

O que é integridade referencial?

Ele garante que todas as referências entre as tabelas em um banco de dados válido. Ela estabelece regras e restrições para os relacionamentos entre chaves primárias e estrangeiras.

Qual a importância da Integridade Referencial?

Ele é crucial para garantir que os relacionamentos entre os dados sejam respeitados, contribuindo para a consistência, confiabilidade e eficiência dos bancos de dados relacionais.

As instruções do SQL permitem criar tabelas `Produtos` e `Categoria_de_Produtos`.

Para criar tabelas você precisa usar esses códigos:

```
CREATE TABLE Categoria_de_Produtos (  
  id_categoria INT AUTO_INCREMENT PRIMARY KEY,  
  nome_categoria VARCHAR (100) NOT NULL  
);  
  
CREATE TABLE Produtos (
```

```
id_produto INT AUTO_INCREMENT PRIMARY KEY,  
nome_produto VARCHAR(100) NOT NULL,  
preco DECIMAL(10, 2) NOT NULL,  
quantidade_em_estoque INT NOT NULL,  
id_categoria INT,  
FOREIGN KEY (id_categoria) REFERENCES  
Categoria_de_Produtos(id_categoria) );
```

CREATE TABLE: é usada para criar uma nova tabela em um banco de dados. Ele também possui um nome e uma estrutura que define como os dados serão armazenados.

```
+-----+  
| Tables_in_sandbox_db |  
+-----+  
| clientes              |  
+-----+
```

FOREIGN KEY: ele é um conceito fundamental em bancos de dados relacionais que garante a integridade referencial entre tabelas. Ela cria um vínculo entre duas tabelas.

PRIMARY KEY: é um conceito fundamental em bancos de dados relacionais. Ela serve para identificar de forma específica cada registro da tabela.

Na linha de comando abaixo mostra um exemplo de 10 produtos de 4 categorias:

-- Criação da Tabela Categoria_de_Produtos

```
CREATE TABLE Categoria_de_Produtos (  
    id_categoria INT PRIMARY KEY,  
    nome_categoria VARCHAR(100) NOT NULL  
);
```

-- Criação da Tabela Produtos

```
CREATE TABLE Produtos (  
    id_produto INT PRIMARY KEY,  
    nome_produto VARCHAR(100) NOT NULL,  
    preco DECIMAL(10, 2) NOT NULL,  
    id_categoria INT,  
    FOREIGN KEY (id_categoria) REFERENCES  
    Categoria_de_Produtos(id_categoria)  
);
```

-- Inserção de Categorias

```
INSERT INTO Categoria_de_Produtos (id_categoria,  
    nome_categoria) VALUES
```

```
(1, 'Eletrônicos'),  
(2, 'Roupas'),  
(3, 'Alimentos'),  
(4, 'Móveis');
```

-- Inserção de Produtos

```
INSERT INTO Produtos (id_produto, nome_produto,  
preco, id_categoria) VALUES
```

```
(1, 'Televisor', 1500.00, 1),  
(2, 'Celular', 800.00, 1),  
(3, 'Camiseta', 50.00, 2),  
(4, 'Calça Jeans', 80.00, 2),  
(5, 'Arroz', 25.00, 3),  
(6, 'Feijão', 20.00, 3),  
(7, 'Sofá', 2000.00, 4),  
(8, 'Mesa de Jantar', 1200.00, 4),  
(9, 'Monitor', 600.00, 1),  
(10, 'Tênis', 120.00, 2);
```

-- Exibir o conteúdo da Tabela
Categoria_de_Produtos

```
SELECT * FROM Categoria_de_Produtos;
```

-- Exibir o conteúdo da Tabela Produtos

```
SELECT * FROM Produtos;
```

Essas operações garantem a criação das tabelas com integridade referencial e inserção dos dados de forma que a relação entre produtos e categorias seja mantida corretamente.

```
+-----+-----+
| id_categoria | nome_categoria |
+-----+-----+
|           1 | Eletrônicos   |
|           2 | Roupas        |
|           3 | Alimentos     |
|           4 | Móveis        |
+-----+-----+
```

```
+-----+-----+-----+-----+
| id_produto | nome_produto  | preco  | id_categoria |
+-----+-----+-----+-----+
|          1 | Televisor     | 1500.00 |          1 |
|          2 | Celular       |  800.00 |          1 |
|          3 | Camiseta      |   50.00 |          2 |
|          4 | Calça Jeans   |   80.00 |          2 |
|          5 | Arroz         |   25.00 |          3 |
|          6 | Feijão        |   20.00 |          3 |
|          7 | Sofá          | 2000.00 |          4 |
|          8 | Mesa de Jantar | 1200.00 |          4 |
|          9 | Monitor       |   600.00 |          1 |
|         10 | Tênis         |   120.00 |          2 |
+-----+-----+-----+-----+
```