

Nome do Aluno: Lucas Rodrigues Amorim
Data: 21/08/2024

Trabalho: Integridade Referencial e Criação de Tabelas SQL

Objetivo:

1. **O que é Integridade Referencial? Qual a importância da Integridade Referencial?**
 - Integridade referencial é um conceito fundamental em bancos de dados relacionais que assegura a consistência e a precisão dos dados entre tabelas relacionadas. Ela garante que os relacionamentos entre as tabelas sejam mantidos de maneira coerente, impedindo que ocorram dados órfãos ou inconsistências.

2. **Instruções SQL:**
 - Criação das tabelas **Produtos** e **Categoria** de **Produtos**.
 - Estabelecimento da relação entre as tabelas usando integridade referencial.
 - Inserção de pelo menos 10 produtos e 4 categorias, diversificando as categorias dos produtos.

```
CREATE TABLE Categoria_de_Produtos (  
    CategoriaID INT PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL  
);
```

CategoriaID: Esta coluna é do tipo INT e atua como a chave primária da tabela. Isso significa que cada valor em CategoriaID deve ser único, garantindo que cada categoria seja identificada de forma exclusiva.

Nome: Esta coluna é do tipo VARCHAR(100), o que permite armazenar até 100 caracteres de texto. A cláusula NOT NULL garante que o nome da categoria deve ser preenchido, ou seja, não pode ser nulo.

```
CREATE TABLE Produtos (  
    ProdutoID INT PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL,  
    Preco DECIMAL(10, 2) NOT NULL,  
    CategoriaID INT,  
    FOREIGN KEY (CategoriaID) REFERENCES Categoria_de_Produtos(CategoriaID)  
);
```

ProdutoID: Este é o identificador único para cada produto. Como chave primária, ele garante que nenhum produto tenha o mesmo ProdutoID.

Nome: Armazena o nome do produto, com um limite de 100 caracteres e não permite valores nulos.

Preço: Esta coluna armazena o preço do produto. O tipo de dado DECIMAL(10, 2) significa que o valor pode ter até 10 dígitos, com 2 deles reservados para casas decimais.

CategoriaID: Esta coluna atua como uma chave estrangeira, ligando cada produto a uma categoria na tabela Categoria_de_Produtos. A integridade referencial é garantida pelo FOREIGN KEY, que força que cada valor de CategoriaID na tabela Produtos corresponda a um valor existente em Categoria_de_Produtos.

```
INSERT INTO Categoria_de_Produtos (CategoriaID, Nome) VALUES (1, 'Eletrônicos');
INSERT INTO Categoria_de_Produtos (CategoriaID, Nome) VALUES (2, 'Roupas');
INSERT INTO Categoria_de_Produtos (CategoriaID, Nome) VALUES (3, 'Alimentos');
INSERT INTO Categoria_de_Produtos (CategoriaID, Nome) VALUES (4, 'Móveis');
```

Essas instruções inserem quatro categorias diferentes na tabela Categoria_de_Produtos, cada uma identificada por um CategoriaID único.

```
INSERT INTO Produtos (ProdutoID, Nome, Preço, CategoriaID) VALUES (1, 'Smartphone', 1500.00, 1);
INSERT INTO Produtos (ProdutoID, Nome, Preço, CategoriaID) VALUES (2, 'TV LED', 2200.00, 1);
INSERT INTO Produtos (ProdutoID, Nome, Preço, CategoriaID) VALUES (3, 'Camiseta', 50.00, 2);
INSERT INTO Produtos (ProdutoID, Nome, Preço, CategoriaID) VALUES (4, 'Calça Jeans', 80.00, 2);
INSERT INTO Produtos (ProdutoID, Nome, Preço, CategoriaID) VALUES (5, 'Arroz', 30.00, 3);
INSERT INTO Produtos (ProdutoID, Nome, Preço, CategoriaID) VALUES (6, 'Feijão', 25.00, 3);
INSERT INTO Produtos (ProdutoID, Nome, Preço, CategoriaID) VALUES (7, 'Sofá', 1200.00, 4);
INSERT INTO Produtos (ProdutoID, Nome, Preço, CategoriaID) VALUES (8, 'Mesa de Jantar', 900.00, 4);
INSERT INTO Produtos (ProdutoID, Nome, Preço, CategoriaID) VALUES (9, 'Notebook', 3500.00, 1);
INSERT INTO Produtos (ProdutoID, Nome, Preço, CategoriaID) VALUES (10, 'Tênis', 150.00, 2);
```

Cada instrução INSERT INTO adiciona um produto na tabela Produtos, associando-o a uma categoria por meio do CategoriaID.

Os produtos são distribuídos entre as categorias Eletrônicos, Roupas, Alimentos, e Móveis.