



## Removendo dados

### Transcrição

[00:00] Fala, aluno. Tudo bom? Anteriormente, nós vimos como inserir um produto a partir da nossa aplicação. Nós criamos a classe `TestaInsercao`, criamos uma cláusula `insert` para passar o mouse como produto que nós queremos agora na nossa base de dados. Porém, efetuando alguns testes, esse produto novo, ele foi inserido várias vezes. Se nós bem lembramos, a nossa tabela ficou com informações repetidas.

[00:36] Sendo que não tem necessidade, eu não quero informações repetidas na minha tabela. Eu quero apagar esses dados. Como agora tudo o que nós fazemos que envolva o banco de dados, é a partir da nossa aplicação, para remover esses dados repetidos também eu vou utilizar a aplicação.

[00:54] Então, para isso, eu vou uma classe "`TestaRemocao`". E eu quero deixar só os produtos que nós inserimos quando estávamos configurando o nosso banco de dados, então vai ser a geladeira azul e o notebook Samsung. Nessa nova classe, nós vamos utilizar todos os conceitos que já vimos ao longo do curso.

[01:18] Eu vou ter que instanciar uma `ConnectionFactory` - calma, sem antes criar o `main` não. Então eu crio um `public static void main(String[] args)` e agora sim eu vou instanciar a minha `ConnectionFactory`, que eu vou precisar de uma conexão aqui - `ConnectionFactory factory = new ConnectionFactory()`.

[01:43] Na minha `factory` eu vou chamar o `Connection connection = factory.recuperarConexao();`. Vou precisar adicionar o `throws SQLException`. Após e

recuperar a conexão, eu preciso criar o Statement, vou receber aqui ele em uma variável chamada `stm`, então fica `Statement stm = connection.createStatement();`.

[02:11] E vou chamar o `stm.execute();`. Como nós queremos deixar só os dois primeiros, então vou fazer a cláusula `("DELETE FROM PRODUTO")` onde eu quero apagar os IDs maior que 2, então fica `stm.execute("DELETE FROM PRODUTO WHERE ID > 2");`. Se eu executar esse código, pode ser que funcione, pode ser que não. Eu preciso de uma confirmação.

[02:40] Qual confirmação eu consigo ter após deletar algumas linhas no nosso banco de dados? Aqui, no Statement, eu tenho um método que chama `getUpdateCount()`, que nos retorna um inteiro. Esse inteiro significa o seguinte, quantas linhas que foram modificadas após o Statement ser executado. Então, quando eu executo esse delete do meu código, quantas linhas ele vai apagar?

[03:14] Isso nós podemos verificar da seguinte forma: eu vou criar uma variável inteira e vou chamar de `linhasModificadas`, ficando `Integer linhasModificadas = stm.getUpdateCount();`. E, para vermos esse resultado melhor, eu vou botar um `System.out.println("Quantidade de linhas que foram modificadas: " + linhasModificadas);`.

[03:41] Se eu executar esse trecho de código, do `TestaRemocao`, quando eu der o "Run As > Java Application", ele vai me mostrar quantas linhas foram excluídas. Então três linhas foram excluídas com essa cláusula SQL. Se de fato nós voltarmos na nossa classe `TestaListagem` e executarmos ela novamente, o resultado vai ser o que nós esperávamos, apenas os dois produtos que foram inseridos desde as primeiras aulas.

[04:11] Se eu voltar na `TestaRemocao` e manda executar o código novamente, ele vai me retornar que foram zero linhas modificadas, porque já não tem mais produtos com ID maior que 2 para serem deletados, então ele não fez nada,

não teve alteração no nosso banco de dados. Então por isso esse valor de resultado é zero.

[04:32] Então, com o `TestaRemocao`, nós fizemos três comandos básicos do SQL, que foi o delete, o insert e o select. A ideia agora é que nós iremos avançar e vamos entrar em conceitos um pouco mais complexos, vamos trabalhar com cláusulas mais complexas, enfim, todo esse passo a passo para entendermos como funciona esse mundo do JDBC. Então, espero que você gostado e até a próxima aula.