Team 85 Project 5 Report

Task 1:

**Connection Pooling:**

Connection Pooling is used to enhance the performance of executing commands on database and make the web applications perform much more efficiently. By reusing the connections through the caching of the connections, we are able to perform the database connections much faster and much more efficiently.

To use the Connection Pooling, we simply overrode the Connection made in the earlier projects. Also, we added the context.xml in the /WebContent/META-INF/context.xml file to add the following line:

<Context path="/project3">

<br/>

<lt;Resource name="jdbc/moviedb" auth="Container" type="javax.sql.DataSource"

initialSize="40" maxActive="100" maxIdle="80" minIdle="40” minEvictableIdleTimeMillis="55000" timeBetweenEvictionRunsMillis="34000"

testOnBorrow="true" validationInterval="60000" validationQuery="SELECT 1"

removeAbandoned="true" removeAbandonedTimeout="55" maxWait="10000"

username="mytestuser" password="mypassword” driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/moviedb?autoReconnect=true&amp;useSSL=

false&amp;cachePrepStmts=true" />

<br/>

</Context>

We also had to add the following code into the /WebContent/WEB-INF/web.xml before the servlet declarations:

<resource-ref>

<description>

Resource reference to a factory for java.sql.Connection

instances that may be used for talking to a particular

database that is configured in the server.xml file.

</description>

<res-ref-name>jdbc/moviedb</res-ref-name>

<res-type>javax.sql.DataSource</res-type>

<res-auth>Container</res-auth>

</resource-ref>

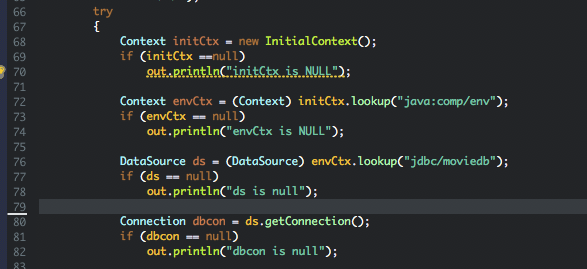
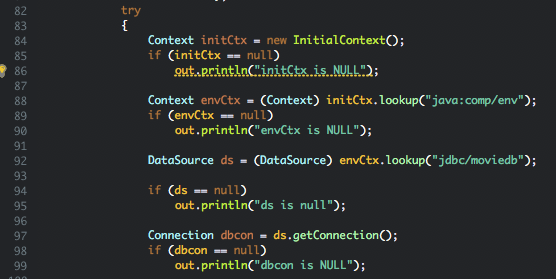
In addition, we had to import the following packages from the Java Library to allow the Connection Pooling in each of our Servlets:

import javax.naming.InitialContext;

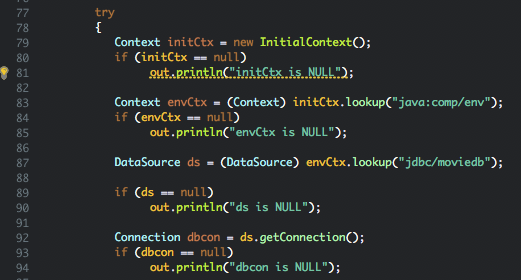
import javax.naming.Context;

import javax.sql.DataSource;

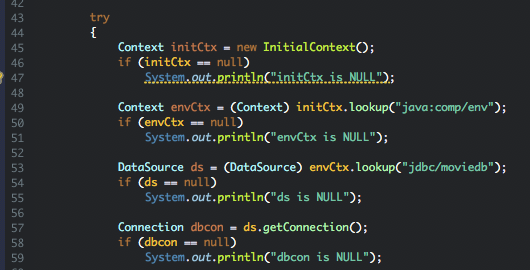
/project3/src/add\_movie.java: /project3/src/addstar.java

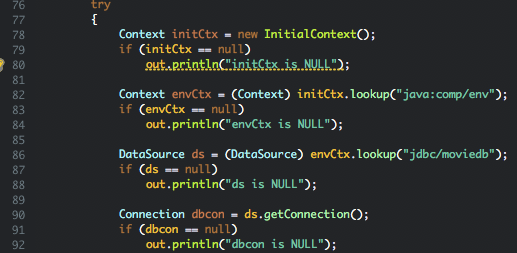
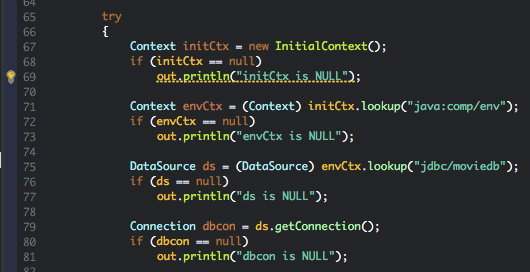
/project3/src/browsegenre.java /project3/src/checkccinfo.java

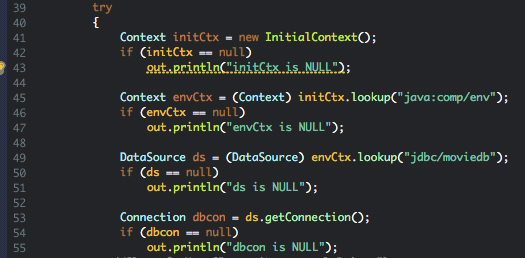
/project3/src/dLogin.java /project3/src/finalcheckout.java

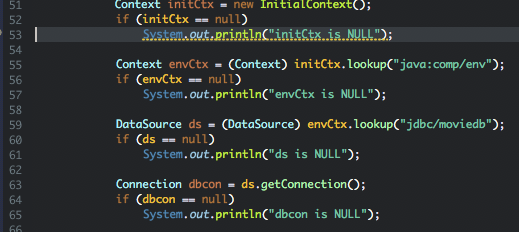
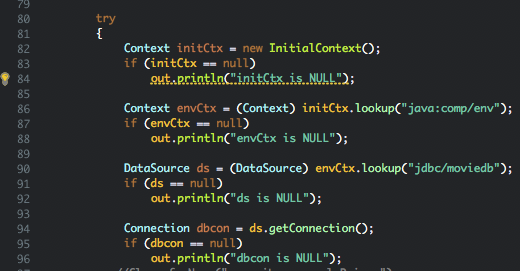
/project3/src/genresearch.java /project3/src/login.java

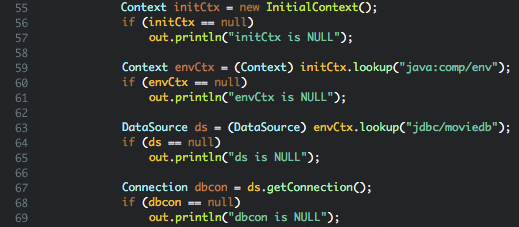
/project3/src/loginapp.java /project3/src/metadata.java

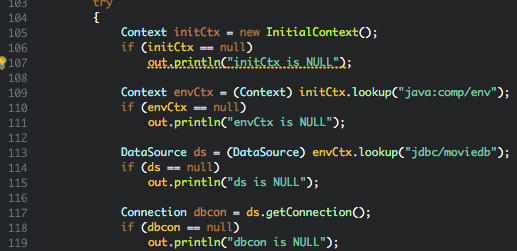
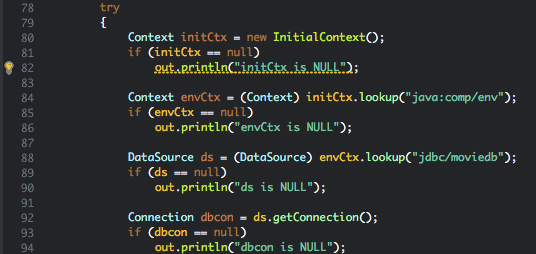
/project3/src/movie\_suggestion.java /project3/src/movieinfo.java

/project3/src/search\_page\_app.java /project3/src/searchpage.java

/project3/src/shoppingcart.java /project3/src/starinfo.java

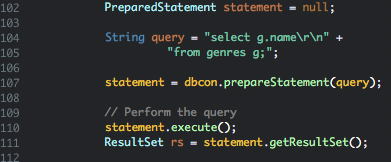
 

**Prepared Statements:**

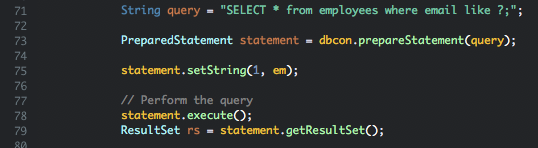
To reduce the execution time of the SQL queries every time we log into the MySQL Server, we used the Prepared Statements. The main benefit of the PreparedStatement is that it sends the query to the DBMS right away where it is compiled.

We implemented this by adding the “import java.sql.PreparedStatement;” package to all the servlets that use the search method and implemented the following lines:

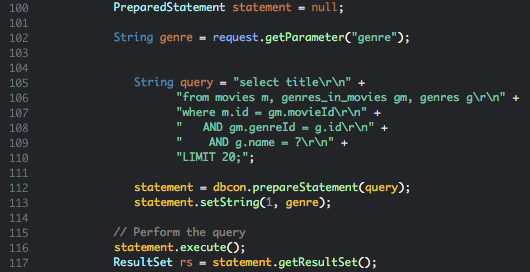
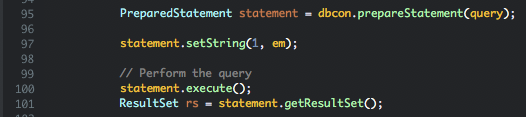
/project3/src/browsegenre.java /project3/src/checkccinfo.java

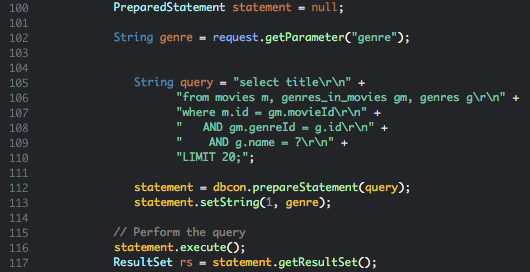
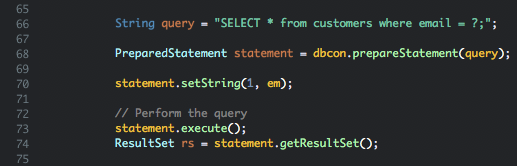
/project3/src/dLogin.java /project3/src/finalcheckout.java

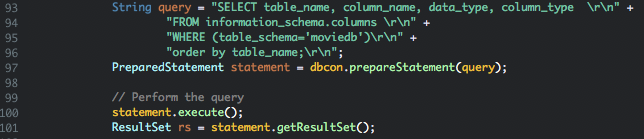
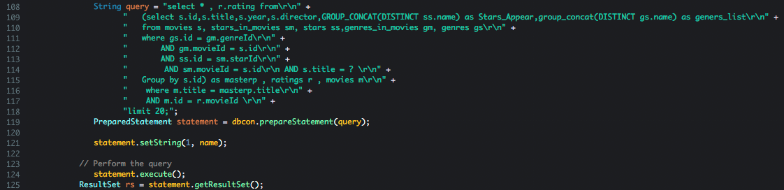
/project3/src/genresearch.java /project3/src/login.java

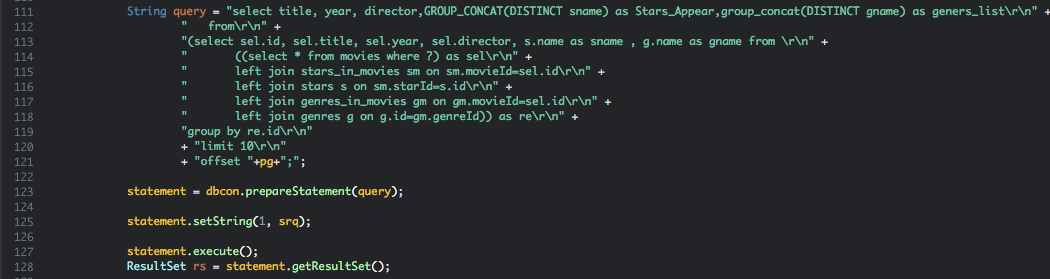
/project3/src/loginapp.java /project3/src/metadata.java

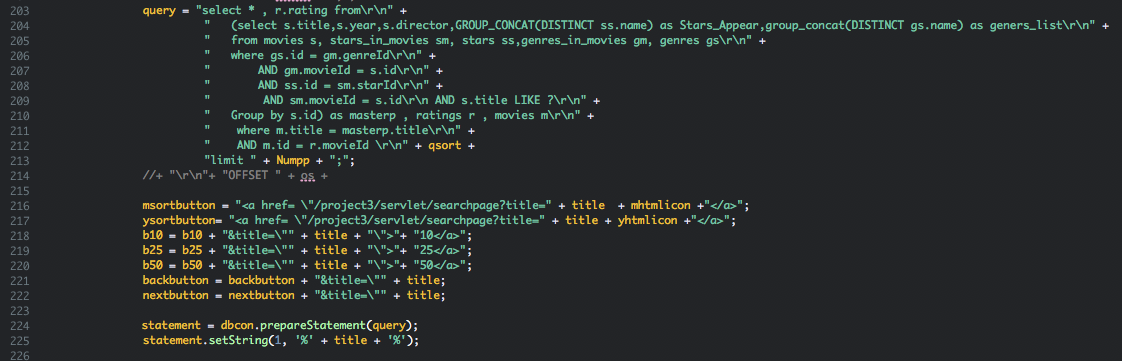
 

/project3/src/movieinfo.java /project3/src/search\_page\_app.java

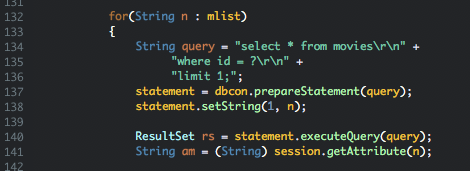
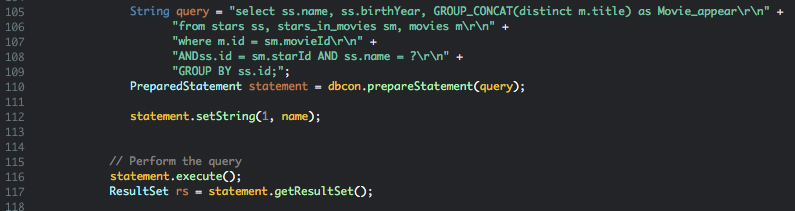
 

/project3/src/searchpage.java





/project3/src/shoppingcart.java /project3/src/starinfo.java

Task 2

Address of AWS and Google instances

AWS address: 52.53.153.231/project3/servlet/welcome

Google address: 35.230.118.114/project3/servlet/welcome

Have you verified that they are accessible? Does Fablix site get opened both on Google’s 80 port and AWS’ 8080 port?

Yes, I had verified that they are accessible.

How connection pooling works with two backend SQL?

The java code has 2 pooling resources that are the master and slave server. The pooling for these two connections allows the servlet to choice which connection is open when doing query operation. The scaled instances use two backend servers that each of them can connect to the master and slave mysql server.

File name, line numbers as in Github

Context.xml in the project3/WebContent/META-INF

Line 4 - 13

Snapshots



How read/write requests were routed?

Read requests are all routed to both master and slave mysql servers. Write request is only routed to the master. Read requests are sent through a preparedstatement, which is to increase the performance and query time.

File name, line numbers as in Github

/project3/src/add\_movie.java - line 61-73 and 84-86

/project3/src/add\_star.java - line 70-82 and 87-89 and 121-130

/project3/src/browsegenre.java - line 77-89 and 98-99

/project3/src/checkccinfo.java - line 58-70 and 84-89 and 124-133

/project3/src/dLogin.java - line 48 - 60 and 67-69 and 92-100

/project3/src/finalcheckout.java - line 57 - 69 and 85-87

/project3/src/genresearch.java - line 77-89 and 101-103

/project3/src/login.java - line 67 -79 and 89-90 and 110-119

/project3/src/loginapp.java - line 36-48 and 58-60 and 76-85

/project3/src/metadata.java - line 68-80 and 91-93

/project3/src/movie\_suggestion.java - line 44-52 and 102 - 105

/project3/src/searchpage.java - line 93-106 and 322 - 326

/project3/src/shoppingcart.java - line 102-114 and 128-130

/project3/src/starinfo.java - line 81-93 and 106-108

Snapshots

from login.java

****

Task 3

Have you uploaded the log file to Github? Where is it located?  
The log files are uploaded and located at the data folder. They are .txt with there name representing the l\_? for Single-instances and 2\_? for scaled version.

Have you uploaded the HTML file to Github? Where is it located?

It is located in the /project3/WebContent File and it’s called jmeter\_report.html

This can also access through the scaled instances http://instance\_ip/project3/jmeter\_report.html

Have you uploaded the script to Github? Where is it located?

The calculation script is in the data folder, and it is written in Python

Have you uploaded the WAR file and README to Github? Where is it located?

The WAR file and the README files are located in the main repository file of the “cs122b-winter18-team-85.”