



Listagem e remoção

Transcrição

[00:00] Fala, aluno. Tudo bom? Dando continuidade ao nosso curso, após conhecermos o `PreparedStatement`, agora podemos alterar todas as nossas classes para utilizá-lo. Então, voltando à nossa classe `TestaListagem`, eu vou alterar o `Statement stm = connection.createStatement();` para `Statement stm = connection.prepareStatement(sql);`.

[00:27] E com o exemplo que fizemos de inserção, no `stm.execute("SELECT ID, NOME, DESCRICAO FROM PRODUTO");`, retiro a minha cláusula `sql`, o `("SELECT ID, NOME, DESCRICAO FROM PRODUTO");` e passo para dentro dos parênteses do `connection.prepareStatement`, ele agora é o responsável por preparar esse `Statement`.

[00:43] Diferente do exemplo da inserção, no `TestaListagem` nós não temos um atributo, mas se um dia eu precisar fazer algum `where`, que vai receber alguma condição, eu já tenho esse `prepareStatement` preparado e eu só preciso adicionar depois o atributo. Então, em vez de eu retornar um `Statement`, eu vou retornar um `PreparedStatement stm = connection.createStatement();`.

[01:19] E eu tiro o `import java.sql.Statement`, que não é mais utilizado. Como nós já vimos, se eu executar o `TestaListagem`, continua do mesmo jeito, nós não tivemos nenhum problema nesse Refactory, nessa alteração. Na classe `TestaRemocao` podemos fazer a mesma coisa. Então em vez de eu ter o `Statement stm = connection.createStatement();`, eu vou ter o `Statement stm = connection.prepareStatement(sql);`.

[01:53] E mais uma vez aqui, só para ficar claro, eu tiro a nossa string SQL (`"DELETE FROM PRODUTO WHERE ID > 2"`) de `stm.execute`, e passo para dentro dos parênteses de `connection.prepareStatement`. Para ficar mais interessante o nosso exemplo, eu vou retirar o `2`, que estava fixo e colocar (`"DELETE FROM PRODUTO WHERE ID > ?"`).

[02:14] E vou passar esse valor do ponto de interrogação através do `setInt`. Então eu vou fazer um `stm.setInt(parameterIndex, x);`, e vou passar nesse primeiro index, eu passo o valor `(1, 2)`. E continuo executando o delete normalmente. Se eu executar o `TestaRemocao`, agora ele vai me mostrar a quantidade de linhas que foram modificadas, duas linhas.

[02:49] Se nós voltarmos no nosso `TestaListagem`, não vai constar mais aqueles produtos que foram inseridos, que tinham os erros, caracteres especiais, enfim, aqueles testes que nós fizemos no `TestaInsercaoComParametro`. Então é isso, pessoal. Mais um Refactory feito no nosso código, espero que vocês tenham gostado e até a próxima aula.