



Inserindo com Statement

Transcrição

[00:00] Fala, aluno. Tudo bom? Vamos dar continuidade ao nosso curso de JDBC. Recapitulando as aulas anteriores, principalmente a nossa classe `TestaListagem`, se nós executarmos ela, nós vamos ver que tem os dois produtos inseridos, um notebook e a geladeira. Esses produtos, eles foram inseridos na época em que estávamos ainda configurando o nosso banco de dados.

[00:24] Então, eles foram inseridos via banco de dados e não via aplicação. A intenção agora é que nós consigamos fazer essa inserção pela a nossa aplicação. Para isso, nós vamos criar uma nova classe chamada `"TestaInsercao"`. Essa classe, ela vai ter um método `public static void main(String[] args)`.

[00:49] E vamos utilizar a nossa `ConnectionFactory` `factory = new ConnectionFactory();` para poder usar o método `recuperarConexao`, que vai ser o que de fato vai retornar a conexão para a nossa classe `TestaInsercao`. Então vamos `Connection connection = factory.recuperarConexao();`.

[01:11] Aqui, como nós não estamos tratando em `.recuperarConexao` a `SQLException`, então nós temos que adicionar o `throws SQLException` no nosso método `main`. Beleza, uma vez que eu tenho a conexão em mãos agora, nós vimos que para executar cláusulas SQL a partir da nossa aplicação, nós precisamos de um `Statement`.

[01:34] Esse Statement, para recuperarmos ele, nós precisamos criar o Statement aqui, que está dentro da connection. Nós temos o `Statement stm = connection.createStatement();`, que vai nos retornar o Statement. Tenho agora o `Statement stm = connection.createStatement();` em mãos, agora eu posso chamar o método `stm.execute("");` e passar para ele qual é a cláusula que nós queremos.

[02:01] Se nós queremos inserir algo, vai ser um `stm.execute("INSERT INTO PRODUTO (nome, descricao) VALUES ('Mouse', 'Mouse sem fio'))");`. Então vamos inserir no produto um nome e uma descrição. Os valores, eu quero aqui um mouse e a descrição dele vai ser um mouse sem fio. Fica então `stm.execute("INSERT INTO PRODUTO (nome, descricao) VALUES ('Mouse', 'Mouse sem fio'))");`.

[02:24] Um ponto de atenção aqui é nas aspas. Lembrando que aspa simples, aqui, nesse caso, é para a string do SQL, então da cláusula do SQL, e as aspas duplas vão ser para a string do Java. Se inverter as aspas vai dar erro, então só para evitar esse tipo de problema mesmo.

[02:48] Nós vimos, anteriormente, que o método `.execute()`, ele nos retorna um booleano. Quando ele retorna uma lista, o booleano vai ser `true`, quando o retorno da cláusula for uma lista, o booleano vai ser `true`, quando não, vai ser `false`. No nosso caso, é um insert, o insert não vai retornar uma lista, então ele tem que ser `false`.

[03:12] Para comprovamos isso que estamos falando, basta fazermos um `System.out.println(resultado);` e ele vai ter que ser `false`. Beleza. Mas para nós não é um resultado muito agradável. Como que eu vou utilizar isso na minha aplicação? De que me serve esse `false`? Não faz sentido.

[03:40] Para mim, faria mais sentido eu ter um resultado, de quando inserisse um produto, de qual foi o produto que foi criado. Qual foi o ID desse produto? Então, para isso, aqui nas nossas classes e métodos JDBC, nós temos uma riqueza de recursos para trabalhar com o banco.

[04:07] Dentro do `stm.execute` , ele me permite criar um novo parâmetro, que vai ser o Statement e dentro dele tem `RETURN_GENERATED_KEYS` , então fica:

```
stm.execute("INSERT INTO PRODUTO (nome, descricao) VALUES ('Mouse', 'Mouse sem fio')", Statement.RETURN_GENERATED_KEYS); .
```

[04:20] Então estou falando para ele o seguinte: quando eu executar essa cláusula insert, eu quero também que ele me retorne a chave gerada, o ID gerado. Para nós, isso vai ser interessante nesse seguinte ponto, eu executo, depois eu pego a chave gerada dentro do meu Statement, com

```
stm.getGeneratedKeys(); .
```

[04:47] E esse `getGeneratedKeys()` , ele vai me retornar um `ResultSet rst = stm.getGeneratedKeys()` ; que nós conhecemos já, que nós sabemos iterar nesse `ResultSet` . Aqui eu vou fazer o `while` , que nós já vimos anteriormente, `while(rst.next())` e aqui - não sei se vocês lembram. Quer dizer, espero que vocês lembrem.

[05:09] Quando nós vamos buscar informação de uma coluna, nós estamos buscando pelo Label dessa coluna. Para ficar mais fácil de relembrar, caso alguém tenha esquecido, dentro da coluna nome, eu passo a Label ("`NOME`") mesmo para recuperar o valor que vai estar naquela coluna.

[05:34] Nós temos uma maneira diferente para fazer isso aqui, e para vocês terem conhecimento, no nosso caso nós queremos saber qual foi o ID criado, então eu vou pegar `Integer id = rst.getInt` , igual anteriormente, só que invés de eu fazer o "`String columnLabel`", eu vou fazer o "`Int columnIndex`".

[05:57] E aqui eu vou passar o index 1, com `Integer id = rst.getInt(1)` ; , que no SQL a primeira coluna, ela é considerada 1, não é 0 igual em uma lista Java, por exemplo. E para eu verificar aqui agora qual é o resultado disso, eu vou botar a seguinte mensagem: `System.out.println("O id criado foi: " + id)`; e vou pegar o ID, que nós estamos recuperando.

[06:30] A partir do momento que eu dou um "Run as", ele vai me mostrar o ID que foi criado, que foi o 18. Se eu for na minha `TestaListagem`, agora eu vou verificar que ele criou para nós o produto mouse sem fio com o ID 18. Se eu executar mais uma vez o código `TestaInsercao`, o ID criado foi 19. E se eu verificar em `TestaListagem`, ele vai mostrar os dois mouses.

[07:09] Então a cada vez que eu testar a minha classe `TestaInsercao`, ele vai criar um novo produto. Só que nesse caso, nós verificamos que ele criou alguns lixos. Nós testamos a `TestaInsercao`, vimos que está inserindo, muito bom, por sinal o nosso resultado. Porém, agora nós temos lixo.

[07:33] Para lixo, nós temos que apagar esses registros repetidos, nós não queremos uma base com registros repetidos, não faz sentido, a nossa tabela não vai ficar concisa. Então isso é um desafio para as próximas aulas, para nós apagarmos esse lixo. Mas, por enquanto, nós temos uma nova classe inserindo um novo produto, retornando o ID desse produto e é isso. Espero que vocês tenham gostado e até a próxima aula.