



## Relacionamento entre tabelas

### Transcrição

[00:00] Fala, aluno. Tudo bom? Até agora nós estamos trabalhando, no nosso sistema, apenas com produtos. Mas estamos fazendo as operações básicas, estamos inserindo, buscando, alterando, removendo, enfim, aquele CRUD que nós viemos trabalhando ao longo do curso. Só que agora chegou uma documentação do nosso chefe, onde ele pede para nós categorizarmos esses produtos.

[00:25] Então agora o produto, além de ter um nome e uma descrição, ele quer também uma categoria. Então vamos no nosso banco de dados, no MySQL 8.0 Command Line Client, para nós vermos como está funcionando a nossa tabela. Se eu fizer `SELECT * FROM PRODUTO`, eu vou selecionar todos os produtos, nós vamos ver aqueles atributos que nós criamos na tabela e o conteúdo que nós adicionamos ao longo do curso.

[00:56] Então o que eu penso aqui? Se o nosso chefe agora quer categorizar esses produtos, eu vou poder criar então uma nova tabela, chamada categoria, e vou passar uma string quando eu for adicionar um novo produto, e eu escrevo se é um eletrônico, se é um eletrodoméstico, enfim, todas essas categorias que tem no nosso dia a dia.

[01:22] Porque, pense bem, eu, João, vou escrever a categoria de notebook Samsung, vou botar "ELETRONICO", tudo maiúsculo, sem acento. A outra desenvolvedora da equipe vai escrever, talvez, um videogame, é um eletrônico também, só que ela vai escrever "eletrônico", tudo minúsculo e com acento.

Então já começamos a ter eletrônico escrito de formas diferentes, começa a duplicar as palavras no nosso banco de dados.

[01:56] Enfim, começa a ficar uma confusão. Então já vejo que essa não é uma boa abordagem. Como eu posso fazer então? Talvez eu tenha uma tabela de categorias e dê um jeito de vincular essas duas tabelas, a de produto e a de categoria. Talvez isso faça mais sentido. Então vamos criar a nossa tabela de categoria e vamos ver como fazer para vincular essas duas tabelas.

[02:25] Então vou botar um `CREATE TABLE CATEGORIA (ID INT AUTO_INCREMENT, NOME VARCHAR (50) NOT NULL, PRIMARY KEY (ID))`, que vai ter um ID do tipo inteiro, Auto increment, vai ter um nome, que vai ser um VARCHAR de 50 caracteres, *not null* e a sua *primary key* vai ser o ID. Também vamos colocar o Engine InnoDB, esse, se vocês lembram, nós falamos sobre esse Engine na primeira aula.

[03:06] Agora, se eu fizer um `SELECT * FROM CATEGORIA`, não vai ter ninguém. Pensando na descrição de cada produto: Notebook Samsung, geladeira, cômoda, nós temos três categorias por enquanto. Tem um eletrônico, eletrodoméstico e móveis. Então vamos fazer logo os inserts dessas categorias.

[03:31] `INSERT INTO CATEGORIA (NOME) VALUES ('ELETRONICOS');`, a categoria de eletrônicos. `INSERT INTO CATEGORIA (NOME) VALUES ('ELETRODOMESTICOS');` e `INSERT INTO CATEGORIA (NOME) VALUES ('MOVEIS');`. Se eu fizer um `SELECT * FROM CATEGORIA` agora, nós temos que ter três categorias. Perfeito.

[04:01] Agora eu quero vincular essa tabela Categoria à essa tabela de Produto. Então eu quero, quando eu fiz um *select* de produto, eu quero que apareça o seguinte: eu tenho no ID 1 o notebook, a descrição é notebook Samsung, e eu quero a referência dessa tabela categoria, que a referência dessa tabela seja o seu ID.

[04:23] Então eu quero que tenha um número 1 na tabela produto, e eu falar: 1, na minha tabela de categoria, é eletrônicos. Para isso, primeiro eu tenho que

criar uma nova coluna na tabela de produto, chamada categoria ID. Então para isso eu vou fazer um `ALTER TABLE PRODUTO ADD COLUMN CATEGORIA_ID INT;` para adicionar uma coluna, que vai se chamar `CATEGORIA_ID` e ela vai ser do tipo inteiro.

[04:53] Criei essa coluna. Se eu fizer um `SELECT * FROM PRODUTO` agora, eu vou ter a "CATEGORIA\_ID", tudo *null*, porque eu ainda não vinculei as duas tabelas. Para eu vincular essas duas tabelas, eu tenho um conceito aqui, no banco de dados, que se chama chave estrangeira. Essa chave estrangeira vai ser a que vai amarrar essas duas tabelas.

```
mysql> SELECT * FROM PRODUTO;
```

id	nome	descricao	CATEGORIA_ID
1	NOTEBOOK	NOTEBOOK SANSUNG	NULL
2	GELADEIRA	GELADEIRA AZUL	NULL
116	Cômoda	Cômoda vertical	NULL

3 rows in set (.00 sec)

[05:22] Então, como eu informei, eu quero que quando eu for adicionar um produto, eu referencie a categoria pelo seu ID. Quem faz isso é essa chave estrangeira. Para criarmos uma chave estrangeira, eu vou ter que criar também uma chave estrangeira. Nós vamos precisar fazer o seguinte, então eu tenho que fazer um `ALTER TABLE` de novo em produto.

[05:44] Eu vou adicionar uma `FOREIGN KEY`, que é a nossa chave estrangeira, falando que é a categoria ID e eu vou fazer o seguinte, referenciando, então é `REFERENCES CATEGORIA` e vou adicionar ID, então fica `ALTER TABLE PRODUTO ADD FOREIGN KEY (CATEGORIA_ID) REFERENCES CATEGORIA (ID);`.

[06:10] Então o que eu estou falando aqui é o seguinte: adicione uma chave estrangeira em Produto, que vai referenciar ao ID da categoria 16. Vamos ver

isso aqui dá certo? Beleza. Agora, se eu fizer um `SELECT * FROM PRODUTO` novamente, continua do mesmo jeito. Mas quando eu for fazer um `UPDATE PRODUTO SET CATEGORIA_ID = 1`, eu vou ter que falar o seguinte, vou falar `UPDATE PRODUTO SET CATEGORIA_ID = 1`.

[07:02] Vamos fazer assim: Update o produto *setando* a `CATEGORIA_ID` igual a 1, que é a nossa eletrônicos. `UPDATE PRODUTO SET CATEGORIA_ID = 1 WHERE ID = 1`; , então o ID do produto é igual a 1. Então se vocês verem a tabela Produtos, onde o notebook está com o ID 1, a `CATEGORIA_ID` vai ser igual a 1, que vai ser `ELETRONICOS`.

[07:26] Se eu executar essa linha. Agora, se eu fizer um `SELECT * FROM PRODUTO` de novo, eu vou ter a `CATEGORIA_ID` do notebook agora é 1, então ele é um eletrônico, da categoria de eletrônicos. A chave estrangeira, ela é legal porque se eu tentar adicionar na tabela Produto uma referência de categoria que não existe na tabela Categoria, por exemplo o número 4, ele vai dar um erro. Então vamos fazer esse teste?

[08:02] Vou fazer o seguinte, eu vou botar `UPDATE PRODUTO SET CATEGORIA_ID = 4 WHERE ID = 2`; , na geladeira. Se eu fizer isso, ele não vai permitir porque não existe um ID 4 em Categoria. Nós não temos uma categoria que está no ID 4. Então essa é a vantagem de termos a chave estrangeira.

[08:32] Então, para terminarmos a nossa aula, vamos só *setar* então qual é a categoria dos outros produtos. A geladeira, ela vai ser `UPDATE PRODUTO SET CATEGORIA_ID = 2 WHERE ID = 2`; , de eletrodoméstico e ela também está no ID 2. Fiz esse código, mandei executar, ele executou com sucesso.

[08:55] Agora eu vou querer o `UPDATE PRODUTO SET CATEGORIA_ID = 3 WHERE ID = 116`; , então onde é a cômoda, ele vai ficar na categoria de móveis. Vou adicionar essa ID, agora quando eu fizer um `SELECT * FROM PRODUTO`; , nós vamos ter a tabela com o produto, já referenciando as suas categorias corretas.

[09:23] Então, aluno, agora nós já aumentamos a complexidade, porque nós estamos trabalhando com relacionamentos de tabelas, com chave estrangeira, enfim, alguns novos conceitos, mas que vão servir justamente para termos uma base dados normatizada, informações sem serem repetidas, nós temos uma tabela de domínio, que chama a de Categoria.

[09:48] Então aprendemos já alguns conceitos bastante interessantes e agora nós já podemos ir para a nossa aplicação e começar a construir ou a estruturar as camadas que vão representar uma categoria. Então por hoje nós ficamos aqui, espero que vocês tenham gostado e até a próxima aula.