Creating an XML File and a CSS file for a Database

It is relatively easy to create an XML file from a database table in Java. If you are provided with the name that is registered for the database driver and the table's name, you can use *metadata* to find out the number of columns in the table and the column names.

Information about a database table can be obtained by using a Java interface called ResultSetMetaData. You first have to get a ResultSet from the table. This can be gotten from either a Select or Select – Where command. From this you can get the metadata:

```
    ResultSetMetaData metaData = rs.getMetaData ();
```

There are a number of things you can learn about the table from the metadata, but the most useful are the number of columns and the column names. These are obtained by:

```
    int numberColumns = metaData.getColumnCount ();
    String columnName = metaData.getColumnName (count);
```

This information can be used to display any database table and to create an XML file from the table. The following program not only creates an XML file, but it also creates a cascading style sheet to display the file in a browser. Not all browsers will do this, but both Netscape 7 and FireFox will. Internet Explorer 6 does not, but most likely future versions of IE will also support this feature. A sample program follows.

```
/*  DatabaseToXML reads in the registered name of a database and the name of a table in the database.
It then produces an xml file with the same name as the database and a css file that will format the xml as
an html table. */

package xml_css;

import java.sql.*;
import java.io.*;

public class DatabaseToXML
{
    public static void main (String [] args)
    {
        try
        {
            // Get the name of the database and the table name.
            BufferedReader stdin = new BufferedReader (new InputStreamReader (System.in));
            System.out.print ("DataBase: ");
            String database = stdin.readLine ();
            System.out.print ("Table name: ");
            String tableName = stdin.readLine ();

            // Get a jdbc-odbc bridge and connect to produce.mdb.
            Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
            Connection con = DriverManager.getConnection ("jdbc:odbc:"  + database);

            // Get a result set and metadata for the table.
            Statement stmt = con.createStatement ();
            String query = "Select * From " + tableName;
            ResultSet rs = stmt.executeQuery (query);
            createXMLFile (rs, database, tableName);          // Create the XML file.
            createStyleSheet (rs, database, tableName); // Create the stylesheet.
            con.close ();
        } catch (ClassNotFoundException e){System.out.println ("Class Not Found exception.\n");}
          catch (SQLException e){System.out.println ("SQL Exception\n");}
          catch (IOException eo) {System.out.println ("IO Exception.");}
    } // main

    // createXMLFile saves a copy of the database formatted as an xml table.
    public static void createXMLFile (ResultSet rs, String database, String tableName)
    {
        PrintWriter xmlfile = null;
        try
        {
            xmlfile = new PrintWriter (new FileOutputStream (database + ".xml"));

            // Get the metadata from the database
            ResultSetMetaData metaData = rs.getMetaData ();

            // Get the number of columns from the metadata.
            int numberColumns = metaData.getColumnCount ();

            // Print out a link to the Stylesheet and the root of the xml document.
            xmlfile.println ("<?xml-stylesheet type='text/css' href='" + database + ".css'?>");

            // Get all the data from the database and write it to the xml file.
            xmlfile.println ("<" + database + ">");
            while (rs.next ())
            {
                xmlfile.println ("    <" + tableName + ">");
                for (int count = 1; count<= numberColumns; count ++)
                {
                    String columnName = metaData.getColumnName (count);
                    xmlfile.println ("          <" + columnName + ">"
                        + rs.getString (count) + "</" + columnName + ">");
                }
                xmlfile.println ("    </" + tableName + ">");
            }
            xmlfile.println ("</" + database + ">");
        } catch (SQLException es) {System.out.println ("SQL Exception");}
        catch (FileNotFoundException e) {System.out.println ("File Not Found Exception");}
        System.out.println ("Data saved in " + database + ".xml");
        xmlfile.close ();
    } // saveData

    // createStyleSheet creates a file containing a stylesheet for the XML file.
    public static void createStyleSheet (ResultSet rs, String database, String tableName)
    {
        PrintWriter cssfile = null;
        try
        {
            cssfile = new PrintWriter (new FileOutputStream (database + ".css"));

            // Get the metadata from the database
            ResultSetMetaData metaData = rs.getMetaData ();
            int numberColumns = metaData.getColumnCount (); // Get the number of columns.

            // Create styles for the whole table.
            cssfile.println (database);
            cssfile.println ("{");
            cssfile.println ("     font-family: 'Times New Roman', serif;");
            cssfile.println ("     display: table;");
            cssfile.println ("     margin-left: 1.0cm;");
            cssfile.println ("     margin-top: 1.0cm");
            cssfile.println ("}");

            // Create the styles for the rows of the table.
            cssfile.println (tableName);
            cssfile.println ("{");
            cssfile.println ("     display: table-row;");
            cssfile.println ("}");

            // Create the styles for the cells in the table.
            for (int count = 1; count <= numberColumns; count++)
            {
                cssfile.print (metaData.getColumnName (count));
                if (count < numberColumns)
                    cssfile.print (", ");
            }
            cssfile.println ();
            cssfile.println ("{");
            cssfile.println ("     display: table-cell;");
            cssfile.println ("     border-style: solid;");
            cssfile.println ("     border-width: thin;");
            cssfile.println ("     padding: 0.3cm");
            cssfile.println ("}");
        } catch (SQLException es) {System.out.println ("SQL Exception");}
        catch (FileNotFoundException e) {System.out.println ("File Not Found Exception");}
        System.out.println ("Data saved in " + database + ".css");
        cssfile.close ();
    }
} //  DatabaseToXML
```

If we now have a database called grocery.mdb with a table called fruit, we can run the program above to create two files, grocery.xml and grocery.css. These are shown below.

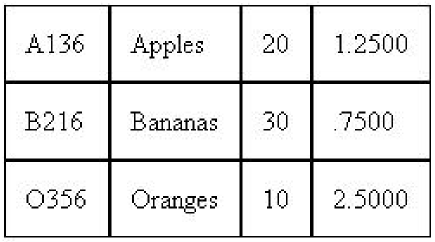
grocery.xml

```
<?xml-stylesheet type='text/css' href='grocery.css'?>
<grocery>
    <fruit>
        <id>A136</id>
        <name>Apples</name>
        <quantity>20</quantity>
        <price>1.2500</price>
    </fruit>
    <fruit>
        <id>B216</id>
        <name>Bananas</name>
        <quantity>30</quantity>
        <price>.7500</price>
    </fruit>
    <fruit>
        <id>O356</id>
        <name>Oranges</name>
        <quantity>10</quantity>
        <price>2.5000</price>
    </fruit>
</grocery>
```

grocery.css

```
grocery
{
    font-family: 'Times New Roman', serif;
    display: table;
    margin-left: 1.0cm;
    margin-top: 1.0cm
}
fruit
{
    display: table-row;
}
id, name, quantity, price
{
    display: table-cell;
    border-style: solid;
    border-width: thin;
    padding: 0.3cm
}
```

When viewed in a browser, the result is:



A136	Apples	20	1.2500
B216	Bananas	30	.7500
O356	Oranges	10	2.5000