02

Listagem com Statement

Transcrição

[00:00] Olá, aluno. Tudo bom? Anteriormente nós vimos como que nós vamos fazer para pegar uma conexão do nosso banco de dados a partir da nossa aplicação. Para isso nós usamos uma classe de teste, chamada `TestaConexao`, onde nós passamos a nossa string de conexão com as informações para a nossa aplicação achar o banco de dados.

[00:20] E apenas fechamos a conexão, porque nós vimos que os recursos do banco de dados, quando acessados pela aplicação, nós devemos fechá-los. Bom, esse código funcionou, se executarmos vamos ver que a nossa conexão está abrindo normalmente, só que a nossa ideia de abrir uma conexão com o banco de dados é para podermos fazer aqui as operações com banco de dados, Inserts, os Selects, os Updates.

[00:49] Enfim, tudo o que nós executamos no banco de dados nós temos que fazer aqui, a partir da nossa aplicação. E eu quero começar aqui trazendo as informações que nós gravamos no banco de dados, lá quando estávamos configurando a nossa Database. Então se viermos no MySQL 8.0 Command Line Client, vou botar a minha senha aqui.

[01:11] Eu vou usar aqui a nossa Database que nós criamos, que é a nossa `use loja_virtual`. Se eu fizer um `select * from produto;`, nós vamos ver que nós temos dois produtos. E é esse produto que eu quero trazer na minha aplicação. Então como nós vamos fazer? Aqui no Eclipse, dentro do nosso projeto, em "loja-virtual-repository > src > (default package)", à esquerda, eu vou criar uma nova classe, que vai se chamar "TestaListagem".

[01:48] Essa classe vai ter um `public static void main(String[] args)` e nós vamos copiar esse conteúdo da classe "TestaConexão". Nós vamos precisar tanto da string de conexão quanto fechar a nossa conexão, então só preciso dar um "Ctrl + C" no código da classe "TestaConexao" e um "Ctrl + V" no código da nova classe. Agora temos o necessário para pegar essa conexão com o banco de dados.

[02:10] Vou adicionar o `throws SQLException`, que nós vimos que é necessário, porque o SQL, ele pode, nessa comunicação com o banco de dados, ele pode dar o SQL Exception, então nós temos que avisar para quem for chamar esses métodos aqui. Então no nosso caso, o main, ele tem que avisar aqui, esse código pode dar um SQL Exception. Como que nós fazemos então para usar os comandos de banco de dados na nossa aplicação?

[02:36] Aquela cláusula que nós usamos no nosso banco de dados, no MYSQL, o `select * from`, o `select` e outros campos que nós queremos trazer para dentro da nossa tabela, eles são considerados no mundo Java como Statements. E como eu faço para criar um Statement? Quando recuperamos a conexão, eu vou ter um método chamado `con.createStatement();`.

[03:05] Esse comando, ele me devolve um Statement. Nós vamos ver aqui, eu tenho uma interface, cuidado para não confundir, que eu tenho uma `Statement` do MySQL, eu quero a "Statement - java.sql", que é o nosso pacote JDBC. Esse `Statement stm = con.createStatement();` agora me devolve um Statement. Como que eu faço então para criar a minha cláusula SQL, o meu Statement SQL?

[03:35] Eu tenho aqui o meu Statement, a minha referência, e eu vou chamar o método `stm.execute();`. Então o que eu passo aqui, entre parênteses? A cláusula que eu quero, que é um select, eu quero trazer o ID do meu produto, o nome do meu produto e a descrição do meu produto. E eu apontei que é da minha tabela produto. Então fica `stm.execute("SELECT ID, NOME, DESCRICAO FROM PRODUTO");`.

[03:57] Então com essa linha nós fizemos mais ou menos o comando que nós executávamos no banco de dados, mas agora nós estamos trabalhando na nossa aplicação. Perfeito? Nós vimos que eu tenho 1 ou N produtos na minha tabela, eu posso ter vários produtos, conforme eu vou adicionando. Então, provavelmente, essa Query `SELECT`, esse nosso Statement vai nos devolver uma lista.

[04:21] Então vamos ver se ele devolve uma lista. Será que é uma lista de string? Vou fazer aqui então, vamos ver se é uma lista de string. Vou botar aqui `List<String> resultados = stm.execute("SELECT ID, NOME, DESCRICAO FROM PRODUTO");`. E se eu fizer esse comando, ele vai dar uma falha na compilação e quando nós vamos ver, ele vai falar que não pode converter um tipo booleano para um tipo string.

[04:42] Então significa que essa linha nos devolve um booleano? Estranho, mas vamos mudar então para o tipo boolean, para vermos o porquê que esse comando nos devolve um booleano. Então fica `boolean resultados = stm.execute("SELECT ID, NOME, DESCRICAO FROM PRODUTO");`. Aparentemente compilou agora. E qual é a motivação desse código, que nos devolveu um booleano?

[05:04] Bom, quando vamos trabalhar com banco de dados, nós temos aquelas cláusulas padrões que nós utilizamos, as mais conhecidas, que é o select, o insert, o update, o delete. E o que acontece? O `.execute`, ele vai nos retornar um booleano true quando o retorno do meu Statement for uma lista.

[05:26] Ou seja, quando eu fizer uma operação com o banco de dados, a partir da minha aplicação, e esse resultado for a lista, que no nosso caso é um select. E quando o retorno for diferente de uma lista, por exemplo, um delete, que não me retorna nada, um insert, que não me retorna nada, um update, que não me retorna nada, esse booleano vai ser *false*.

[05:46] Então, para testarmos isso, eu posso dar um `System.out.println(resultado);` e nós vamos ver se realmente ele nos retorna

um true. Se eu mandar executar esse código, vou clicar com o botão direto, dar um "Run as > Java Application" e está aqui o resultado true, porque de fato ele é uma lista. Só que para eu pegar essa lista, eu não pego diretamente aqui igual nós queríamos, uma lista de string. Como eu tenho que fazer?

[06:12] Eu vou tirar essa variável resultado daqui, `boolean resultados = stm.execute("SELECT ID, NOME, DESCRICAO FROM PRODUTO");` e nós vamos utilizar o `Statement` para pegar os resultados dessa lista. Para eu pegar os resultados dessa lista com `Statement`, eu tenho um `getResultSet();`. Com esse `stm.getResultSet();`, eu tenho também uma interface chamada `ResultSet`, também do pacote java.sql, do nosso JDBC.

[06:44] E com `ResultSet rst stm.getResultSet();` eu consigo pegar todo o conteúdo dos meus produtos adicionados à minha tabela. Então você pode ver que agora está compilando. E como que eu faço então para pegar os meus resultados da minha tabela? Eu tenho que verificar se quando eu for buscar, na minha tabela, se eu tenho um próximo. Então como que eu faço?

[07:11] Eu quero pegar o primeiro produto. Beleza, eu pego o primeiro produto. Eu tenho um próximo? Tenho. Então vai buscar o próximo produto. Eu tenho um próximo? Então eu busco o próximo produto. Quando não tiver mais produtos, ele sai do laço. Então a palavra-chave aqui é um laço e se eu tenho próximo. Eu posso usar como laço aqui um `while`.

[07:33] E, para facilitar a nossa vida, o próximo que sempre vamos procurar o `ResultSet`, ele já nos fornece com esse método `while(rst.next()) {}`. Então ele vai pegar na primeira posição. Tem um próximo? Para ficar mais claro, vamos no MySQL 8.0 Command Line Client. O ID vai estar como se fosse uma posição zero aqui na tabela. Quando eu chamo o `(srt.next());`, ele vai perguntar: tem um próximo cara que vai ser o primeiro produto? Tem.

[08:01] Então pega esse cara, o primeiro produto. Tem um próximo cara? Tem. Então pego esse segundo produto. Tem um próximo? Não. Então eu saio do laço. Então, dessa forma, nós conseguimos trazer os dois produtos. Mas ainda

não terminamos, precisamos pegar os atributos, nós precisamos montar aqui, na verdade, os atributos no mundo Java com as informações do MySQL que nós temos na tabela produto. Então, como eu faço isso?

[08:26] Nosso primeiro atributo é o ID, então eu pego ele como `integer id`. E o `rst`, o ResultSet, vamos ter um `srt.getInt` está vendo aqui? E nós temos duas formas de buscar esse `.getInt`, que é de fato, a coluna que queremos pegar. Então eu quero pegar o resultado da coluna ID, eu tenho duas formas para fazer isso: informando com `(int columnIndex)`, que no MySQL vai ser o primeiro Index ou com o `(String columnLabel)`.

[09:09] Para ficar claro, o que acontece? Eu vou pegar o `(String columnLabel)` e vou passar o ID. Se eu quisesse pegar pelo `(int columnIndex)`, o Index do nosso ID é o 1. Lembrando que é diferente de uma lista do Java, por exemplo, que começa no Index 0, aqui vai começar do Index 1. Então tenho o ID 1, o Index 1, o nome o Index 2 e a descrição o Index 3.

[09:36] Se eu quiser pegar pelo Label, eu vou usar ID, nome e descrição, e eu posso utilizar ele como ID maiúsculo que vai funcionar. Se eu fizer um `Integer id = rst.getInt("ID");` e `System.out.println(id);`, nós vamos ver o ID. Pra eu pegar o nome, o nosso conteúdo vai ser uma string, então eu uso o `String nome = rst.getString();`. Então você vê que é bem fácil de trabalhar com a recuperação dos dados da nossa tabela a partir da nossa aplicação.

[10:04] E eu informo aqui o nome, o `column label` da nossa tabela. Então fica `String nome = rst.getString("NOME");`. E depois também dou um `System.out.println(nome);` passando o nome. E, por último, que nós precisamos, é a nossa descrição, que também vai ser um `rst.getString()`, e vamos passar ele como `String.descricao = rst.getString("DESCRICAO");`.

[10:33] Se eu der um `System.out.println(descricao);`, nós vamos pegar a descrição e vamos então printar na nossa tela. Então, dessa forma você vê que o método está bem explicado. Eu executo a Query em `stm.execute("SELECT I`

`NOME, DESCRICAO FROM PRODUTO");` , pego o resultado em `ResultSet rst = stm.getResultSet();` .

[10:48] Faço o laço nesse resultado, em `while(rst.netx()){` e imprimo o ID, o nome e a descrição do meu produto, que está na tabela. Se eu mandar executar esse laço, vou dar um "Run as > Java Application", ele vai nos trazer os dois produtos que estavam na nossa tabela. Então agora já temos o nosso primeiro método de operação, fazendo mesmo a operação com o nosso banco de dados a partir da nossa aplicação.

[11:14] O intuito agora é que continuemos com esse trabalho, continuemos realizando as outras operações, como delete, insert, entre outras que nós vamos ver ao longo do curso. Mas nessa aula, a nossa intenção era criar aqui a listagem. Dou a aula como concluída e eu vejo você no próximo vídeo. Obrigado, aluno.