



Testando o pool de conexões

Transcrição

[00:00] Fala, aluno. Tudo bom? Vamos dar continuidade no assunto Pool de conexões. Até agora nós vimos o quê sobre o Pool de conexões? Nós vimos como ele funciona, para isso nós fizemos um desenho mostrando que o Pool de conexões, ele já tem um número X de conexões abertas para processar as requisições.

[00:24] Após entendermos esse desenho, nós partimos para o código e usamos o C3P0 para configurar o nosso Pool. Tem a interface DataSource agora, que é ela quem vai expor as informações do nosso Pool de conexões. Só que até agora, nós não conseguimos ver de fato o Pool de conexões funcionando. Como é que ele funciona?

[00:47] Beleza, eu vi que eu tenho um número X de conexões já abertas e que essas requisições, quando forem feitas, já vão ser processadas pelo Pool. Mas eu não vi isso funcionando. O que eu preciso fazer? Então, para conseguirmos testar o nosso Pool de conexões, a primeira coisa que teremos que fazer é ir na `ConnectionFactory`, nas configurações do Pool.

[01:11] Eu vou ter que *setar* o número máximo de conexões que eu quero permitir que sejam abertas. Então eu tenho um parâmetro, um método: `.setMaxPoolSize()`. Eu vou definir ele como `comboPooledDataSource.setMaxPoolSize(15);`. Então quando eu instanciar a minha Connection factory, que ele carregar essas minhas informações no Pool, eu quero que ele já carregue o Pool com 15 conexões disponíveis.

[01:40] Mas como eu testo? Então eu deixei uma classe `public class TestaPoolConexoes` criada e ela só tem o `main`. Então vocês vão criar essa classe no projeto de vocês e para isso nós vamos instanciar a `ConnectionFactory`, que vou chamar de `connectionFactory`, e vou instancia-la: `ConnectionFactory connectionFactory = new ConnectionFactory();`.

[02:03] Quando eu instancio, mais uma vez nós vamos ver que ele já vai carregar então todas as informações do nosso Pool de conexões. Só que eu quero fazer o quê? Eu quero fazer um laço, fazer um `for (int i = 0; i < 20; i++)`, e eu quero requisitar 20 conexões. Eu quero fazer 20 requisições para o meu Pool de conexões.

[02:25] Como que esse `ConnectionFactory.recuperarConexão();` vai se comportar? Deixa só eu adicionar o `throws SQLException`. Então eu instanciei o meu `ConnectionFactory`, peguei as informações do meu Pool e agora eu estou solicitando 20 conexões. Mas eu só tenho 15 conexões disponíveis no meu Pool.

[02:48] Vamos ver como é o comportamento do Pool de conexões. Então eu vou botar aqui `System.out.println("Conexão de número: " + i);` e vou concatenar a nossa variável que está contando o número de conexões abertas no nosso Pool - com o nosso banco de dados, na verdade. Então vamos lá.

[03:16] Se eu mandar executar esse `TestaPoolConexoes`, vou dar um "Run As > Java Application", ele vai nos trazer as informações e está aqui o nosso Pool de conexões funcionando. Veja bem, eu tenho 15 conexões na minha configuração do Pool, então tenho 15 conexões disponíveis.

[03:37] No teste, eu solicitei 20 requisições, eu falei para ele me fazer 20 requisições. Como que isso vai acontecer? Mais uma vez testando. Está aqui, então ele só abriu de fato as 15 conexões com o nosso banco de dados, Você pode estar me perguntando: como é isso na vida real? Como funciona? E as outras requisições?

[04:02] As outras requisições, elas vão esperar o processamento das que já estão ocupando, digamos assim, uma das conexões do Pool que estão abertas. Então essa é aquela maneira que nós vimos, nós evitamos abrir descontroladamente várias conexões e vamos limitar essa conexão. Só que não vamos enfileirar igual nós enfileiraríamos se tivéssemos apenas uma conexão aberta, porque o processamento é muito rápido.

[04:33] Quando processar a requisição que está na primeira conexão aberta, processou, se tiver alguém chegando para ser enfileirado, quando ela já for liberada, essa que estava chegando já pode utilizar uma conexão aberta. Então seria mais ou menos esse o mecanismo, nós chegamos a ver isso no desenho.

[04:56] Se eu vir no MySQL 8.0 Command Line Client, eu já tinha feito uns testes, chamando o `show processlist;`. Só que agora é basicamente isso, se eu mandar executar o `show processlist;` de novo, eu vou ter 30 conexões, porque eu executei duas vezes o meu Pool de conexões. Então se nós contarmos, vai ter 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30.

[05:32] Lembrando que essa primeira da lista não é da nossa Database loja virtual e essa última, está como "starting", é o próprio MySQL 8.0 Command Line Client. Então ele abriu as 30 requisições porque nós chamamos duas vezes a nossa `TestaPoolConexoes`. Então está aqui um exemplo de como funciona o Pool de conexões.

[06:00] Esse é um teste que dá para vocês brincarem bastante aumentando o número máximo do Pool, pode colocar - temos outros parâmetros, como o número mínimo do Pool. Então, como eu já tinha falado, a documentação também da C3P0 tem explicando todos os métodos, o que cada um faz, o que cada configuração vai mudar e no seu Pool de conexões.

[06:28] Então eu recomendo que você dê uma lida nessa documentação para poder, de fato, aproveitar todos os recursos desse Pool de conexões. Hoje foi mais para mostrar mesmo como ele funciona, limitando o número de

conexões, fazer um teste fazendo mais requisições que o número máximo disponível de conexões.

[06:51] Enfim, foi mais para nós brincarmos mesmo um pouco, ver o Pool de conexões funcionando, mas nada impede vocês de estudarem a documentação e implementarem do jeito que vocês gostarem na aplicação de vocês. Então é isso, aluno. Espero que vocês tenham gostado e até o próximo vídeo.