**Database Application Development**

**CSC317 Database Systems II**

**Professor: Dr. Kurt Diesch**

**By Odiscious Dozier**

**Trident University**

**CASE 03 - Java and DBMS II**

**Summary**

**Summary**

This assignment is an extension of the Case 03 assignment and was an absolute joy to work through. I did end up suffering from a head cold that lasted over a week and along with working 3 different jobs and interviewing for a 4th that will be replacing 2 of the 3 jobs, my car ended up breaking down. This assignment was not that difficult to accomplish on its own, but with the, afore mentioned, issues… this assignment has taken much longer than it needed to.

The work accomplished in this assignment was an add-on to Case 03, allowing for a user to query the database from a graphical user interface (GUI) via a web browser. The user is now able to retrieve records through the use of a filter, insert, update, and delete, records as they choose.

A connection to the database is created using the getConnection() function on the OracleDataSource object. SQL statements are then defined, created with the createStatement() function on the connection object. Then, the executeQuery() function is used to run queries on the database and the series of rows and columns that is returned is contained in the ResultSet object. This object is then used to display the data in the application’s pages.

One of the more common errors I came across when working with JDeveloper is that the hierarchical structure of the files within a given project are organized a bit different than other IDEs that I’ve become accustomed to like, Eclipse. In section 4.3 of the assignment, I had to modify the path using the browse button within the JSP dialog box.

An interesting take away from this portion of the assignment was the use of the JSP useBean tag. The UseBean tag was new for me and it seems to provide the ability to make static web pages dynamic.

In section 4.4.2, I had to find a different way to create a Result Set – the directions in the tutorial did not work for my JDeveloper instance. In the employees.jsp – Structure window, I had to navigate to the ‘page’ directory. Then, I clicked on ‘import’ in the Properties window. In doing so, the Edit Property: Import window popped up and then I selected the Hierarchy tab and navigated from java to sql to sql to ResultSet and clicked ‘ok’. The ResultSet was now created.

In section 4.5.4, I was unable to find a way to open the Scriplet Properties to modify the Script I had already written. After a quick bit of searching around, I realized I could probably find the source code and modify it. I did so by clicking on the ‘source’ tab located next to the design tab and was able to make the necessary modifications.

In section 5.2.1, creating a method to identify an employee record, the tutorial had left out the last curly brace. After a bit of troubleshooting, I came back to the code segment and self-compiled the given function. It didn’t take me a long time to resolve this issue and I was pleased to move on.

Additionally, the job ID was skipped for this section of the tutorial. After I added to logic to the program, I was able to move on as expected.

In section 5.2.5, the tutorial suggests navigating to the Page Directive at the top corner of the page. This didn’t work in my case, I had to, navigate to the update\_action.jsp – Structure window, click on ‘page’, and then go to ‘import’ on the Properties window to successfully complete the task.

Additionally, I had experienced another path issue in this section. The updateEmployee() function was broken for a significant amount of time because, I had placed the update\_action.jsp file in the wrong directory. Once resolved, I was able to modify Steven King to Stevena King and back again.

Fatigue had taken its toll on my, in this project, and I had made a very simple oversight that cost me a significant amount of time. In section 5.4.2, the tutorial calls for adding a link to delete an employee to the main table in the application. After inserting the code appropriately, I expected to see the edit/delete column header change, it did not; instead, after needlessly troubleshooting for an extended period of time, I realized that all of the rows now included the ‘delete’ link next to the ‘edit’ link.

In section 7.2, I struggled with getting my program to close the open objects; for example, when I had launched the Web logic server, by running the program, the web browser opened to the login.jsp page, but when clicking submit after entering the appropriate data an error page was displayed. This was caused by the ResultSet closing too soon. After troubleshooting, I had realized that this error was caused by the empsbean.closeAll() function being placed inside the while loop that was trying to populate the table from the ResultSet. I quickly moved the empsbean.closeAll() function outside of the while loop and the application now operates successfully .

REFERENCES

4 Querying for and Displaying Data. (2016). Retrieved March 03, 2016, from <http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/querdata.htm#TDPJD148>

5 Updating Data. (2016). Retrieved March 07, 2016, from <http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/upddata.htm#TDPJD186>

7 Getting Unconnected from oracle Database XE. (2016). Retrieved March 18, 2016, from

<http://docs.oracle.com/cd/E17781_01/appdev.112/e18805/getuncon.htm#TDPJD249>