

Exercício 3:

Comando:

```
DROP TABLE IF EXISTS Endereco CASCADE;  
CREATE TABLE Endereco(  
    idEndereco SERIAL PRIMARY KEY,  
    Logradouro VARCHAR(100) NOT NULL,  
    Cidade VARCHAR(40) NOT NULL,  
    Estado VARCHAR(40) NOT NULL,  
    Pais VARCHAR(40) NOT NULL  
);
```

```
DROP TABLE IF EXISTS Usuario CASCADE;  
CREATE TABLE Usuario(  
    Nome VARCHAR(40) NOT NULL,  
    Telefone VARCHAR(20) UNIQUE NOT NULL,  
    Email VARCHAR(40),  
    Senha VARCHAR(40),  
    idEndereco INT NOT NULL,  
    Sexo VARCHAR(20),  
    DataNasc DATE,  
    PRIMARY KEY (Nome, Telefone),  
    CONSTRAINT FK_usuario_endereco  
    FOREIGN KEY (idEndereco)  
    REFERENCES Endereco(idEndereco)  
);
```

```
DROP TABLE IF EXISTS Propriedade CASCADE;  
CREATE TABLE Propriedade(  
    Nome_prop VARCHAR(40) NOT NULL,  
    idEndereco INT NOT NULL,  
    Nome_usuario VARCHAR(40),  
    Tel_usuario VARCHAR(20),  
    Classe VARCHAR(20),  
    PrecoDiaria DECIMAL(10),  
    TaxaLimpeza DECIMAL(10),  
    MaxHospedes INT,  
    MaxNoites DECIMAL(2),  
    MinNoites DECIMAL(2),  
    QtdBanheiros INT,  
    QtdQuartos INT,  
    Data_inicial DATE,  
    Data_final DATE,  
    Regras VARCHAR(40),  
    PRIMARY KEY (Nome_prop, idEndereco),  
    CONSTRAINT FK_prop_usuario  
    FOREIGN KEY (Nome_usuario, Tel_usuario)  
    REFERENCES Usuario(Nome, Telefone),
```

```
CONSTRAINT FK_prop_endereco  
FOREIGN KEY (idEndereco)  
REFERENCES Endereco(idEndereco)  
);
```

```
DROP TABLE IF EXISTS Locacao CASCADE;  
CREATE TABLE Locacao(  
    idLocacao INT PRIMARY KEY,  
    DtCheckIn DATE,  
    DtCheckOut DATE,  
    Dt_Reserva DATE,  
    QtdHospedes DECIMAL(2),  
    Valor DECIMAL(10),  
    Cancelado BOOL  
);
```

```
DROP TABLE IF EXISTS Mensagem CASCADE;  
CREATE TABLE Mensagem(  
    Timestamp timestamp,  
    Usuario_origem VARCHAR(40) NOT NULL,  
    Tel_origem VARCHAR(20) NOT NULL,  
    Usuario_destino VARCHAR(40) NOT NULL,  
    Tel_destino VARCHAR(20) NOT NULL,  
    Texto VARCHAR(200),  
    CONSTRAINT FK_msg_usuario_origem  
    FOREIGN KEY (Usuario_origem, Tel_origem)  
    REFERENCES Usuario(Nome, Telefone),  
    CONSTRAINT FK_msg_usuario_destino  
    FOREIGN KEY (Usuario_destino, Tel_destino)  
    REFERENCES Usuario(Nome, Telefone)  
);
```

```
DROP TABLE IF EXISTS Comodidade CASCADE;  
CREATE TABLE Comodidade(  
    idComodidade SERIAL PRIMARY KEY,  
    Nome_prop VARCHAR(40) NOT NULL,  
    Endereco_prop INT NOT NULL,  
    Descricao VARCHAR(50) NOT NULL,  
    CONSTRAINT FK_comodidade_propriedade  
    FOREIGN KEY (Nome_prop, Endereco_prop)  
    REFERENCES Propriedade(Nome_prop, idEndereco)  
);
```

```
DROP TABLE IF EXISTS Avaliacao CASCADE;  
CREATE TABLE Avaliacao(  
    Nome_prop VARCHAR(40) NOT NULL,  
    Endereco_prop INT NOT NULL,
```

```

Mensagem VARCHAR(200),
Nota_comunicacao DECIMAL(2),
Nota_limpeza DECIMAL(2),
Nota_localizacao DECIMAL(2),
Nota_valor DECIMAL(2),
CONSTRAINT FK_aval_propriedade
FOREIGN KEY (Nome_prop, Endereco_prop)
REFERENCES Propriedade(Nome_prop, idEndereco)
);

```

```

DROP TABLE IF EXISTS ContaBancaria CASCADE;
CREATE TABLE ContaBancaria(
    Numero INT PRIMARY KEY,
    Nome_usuario VARCHAR(40) NOT NULL,
    Tel_usuario VARCHAR(20) NOT NULL,
    Tipo VARCHAR(20),
    Numero_roteamento INT,
    CONSTRAINT FK_conta_usuario
    FOREIGN KEY (Nome_usuario, Tel_usuario)
    REFERENCES Usuario(Nome, Telefone)
);

```

```

DROP TABLE IF EXISTS Locar CASCADE;
CREATE TABLE Locar(
    Nome_prop VARCHAR(40) NOT NULL,
    Endereco_prop INT NOT NULL,
    Nome_usuario VARCHAR(40) NOT NULL,
    Tel_usuario VARCHAR(20) NOT NULL,
    idLocacao INT NOT NULL,
    CONSTRAINT FK_locar_propriedade
    FOREIGN KEY (Nome_prop, Endereco_prop)
    REFERENCES Propriedade(Nome_prop, idEndereco),
    CONSTRAINT FK_locar_usuario
    FOREIGN KEY (Nome_usuario, Tel_usuario)
    REFERENCES Usuario(Nome, Telefone),
    CONSTRAINT FK_locar_idLocacao
    FOREIGN KEY (idLocacao)
    REFERENCES Locacao(idLocacao)
);

```

Explicação:

Seguimos o mapeamento dos exercícios anteriores para a criação das tabelas.

- **CREATE TABLE:**
 - Este comando SQL é usado para criar uma nova tabela no banco de dados.
 - Ele especifica o nome da tabela, seguido por uma lista de colunas e suas definições.

- **DROP TABLE IF EXISTS:**
 - Este comando SQL é usado para excluir uma tabela do banco de dados, caso ela exista.
 - A cláusula IF EXISTS verifica se a tabela existe antes de tentar excluí-la, o que evita erros caso a tabela não exista.
 - Por exemplo: DROP TABLE IF EXISTS Usuario; exclui a tabela "Usuario" se ela existir.
- **CONSTRAINT FK_prop_usuario FOREIGN KEY (Nome_usuario, Tel_usuario) REFERENCES Usuario(Nome, Telefone):**
 - Este comando define uma restrição de chave estrangeira (foreign key constraint) na tabela.
 - A restrição especifica que as colunas Nome_usuario e Tel_usuario na tabela atual (Propriedade) são chaves estrangeiras que referenciam as colunas Nome e Telefone na tabela "Usuario".
 - Isso garante que os valores nas colunas Nome_usuario e Tel_usuario na tabela "Propriedade" só possam ser inseridos se corresponderem a valores válidos nas colunas Nome e Telefone da tabela "Usuario".
 - Isso é útil para manter a integridade referencial entre as tabelas e garantir que não haja dados inconsistentes.
- **PRIMARY KEY:**
 - A cláusula PRIMARY KEY é usada para definir uma ou mais colunas como chaves primárias da tabela.
 - Uma chave primária é usada para identificar exclusivamente cada registro em uma tabela.

Resultado:

[illegible]