##### APT1050B : Database Management Systems

**Lab #6:** Working with Java, JSP and Oracle Connectivity

*Instructor: G. Chege, PhD Given on 20th July 2022*

**Complete this Lab using NETBEANS and ORACLE and submit your results via blackboard.**

We need to develop a three tier client server web solution using a browser as the front end. Tools required include Java, JSP (both in the Netbeans IDE) and Oracle Express Edition (11g or 12g).

Access Oracle XE from a URL using <http://127.0.0.1:8080/apex/f?p=4950>

**Part I – How to Access DB Data and Display on a Browser**

1. Create an Oracle Database with a table called PARTS whose structure is:

**PARTS (PNO, PNAME, COLOR, WEIGHT, STATUS)**

Where:

PNO varchar (10),

PNAME varchar(20),

COLOR varchar (20),

WEIGHT varchar (15),

STATUS varchar (15)

Enter Sample data with about 10 records into Table **Parts**. (NB - STATUS has values: Available, On-Order, Out-of-Stock).

1. Now create a simple Web Application using **JSP in Netbeans**. Call the application

**javaDBJSPConnectivity.** Create a JSP file called javaDBJSPConnectivity.jsp and enter the code given below and save:

<%--

Document : javaDBJSPConnectivity

Created on : June 23, 2022, 03:16:27 PM

Author : Gerald Chege

--%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<%@ page contentType="text/html;charset=windows-1252" import="java.sql.\*"%>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=windows-1252"/>

<title>APT1050 JAVA/ORACLE DATABASE CONNECTIVITY </title>

</head>

<body>

<jsp:useBean id="dataBean" class="APT1050.DataHandler" scope="session"/>

<h2> APT1050 JAVA/ORACLE DATABASE CONNECTIVITY </h2>

<%

Connection conn;

ResultSet rset;

//String pno, pname, colour, weight, status

// get a connection

conn = dataBean.getDBConnection();

rset = dataBean.getData(conn);

%>

<table cellspacing="0" cellpadding="1" border="1" width="100%">

<% while (rset.next()) {

%>

<tr bgcolor="#00adff">

<td width="12%"> <%= rset.getString("PNO")%></td>

<td width="12%"> <%= rset.getString("PNAME")%></td>

<td width="12%"> <%= rset.getString("COLOR")%></td>

<td width="12%"> <%= rset.getString("WEIGHT")%></td>

<td width="12%"> <%= rset.getString("STATUS")%></td>

</tr>

<%} %>

<% rset.close();%>

</table></body>

</html>

1. Now create a Java class to retrieve data from your Oracle database. Call the class **DataHandler.java**. A sample code is given below (NB- Netbeans will be explained in class). Add the provided Oracle connection driver java archive (jar) into the Library.

/\*\*

\*

\* @author gchege

\*/

package APT1050;

import java.sql.\*;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import oracle.jdbc.pool.OracleDataSource;

import java.sql.Statement;

import javax.servlet.http.HttpSession;

import java.lang.StringBuffer;

public class DataHandler {

// class constructor method

public DataHandler() {

}

// in general connection declared as:

String jdbcUrl = "jdbc:oracle:thin:@127.0.0.1:1521:XE";

//String userid = "SYSTEM";

//String password = "123456"; //We need to confirm password (usually the default is called oracle)

String userid = "APT1050"; //name of workspace

String password = "gchege"; // workspace pwd

Connection conn;

Statement stmt;

ResultSet rset;

String query;

String sqlString;

/\* User Authentication done here – this is the general format for production systems:

Method authenticateUser checks if the userid, password, and host values supplied by a user are valid:

requires import of the javax.servlet.http.HttpSession class

\*/

/\*public boolean authenticateUser(String jdbcUrl, String userid, String password,

HttpSession session) throws SQLException {

this.jdbcUrl = jdbcUrl;

this.userid = userid;

this.password = password;

try {

OracleDataSource ds;

ds = new OracleDataSource();

ds.setURL(jdbcUrl);

conn = ds.getConnection(userid, password);

return true;

}

catch ( SQLException ex ) {

System.out.println("Invalid user credentials");

session.setAttribute("loginerrormsg", "Invalid Login. Try Again...");

this.jdbcUrl = null;

this.userid = null;

this.password = null;

return false;

}

}

\*/

// our getDBConnection connection method defined here

public Connection getDBConnection() throws SQLException {

try {

OracleDataSource ds;

ds = new OracleDataSource();

ds.setURL(jdbcUrl);

conn = ds.getConnection(userid, password);

} catch ( SQLException ex ) {

//logException( ex );

}

return conn;

}

// READ DATA from your Database (e.g. from PARTS Table)

public ResultSet getData(Connection conn) throws SQLException{

try {

stmt = conn.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,

ResultSet.CONCUR\_READ\_ONLY);

query = "SELECT PNO, PNAME, COLOR, WEIGHT, STATUS FROM PARTS";

rset = stmt.executeQuery(query);

}

catch ( SQLException ex ) {

//logException( ex );

}

return rset;

}

}

1. Now run your application and see sample results.

**Part II Lab Assignment**

Modify your application so as to:

i) Include two other tables called **SUPPLIER** and **SHIPMENT**

where the relations are:

**SUPPLIER** (SNo, SName, City, Status) – all as VARCHARs

The primary key for relation SUPPLIER is SNo.

and

**SHIPMENT** (PNo, SNo, quantity) – all as VARCHARS;

The primary key for relation SHIPMENT is (PNo, SNo).

Supplier is related to Parts through Shipment :

Parts -> Shipment (1 to M)

Supplier -> Shipment (1 to M)

Enter sample data into the new tables.

(ii) Design a web form that allows you to insert data records into any of the three tables (use the SQL INSERT statement);

iii) design another form that allows you to retrieve data meeting two different criteria of your choice.