

Exercise: Simple JDBC

The Exercise

This simple exercise will give you some familiarity with using Java's JDBC facility to place a simple query against a small database.

Notes:

- Your instructor should be able to supply you with the database file needed for this exercise. You will need to set up your database system before attempting this exercise.
- As written, this exercise uses Sun's JDBC-ODBC 'bridge' driver. If a different database driver is to be used, you will need to modify the DRIVER and URLPREFIX final variables accordingly. Your instructor should be able to tell you what modifications are needed.

1. Create the Java Source Code

Create a text file called *ShowTable.java*, containing the following Java code:

```
import java.sql.*;
import java.io.*;
import java.util.*;

public class ShowTable
{
    private static final PrintWriter out = new PrintWriter (System.out, true),
                                err = new PrintWriter (System.err, true);
    private static String DRIVER = "sun.jdbc.odbc.JdbcOdbcDriver",
                        URLPREFIX = "jdbc:odbc:";

    static
    {
        try
        {
            Class.forName (DRIVER);
        }
        catch (ClassNotFoundException cnfe)
        {
            cnfe.printStackTrace (err);
            System.exit (-1);
        }
    }

    private static void displayQueryResults (ResultSet rs)
        throws SQLException
    {
        ResultSetMetaData rsmd = rs.getMetaData ();

        int numCols = rsmd.getColumnCount ();

        for (int i = 1; i <= numCols; i++)
        {
            if (i > 1)
                out.print (",");
            out.print (rsmd.getColumnLabel (i));
        }

        out.println ();

        while (rs.next ())
        {
            for (int i = 1; i <= numCols; i++)
            {
                if (i > 1)
                    out.print (",");
                out.print (rs.getString (i));
            }
            out.println ();
        }
    }

    private static void displayDBMetaData (DatabaseMetaData dbMeta)
```

```

throws SQLException
{
out.println ();
out.println ("Connected to: " + dbMeta.getURL ());
out.println ("Driver: " + dbMeta.getDriverName ());
out.println ("Driver version: " + dbMeta.getDriverVersion ());
out.println ("Product name: " + dbMeta.getDatabaseProductName ());
out.println ("Product version: " + dbMeta.getDatabaseProductVersion ());
}

private static void dumpSQLException (SQLException sqle)
{
err.println ();
err.println ("--- SQLException caught ---");
err.println ();
while (sqle != null)
{
err.println ("SQLState: " + sqle.getSQLState ());
err.println ("ErrorCode: " + sqle.getErrorCode ());
err.println ("Message: " + sqle.getMessage ());
err.println();
sqle = sqle.getNextException ();
}
}

public static void main (String [] args)
{
Connection dbConn = null;
try
{
Properties info = new Properties ();

info.put ("user", args [0]);
info.put ("password", args [1]);

dbConn = DriverManager.getConnection (URLPREFIX + args [2], info);
out.println (URLPFX + args [2]);
displayDBMetaData (dbConn.getMetaData ());
Statement statement = null;
try
{
statement = dbConn.createStatement ();
displayQueryResults
(statement.executeQuery ("SELECT * FROM " + args [3]));
}
catch (SQLException se)
{
throw se;
}
finally
{
try { statement.close (); } catch (Exception e) { /* SQUELCH! */ }
}
catch (SQLException sqle)
{
dumpSQLException (sqle);
}
finally
{
try { dbConn.close (); } catch (Exception e) { /* SQUELCH! */ }
}
}
}

```

2. Compile the Java Source

```

prompt> javac ShowTable.java
prompt>

```

3. Run the Application

```

prompt> java ShowTable username password dbname tablename
prompt>

```