IBM Inter-University
Programming Contest 2012

Question 1 - Multiple Choice

1.1, By using which of the following features it is possible to improve the speed of a RDBMS from disk speed to memory speed?

A. Bufferpool
B. Instance
C. Table
D. SMS

1.2, Integrated Development Environment (IDE) consists of:

1) Source code editor
2) Compiler and/or an interpreter
3) Code visualisation tool
4) Build automation tools
5) Debugger or Profiler
6) Version Control tool or adaptor
7) Runtime test environment integration

A) 1, 2, 5 & 6
B) 1, 2, 3, 4 & 5
C) 1, 3, 5, 6 & 7
D) 1, 2, 3, 4, 5, 6 & 7

1.3, Which one is the correct Cloud deployment models?

a. IaaS, PaaS, SaaS
b. IaaS, QaaS, SaaS
c. PaaS, CaaS, SaaS
d. IaaS, PaaS, ZaaS

1.4, Which of the following statements regarding loggings is correct?

A. A message entry is a clear and concise message intended for end users and administrators to view, whereas a trace entry is a complex and detailed message intended for engineers or developers to use.
B. A message entry is a clear and concise message intended for engineers or developers to use, whereas a trace entry is a complex and detailed message intended for end users and administrators to view.
C. A message entry is a complex and detailed message intended for engineers or developers to use, whereas a trace entry is a clear and concise message intended for end users and administrators to view.
D. A message entry is a complex and detailed message intended for end users and administrators to view, whereas a trace entry is a clear and concise message intended for engineers or developers to use.

IBM Inter-University
Programming Contest 2012

1.5 Business Intelligence system does NOT help Retails store management

A. Find out insight from historical data
B. Discover trend of Sales pattern by season
C. Calculate discount for customer at cashier
D. Improve the efficiency for the marketing promotion

C 1.6, Which of the followings is NOT the responsibility of On-Demand Router (ODR)?

A. Keeping track of application/user/etc resource usage
B. Ensuring higher priority applications/users/etc are serviced first
C. DMZ XML firewalling and threat protection
D. Starting/Stopping JVM's to meet demand

D 1.7, Which of the following is NOT a feature of IBM WebSphere Portal?

A. Deliver highly personalized experience
B. Leverage existing investments
C. Deliver a front-end to Service Oriented Architecture (SOA)
D. Help software developers design and develop applications

1.8, Which is not a component in IBM SmartCloud Provisioning setup?

a. Storage Node
b. Compute Node
c. First Box
d. Firewall

C 1.9, What kind of technology that is using in Cloud Provisioning (eg IBM SmartCloud)?

a. Google File System
b. OpenSource.org Hadoop
c. Apache Hadoop
d. IBM GFS

1.10, Which of the following out-of-the-box feature of WebSphere Portal has rich text editor and helps you publish news, announcement and company information?

Marco
B
A. Links and Bookmarks Portlet
B. Web Content Manager
C. Web Application Integrator
D. Google Gadget Portlet

最好關 Web Sphere 商

IBM Inter-University
Programming Contest 2012

1.11, Which of the following is NOT benefit of Static Analysis

A) Understanding software component behavior by using data collected during the execution of the component.
B) Ensure that the source adheres to a predefined coding standard
C) Detect common performance problems
D) Understand the dependencies of the imports of each class

1.12, What are the characteristics pre-Cloud middleware have?

1 High TCO
2 Low utilization
3 High maintenance cost
4 Fast time-to-market

A. 1 & 3
B. 1, 2 & 3
C. 1, 2 & 4
D. All of the above

1.13, Which of the following is NOT design pattern state in Design Patterns: Elements of Reusable Object-Oriented Software

A) Factory Method
B) Bridge
C) Null Object
D) Command

1.14, Which of the following is unstructured data?

A. Sales transaction record for retails store
B. No of minutes usage per mobile device
C. Sickness description in Patient records
D. Average speed per trains in China for each track

1.15, The Benefits of Business Intelligence is/are:

A) Helps align the organization towards its key objectives
B) Improve Database performance by leveraging the temp tables
C) Monitor the servers performance by actively scanning the key components
D) All of the above

Question 2.1 (50 Points)

Topic

Login logic for our eCommence Portal.

Requirements

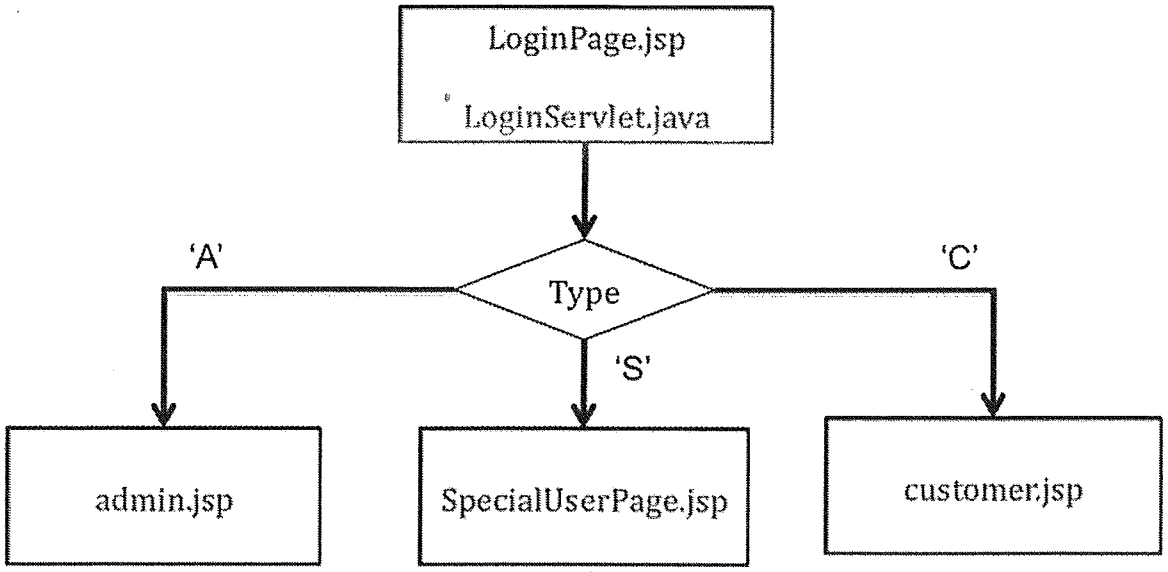
Add logic in the login screen to direct the user to the correct page based on his/her user type.

The login screen is implemented by *LoginPage.jsp* in which it calls the java program *LoginServlet.java*.

In *LoginServlet.java*, fill in the missing part to

- Redirect the admin user (with user type "A") to the page *admin.jsp*,
- Redirect the special user (with user type "S") to the page *SpecialUserPage.jsp*,
- Redirect the customer (with user type "C") to the page *customer.jsp*.

The above logic is illustrated in the following diagram:

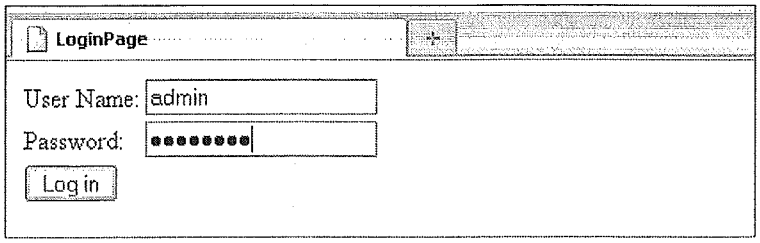


IBM Inter-University
Programming Contest 2012

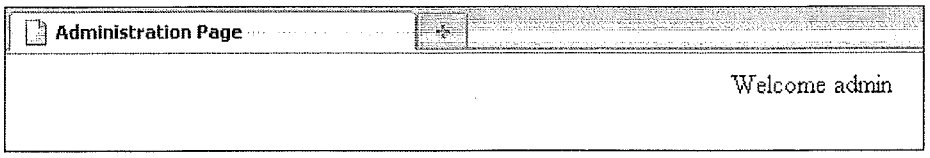
Target Results

To score points for this question, achieve the followings: (Score will be given based on the each type of user login)

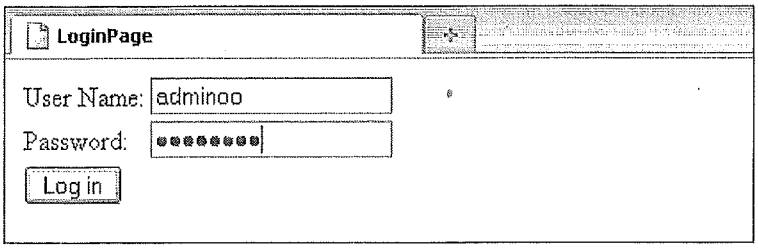
1. Log in the page using user name "admin" and password "password" in Firefox **http://localhost:9080/Q1Web/LoginPage.jsp**.



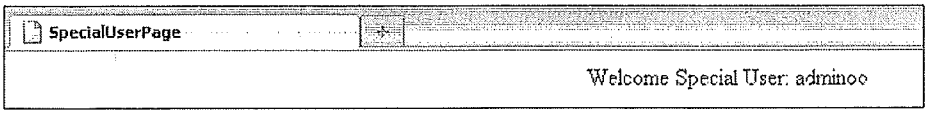
The user will be redirected the page *admin.jsp*:



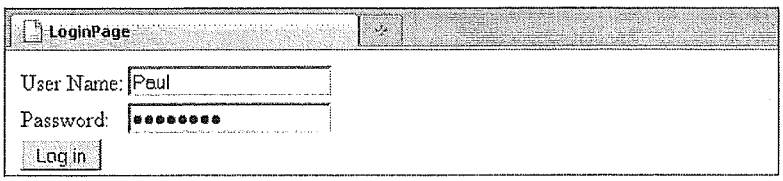
2. Login the page using user name "adminoo" and password "Abcdef12" in Firefox **http://localhost:9080/Q1Web/LoginPage.jsp**.



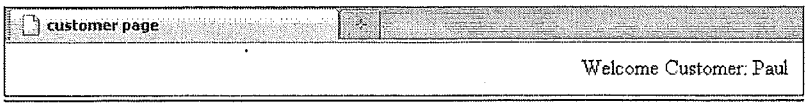
The user will be redirected to *SpecialUserPage.jsp*:



3. Login the page using user name "Paul" and password "password" in Firefox **http://localhost:9080/Q1Web/LoginPage.jsp.**



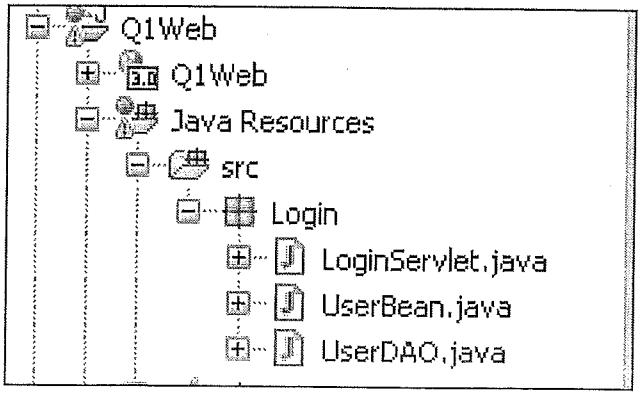
The user will be redirected to *customer.jsp*:



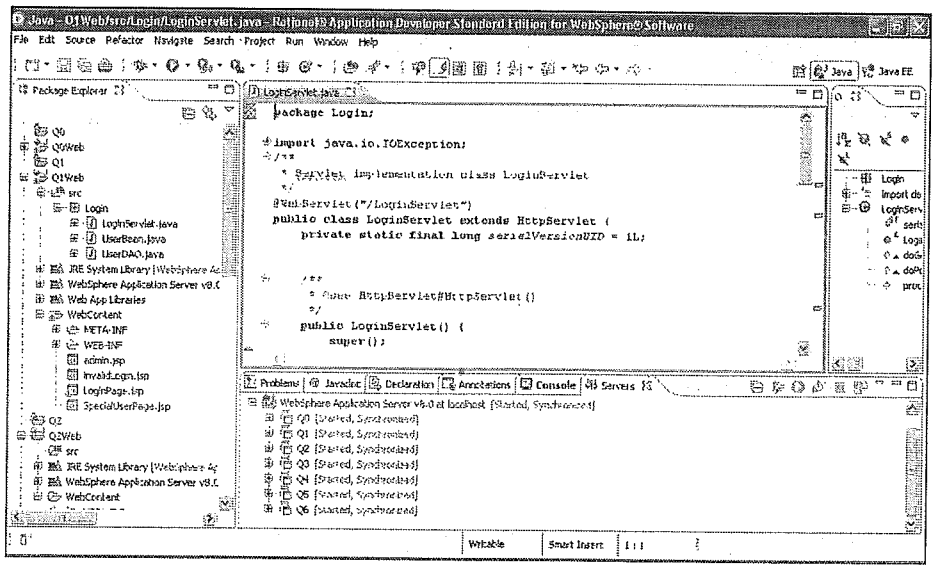
IBM Inter-University
Programming Contest 2012

Hints and tips

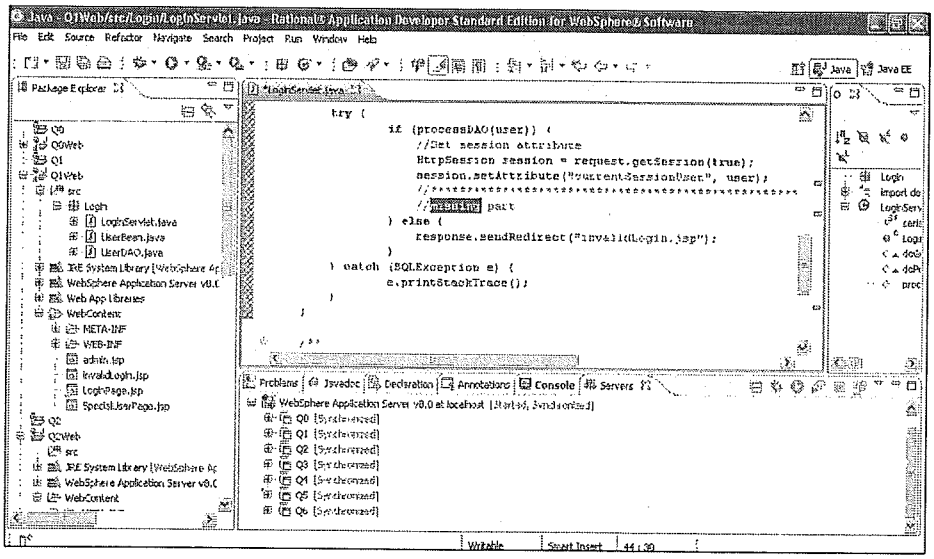
The program *LoginServlet.java* can be accessed under Q1Web in RAD Package Explorer on the left panel.



Double click to open *LoginServlet.java*:



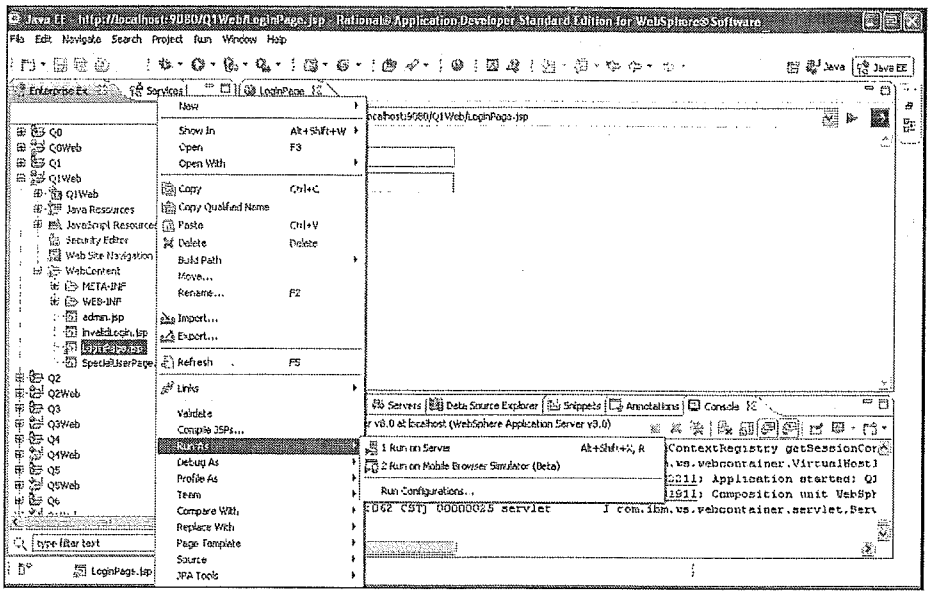
Press Ctrl-F to start the searching. Enter "missing":



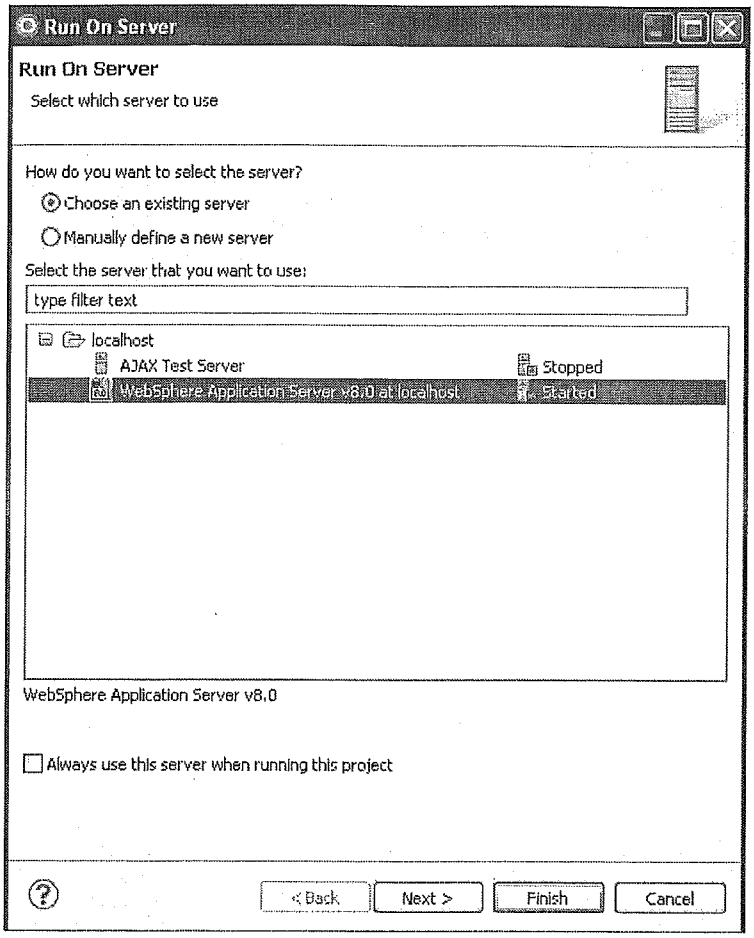
IBM Inter-University
Programming Contest 2012

Fill in the missing part with the conditional statement for checking user type of the login user and perform the page redirection.

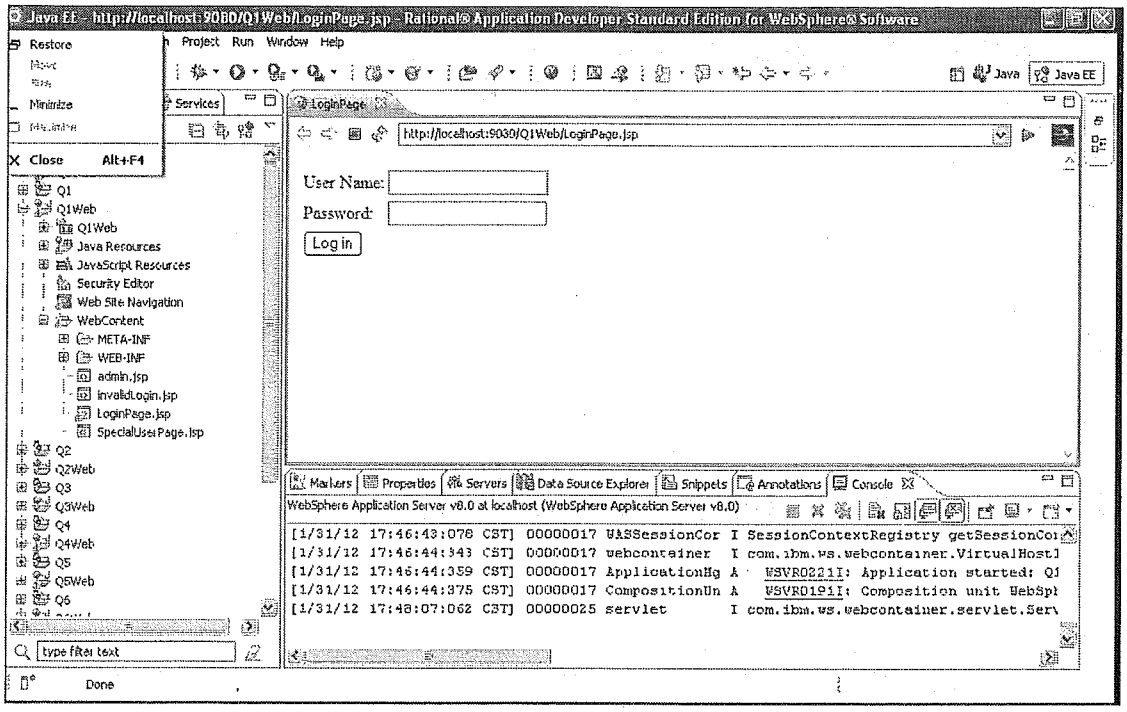
After completing your coding, you can see your result by right click on the *LoginPage.jsp* and select "Run as" -> "Run on server".



Select WebSphere Application Server v8.0 at localhost and press "Finish".



You will see the page appear in your console and you can start testing your result.



IBM Inter-University
Programming Contest 2012

Appendix

Source code for *LoginServlet.java*

```
package Login;

import java.io.IOException;
import java.sql.SQLException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
/**
 * Servlet implementation class LoginServlet
 */
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;


    /**
     * @see HttpServlet#HttpServlet()
     */
    public LoginServlet() {
        super();
        // TODO Auto-generated constructor stub

    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        //
        // The object "user" will store the username and password in the login screen
        //
        UserBean user = new UserBean();

        try {
            //
            // Obtain username and password and store in the object "user"
            //
            user.setUserName(request.getParameter("un"));
            user.setPassword(request.getParameter("pwd"));
            //
            // To check whether the username and password is valid
            //
            user = UserDAO.login(user);

            if (user.isValid()) {
                //
                //Set session attribute
                //
                HttpSession session = request.getSession(true);
                session.setAttribute("currentSessionUser", user);

    //*************************************************************************
***********
                // Missing part:
                // - You should redirect the user to the correct page based on the user type
                // - User type can be get using the function user.getType()

    //*************************************************************************
***********
            } else {
                response.sendRedirect("invalidLogin.jsp");
```

```
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
            // TODO Auto-generated method stub
        }
}
```

Question 2.2 (50 points)

Topic

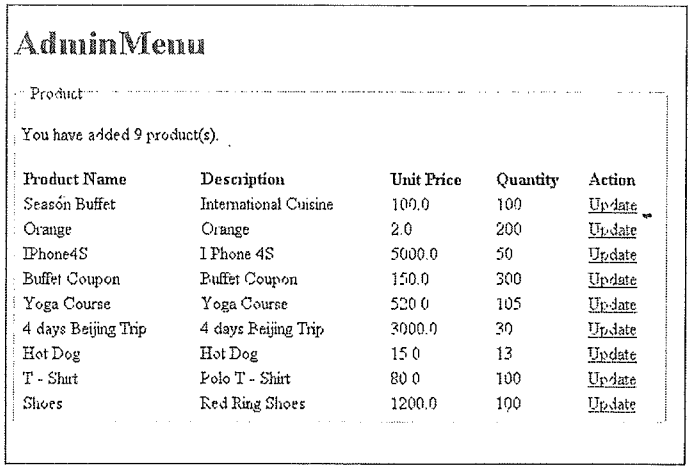
Update Product Information for Product Maintenance.

Requirements

Add logic to complete the administration menu for updating product information.

The administration menu consists of 2 webpages, which are *AdminMenu.jsp* & *Update.jsp*.

Run the administration menu using the *AdminMenu.jsp*.



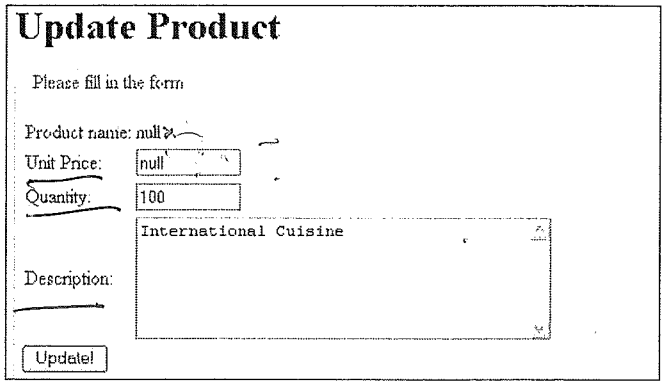
AdminMenu

Product

You have added 9 product(s).

Product Name	Description	Unit Price	Quantity	Action
Seasón Buffet	International Cuisine	100.0	100	Update
Orange	Orange	2.0	200	Update
IPhone4S	I Phone 4S	5000.0	50	Update
Buffet Coupon	Buffet Coupon	150.0	300	Update
Yoga Course	Yoga Course	520.0	105	Update
4 days Beijing Trip	4 days Beijing Trip	3000.0	30	Update
Hot Dog	Hot Dog	15.0	13	Update
T - Shirt	Polo T - Shirt	80.0	100	Update
Shoes	Red Ring Shoes	1200.0	100	Update

By clicking the "Update" links in the "Action" column, the *Update.jsp* will be executed and displays the details of the product. Two fields (with null value) have not receive the value selected from database as shown below.



Update Product

Please fill in the form

Product name: null

Unit Price: null

Quantity: 100

Description: International Cuisine

Update!

Update product
Set UnitPrice = " " decimal
QUANTITY = " "
Description = " " char
where ID

In *Update.jsp*, fill in the missing part to

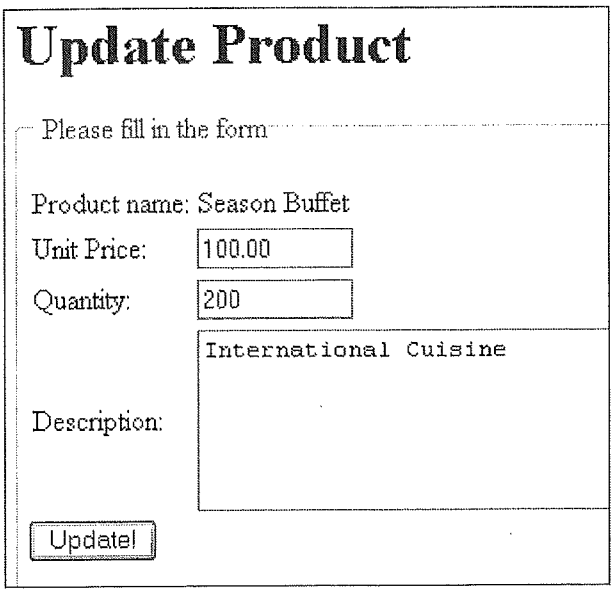
- Setup the two fields receive the value selected from database.

IBM Inter-University
Programming Contest 2012

Target Results

To score points for this question, achieve the followings: (Score will be given based on the result returned for different fields)

1. Access the administration menu in Firefox **http://localhost:9080/Q2Web/AdminMenu.jsp**.and press the "Update" link, the product information should be listed in the update form.

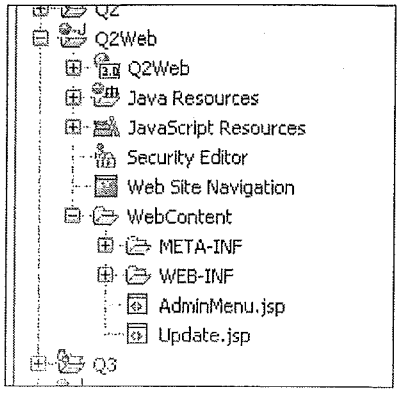


2. The program should update the product information on the Admin Menu, after inputting new Product information at the update form and press the update button.

IBM Inter-University
Programming Contest 2012

Hints and tips

Open *Update.jsp* and search for the missing part, you can use the steps described in Question1 to open the *Update.jsp* in the RAD console and search for missing part



Fill in the missing part for variables "Pname" and "Uprice" to get the value of result set.

```
}
if(Descip == null){
    Descip = rs.getString("DESCRIPTION");
}
/* missing part

Please Complete this missing part

This part is to receive the value of product name and unit price f
*/

if (rs != null) {
    rs.close();
```

After you completed your coding, you can use the steps described in Question1 to open the *Update.jsp* in the RAD console and test the result.

We assume that the data type will be inputted correctly. The program will not provide any data type checking process. For example, user will only input decimal with 7 digits at"UnitPrice".

Appendix

This is the source code for Update.jsp

```
<%@page import="java.lang.String, java.lang.StringBuffer" %>
<%@page import="java.sql.Connection, javax.sql.DataSource, java.sql.PreparedStatement" %>
<%@page import="java.sql.ResultSet, java.sql.Statement" %>
<%@page import="java.sql.SQLException, javax.naming.NamingException,
javax.naming.InitialContext" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"><%@page
      language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1" %>
<html>
<head>
<title>Update</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
</head>
<body>
<h1>Update Product</h1>
<div style='width:600px'>
<fieldset>
<%
try {

      // Receive Parameter
      String PID = request.getParameter("PID");
      String Pname = request.getParameter("Pname");
      String Descip = request.getParameter("Descip");
      String Uprice = request.getParameter("Uprice");
      String Quantity = request.getParameter("Quantity");

      if (PID != null && !PID.equalsIgnoreCase("") &&
      Pname != null && !Pname.equalsIgnoreCase("") &&
      Descip != null && !Descip.equalsIgnoreCase("") &&
      Uprice !=null && !Uprice.equalsIgnoreCase("") &&
      Quantity != null && !Quantity.equalsIgnoreCase("")) {
            //Update database ,Process after received Parameter


            InitialContext ctx = new InitialContext();
            //obtain data source from application server
            DataSource windowsDS = (DataSource)ctx.lookup("jdbc/ecomm");

            //Get connections from application server
            Connection windowsConn = windowsDS.getConnection();

            //Prepare statements in connection

            PreparedStatement pstmt = null ;
            pstmt = windowsConn.prepareStatement("update product set NAME='" + Pname +
            "' , DESCRIPTION='" + Descip +"' , UNITPRICE="+ Uprice + "," +
       "QUANTITY=" + Quantity + " where PROID = " + PID );

    boolean flag = pstmt.execute();
    if (flag == false) {
%>
<!-- Process if update successfully -->
<legend>The Product is successfully updated.</legend>
<table>
      <tr>
            <td><p>Product Name:</p></td>
            <td> <%= Pname %></td>
      </tr>
      <tr>
            <td><p>Description:</p></td>
            <td> <%= Descip %></td>
      </tr>
      <tr>
            <td><p>Unit Price:</p></td>
```

```
                <td> <%= Uprice %></td>
        </tr>
        <tr>
                <td><p>Quantity:</p></td>
                <td> <%= Quantity %></td>
        </tr>
</table>
<%
                } else {
%>
<legend>ERROR: The Product is failed to update.</legend>
<%
      }

                if (pstmt != null) {
                        pstmt.close();
                }
      if (windowsConn != null) {
        windowsConn.close();
      }
    } else if(PID !=null){


        InitialContext ctx = new InitialContext();
                //obtain data source from application server
                DataSource windowsDS = (DataSource)ctx.lookup("jdbc/ecomm");

                //Get connections from application server
                Connection windowsConn = windowsDS.getConnection();

                //Create statements in connection
                Statement stmt=windowsConn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
                ResultSet rs = stmt.executeQuery("Select * FROM Product WHERE PROID=" + PID);

                int numRow = 0;
                if (rs != null && rs.last() != false) {
                  numRow = rs.getRow();
                  rs.beforeFirst();
                  rs.next();
                }


                if(Quantity == null){
                        Quantity = rs.getString("QUANTITY");
                }
                if(Descip == null){
                        Descip = rs.getString("DESCRIPTION");
                }
                /* missing part

                Please Complete this missing part

                This part is to receive the value of product name and unit price from the upper select
SQL statement.
                */

                if (rs != null) {
                        rs.close();
                }
                if (stmt != null) {
                        stmt.close();
                }
                if (windowsConn != null) {
                        windowsConn.close();
                }
```

```
%>
<legend>Please fill in the form</legend>

<!-- The fill in form -->
<form method='POST' action='<%= request.getRequestURI() %>'>
<input name='PID' type='hidden' value='<%= PID %>' />
<%
        if(Pname != null && !Pname.equalsIgnoreCase("") ){
%>
<input name='Pname' type='hidden' value='<%= Pname %>' />
<%
        }
%>
<table>
    <tr>
    <td><p>Product name:</p></td><td><%= Pname %></td>
    </tr>
    <tr>
    <td><p>Unit Price:</p></td>
    <td><input name='Uprice' type='text' size='10' maxlength='7' value='<%= Uprice %>' /></td>
    </tr>
    <tr>
    <td><p>Quantity:</p></td>
    <td><input name='Quantity' type='text' size='10' maxlength='4' value='<%= Quantity %>'
/></td>
    </tr>
    <tr>
    <td><p>Description:</p></td>
    <td>
    <TEXTAREA name="Descip" COLS=40 ROWS=6><%= Descip %></TEXTAREA></td>
    </tr>
</table>
<input type='submit' value='Update!' />

</form>
<%

    } else {
%>
  Please select a product to update.
<%
  }
} catch (SQLException e) {
%>
<div style='color: red'><%= e.toString() %></div>
<%
}
%>
<br/><a href='<%= request.getContextPath() + "/AdminMenu.jsp"%>'>Back to AdminMenu</a>
</fieldset>
</div>
</body>
</html>
```

IBM Inter-University
Programming Contest 2012

Question 2.3 (100 points)

Topic

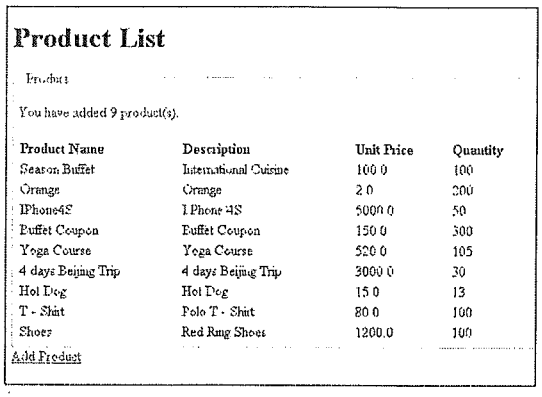
Adding product for sell on to our eCommence portal

Requirements

Add logic to complete the adding product process.

The process consists of 2 webpages, which are *Productlist.jsp* and *AddProduct.jsp*.

Start the process by using the *Productlist.jsp*:



Product List

Product

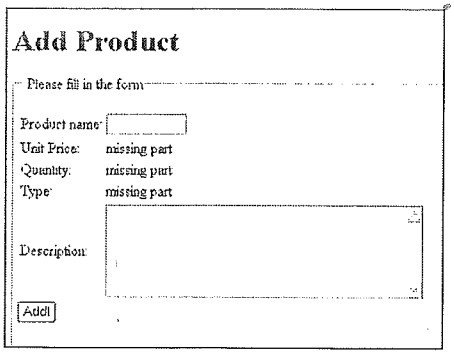
You have added 9 product(s).

Product Name	Description	Unit Price	Quantity
Season Buffet	International Cuisine	100.0	100
Orange	Orange	2.0	200
IPhone4S	I Phone 4S	5000.0	50
Buffet Coupon	Buffet Coupon	150.0	300
Yoga Course	Yoga Course	520.0	105
4 days Beijing Trip	4 days Beijing Trip	3000.0	30
Hot Dog	Hot Dog	15.0	13
T - Shirt	Polo T - Shirt	80.0	100
Shoes	Red Rang Shoes	1200.0	100

Add Product

By clicking the "Add Product" links at the bottom of the page, the *AddProduct.jsp* will be executed for inputting a new product and saving to the database.

There are 3 fields with "missing part" as shown below.



Add Product

Please fill in the form

Product name:

Unit Price: missing part

Quantity: missing part

Type: missing part

Description:

Add!

In *AddProduct.jsp*, fill in the missing part to

- Get the sum of Sold quantity (*BQUAN*) from *TRANSACTION* table for the Type (*CATEGORYID*) of newly added product. Fill in the missing SQL statement and the missing logic to set the value for *SumQuan*.
- Calculate a unique *PROID* (next number in ascending sequence). Fill in the missing SQL statement and the missing logic to set the value for *tmpPROID*.
- Complete the form which includes the following fields
  - Unit price (Uprice)– text
  - Quantity – text
  - Type – drop down box

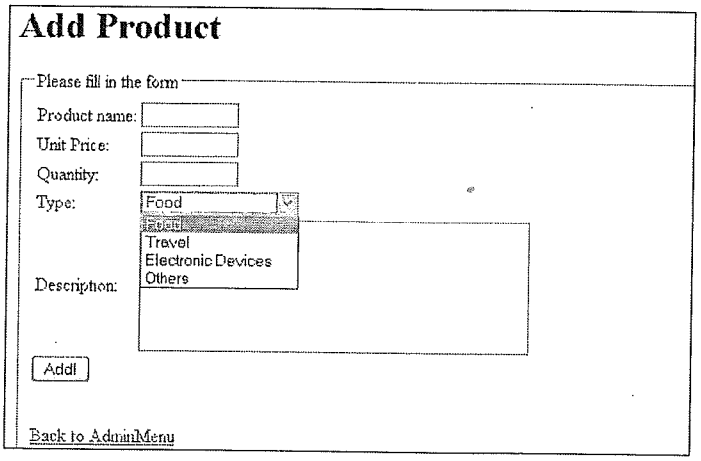
For the Type parameter, the values should refer to the CATEGORY table in the database:

Value	Display Name
100	*Food*
101	*Travel*
103	*Electronic Devices*
102	*Others*

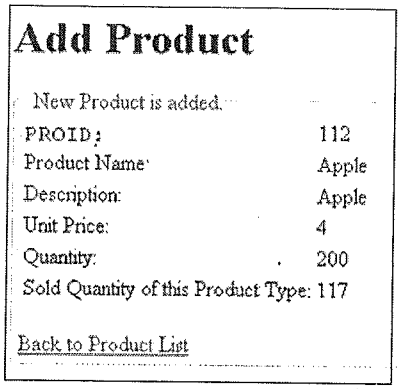
Target Results

To score points for this question, achieve the followings: (Score will be given based on the fields returned in the form, the creation of the unique *PROID* and the correct display of *BQUAN*)

1. Access the administration menu in Firefox **http://localhost:9080/Q3Web/ProductList.jsp** and press the "Add Product" link, the add product form should be displayed as shown below.



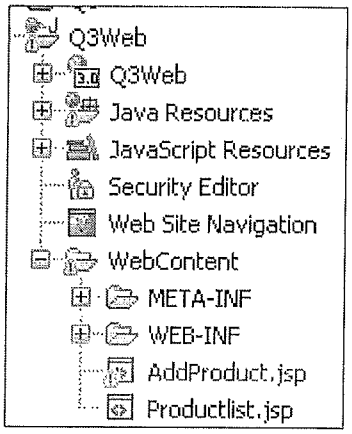
2. After entering the new product information and pressing the "Add" button, the new product information will be shown in the summary with the correct *PROID* (ie, next number in ascending sequence) and the sold quantity of this product type *BQUAN* as shown below.



IBM Inter-University
Programming Contest 2012

Hints and tips

Open *AddProduct.jsp* and search for the missing part, you can use the steps described in Question1 to open the *AddProduct.jsp* in the RAD console and search for missing part



Fill in the missing part for calculating the "SumQuan", unique "PROID" and complete an input form.

After you completed your coding, you can use the steps described in Question1 to open the *AddProduct.jsp* in the RAD console and test the result.

For joining tables, TRANSACTION.PID = PRODUCT.PROID

We assume that the data type will be inputted correctly. The program will not provide any data type checking process. For example, user will only input decimal with 7 digits at "UnitPrice".

A sample drop down box syntax is shown below:

<select name='form name'>
<option value ='database value'>display words
<option value ='database value'>display words
...
</select>

Appendix

This is the source code for *AddProduct.jsp*

```
<%@page import="java.lang.String, java.lang.StringBuffer" %>
<%@page import="java.sql.Connection, javax.sql.DataSource, java.sql.PreparedStatement" %>
<%@page import="java.sql.ResultSet, java.sql.Statement" %>
<%@page import="java.sql.SQLException, javax.naming.NamingException,
javax.naming.InitialContext" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"><%@page
      language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1" "%>
<html>
<head>
<title>AddProduct</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
</head>
<body>
<h1>Add Product</h1>
<div style='width:600px'>
<fieldset>
<%
try {
      // Receive Parameter
      String UID = request.getParameter("UID");
      String PROID = request.getParameter("PROID");
      String Pname = request.getParameter("Pname");
      String Descip = request.getParameter("Descip");
      String Uprice = request.getParameter("Uprice");
      String Quantity = request.getParameter("Quantity");
      String Type = request.getParameter("Type");
      String SumQuan = null;
      String Visted = "N";
      Visted = request.getParameter("Visted");
      if (PROID != null && !PROID.equalsIgnoreCase("") &&
            Pname != null && !Pname.equalsIgnoreCase("") &&
            Descip != null && !Descip.equalsIgnoreCase("") &&
            Uprice !=null && !Uprice.equalsIgnoreCase("") &&
            Quantity != null && !Quantity.equalsIgnoreCase("") &&
            Type != null && !Type.equalsIgnoreCase("") &&
            !PROID.equals("0")) {

            InitialContext ctx = new InitialContext();
            //obtain data source from application server
            DataSource windowsDS = (DataSource)ctx.lookup("jdbc/ecomm");

            //Get connections from application server
            Connection windowsConn = windowsDS.getConnection();

            //Create statements in connection
            PreparedStatement pstmt = windowsConn.prepareStatement("INSERT INTO product
VALUES ("+ PROID +" , '"+ Pname +"', '"+Descip+"', "+Uprice+", "+Quantity+", "+UID+" , " + Type +" , 'F' ,
'F')");

            boolean flag = pstmt.execute();

            Statement stmt=windowsConn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
            ResultSet rset = null;
            /*
            rset = stmt.executeQuery("Please Complete this missing SQL statement. ");

            rset.next();

            SumQuan = reset.getString("Your column name");

            */
            if (pstmt != null) {
                  pstmt.close();
```

```
                }
                if (windowsConn != null) {
                        windowsConn.close();
                }
                if (flag == false) {
%>
<legend>New Product is added.</legend>
<table>
        <tr>
                <td><p>PROID:</p></td>
                <td> <%= PROID %></td>
        </tr>
        <tr>
                <td><p>Product Name:</p></td>
                <td> <%= Pname %></td>
        </tr>
        <tr>
                <td><p>Description:</p></td>
                <td> <%= Descip %></td>
        </tr>
        <tr>
                <td><p>Unit Price:</p></td>
                <td> <%= Uprice %></td>
        </tr>
        <tr>
                <td><p>Quantity:</p></td>
                <td> <%= Quantity %></td>
        </tr>
        <tr>
                <td><p>Sold Quantity of this Product Type:</p></td>
                <td> <%= SumQuan %></td>
        </tr>
</table>
<%
                } else {
%>
<legend>ERROR: New Product is failed to add.</legend>
<%
                }
        } else if(PROID != null && !PROID.equalsIgnoreCase("") && Visted.equals("Y"))
                out.println("Error!");
        else {
                if (UID == null)
                        UID = "1004";

                Visted = "Y";
                InitialContext ctx = new InitialContext();
                //obtain data source from application server
                DataSource windowsDS = (DataSource)ctx.lookup("jdbc/ecomm");
                //Get connections from application server
                Connection windowsConn = windowsDS.getConnection();

                //Create statements in connection
                Statement stmt=windowsConn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
                ResultSet rs = null;
                int tmpPROID =0;
                /* missing part

                //This part is to calculate the unqiue PROID.
                rs = stmt.executeQuery("Please Complete this missing SQL statement.");


                rs.next(); .

                //Please complete this missing logic of setting value to tmpPROID
                */

%>
```

```
<legend>Please fill in the form</legend>
<form method='POST' action='<%= request.getRequestURI() %>?UID=<%=UID%>'>
<!-- Please complete this missing part -->
<input name='Visted' type='hidden' value='<%=Visted %>' />
<input name='PROID' type='hidden' value='<%=tmpPROID %>' />
<table>
      <tr>
      <td><p>Product name:</p></td>
      <td><input name='Pname' type='text' size='10' maxlength='30' /></td>
  </tr>
  <tr>
      <td><p>Unit Price:</p></td>
      <td>missing part</td>
  </tr>
  <tr>
      <td><p>Quantity:</p></td>
      <td>missing part</td>
  </tr>
  <tr>
      <td><p>Type:</p></td>
      <td>missing part
      <!-- sample drop down box syntax
            <select name='form name'>
                  <option value ='database value'>display words
                  <option value ='database value'>display words
                  ...
                  </select>
      -->
            </td>
  </tr>
  <tr>
      <td><p>Description:</p></td>
      <td><TEXTAREA name="Descip" COLS=40 ROWS=6></TEXTAREA></td>
  </tr>
</table>

<input type='submit' value='Add!' />
</form>
<%
            if (rs != null) {
                  rs.close();
            }
            if (stmt != null) {
                  stmt.close();
            }
            if (windowsConn != null) {
                  windowsConn.close();
            }
      }
} catch (SQLException e) {
%>
<div style='color: red'><%= e.toString() %></div>
<%
}
%>
<br/><a href='<%= request.getContextPath() + "/Productlist.jsp"%>'>Back to Product List</a>
</fieldset>
</div>
</body>
</html>
```

IBM Inter-University
Programming Contest 2012

Question 2.4 (200 Points)

Topic

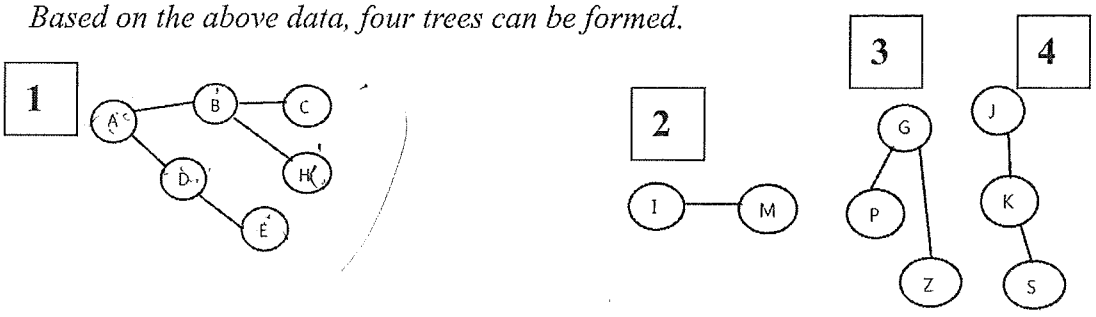
After the website been running for while, data on how customer behaves has been collected. Your management would like to use this information to increase the cross sell opportunity. All information is stored in the ecomm database "productrelation" table.

*The **"productrelation"** table consists of the following data.*

All category relationships are stored in this table. Each row in the table represent the element on the left relates to that on the right.

CATNAME1	CATNAME2
A	B
A	D
B	C
B	H
D	E
G	P
G	Z
I	M
J	K
K	S

Based on the above data, four trees can be formed.



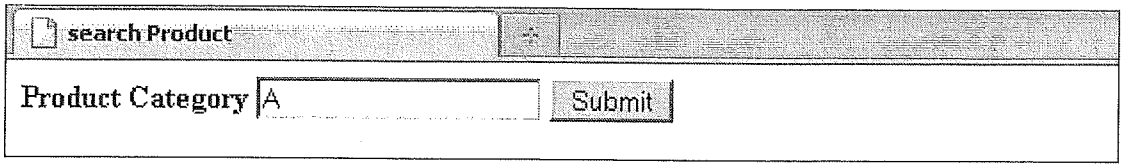
*When you input **S** (name of the category) in the search text box, the expected result should be "**K, J**" which is the **complete set of category** in tree 4. Since **(J, K)** & **(K, S)** has a common category "**K**", **(J, S)** is a transitive relationship or vice versa. (e.g., when J is search, S, K should return)*

IBM Inter-University
Programming Contest 2012

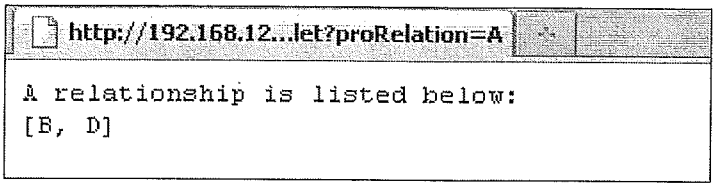
Requirements

Modify the existing *getSubSet()* function in *DB2CategoryDAO.java*.

The searching process is implemented by the java program *DB2CategoryDAO.java*, and run the login logic using the *searchProduct.jsp*.



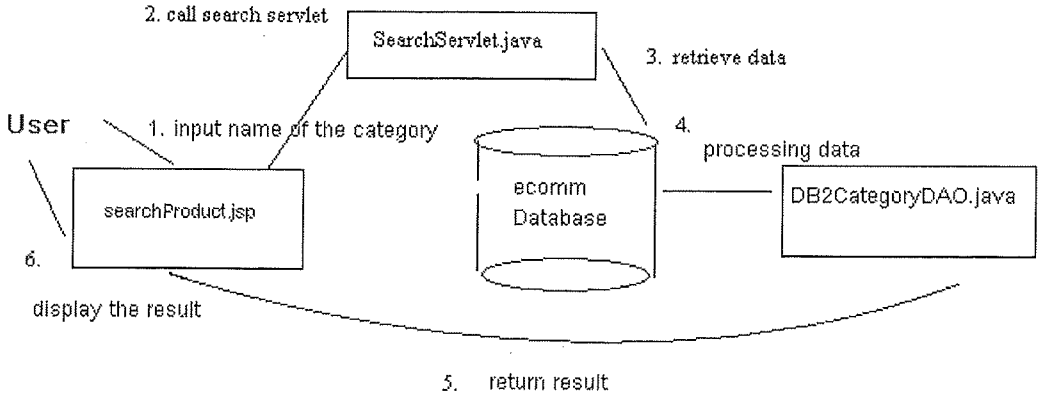
When input "A" in the Product Category box and press submit, the current program only return "B" and "D".



In *DB2CategoryDAO.java,* fill in the missing part to

- fulfill the requirement listed in the background session which is returning the **complete** set of category which is in the same tree whenever one of the category is seached. For example, when A is searched, it should return B, C, D, E & H.

The following diagram illustrates the searching procedure.



IBM Inter-University
Programming Contest 2012

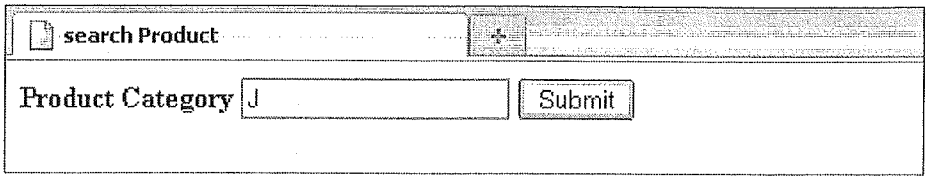
Target Results

To score points for this question, achieve the followings: (partial score will be rewarded for each relationship searching)

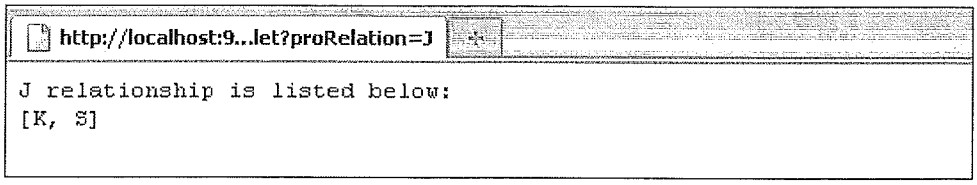
1. Access the search product in Firefox

http://localhost:9080/Q4Web/searchProduct.jsp

Input **J** in the text box.



If your implementation is correct, the result should look this



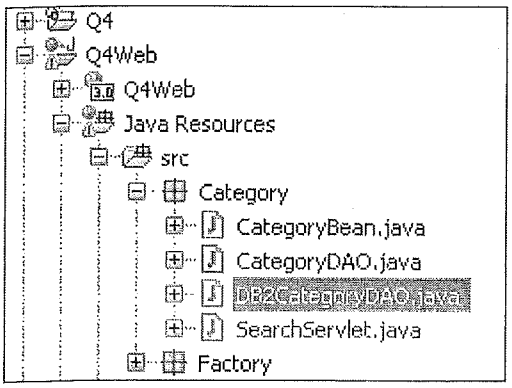
Please note the following

- **You are not allowed to insert/modify/delete any record stored in "*productrelation*" table, otherwise no point will be awarded for this question**
- **Duplicated result will deduct points**
- **Order of result is not required**

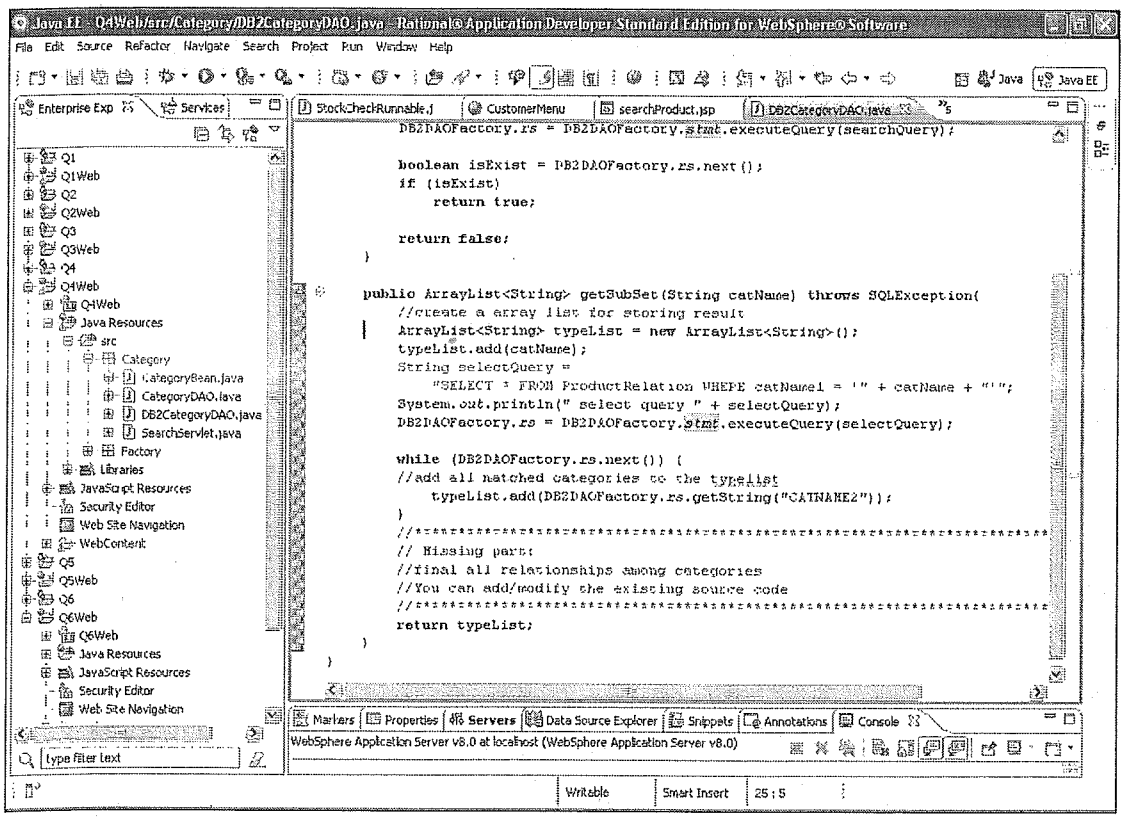
IBM Inter-University
Programming Contest 2012

Hints and tips

Open *DB2CategoryDAO.java* and search for the missing part, you can use the steps described in Question1 to open the *DB2CategoryDAO.java* in the RAD console and search for missing part



Fill in the missing part for search logic



IBM Inter-University
Programming Contest 2012

Appendix

This is the source code for DB2CategoryDAO.java

```
package Category;

import java.sql.SQLException;
import java.util.ArrayList;

import Factory.*;

public class DB2CategoryDAO implements CategoryDAO{
	public boolean isCategory(CategoryBean bean)throws SQLException {
		String searchQuery =
			"SELECT * FROM ProductRelation WHERE catName1 = '" + bean.getCatName() + "'
OR " +
				" catName2 = '" + bean.getCatName() + "'";
//	System.out.println(" search query " + searchQuery);
		DB2DAOFactory.rs = DB2DAOFactory.stmt.executeQuery(searchQuery);

		boolean isExist = DB2DAOFactory.rs.next();
		if (isExist)
			return true;

		return false;
	}

	public ArrayList<String> getSubSet(String catName) throws SQLException{
		//create a array list for storing result
		ArrayList<String> typeList = new ArrayList<String>();
		typeList.add(catName);
		String selectQuery =
			"SELECT * FROM ProductRelation WHERE catName1 = '" + catName + "'";
		System.out.println(" select query " + selectQuery);
		DB2DAOFactory.rs = DB2DAOFactory.stmt.executeQuery(selectQuery);

		while (DB2DAOFactory.rs.next()) {
		//add all matched categories to the typelist
			typeList.add(DB2DAOFactory.rs.getString("CATNAME2"));
		}

	//*************************************************************************
************
		// Missing part:
		//final all relationships among categories
		//You can add/modify the existing source code

	//*************************************************************************
************
		return typeList;
	}
}
```

This is the source code for SearchServlet.java

```
package Category;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import Factory.DAOFactory;
import Factory.DB2DAOFactory;


/**
 * Servlet implementation class searchServlet
 */
@WebServlet("/searchServlet")
public class SearchServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public SearchServlet() {
        super();
        // TODO Auto-generated constructor stub
    }
     /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
            CategoryBean cat = new CategoryBean();
            PrintWriter out = response.getWriter();

            String catName = request.getParameter("proRelation");

            try {
                out.println(processDAO(cat, catName));
            } catch (SQLException e) {
                e.printStackTrace();
            }
    }
    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
    }

    protected String processDAO(CategoryBean cat, String catName) throws SQLException {
        //store category name
        String categoryName;
        //create the required DAO Factory
        DAOFactory db2Factory= DAOFactory.getDAOFactory(DAOFactory.DB2);
        //open a new DB2 connection
        DB2DAOFactory.createConnection();
        //Create a DAO
        CategoryDAO categoryDAO = db2Factory.getCategoryDAO();

        if (!catName.isEmpty()) {
```

```
                categoryName = catName.toUpperCase();
                cat.setCatName(categoryName);
                boolean isFound = categoryDAO.isCategory(cat);

                if (isFound) {
                        ArrayList<String> result = categoryDAO.getSubSet(categoryName);
        //Close DB2 connection
                        DB2DAOFactory.closeConnection();
                        return catName + " relationship is listed below: \n" + result;
                }
                        return "Sorry, category " + catName + " does not have any relation";
        }
        return "Please input the name of the category.";
    }
}
```

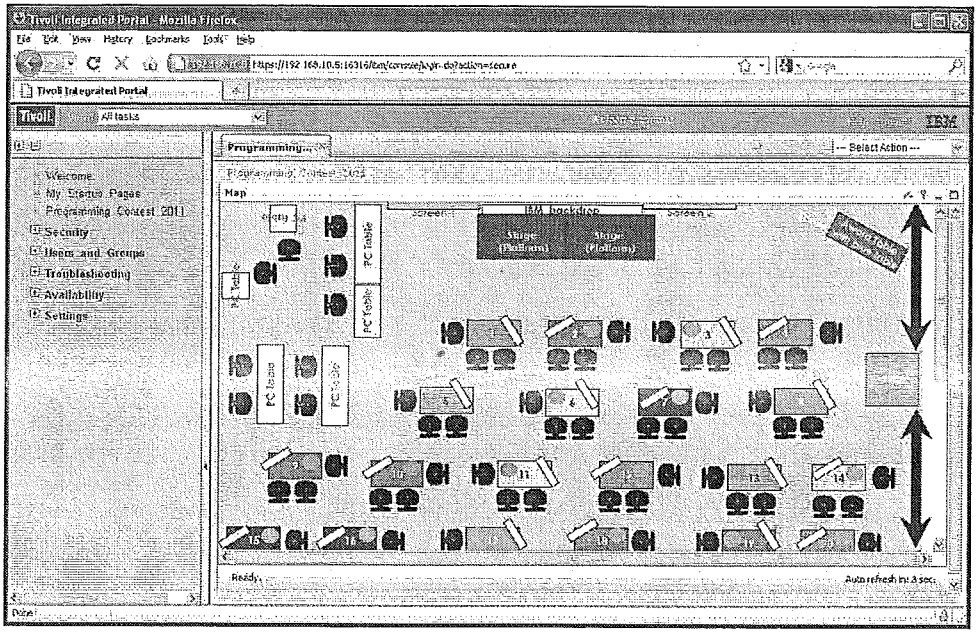
Question 2.5 (80 Points)

Topic

Your website may offer deals that exceed the capability the offer owner can handle. To avoid this situation, the system should notify the owner as soon as any of the product sold exceeds the safety threshold set by the owner.

A java program called StockCheckManager.java will

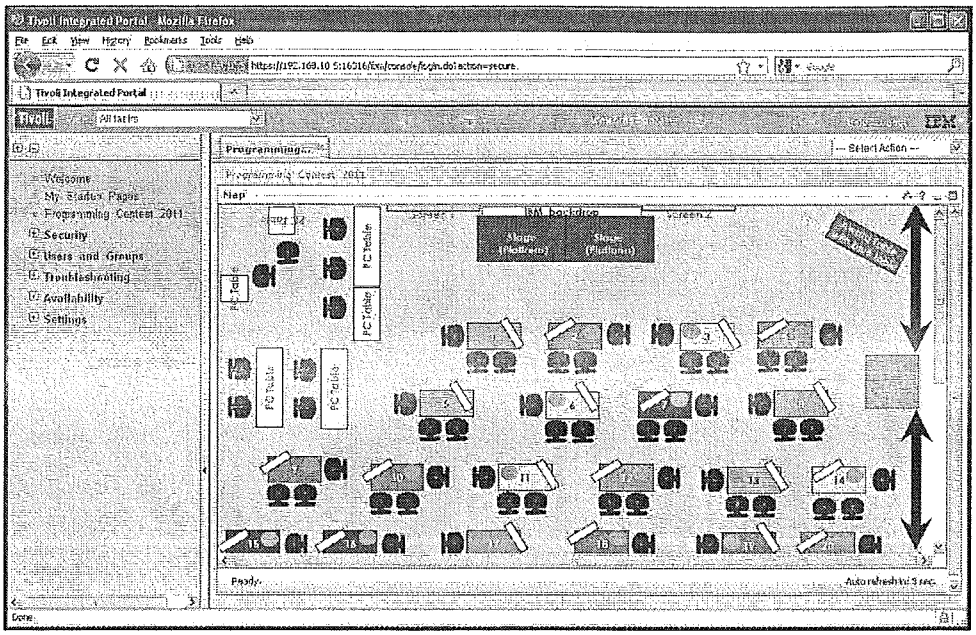
-Retrieve the database value and
-Determine the type of message to be sent
-Send the message to Netcool Omnibus



IBM Inter-University
Programming Contest 2012

Requirements

Modify two functions, *sendMessage()*, *findMsgType()* in *StockCheckManager.java* which will make the Netcool Omnibus alerting system able to handle the alerts

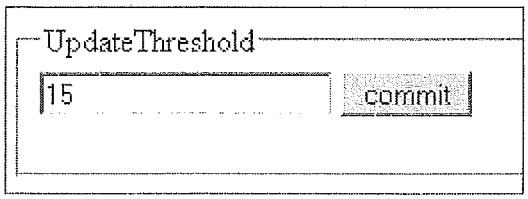


The alerting process is implemented by the java program *StockCheckRunnable.java*, Open *ChangeSafety.jsp* to modify the "Safety Threshold". Press the "Update" link to update the "Safety Threshold". The red alert will be shown when any product is in a state that "safety threshold" <= "sold quantity". The alert will return to green when all the uncutoff (Cutoff state = False) products' "safety threshold" > "sold quantity"

Change Safety

Product	Sold Quantity	Safety Threshold	Cutoff state	Update Threshold	Cutoff
Season Buffet	8	200	False	update	cutoff
Orange	74	80	False	update	cutoff
IPhone4S	12	50	False	update	cutoff
Buffet Coupon	21	300	False	update	cutoff
Yoga Course	16	105	False	update	cutoff
4 days Beijing Trip	4	30	False	update	cutoff
Hot Dog	14	16	False	update	cutoff

undo cutoff



In *StockCheckRunnable.java,* fill in the missing part for

- *sendMessage()*, a function that according the given status, send message to Netcool Omnibus System; modify the parameter so that the message respresenting your team will be shown. Please find detail in Hints and Tips.

- *findMsgType ()* , a function that determine the type of message the system should send to Netcool Ominibus; setup logic to return a boolean value, (False – green, True – red) for alerting purpose.

Target Results

To score points for this question, achieve the followings: (partial score will be rewarded for each condition alert), please refresh the page after the submission

1. Access the "Safe Threshold" in Firefox **http://localhost:9080//Q5Web/changeSafety.jsp** to change the Safety Threshold accordingly.

Change Safety

Product	Sold Quantity	Safety Threshold	Cutoff state	Update Threshold	Cutoff
Season Buffet	8	200	False	update	cutoff
Orange	74	80	False	update	cutoff
IPhone4S	12	50	False	update	cutoff
Buffet Coupon	21	300	False	update	cutoff
Yoga Course	16	105	False	update	cutoff
4 days Beijing Trip	4	30	False	update	cutoff
Hot Dog	14	16	False	update	cutoff

undo cutoff

If your implementation is correct, the result should look this

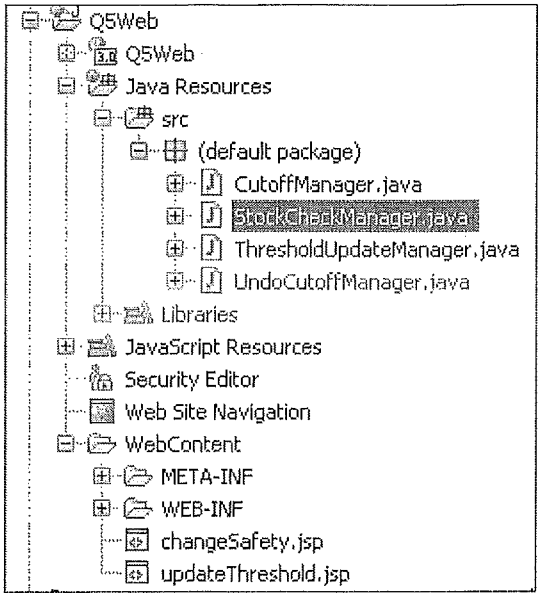
1, When "Safety Threshold" of any of the product is smaller or equal to the "Sold Quantity", the table that represent your team will lights up a Red alert on the Omnibus dashboard.

2. The table of your team on Omnibus dashboard will display back to a Green status
   - when the alerting product is cutoff **AND**
   - All product with a "False" cutoff state has the "Safety Threshold" greater than "Sold Quantity".

IBM Inter-University
Programming Contest 2012

Hints and tips

Open *StockCheckManager.java* and search for the missing part, you can use the steps described in Question1 to open the *StockCheckManager.java* in the RAD console and search for missing part



Fill in the missing part for alerting logic.

```
private void sendMessage (String status) throws InterruptedException, IOException{
    //missing part
    //Modify the parameters to send message representing your team
    Runtime.getRuntime()
    .exec("C:\\postzmsg.exe -f C:\\my.conf -r CRITICAL -m contest EventID=1XX situ
    .waitFor();
}
private boolean isRed(int totalboughtquantity, int threshold,
                      boolean cutoff, boolean outstanding){
    //missing part
    //modify to return boolean value that satisfy conditions specified
    return false;
}
private boolean isGreen(int totalboughtquantity, int threshold,
                      boolean cutoff, boolean outstanding){
    //missing part
    //modify to return boolean value that satisfy conditions specified
    return false;
}
```

You are advised to skim the comment in StockCheckRunnable.java and Database schema.

Here show two commands for the completing the background process. The first command is to change the traffic light of team<Y> to red, and the second command is to change the traffic light of team<Y> back to green. You can modify the two commands by charige Y to your team number, and XX = Y with a prefix 0 if Y is single digit. For example, if Y = 1, then XX = 01. Then the two commands will become:

```
postzmsg.exe -f my.conf -r CRITICAL -m contest EventID=101
   situation_name="teamXX" satuation_status="Y"
integration_type="U"
   situation_origin=team1 situation_displayitem='team1' team1 ITM
```

and

```
postzmsg.exe -f my.conf -r CRITICAL -m contest EventID=101
   situation_name="teamXX" satuation_status="N"
integration_type="U"
   situation_origin=team1 situation_displayitem='team1' team1 ITM
```

respectively.

In Java, the function Runtime.getRuntime().exec().waitFor() is used to execute a system command. Absolute paths must be specified for the program "postzmsg.exe" and the file "my.conf". For simplicity, copies of both files are already copied into C:\

As a result, the complete Java statement to send a GREEN alert is shown below:

```
Runtime.getRuntime().exec("C:\\postzmsg.exe -f C:\\my.conf -r
CRITICAL -m contest EventID=101 situation_name=\"team1\"
satuation_status=\"N\" integration_type=\"U\" situation_origin=team1
situation_displayitem='team1' team1 ITM").waitFor();
```

IBM Inter-University
Programming Contest 2012

Appendix

This is the source code for *StockChangeManager.java*

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.sql.DataSource;


public class StockCheckManager {

	private static void sendMsg (String status) throws InterruptedException, IOException{
		//missing part
		//Modify the parameters to send message representing your team
		Runtime.getRuntime()
		.exec("C:\\postzmsg.exe -f C:\\my.conf -r CRITICAL -m contest EventID=1XX
situation_name=\"teamY\" situation_status=\""+status+"\" integration_type=\"U\" situation_origin=teamY
situation_displayitem='teamY' teamY ITM")
		.waitFor();
	}
	private static boolean findMsgType(ArrayList<Integer> soldQuantityList,
	                                   ArrayList<Integer> thresholdList,
	                                   ArrayList<Boolean> cutoffList){
		//missing part
		//Modify to return a type, true represent red message, false represent green message
		return false;
	}
	public static void run(){
		String sql =
			"select	* "+
			"from (select	* "+
				"from product "+
				"where	uid = 1004 "+
				")availProd join "+
				"(select	pid, sum (bquan) as soldQuantity "+
				"from transaction "+
				"group by pid "+
				") tranCnt on availProd.proid = tranCnt.pid ";
		try {
			InitialContext	context	= new InitialContext();
			DataSource	source	= (DataSource) context.lookup ("jdbc/ecomm");
			Connection	connection	= source.getConnection();
			PreparedStatement statement	= connection.prepareStatement(sql);
			ResultSet	results	= statement.executeQuery();
			ArrayList<Integer> soldQuantityList = new ArrayList<Integer>();
			ArrayList<Integer> thresholdList = new ArrayList<Integer>();
			ArrayList<Boolean> cutoffList	= new ArrayList<Boolean>();
			while (results.next()){
				soldQuantityList.add(results.getInt("soldQuantity"));
				thresholdList.add(results.getInt("quantity"));
				cutoffList.add(results.getString("cutoff").equals("T"));
			}
			boolean type = findMsgType(soldQuantityList, thresholdList, cutoffList);
			sendMsg(type?"Y":"N");
		} catch (NamingException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		} catch (SQLException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
```

```
                e.printStackTrace();
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
}

}
```

Question 2.6 (145 point)

Apply various discount to the buying process when different payment method is selected.
Access customer transaction record and print invoice from the website

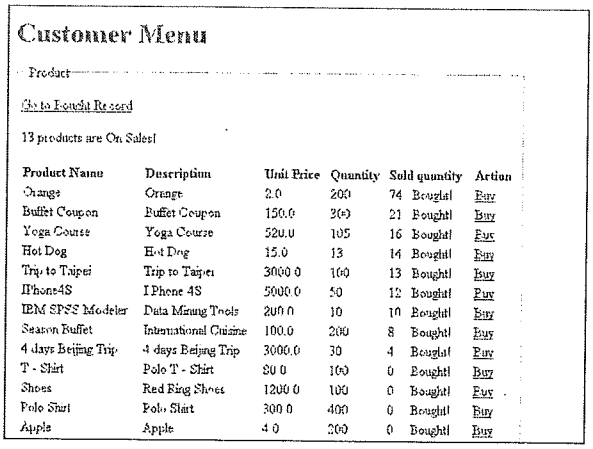
Requirements

There are 2 parts for the requirement.

Add logic to complete the buying product process for the 1st part.

The customer menu consists of 2 webpages, which are *CustomerMenu.jsp* and *Buy.jsp*.

Run the customer menu using the page *CustomerMenu.jsp*:



Customer Menu

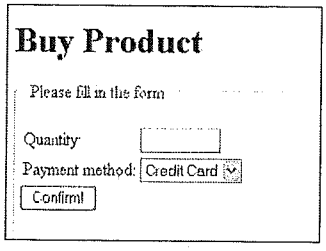
Product

Go to Bought Record

13 products are On Sales!

Product Name	Description	Unit Price	Quantity	Sold quantity	Action
Orange	Orange	2.0	200	74	Bought! Buy
Buffet Coupon	Buffet Coupon	150.0	300	21	Bought! Buy
Yoga Course	Yoga Course	520.0	105	16	Bought! Buy
Hot Dog	Hot Dog	15.0	13	14	Bought! Buy
Trip to Taipei	Trip to Taipei	3000.0	100	13	Bought! Buy
IPhone4S	I Phone 4S	5000.0	50	12	Bought! Buy
IBM SPSS Modeler	Data Mining Tools	200.0	10	10	Bought! Buy
Season Buffet	International Cuisine	100.0	200	8	Bought! Buy
4 days Beijing Trip	4 days Beijing Trip	3000.0	30	4	Bought! Buy
T - Shirt	Polo T - Shirt	80.0	100	0	Bought! Buy
Shoes	Red Ring Shoes	1200.0	100	0	Bought! Buy
Polo Shirt	Polo Shirt	300.0	400	0	Bought! Buy
Apple	Apple	4.0	200	0	Bought! Buy

By clicking the "Buy" links in the "Action" column, the *Buy.jsp* will be executed for inputting quantity and Payment method for a new buy.



Buy Product

Please fill in the form

Quantity:

Payment method: Credit Card

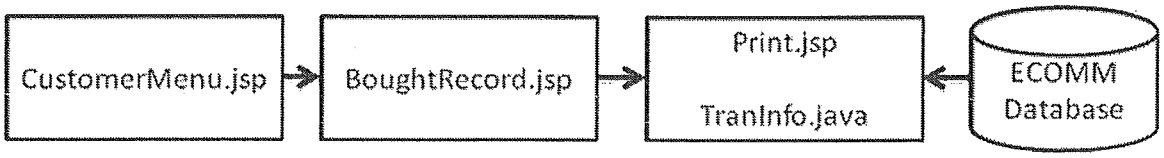
Confirm!

In *Buy.jsp*, there are 2 missing parts. Fill in the missing part to

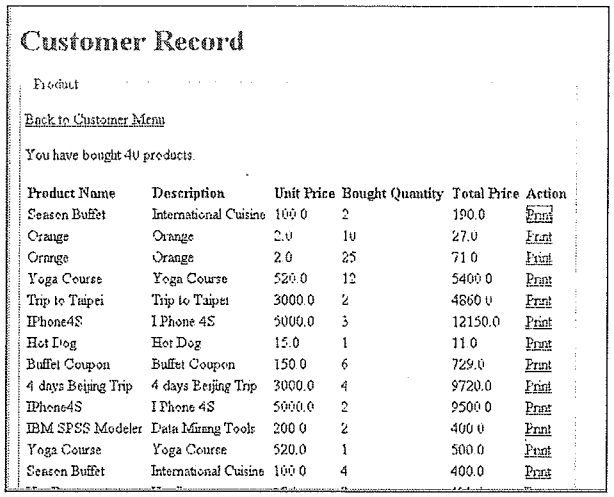
- Insert a new record to transaction table.
- Calculate the total price with different discount for different.
  - A variable **double Price =0;** is initialized for you.
  - You will get some parameters from CustomerMenu.jsp which include:
    -PID –UID –Uprice –Pname -Descip
  - For variable "PayMethod", there are 4 types:
    P – paypal C – credit card S – PPS E – E banking
  - There are some discounts for paypal and credit card:
    Paypal - 0.95 Credit Card - 0.9

For the 2nd part, add logic to complete the invoice printing process

The print process is implemented by *Print.jsp* in which it calls the java program *TranInfo.java*.



Open the *CustomerMenu.jsp*, click on the "Go to Bought Record" link on the top of the screen to open the *BoughtRecord.jsp* which display the customer transaction record



Customer Record

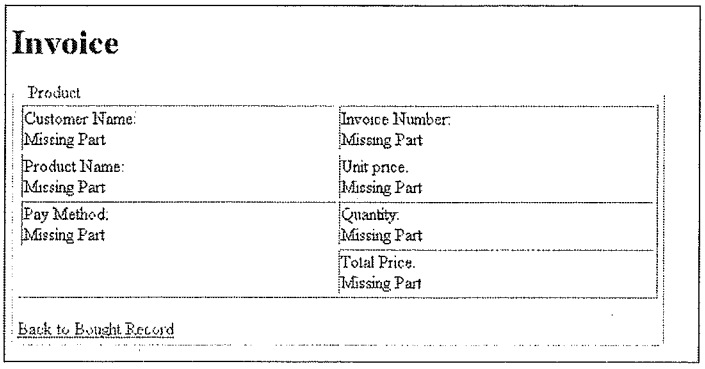
Product

Back to Customer Menu

You have bought 40 products.

Product Name	Description	Unit Price	Bought Quantity	Total Price	Action
Season Buffet	International Cuisine	100.0	2	190.0	Print
Orange	Orange	2.0	10	27.0	Print
Orange	Orange	2.0	25	71.0	Print
Yoga Course	Yoga Course	520.0	12	5400.0	Print
Trip to Taipei	Trip to Taipei	3000.0	2	4860.0	Print
IPhone4S	I Phone 4S	5000.0	3	12150.0	Print
Hot Dog	Hot Dog	15.0	1	11.0	Print
Buffet Coupon	Buffet Coupon	150.0	6	729.0	Print
4 days Beijing Trip	4 days Beijing Trip	3000.0	4	9720.0	Print
IPhone4S	I Phone 4S	5000.0	2	9500.0	Print
IBM SPSS Modeler	Data Mining Tools	200.0	2	400.0	Print
Yoga Course	Yoga Course	520.0	1	500.0	Print
Season Buffet	International Cuisine	100.0	4	400.0	Print

By clicking the "Print" links in the "Action" column, the *Print.jsp* will be executed for displaying the transaction record selected, all the fields are displaying "Missing Part" are shown as below



Invoice

Product

Customer Name: Missing Part	Invoice Number: Missing Part
Product Name: Missing Part	Unit price. Missing Part
Pay Method: Missing Part	Quantity: Missing Part
	Total Price. Missing Part

Back to Bought Record

In *TranInfo.java*, fill in the missing part to:

- Create the constructor *TranInfo(int TID)* to connect to the *ECOMM* database and get the transaction details with the transaction ID (*TID)* passed to the constructor.
- Create the following method inside constructor *TranInfo* created above for *Print.jsp* to get the value to display in the form
    - getTID()

- getPname()
- getUprice()
- getPaymethod()
- getBquan()
- getPrice()
- getUname()

In *Print.jsp*, fill in the missing part to:

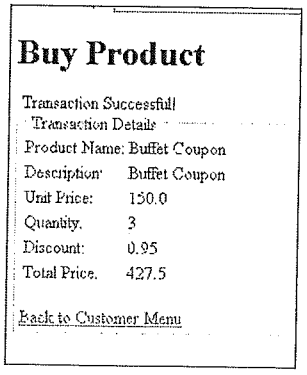
- Create an object to execute the *TranInfo.java* class which created above and pass the TID to the get the transaction detail record.
- Call the method created above in TranInfo class to display corresponding value at form respectively.

Target Results

To score points for this question, achieve the followings: (Score will be given based on the different discount apply to different payment method for the 1st part and different fields in the invoice)

1. Access the customer menu in Firefox **http://localhost:9080/Q6Web/CustomerMenu.jsp** and press the "Buy" link, the *Buy.jsp* will be displayed. Input the buying quantity and press "Confirm" button. If customer selects "*PayPal*", a 5% discount will be applied to the transaction amount. If customer selects "*Credit Card*", a 10% discount will be applied to the transaction amount. All other method will have no discount.

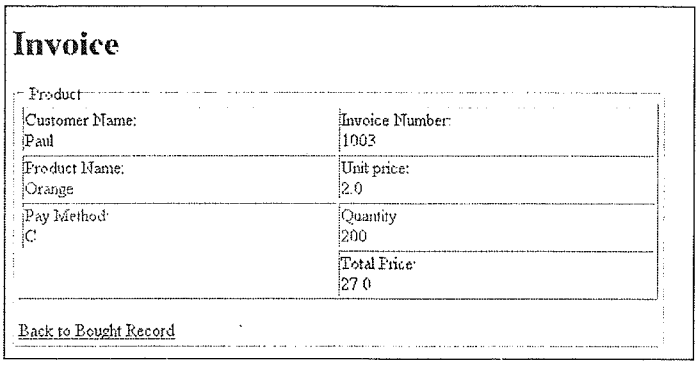
If your implementation is correct, the correct discount will be shown as shown below



2. Access the customer menu in Firefox

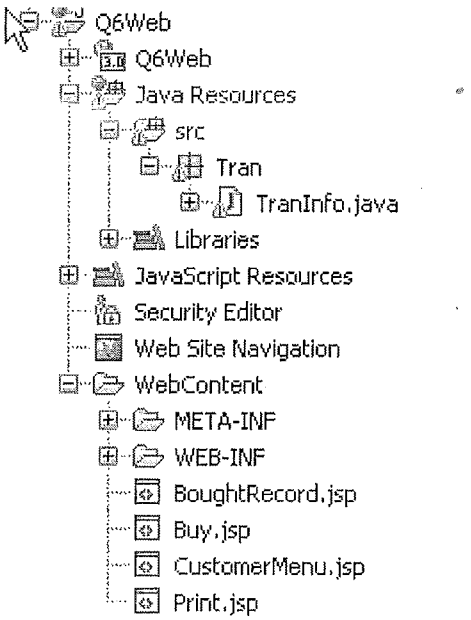
http://localhost:9080/Q6Web/CustomerMenu.jsp and press the "Go to Bought Record" link on the top of the screen and press "Print" link at the *BoughtRecord.jsp*,

If your implementation is correct, the correct transaction information will be shown as below



Hints and tips

Open *Buy.jsp* and search for the missing part, you can use the steps described in Question1 to open the *Buy.jsp* in the RAD console and search for missing part



To continue with the 2nd part, open *Transinfo.java* and *Print.jsp* and search for the missing part.

After you completed your coding you can use the steps described in Question1 to open the *CustomerMenu.jsp* in the RAD console and test the result.

IBM Inter-University
Programming Contest 2012

Appendix

This is the source code for *Buy.jsp*

```
<%@page import="java.lang.String, java.lang.StringBuffer" %>
<%@page import="java.sql.Connection, javax.sql.DataSource, java.sql.PreparedStatement" %>
<%@page import="java.sql.ResultSet, java.sql.Statement" %>
<%@page import="java.sql.SQLException, javax.naming.NamingException,
javax.naming.InitialContext" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"><%@page
        language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<html>
<head>
<title>Buy</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
</head>
<body>
<h1>Buy Product</h1>
<div style='width:600px'>
<fieldset>
<%
try {
        //define variable and
        double Price = 0;
        boolean flag = false;
        String TID = request.getParameter("TID");
        String UID = request.getParameter("UID");
        String PID = request.getParameter("PID");
        String Uprice = request.getParameter("Uprice");
        String Bquan = request.getParameter("Bquan");
        String Paymethod = request.getParameter("Paymethod");
        String Pname = request.getParameter("Pname");
        String Descip = request.getParameter("Descip");


        if(PID == null){
%>
Please Select a product.
<%
        }else if (TID != null && !TID.equalsIgnoreCase("") &&
                UID != null && !UID.equalsIgnoreCase("")&&
                PID != null && !PID.equalsIgnoreCase("") &&
                Uprice != null && !Uprice.equalsIgnoreCase("") &&
                Bquan != null && !Bquan.equalsIgnoreCase("") &&
                Paymethod != null && !Paymethod.equalsIgnoreCase("")){

                // Process after submitting the Buying Form below
                /*      missing part
                Please Complete this part
                Calculate the Price with payment method discount.
                */
                if (Price !=0){
                        InitialContext ctx = new InitialContext();

                        //obtain data source from application server
                        DataSource windowsDS = (DataSource)ctx.lookup("jdbc/ecomm");

                        //Get connections from application server
                        Connection windowsConn = windowsDS.getConnection();

                        PreparedStatement pstmt = null;
                        /* missing part
                        pstmt = windowsConn.prepareStatement("Please Complete this missing SQL
statement");
                        */
                        flag = pstmt.execute();

                        if (pstmt != null) {
```

double discount = 1;
if (Paymethod)) discount=0.95
if ("p".equals(Paymethod)) discount=0.9
Price = Uprice * Bquan * discount
insert (TID, PID, VID,) into Transaction
Paymethod, Bquan, price

```
                    pstmt.close();
                }
                if (windowsConn != null) {
                    windowsConn.close();
                }
                out.println("Transaction Successful!");
        } else
         out.println("Total Price is 0. Transaction Fail! Please Complete the missing part.");
                if (flag == false) {
%>
<legend> Transaction Details</legend>
<table>
        <tr>
                <td><p>Product Name:</p></td>
                <td> <%= Pname %></td>
        </tr>
        <tr>
                <td><p>Description:</p></td>
                <td> <%= Descip %></td>
        </tr>
        <tr>
                <td><p>Unit Price:</p></td>
                <td> <%= Uprice %></td>
        </tr>
        <tr>
                <td><p>Quantity:</p></td>
                <td> <%= Bquan %></td>
        </tr>
        <tr>
                <td><p>Discount:</p></td>
                <td>
<%
                if (Paymethod.equals("P"))
                        out.println("0.95");
                else if (Paymethod.equals("C"))
                        out.println("0.9");
                else
                        out.println("No");

%>
                </td>
        </tr>
        <tr>
                <td><p>Total Price:</p></td>
                <td> <%= Price %></td>
        </tr>
</table>
<%

                } else {
%>
<legend>ERROR: Transaction Error.</legend>
<%
                }
        } else {
                InitialContext ctx = new InitialContext();

                //obtain data source from application server
                DataSource windowsDS = (DataSource)ctx.lookup("jdbc/ecomm");

                //Get connections from application server
                Connection windowsConn = windowsDS.getConnection();

                Statement stmt=windowsConn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
                ResultSet rs = stmt.executeQuery("Select max(TID) as TID FROM transaction");

                int numRow = 0;
                if (rs != null && rs.last() != false) {
```

```
                    numRow = rs.getRow();
                    rs.beforeFirst();
                    rs.next();
                }
                int tempTID = rs.getInt("TID")+1;

                if (rs != null) {
            rs.close();
                }
                if (stmt != null) {
                   stmt.close();
                }
                if (windowsConn != null) {
                   windowsConn.close();
                }

%>
<legend>Please fill in the form</legend>
<form method='POST' action='<%= request.getRequestURI()%> '>
<input name='TID' type='hidden' value='<%=tempTID%>' />
<input name='PID' type='hidden' value='<%=PID%>' />
<input name='UID' type='hidden' value='<%=UID%>' />
<input name='Pname' type='hidden' value='<%=Pname%>' />
<input name='Descip' type='hidden' value='<%=Descip%>' />
<input name='Uprice' type='hidden' value='<%=Uprice%>' />
<table>
      <tr>
            <td><p>Quantity:</p></td>
            <td><input name='Bquan' type='text' size='8' maxlength='8' /></td>
      </tr>
      <tr>
            <td><p>Payment method:</p></td>
            <td><select name='Paymethod'>
                  <option value ='C'>Credit Card
                  <option value ='P'>PayPal
                  <option value ='S'>PPS
                  <option value ='E'>E Banking
                  </select>
            </td>
      </tr>
</table>
<input type='submit' value='Confirm!' />
</form>
<%
      }
%>
<br/><a href='<%= request.getContextPath() + "/CustomerMenu.jsp"%>'>Back to Customer Menu</a>

<%

} catch (SQLException e) {
%>
<div style='color: red'><%= e.toString() %></div>
<%
}
%>
</fieldset>
</div>
</body>
</html>
```

IBM Inter-University
Programming Contest 2012
This is the source code for *TranInfo.java*

```
package Tran;

import java.sql.*;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.sql.DataSource;

public class TranInfo {
        private int TID;
        private String Pname;
        private double Uprice ;
        private String Paymethod;
        private int Bquan;
        private double Price;
        private String Uname;

        public TranInfo(int TID) throws SQLException, NamingException{
/*
                Please Complete this Missing Constructor
                There are some codes for your reference

                try{
                        InitialContext ctx = new InitialContext();
                         //obtain data source from application server
                        DataSource windowsDS = (DataSource)ctx.lookup("jdbc/ecomm");

                        //Get connections from application server
        Connection windowsConn = windowsDS.getConnection();

        //Create statements in connection
        Statement stmt=windowsConn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);

                }
      catch( SQLException e) {
                        System.out.println(e);
      }
      catch (NamingException e){
        System.out.println(e);
      }
*/
        }
/*
                Please Complete this missing part for calling get method

*/


}
```

Source get/set

IBM Inter-University
Programming Contest 2012

This is the source code for *Print.jsp*

```
<%@page import="java.lang.String, java.lang.StringBuffer" %>
<%@page import="java.sql.Connection, javax.sql.DataSource" %>
<%@page import="java.sql.ResultSet, java.sql.Statement" %>
<%@page import="java.sql.SQLException, javax.naming.NamingException,
javax.naming.InitialContext" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"><%@page
      language="java" contentType="text/html; charset=ISO-8859-1"
      pageEncoding="ISO-8859-1"
      import="Tran.TranInfo" %>
<html>
<head>
<title>Print</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
</head>
<body>
<h1>Invoice</h1>
<div style='width:600px'>
<fieldset>
<legend>Product</legend>
<%
try {
      String TID = request.getParameter("TID");
      // missing part : Create object of TranInfo class
%>
<table width="100%" border="1">
      <tr>
            <td width="50%">
                  Customer Name:
                  <br />Missing Part
            </td>
            <td>
                  Invoice Number:
                  <br />Missing Part
            </td>
      </tr>
      <tr>
            <td>
                  Product Name:
                  <br />Missing Part
            </td>
            <td>
                  Unit price:
                  <br />Missing Part
            </td>
      </tr>
      <tr>
            <td>
                  Pay Method:
                  <br />Missing Part
            </td>
            <td>
                  Quantity:
                  <br />Missing Part
            </td>
      </tr>
      <tr>
            <td></td>
            <td>
                  Total Price:
                  <br />Missing Part
            </td>
      </tr>
</table>
<%
} catch (NumberFormatException e){
%>
<div style='color: red'><%= e.toString() %></div>
```

P.5 , P.8

TranInfo info = new TranInfo(TID);

info.

<%= Uname %>

<Pname>

Uprice

Paymethod

Bquan

Price

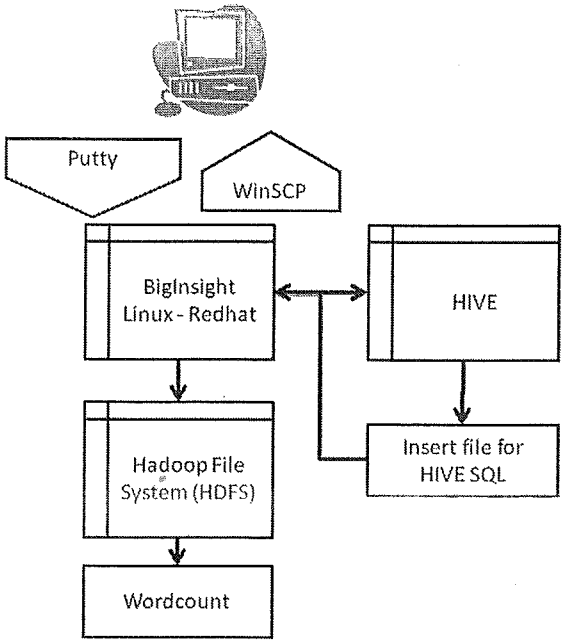
```
<br/>Please Select a transaction.
<%
}
%>
<br/><a href='<%= request.getContextPath() + "/BoughtRecord.jsp"%>'>Back to Bought Record</a>
</fieldset>
</div>

</body>
</html>
```

Question 2.7 (50 Points)

After the website has been running for a few months, your marketing team wants to understand more on the behaviour on the customer accessing the website. You have the log files of the website on when, who and what the customer has done. However the information is in a text file format and unstructured, therefore it is not easy to analysis.

Requirements



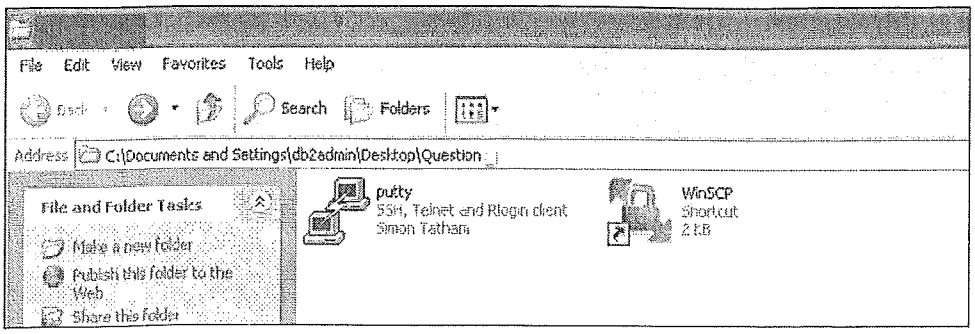
Leverage *Putty* to access the *BigInsight* installed on our *IBM Enterprise Cloud*. In this question complete the following tasks

- Use the wordcount function inside *Hadoop File System (HDFS)* to count the words in the "Weblog.txt" already stored,
- Use *HIVE SQL* and its function to select the **Top 5** customers who have accessed the website and the **Top 10** Products who have been browsed. The word count result is stored in the "wordcount" table inside *HIVE*.

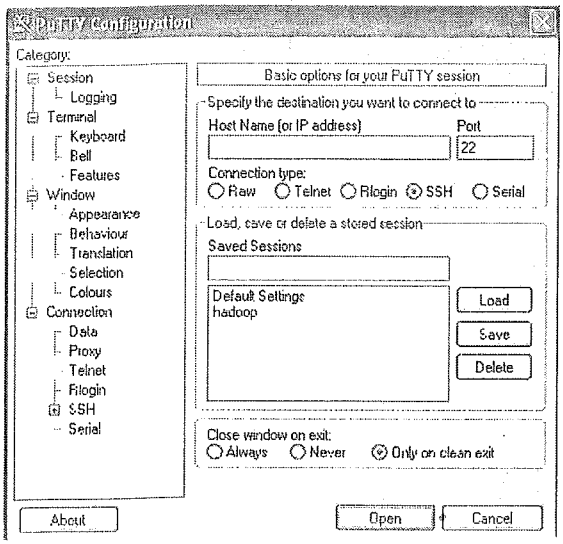
Leverage *WinSCP* to get the output files to your local machine.

IBM Inter-University
Programming Contest 2012

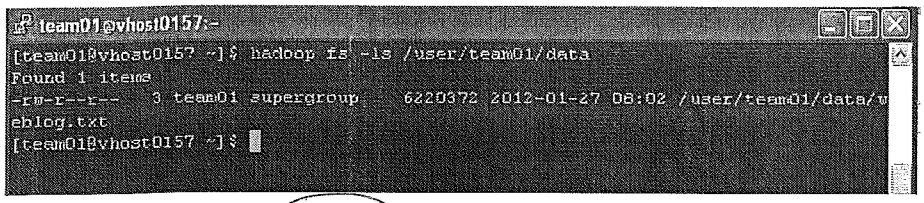
To start this question,



open Question 7 folder on desktop, run *Putty* and click on hadoop session.



Run the wordcount function in *HDFS* with the file provided inside the *HDFS* as shown below.



The web log files "weblog.txt" is located at the "data" folder under your team number folder, eg /user/team01/data as shown above.

Leverage the training material to access *HDFS* and run the wordcount function. The wordcount function is stored at the same jar files as in training, and the location is (*/mnt/biginsights/opt/ibm/biginsights/IHC/hadoop-0.20.2-examples.jar*).

After you run the wordcount process, writre down the "Job Number". Get into *HIVE* system to run the select statement for the top 5 customers and top 10 products browsed.

```
[team018@host0157 ~]$ hive
2012-01-31 13:40:07,633 GMT : Connection obtained for host: 170.225.160.157, por
t number 1528.
log4j:ERROR setFile(null,true) call failed.
```

Please ignore the error message and run the *HIVE* function in the wordcount table.

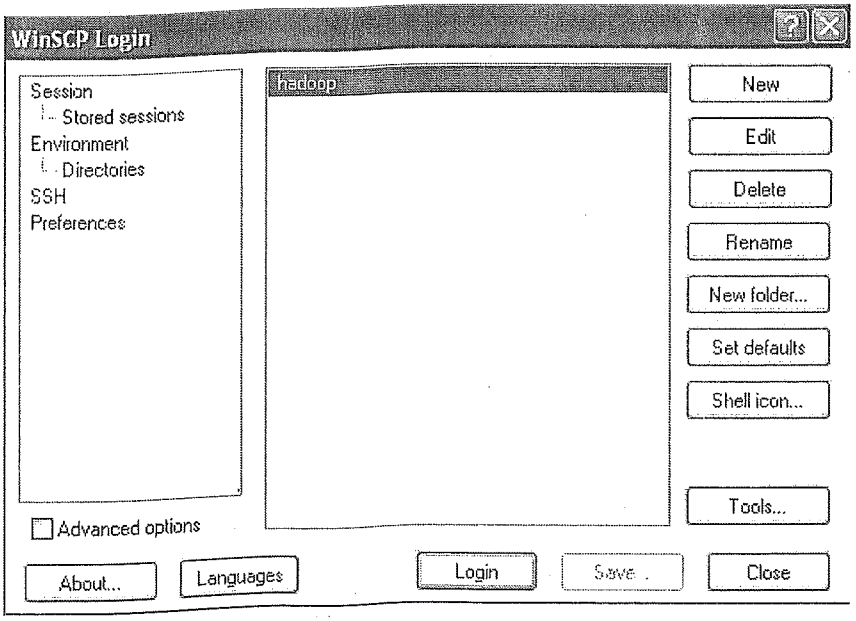
```
hive> show tables;
OK
wordcount
Time taken: 2.206 seconds
hive>
```

The wordcount table is located inside the *HIVE* as shown above.

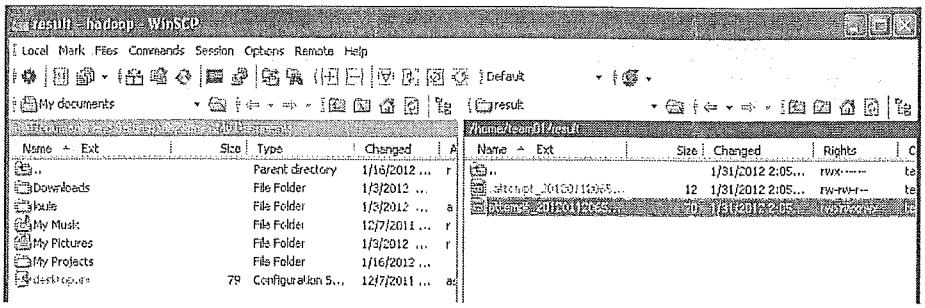
There will be 2 columns for the "wordcount" table (**word - string, count - integer**). The "customer id" is in the format of CID_____ (eg CID12345) and the "product id" is in the format of PID_____ (eg PID12345)
The output file should stored in the path of the folder of your team folder which access have been granted to which is your team, (ie, "/home/teamXX/result").

Here is the insert to local directory and select statement syntax in *HIVE*
INSERT OVERWRITE LOCAL DIRECTORY "*local directory path*"
SELECT *[ALL | DISTINCT] select_expr, select_expr, ...*
FROM *table_reference*
[WHERE *where_condition*]
[GROUP BY *col_list*]
[SORT BY *col_list*] *desc|asc*
[LIMIT *number*]

After complete the *HIVE SQL* statements you can use *WinSCP* located in the Question 7 folder same as *Putty*, select hadoop and login accordingly to connect



The output file will be stored in the path in the *HIVE SQL* statement



Results

In order to score points for this question, you will need to provide the following to the judge. (score will be rewarded for each complete task)

1, The job number you have run the wordcount (please send the job number to the judge via Sametime), judge will check the status via on the "Job status" page on the *BigInsight* dashboard (URL should refer to the training material)

2, You will need to run 2 queries against the "wordcount" table on the HIVE system to get the top 5 customer_id and top 10 products_id into text files and send it to the judge for scores via Sametime. You should use the insert to local directory function to put the result into text file.

Hints and tips

You will need to use the following tools

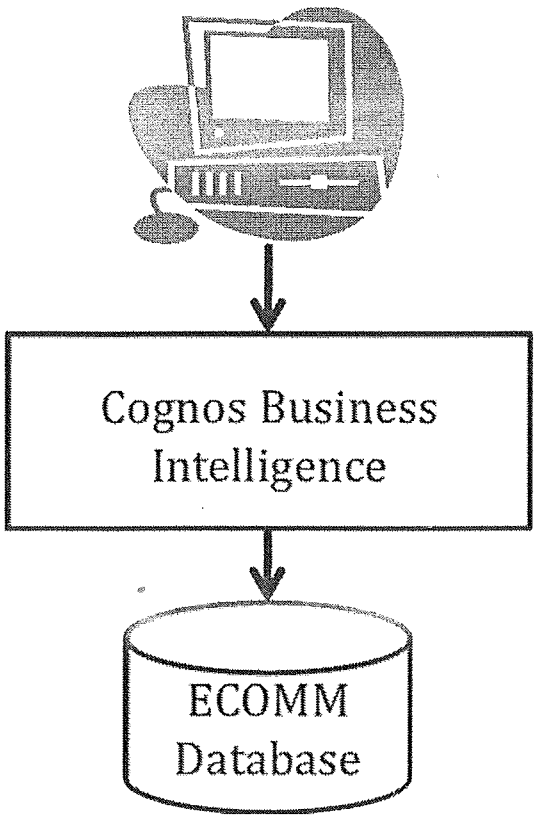
- Putty to access the server which store the log file, please log in Putty using the profile "hadoop"
- WinSCP to transfer the files from Server to local system, please use the hadoop profile in WinSCP Login

Your login username and password to the server (170.225.160.157) has passed to you. You need to access the HDFS and HIVE to perform the task for this question, you only have access of the folder of your team number. (eg team01)

Question 2.8 (50 Points)

Information about the product has been stored at the database for the eComm portal and it is useful for management to understand the business. In this question, you will leverage the Business Intelligence tools to extract the data to present to the management.

Requirements



Leverage the Cognos 10 Business Intelligence installed on your local machine. In this question complete the following tasks

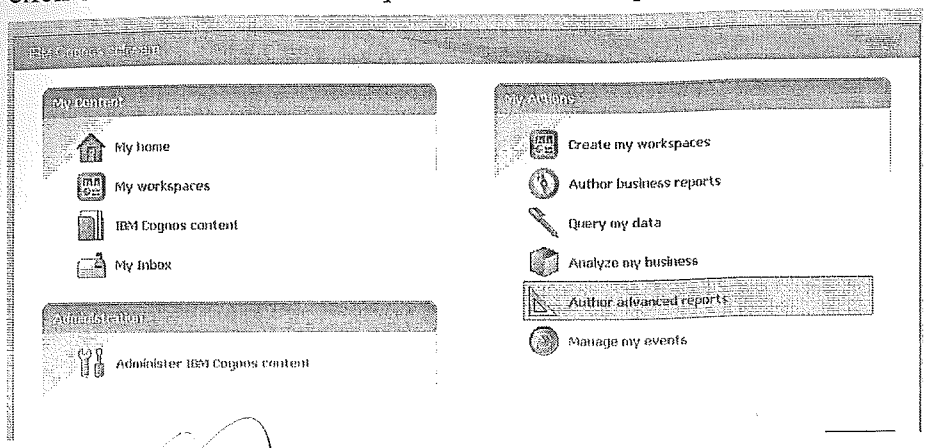
- Create a report with a crosstab and a chart and save in the Cognos Public folder structure.

IBM Inter-University
Programming Contest 2012

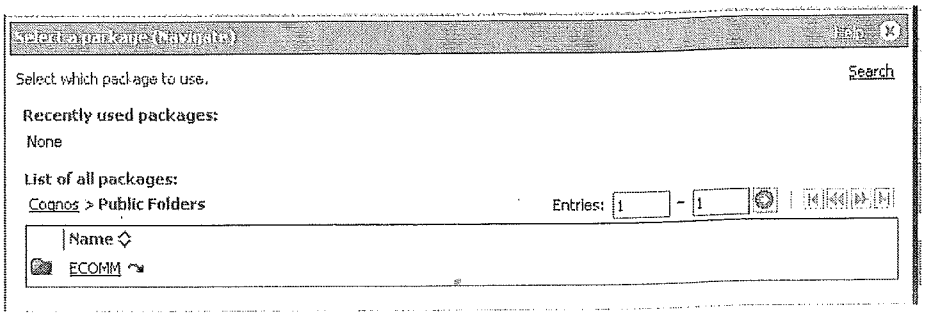
To start the question, you will open your browser and access the following site

http://localhost/ibmcognos,

click on "Author advanced reports" to launch Report studio

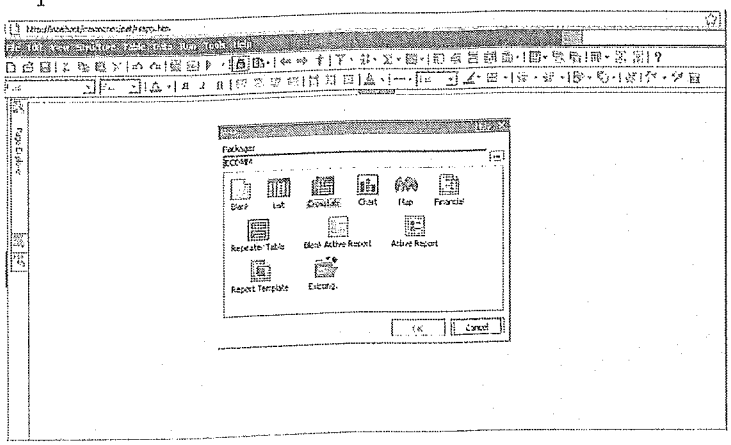


Select the "ECCOM" package for your report, if pop up is disable on your browser, you will need to enable pop up AND repeat the previous step.



IBM Inter-University
Programming Contest 2012

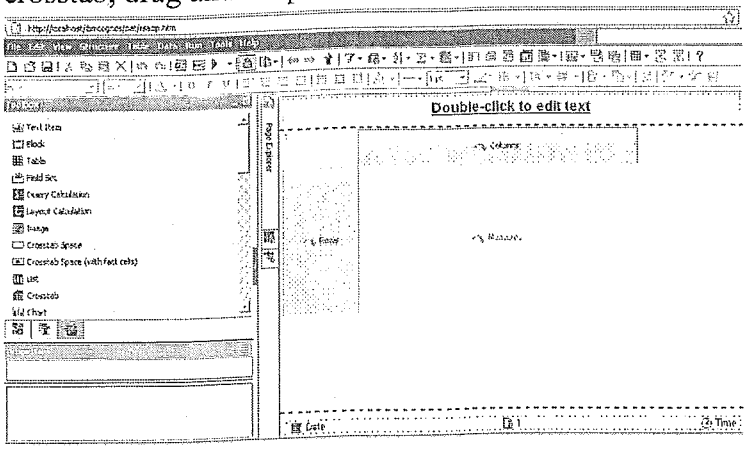
Create your report in report studio and select crosstab for the 1st part of the requirement.



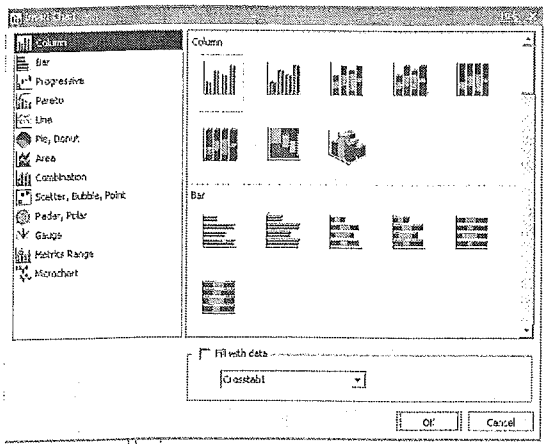
The upper part of the report is a "Crosstab" with

- "Name" in the "Product" table for the "rows" and
- "Unit Price" & "Quantity" in the "Product" tables for the "Column"
- The Crosstab is filtered just to show the 6 "Products" selected as shown in the result

For the 2nd part of the requirement you need to create a column chart below the crosstab, drag and drop chart from the toolbox to the report area



Select column chart at the chart selection box



Create the chart with the following requirement.

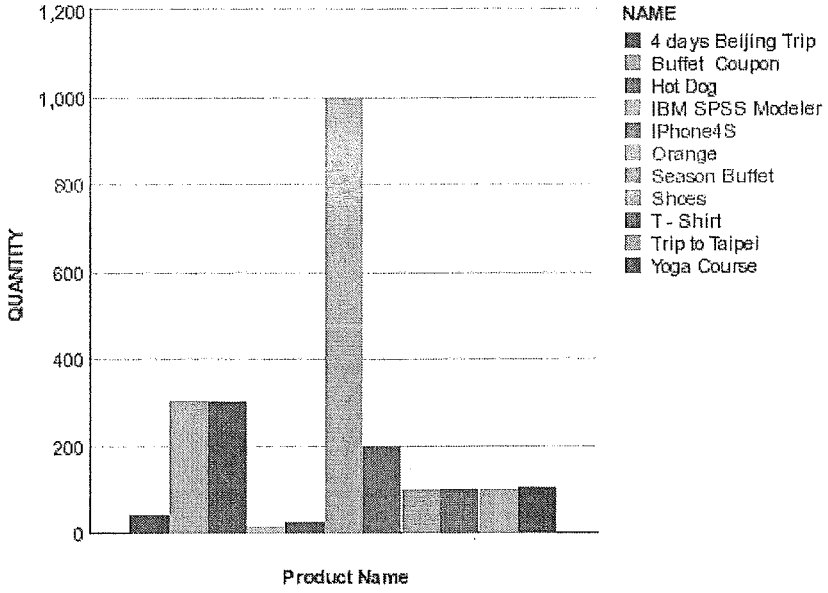
The lower part of the report is "Column Chart" with

- "Name" in the "Product" table as the "Series" and
- "Quantity" as the "default measure".
- The X axis is labeled as "Product Name"

After you complete the question, you save the report as **teamXX** in the ECOMM folder

Results

	UNITPRICE	QUANTITY
4 days Beijing Trip	3,000	40
Buffet Coupon	150	300
Hot Dog	15	300
IBM SPSS Modeler	200	10
Trip to Taipei	3,000	100
Yoga Course	520	105



The report should be saved at the Public Folder->ECCOM folder as "TeamXX"
Please refer to the Hint and tips session for the details of the report requirements

Hints and tips

You will need to use the following tools

- Cognos Business Intelligence Tools – To access the tools access the following website from your browser

Question 2.9 (50 point)

Create the administration portal

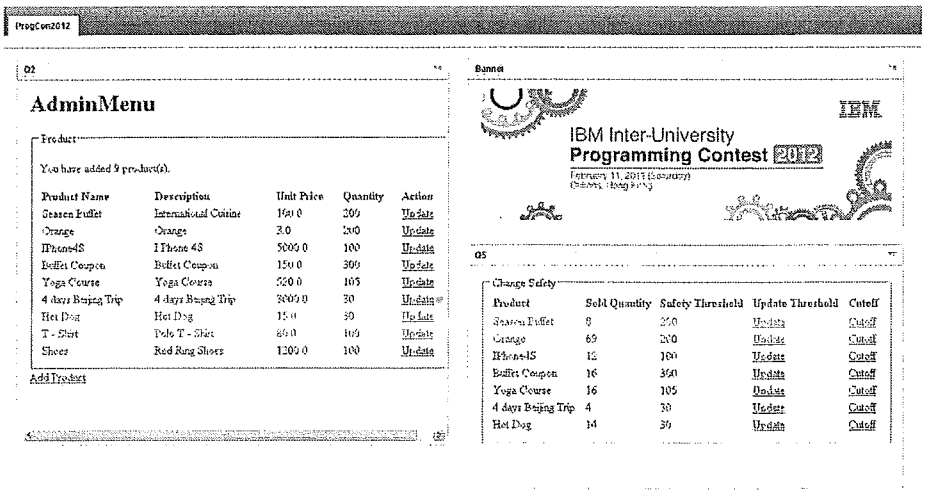
Requirements

Go to **http://portal.ibm.com:10039/wps/myportal** and login with your own ID & password.

Construct a portal screen which consists of 3 portlets constructed in Q2, Q5 and Programming Contest Logo.

Results

To score points for this question, you need to achieve the followings: (score will be rewarded for each complete task)



1, Go to http://portal.ibm.com:10039/wps/myportal and login with your own ID & password. You should create 3 portlets as the upper picture.

2, The titles of those portlets should be Banner, Q2 and Q5.

3, The heights of portlets should be 560 pixels for Q2, 200 pixels for Banner and 300 pixels for Q5.

IBM Inter-University
Programming Contest 2012

Hints and tips

You will need to use the IBM WebSphere Portal for editing your own page

- For portlets for Q2 and Q5, the URL should be:
  http://<your IP>:9080/Q2Web/AdminMenu.jsp.
  http://<your IP>:9080/Q5Web/changeSafety.jsp.
  Don't use localhost as the judge cannot see it.
- The URL of Banner should be http://portal.ibm.com:10039/wps/banner.jpg

Appendix I

ecomm database schema

Table Name: USER

Name	Data Type	Description
USERID	INTEGER(4)	Unique Identity
USERNAME	VARCHAR(10)	User Name for login
PASSWORD	VARCHAR(10)	
TYPE	CHARACTER(1)	User Type('S','A', "C") S means "Special" A means "Admin" C means "Customer"
ADDRESS	VARCHAR(40)	Customer Address
PHONE	DECIMAL(8)	Telephone Number

Table Name: PRODUCT

Name	Data Type	Description
PROID	INTEGER(4)	Product Identity
NAME	VARCHAR(30)	Product Name
DESCRIPTION	VARCHAR(250)	Product Description
UNITPRICE	DECIMAL(7)	Unit price of products
QUANTITY	INTEGER(4)	The expected cut off quantity / Threshold
UID	INTEGER(4)	User ID of product owner (Seller)
CATEGORYID	INTEGER(4)	Product Category ID (100 / 101 /102 / 103)
CUTOFF	CHARACTER(1)	Status whether the offer of product has been cut off
OUTSTANDING	CHARACTER(1)	Status that the sold quantity is more than or equal to the expected cut-off quantity / threshold and that a green omnibus restore message has not been sent.

Table Name: TRANSACTION

Name	Data Type	Description
TID	INTEGER(4)	Transaction Identity
PID	INTEGER(4)	Product Identity
UID	INTEGER(4)	User ID of Buyer
PAYMETHOD	CHARACTER(1)	Payment method C – Credit Card P – Paypal S – PPS E – E banking
BQUAN	INTEGER(4)	The Bought Quantity of each transaction
PRICE	DECIMAL(7)	Total price of each transaction

Table Name: CATEGORY

Name	Data Type	Description
CATEGORYID	INTEGER(4)	Category Identity
NAME	VARCHAR(30)	Name of Category 100 – Food 101 – Travel 102 – others 103 – Electronic Devices

Table Name: PRODUCTRELATION

Name	Data Type	Description
CATNAME1	VARCHAR(20)	Category name 1
CATNAME2	VARCHAR(20)	Category name 2