

Nome: **Ariel Freitas dos Santos**

Matrícula: **202208291694**

Curso: **Desenvolvimento Full Stack**

Turma : **2023.3**

Campus: **POLO ST MORADA DO SOL - GOIÂNIA - GO**

Github: https://github.com/ArielFSantos/SQL_Estacio_Nivel_2



Missão Prática | Mundo 3 | Nível 2
RPG0015 - Vamos manter as informações!

Objetivo:

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 dados (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 Server.

Criação de Tabelas:

1.Criação de Tabelas...C.master (loja (61))

```
CREATE TABLE Usuarios (
    ID INT PRIMARY KEY,
    Login NVARCHAR(50),
    Senha NVARCHAR(100),
);

CREATE TABLE Pessoas (
    ID INT PRIMARY KEY,
    Nome NVARCHAR(255),
    Logradouro NVARCHAR(255),
    Cidade NVARCHAR(255),
    Estado NVARCHAR(255),
    Telefone NVARCHAR(20),
    Email NVARCHAR(255)
);

CREATE TABLE PessoasFisicas (
    CPF NVARCHAR(14) PRIMARY KEY,
    PessoaID INT UNIQUE,
    FOREIGN KEY (PessoaID) REFERENCES Pessoas(ID)
);

CREATE TABLE PessoasJuridicas (
    CNPJ NVARCHAR(18) PRIMARY KEY,
    PessoaID INT UNIQUE,
    FOREIGN KEY (PessoaID) REFERENCES Pessoas(ID)
);

CREATE TABLE Produtos (
    ID INT PRIMARY KEY,
    Nome NVARCHAR(255),
    QuantidadeEstoque INT,
    PrecoVenda DECIMAL(10, 2)
);

CREATE TABLE MovimentosCompra (
    ID INT PRIMARY KEY,
    IDUsuario INT,
    IDPessoaJuridica NVARCHAR(18),
    IDProduto INT,
    QuantidadeComprada INT,
    PrecoUnitarioCompra DECIMAL(10, 2),
    DataCompra DATE,
    FOREIGN KEY (IDUsuario) REFERENCES Usuarios(ID),
    FOREIGN KEY (IDPessoaJuridica) REFERENCES PessoasJuridicas(CNPJ),
    FOREIGN KEY (IDProduto) REFERENCES Produtos(ID)
);

CREATE TABLE MovimentosVenda (
    ID INT PRIMARY KEY,
    IDUsuario INT,
    IDPessoaFisica NVARCHAR(14),
    IDProduto INT,
    QuantidadeVendida INT,
    DataVenda DATE,
    FOREIGN KEY (IDUsuario) REFERENCES Usuarios(ID),
    FOREIGN KEY (IDPessoaFisica) REFERENCES PessoasFisicas(CPF),
    FOREIGN KEY (IDProduto) REFERENCES Produtos(ID)
);
```

Criação de Sequence:

```
2.Criação de Sequen....master (loja (63)) X 1.Cr
CREATE SEQUENCE SequenciaUsuarios
START WITH 1
INCREMENT BY 1
MINVALUE 1;

CREATE SEQUENCE SequenciaProdutos
START WITH 1
INCREMENT BY 1
MINVALUE 1;

CREATE SEQUENCE SequenciaPessoa
START WITH 1
INCREMENT BY 1
MINVALUE 1;
```

Alimentando Tabelas:

```
Alimentando as Ta...C.master (loja (60)) X 2.Criação de Sequen....master (loja (63)) 1.Criação de Tabelas...C.master (loja (61))
INSERT INTO Usuarios (ID, Login, Senha)
VALUES
(1, 'op1', 'op1'),
(2, 'op2', 'op2');

INSERT INTO Produtos (ID, Nome, QuantidadeEstoque, PrecoVenda)
VALUES
(1, 'Produto A', 100, 10.99),
(2, 'Produto B', 50, 24.99),
(3, 'Produto C', 200, 5.99);

DECLARE @ProximoID INT;
SET @ProximoID = NEXT VALUE FOR SequenciaPessoa;

INSERT INTO Pessoas (ID, Nome, Logradouro,Cidade,Estado, Telefone,Email)
VALUES (@ProximoID, 'João Silva', 'Rua Principal 123','Goiania','GO', '1111-2222','joao@gmail.com');

INSERT INTO PessoasFisicas (CPF, PessoaID)
VALUES ('12345678901', @ProximoID);

DECLARE @ProximoIDJuridica INT;
SET @ProximoIDJuridica = NEXT VALUE FOR SequenciaPessoa;

INSERT INTO Pessoas (ID, Nome, Logradouro,Cidade,Estado, Telefone,Email)
VALUES (@ProximoIDJuridica, 'Marcos Souza', 'Rua Principal 231','Teresina','PI', '2121-1212','marcos@gmail.com');

INSERT INTO PessoasJuridicas (CNPJ, PessoaID)
VALUES ('41247954631874', @ProximoIDJuridica);

INSERT INTO MovimentosCompra (ID, IDUsuario, IDPessoaJuridica, IDProduto, QuantidadeComprada, PrecoUnitarioCompra, DataCompra)
VALUES
(1, 1, '41247954631874', 1, 50, 9.99, '2023-09-20'),
(2, 2, '41247954631874', 2, 30, 20.49, '2023-09-20');

INSERT INTO MovimentosVenda (ID, IDUsuario, IDPessoaFisica, IDProduto, QuantidadeVendida, DataVenda)
VALUES
(1, 1, '12345678901', 1, 20, '2023-09-21'),
(2, 2, '12345678901', 2, 10, '2023-09-21');
```

Consulta de Datos :

4.Consulta de Dados.....PC.loja (loja (74))*

```

select * from Pessoas
select * from PessoasFisicas
select * from PessoasJuridicas
select * from MovimentosCompra
select * from MovimentosVenda
select * from Produtos
select * from Usuarios
    
```

100 %

Resultados Mensagens

ID	Nome	Logradouro	Cidade	Estado	Telefone	Email
1	João Silva	Rua Principal 123	Goiania	GO	1111-2222	joao@gmail.com
2	Marcos Souza	Rua Principal 231	Teresina	PI	2121-1212	marcos@gmail.com

CPF	PessoaID
12345678901	1

CNPJ	PessoaID
41247954631874	2

ID	IDUsuario	IDPessoaJuridica	IDProduto	QuantidadeComprada	PrecoUnitarioCompra	DataCompra
1	1	41247954631874	1	50	9.99	2023-09-20
2	2	41247954631874	2	30	20.49	2023-09-20

ID	IDUsuario	IDPessoaFisica	IDProduto	QuantidadeVendida	DataVenda
1	1	12345678901	1	20	2023-09-21
2	2	12345678901	2	10	2023-09-21

ID	Nome	QuantidadeEstoque	PrecoVenda
1	Produto A	100	10.99
2	Produto B	50	24.99
3	Produto C	200	5.99

ID	Login	Senha
1	op1	op1
2	op2	op2

Análise e Conclusão:

Quais as diferenças no uso de sequence e identity?

Sequências (sequences) e colunas de identidade (identity columns) são usadas para gerar valores automáticos em um banco de dados.

Sequências oferecem mais opções de personalização, enquanto colunas de identidade têm um uso mais simples e podem ser menos eficientes em cenários de alto volume. A escolha depende das necessidades do projeto e do sistema de gerenciamento de banco de dados.

Qual a importância das chaves estrangeiras para a consistência do banco

As chaves estrangeiras são elementos que garantem a integridade e consistência dos dados em um banco de dados relacionado, estabelecendo relações entre tabelas. Elas impedem a inserção de registros "órfãos" e asseguram que os dados reflitam com precisão as relações entre entidades. Isso torna o banco de dados mais confiável e facilita a manutenção e consulta dos dados.

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

Na álgebra relacional, os operadores incluem projeção, seleção, união, interseção, diferença, produto cartesiano e junção. Por outro lado, o cálculo relacional não possui operadores no mesmo sentido, mas é baseado em predicados e quantificadores para recuperar dados. Em resumo, a álgebra relacional possui operadores específicos, enquanto o cálculo relacional é baseado em predicados e quantificadores.

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

A cláusula "GROUP BY" em consultas SQL é usada para agrupar resultados com base em uma ou mais colunas. Um requisito obrigatório é que todas as colunas não agregadas no "SELECT" estejam presentes no "GROUP BY". Isso permite resumir dados e aplicar funções de agregação para obter informações consolidadas.