovimento) references movimento  
(idMovimento)
```

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



1X1: Usando uma chave estrangeira em uma das tabelas para se relacionar com a chave primária da outra. Cada registro em uma tabela está relacionado a um único registro na outra tabela.

1XN: Adicionando uma chave estrangeira na tabela "N" que se relaciona com a chave primária da tabela "Um". Cada registro na tabela "N" está associado a apenas um registro na tabela "Um", mas cada registro na tabela "Um" pode estar relacionado a muitos registros na tabela "N".

NxN: Criando uma tabela de junção que contém chaves estrangeiras relacionando as chaves primárias das duas tabelas que você deseja relacionar. Isso permite que muitos registros de uma tabela estejam associados a muitos registros da outra tabela.

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

O modelo "Herança de Tabela Única". Uma única tabela armazena todos os tipos de entidades com uma coluna de "tipo" para distinguir entre eles. Cada registro é associado a uma subclasse através dessa coluna.

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

Oferece uma interface amigável, edição de dados, modelagem de banco de dados, ferramentas de administração e depuração de consultas, tudo em uma única ferramenta.

Observe que os tópicos acima seguem exatamente o que está na Atividade Prática exigida.



2º Procedimento – Alimentando a Base

Inserir neste campo, **de forma organizada**, todos os códigos do roteiro do 2º Procedimento da Atividade Prática, os resultados da execução do código e a Análise e Conclusão:

use LOJA

```
insert into usuario (login, senha)
```

```
values ('op1', 'op1');
```

```
values ('op2', 'op2');
```

```
insert into pessoa (nome, cidade, endereco, uf, telefone, tipoPessoa, idUser)
```

```
values ('João', 'Poa', 'rua 2', 'rs', '111111111', 'PF', '1');
```

```
values ('maria', 'floripa', 'rua 36', 'sc', '222222222', 'PJ', '2');
```

```
insert into pessoa_fisica (cpf, dt_nasc, idPessoa)
```

```
values ('77777777777', '01/01/1998', '4')
```

```
insert into pessoa_juridica (cnpj, razaoSocial, idPessoa)
```

```
values ('999999999999999', 'Mercado Maria', '5');
```

```
insert into produto (nomeProduto, qtd, valorUnitario, tipo, idJuridica)
```

```
values ('Banana', '100', '5.00', 'S', '14');
```



```
values ('Laranja', '500', '2.00','S' , '14');
```

```
values ('Manga', '800', '4.00','S' , '14');
```

```
insert into movimento (qtd_pedido, precoTotal, idFisica)
```

```
values ('2', '4.00', '12')
```

```
insert into produto_movimento (idProduto, idMovimento)
```

```
values ('19', '21')
```

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

Identity: É específico do SQL Server, simples, menos personalizável e associado a colunas específicas.

Sequence: Suportado por várias plataformas, mais flexível, personalizável e pode ser compartilhado entre várias tabelas ou bancos de dados.

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

Manter a integridade dos dados, Evitar erros e inconsistências,

Garantir que as relações entre as tabelas sejam mantidas, Facilitar consultas e garantir dados confiáveis.

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

Álgebra Relacional:

SELECT, PROJECT, JOIN, UNION, INTERSECT, EXCEPT

Cálculo Relacional:

WHERE, EXISTS, FOR ALL, ANY/SOME



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

O agrupamento é feito usando 'GROUP BY'. Um requisito obrigatório é listar todas as colunas não agregadas no 'SELECT' na cláusula 'GROUP BY'. Isso ajuda a organizar os dados em grupos com base em certas colunas e aplicar funções de agregação, como soma ou contagem, a esses grupos.

Observe que os tópicos acima seguem exatamente o que está na Atividade Prática exigida.

Conclusão

Este trabalho abordou alguns dos principais conceitos de bancos de dados, incluindo modelagem, feita no DBDesigner, normalização, herança, sequências, chaves estrangeiras e o uso do SQL Server Management Studio facilitou na criação do banco. Além disso, é fundamental destacar a necessidade de encontrar o equilíbrio entre a estrutura do banco e o desempenho.