



Assumindo o controle

Transcrição

[00:00] Fala, aluno. Tudo bom? Anteriormente, nós vimos como utilizar o `PreparedStatement` no nosso código. Nós vimos que com ele é possível evitar os SQL Injections, que são capazes de acabar com a nossa aplicação. Vimos também que o nosso código fica mais sucinto, porque não precisamos mais ficar controlando a concatenação de variáveis com a nossa cláusula SQL.

[00:31] Enfim, todo aquele problema que tínhamos com validação de Query, enfim, temos várias vantagens em utilizá-lo. Porém agora eu quero continuar utilizando essa classe `TestaInsercaoComParametro`, porque agora, ao invés de adicionar um único produto, o produto que estará guardado nas nossas variáveis aqui, eu quero adicionar mais de um produto, eu quero ter opção de adicionar dois, três, quatro produtos, que for.

[01:00] Hoje o nosso código não está preparado para isso. Nós temos aqui dois `setString`, na verdade, que vão adicionar o nome e a descrição, que são os dois atributos que a nossa query espera. Vamos ter um método `execute()`, que executa essa cláusula SQL do insert e temos o método `getGeneratedKeys`, que retorna a chave daquele objeto que foi criado, aquele produto que foi criado.

[01:25] Então, para isso ficar melhor para a nossa refatoração, para adicionar mais do que um produto, eu vou extrair esse pedaço de código para um método. Se eu der um "Ctrl + 3" no Eclipse, no meu programa já está o "Extract Method", mas vocês podem procurar essa mesma opção, que vai ter no programa de vocês.

[01:47] Eu quero colocar o nome do método de "adicionarVarivel". Ele já está chamando o método e passando os atributos que nós vamos precisar, que é o nome, para *setar* no `stm.setString(1, nome);`; descrição para *setar* no segundo `setString`, o `stm.setString(2, descricao)`, e o Statement, o `stm`, que é o que vai de fato executar a nossa cláusula SQL, que vai retornar as chaves geradas, enfim, nós precisamos desses três atributos.

[02:19] Para ficar melhor o nosso código, eu não vou passar o `String nome = ""`; e o `String descricao = ""`; como valor na variável, eu vou passar diretamente as strings na `adicionarVariavel("", "", stm);`. Vou copiar esse trecho da `adicionarVariavel`, porque agora eu quero adicionar dois produtos.

[02:37] Então, o primeiro produto eu quero uma `("SmartTV", "", stm);`, eu vou botar `("SmartTV", "45 polegadas", stm);`. E vou botar um `("Radio", "Radio de bateria", stm);`. Vocês podem dar o nome que vocês quiserem para os produtos, é só para exemplificar mesmo. Com esse nosso código, não é para ter nenhuma diferença dos outros códigos que nós já viemos executando apenas com um produto.

[03:12] Ele tem que adicionar os dois produtos e nos retornar duas chaves. Vamos verificar? Vou mandar executar a minha aplicação. Beleza, olha, ele criou dois IDs para mim, o 68 e o 69. Para conferirmos se os produtos foram de fato inseridos, vamos na `TestaListagem` e vamos verificar se os dois produtos estão lá. Já garantimos que nós temos os dois produtos novos inseridos na base.

[03:21] Só que nós mandamos executar aqui a classe

`TestaInsercaoComParametro` e ele adicionou os dois produtos. O que aconteceria se no momento em que eu estivesse adicionando o produto rádio, eu tivesse uma exceção? Qual seria o comportamento da minha aplicação? Ela adicionaria apenas o primeiro produto ou ela não adicionaria ninguém, por conta da exceção que deu na hora de adicionar o segundo produto?

[04:13] Qual seria esse comportamento? Para descobrirmos, eu vou forçar um pouco no código, para ele de fato dar uma exceção. Então eu vou verificar se o nome do produto é igual à `Radio`, com `if(nome.equals("Radio"))`, porque se for, eu vou dar uma `throw new RuntimeException("");`, eu vou forçar aqui um erro. E vou botar uma mensagem que `("Não foi possível adicionar o produto");`.

[04:47] Eu vou mandar agora remover esses dois produtos que nós inserimos, com `stm.setInt(1, 2);` em `TestaRemocao`. Apagamos os produtos. Agora, em `TestaInsercaoComParametro` eu vou mandar novamente inserir os meus dois produtos. Quando eu mandar executar a classe, o ID criado foi o 70, mas estourou uma exceção, informando que não foi possível adicionar o produto.

[05:17] Agora para eu descobrir qual foi o comportamento do meu sistema, se ele adicionou só o primeiro, se ele não adicionou ninguém, vamos listar novamente no `TestaListagem`. Vou dar um "Run As > Java Application". Ele adicionou o primeiro produto, como ele tinha informado no momento da inserção, só que o segundo o produto não consta aqui.

[05:37] Então o que a nossa aplicação está fazendo? Então ela está pegando o primeiro produto que pedi para inserir, ela abre uma transação, vai concatenar o nome e a descrição na minha cláusula SQL, vai mandar executar essa cláusula, vai dar o `ResultSet` e fecha a transação. No segundo produto, ela vai abrir uma nova transação, vai concatenar o nome e a descrição do produto.

[06:18] Só que vai dar uma exceção, porque o nome é `Radio` e falamos que se fosse `Radio` tinha que dar uma `SQL Exception`. Mas olha só: nós abrimos uma transação, fechamos uma transação. Abre uma transação, fecha uma transação. A meu ver, isso poderia estar de uma forma melhor no nosso código. Então, o que acontece?

[06:43] Eu posso assumir o controle dessa transação. Eu não quero mais que o JDBC faça dessa maneira. Porque quando eu tenho o controle da minha transação, no momento em que der erro, eu posso não incluir o segundo

produto que deu erro, mas também posso falar para ele apagar o primeiro produto que foi inserido, digamos que uma transação única, enfim. Para fazer isso, o que eu preciso fazer?

[07:19] Eu tenho que tirar a responsabilidade, o Commit, ou seja, da inserção do meu produto, das mãos do JDBC. E eu vou fazer o seguinte:

`connection.setAutoCommit(false);` , vou falar: `connection` , o `setAutoCommit` agora vai ser *false*, ou seja, eu vou controlar o momento do Commit da minha aplicação, no momento da minha transação. Então com esse

`connection.setAutoCommit(false)` , agora eu que controlo a minha transação.

[07:48] Eu posso então, agora, fazer o seguinte: eu vou apagar o meu produto "SmartTV" que foi adicionado, em `TestaRemocao` , e eu vou adicionar de novo, em `TestaInsercaoComParametros` . Vamos ver o comportamento agora com esse `connection.setAutoCommit(false);` . Ele criou o ID 71, deu erro, um cenário bem parecido com o anterior.

[08:13] Vamos ver o que aconteceu, aqui na nossa `TestaListagem` . Não incluiu nenhum produto. Então vamos fazer o seguinte, eu vou comentar aqui essa exceção, o `if(nome.equals("Radio"))` , que ele pode dar por conta dessa exceção. Então vou comentar aqui. Agora vou novamente mandar inserir e vamos ver o resultado.

[08:47] O ID criado foi o 72, o ID criado foi o 73. Um cenário igual ao que estava no começo da aula. Deixa eu testar no `TestaListagem` . Não adicionou nenhum dos dois. Bom, temos o seguinte cenário: agora nós tiramos a responsabilidade do JDBC de fazer os Commits das nossas transações, de fato inserir o produto na nossa tabela. Só que agora, com isso, não estamos conseguindo mais adicionar nenhum produto.

[09:28] Por que será que isso está acontecendo? Resolvemos um problema, porém agora nós temos que tratar esse segundo problema. Isso nós já vamos ter insumo para na próxima aula já tratar esse problema e verificar o motivo

não estar inserindo esse produto e como nós vamos tratar isso. Então espero que vocês tenham gostado dessa aula. Até o próximo vídeo.