PC Grid Database

MySQL –

There were two databases associated with the PC Grid Database: Cluster and Grid The Grid database was to a table named GRIDRCS. This table holds various forms of information about the user, hours, memory, etc. The Cluster database has a GRIDRCS table, a table named PARENT, and a table named CLUSTER. The PARENT table held only the IP name. The CLUSTER on the other hand held the IP Name, CPU Utilization, Free Memory, Available Disk, and the Actual-Intra-Network Bandwidth. The tables are shown below as they were inserted:

GRIDRCS Table-

```
CREATE TABLE GRIDRCS(IPName VARCHAR(25),IPAddress
VARCHAR(25),
CurrentIPAddress VARCHAR(25),Owner VARCHAR(50),Phone
VARCHAR(20),
Email VARCHAR(50),TimeZone TINYINT(5),OfferedHours1 VARCHAR(30),
OfferedHours2 VARCHAR(30),OfferedHours3 VARCHAR(30),
CPUArchitecture VARCHAR(50),CPUPerformance INT(30),
OSType VARCHAR(30),Memory INT(30),FreeMemory INT(30),
NumberOfCPUs INT(5),ActualNumberOfCPUs INT(5),CPUUtilization INT(5),
Disk FLOAT(30),AvailableDisk INT(30),InterNetworkDevice VARCHAR(30),
InterNetworkBandwidth INT(30),ActualInterNetworkBandwidth INT(30)
,IntraNetworkDevice VARCHAR(30),IntraNetworkBandwidth INT(30),
ActualIntraNetworkBandwidth INT(30),Libraries VARCHAR(30),
Application VARCHAR(50));
```

PARENT Table-

```
CREATE TABLE PARENT(IPName VARCHAR(50));
```

CLUSTER Table-

```
CREATE TABLE CLUSTER(IPName VARCHAR(25),CPUUtilization INT(30),
FreeMemory INT(30),AvailableDisk INT(30),ActualIntraNetworkBandwidth
INT(30));
```

Currently the tables work fine, but none of the tables have a primary key. The Console Manager that MySQL uses is not fully functional, crashing whenever a user wants to change the field of the table. The cause of this error is unknown and has nothing to do with any code that has been written this quarter. During the course of the quarter the database was always active. This allowed us to manipulate the database as long as there was access to the web.

Java Servlets/Tomcat Server-

There were four servlets written for the tomcat server: Insert, Delete, Update, and Display. The servlets are placed in the webapps folder in the tomcat server. Inside the webapps folder are other folders which have been added to the configuration file. The user can type in localhost and the folder name in their browser to access the html and servlet files. For instance, the user would type http://localhost/mystuff/servlet/index to access the index servlet.

To add these servlets to a tomcat server the user can use Ant to help create the folder or use one of the other folders that is already present. You could take the folder, copy it, place your servlets in the classes folder, and add the folder to the configuration file. You would then be able to access the servlets through the tomcat server.

The servlets work by displaying the first of the two functions to the user. The function runs through and displays the HTML to the user. The user then enters the information into the fields and presses the appropriate button; in this case it would be the "submit" button. The form will then run through the servlet again, gathering the information in the fields and manipulating the database based upon the user entries. Each servlet is summarized below:

Insert.java-

Insert begins by prompting the user to enter information into the fields on the page. The user enters the information into the fields and presses the submit button after all the fields have been filled out. The servlet then runs the second function named doGet. This function runs through the first function grabbing all the information the user entered. This information is then placed in variables matching the fields in the table. (Such as GRIDRCS, PARENT, etc.) The variables are then placed in SQL syntax to correctly insert the information into the database. If everything was completed successfully, a message is displayed to the user. If an error occurred, the appropriate exception was thrown.

Delete.java-

Delete was designed according to Professor Fukuda's specification. The user is prompted for a row to delete. The servlet then goes through the first function again and retrieves the row from the text field and uses the information gathered in SQL syntax to

delete the row. If the row exists, the row is removed from the database and a message is displayed to the user that the delete was successful. If the delete wasn't successful, for example; If the user chose to delete a row that didn't exist, an exception would be thrown.

Display.java-

Display has more components than the other servlets. There are two options in this servlet. One is to enter in the range of rows that you wish to see, and the other is to display all of the rows in the entire database. After the user hits the submit button, the servlet first looks to see if "display all" was checked. If it was, all the information is formatted into rows and displayed on the screen. If "display all" wasn't checked, the servlet retrieves the rows the user entered from the text box and places them in SQL syntax. The result returned is then formatted and placed in a formatted table.

Update.java-

Update is similar to the Insert.java, but it asks for the row to update in addition to all of the other information. After the row and information is entered into the fields, the servlet checks to see what values are null. If the field is null, the variable to be used in the SQL syntax is set to null, otherwise the variable is set to the value in the text field. This process occurs for every field in the servlet. After this process has concluded, the SQL syntax is created and the appropriate row is updated. If the user enters in a row that does not exist, an exception is thrown to the console indicating the error has occurred.

Servlet Source Code-

These servlets were created by Eric Nelson and Jon Hagen for use with the PC Grid Database. This code reflects the servlets as of 6/10/2003. There are still bugs with some of the servlets relating to the delete.java and update.java with regards to the insert row. These bugs should be removed shortly.

Insert.java-

```
/*
 * Name : Insert.java
 * Date : 4/20/2003
 * Author : Jon Hagen & Eric Nelson
 *
 * Description:  This is the servlet used to insert information into the
 * resource database.
 *
```

```
 */

import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.text.DateFormat;
import java.lang.Integer;
import java.lang.Double;


public class Insert extends HttpServlet {


   public void doGet(HttpServletRequest request,
                     HttpServletResponse response)
      throws IOException, ServletException
   {
System.out.println("~/system_new/tomcat/webapps/metacomputer/WEB-
INF/classes/Insert.java ... doGet()\n");
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();
      out.println("<html>");
      out.println("<body>");
      out.println("<head>");

      String title = new String("Insert into Resource Database");
      out.println("<title>" + title + "</title>");
      out.println("</head>");
      out.println("<body bgcolor=\"white\">");


                  //Print out the title to the screen
      out.println("<h3>" + title + "</h3>");

      //Assign names to string variables
      String firstName = request.getParameter("firstname");
      String lastName = request.getParameter("lastname");
      String checkBox = request.getParameter("checkbox");
      String ipName = request.getParameter("ipname");
      String ipAddress = request.getParameter("ipaddress");
      String curIp = request.getParameter("curip");
      String eOwner = request.getParameter("eowner");
      String phone = request.getParameter("phone");
```

```
String email = request.getParameter("email");
String timeZone = request.getParameter("timezone");
String offerOne = request.getParameter("offerone");
String offerTwo = request.getParameter("offertwo");
String offerThree = request.getParameter("offerthree");
String cpuArch = request.getParameter("cpuarch");
String cpuPerf = request.getParameter("cpuperf");
String osType = request.getParameter("ostype");
String memory = request.getParameter("memory");
String freeMemory = request.getParameter("freememory");
String numCpu = request.getParameter("numcpu");
String actualCpu = request.getParameter("actualcpu");
String cpuUtil = request.getParameter("cpuutil");
String disk = request.getParameter("disk");
String availDisk = request.getParameter("availdisk");
String interDevice = request.getParameter("interdevice");
String interBand = request.getParameter("interband");
String actualInter = request.getParameter("actualinter");
String intraDevice = request.getParameter("intradevice");
String intraBand = request.getParameter("intraband");
String actualIntra = request.getParameter("actualintra");
String libraries = request.getParameter("libraries");
String application = request.getParameter("application");


//If String variables received values, then insert completed
//****USED TO TEST****
if (ipName!= null || ipAddress != null) {

        String jdbcDriver = "com.mysql.jdbc.Driver";

        String dbURL ="jdbc:mysql:///grid";
        Connection dbConn = null;
        Statement stmt = null;

            try
            {
                Class.forName(jdbcDriver) .newInstance() ;
                dbConn = DriverManager.getConnection(dbURL,
                "jhagen", "jhagen1");


                //Cast strings variables to int to be placed in database
                int timeZone_i = Integer.parseInt(timeZone);
                int cpuPerf_i = Integer.parseInt(cpuPerf);
                int memory_i = Integer.parseInt(memory);
```

```
                        int freeMemory_i = Integer.parseInt(freeMemory);
                        int numCpu_i = Integer.parseInt(numCpu);
                        int actualCpu_i = Integer.parseInt(actualCpu);
                        int cpuUtil_i = Integer.parseInt(cpuUtil);
                        double disk_d = Double.parseDouble(disk);
                        int availDisk_i = Integer.parseInt(availDisk);
                        int interBand_i = Integer.parseInt(interBand);
                        int actualInter_i = Integer.parseInt(actualInter);
                        int intraBand_i = Integer.parseInt(intraBand);
                        int actualIntra_i = Integer.parseInt(actualIntra);

                        String sql = "insert into GRIDRCS " +
                                "(IPName,IPAddress,CurrentIPAddress," +

"Owner,Phone,Email,TimeZone,OfferedHours1,OfferedHours2," +

"OfferedHours3,CPUArchitecture,CPUPerformance,OSType," +

"Memory,FreeMemory,NumberOfCPUs,ActualNumberOfCPUs," +

"CPUUtilization,Disk,AvailableDisk,InternetworkDevice," +

"InternetworkBandwidth,ActualInternetworkBandwidth," +

"IntraNetworkDevice,IntraNetworkBandwidth," +

"ActualIntraNetworkBandwidth,Libraries,Application,id) " +
                                        "values " +
                                            "('" + ipName + "','" + ipAddress + "','" +
                                            curIp + "','" +
                                            eOwner + "','" + phone + "','" + email + "',"
                                            + timeZone_i + ",'" +
                                            offerOne + "','" + offerTwo + "','" +
                                            offerThree + "','" +
                                            cpuArch + "'," + cpuPerf_i + ",'" + osType +
                                            "'," + memory_i + "," +
                                            freeMemory_i + "," + numCpu_i + "," +
                                            actualCpu_i + "," +
                                            cpuUtil_i + "," + disk_d + "," + availDisk_i
                                            + ",'" +
                                            interDevice + "'," + interBand_i + "," +
                                            actualInter_i + ",'" +
                                                    intraDevice + "'," + intraBand_i + ","
                                            + actualIntra_i + ",'" +
                                    libraries + "','" + application + "'," + 3 +  ");";
```

```
                                stmt = dbConn.createStatement();
                                boolean results = stmt.execute(sql);
                                if(results) System.out.println("results = TRUE\n");
                                else System.out.println("results = FALSE\n");
                                stmt.close();


                                /*stmt =
dbConn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCU
R_UPDATABLE);
                                String sql = "select " +
                                                "IPName,IPAddress,CurrentIPAddress," +

        "Owner,Phone,Email,TimeZone,OfferedHours1,OfferedHours2," +

        "OfferedHours3,CPUArchitecture,CPUPerformance,OSType," +

        "Memory,FreeMemory,NumberOfCPUs,ActualNumberOfCPUs," +

        "CPUUtilization,Disk,AvailableDisk,InternetworkDevice," +

        "InternetworkBandwidth,ActualInternetworkBandwidth," +

        "IntraNetworkDevice,IntraNetworkBandwidth," +

        "ActualIntraNetworkBandwidth,Libraries,Application " +
                                        "from GRIDRCS;";
                                ResultSet rs = stmt.executeQuery(sql);
                                //move to insert row
                                rs.moveToInsertRow();

                                //update all columns in insert row
                                rs.updateString("IPName", ipName);
                                rs.updateString("IPAddress", ipAddress);
                                rs.updateString("CurrentIPAddress", curIp);
                                rs.updateString("Owner",eOwner);
                                rs.updateString("Phone", phone);
                                rs.updateString("Email", email);
                                rs.updateInt("TimeZone", timeZone_i);
                                rs.updateString("OfferedHours1", offerOne);
                                rs.updateString("OfferedHours2", offerTwo);
                                rs.updateString("OfferedHours3", offerThree);
                                rs.updateString("CPUArchitecture", cpuArch);
                                rs.updateInt("CPUPerformance", cpuPerf_i);
                                rs.updateString("OSType", osType);
                                rs.updateInt("Memory", memory_i);
```

```
            rs.updateInt("FreeMemory", freeMemory_i);
            rs.updateInt("NumberOfCPUs", numCpu_i);
            rs.updateInt("ActualNumberOfCPUs", actualCpu_i);
            rs.updateInt("CPUUtilization", cpuUtil_i);
            rs.updateDouble("Disk",disk_d);
            rs.updateInt("AvailableDisk", availDisk_i);
            rs.updateString("InterNetworkDevice", interDevice);
            rs.updateInt("InterNetworkBandwidth", interBand_i);

            rs.updateInt("ActualInterNetworkBandwidth",actualInter_i)
            ;
            rs.updateString("IntraNetworkDevice", intraDevice);
            rs.updateInt("IntraNetworkBandwidth", intraBand_i);

            rs.updateInt("ActualIntraNetworkBandwidth",actualIntra_i)
            ;
            rs.updateString("Libraries", libraries);
            rs.updateString("Application", application);

            rs.insertRow(); //insert row into ResultSet and database
          //rs.close();
            stmt.close();*/

}
catch (ClassNotFoundException e)
{
            System.out.println("JDBC driver not found: " +
            jdbcDriver);
            System.out.println(e.getMessage());
}
catch   (SQLException e)
{
            System.out.println("Error retrieving data from: " +
            dbURL);
            System.out.println(e.getMessage());
}
catch (Exception e)
{
            System.out.println("Error: " + e);
            System.out.println(e.getMessage());
}
finally
{
            try
            {
                        if (dbConn != null)
```

```
                                        {
                                                dbConn.close();
                                        }
                                }
                        catch (SQLException ignored) {}
                        }

    out.println("Insert was completed successfully!");
                        out.println("<br><br><a href=\"/metacomputer/servlet/Index\">");
                        out.println(" Back to Main Menu </a>");
                ipName = null;
    ipAddress = null;
}
else{
        out.println("<P>");
        out.print("<form action=\"");
        out.print("Insert\" ");
        out.println("method=POST>");
        out.println("IP Name:  ");
        out.println("<input type=text size=40 name=ipname>");
        out.println("<br>");
        out.println("IP Address:  ");
        out.println("<input type=text size=40 name=ipaddress>");
        out.println("<br>");
        out.println("Current IP Address:  ");
        out.println("<input type=text size=40 name=curip>");
        out.println("<br>");
        out.println("Owner:  ");
        out.println("<input type=text size=40 name=eowner>");
        out.println("<br>");
        out.println("Phone:  ");
        out.println("<input type=text size=40 name=phone>");
        out.println("<br>");
        out.println("Email:  ");
        out.println("<input type=text size=40 name=email>");
        out.println("<br>");
        out.println("Time Zone:  ");
        out.println("<input type=text size=40 name=timezone>");
        out.println("<br>");
        out.println("Offered Hours 1:  ");
        out.println("<input type=text size=40 name=offerone>");
        out.println("<br>");
        out.println("Offered Hours 2:  ");
        out.println("<input type=text size=40 name=offertwo>");
        out.println("<br>");
        out.println("Offered Hours 3:  ");
```

```
out.println("<input type=text size=40 name=offerthree>");
out.println("<br>");
out.println("CPU Architecture:  ");
out.println("<input type=text size=40 name=cpuarch>");
out.println("<br>");
out.println("CPU Performance:  ");
out.println("<input type=text size=40 name=cpuperf>");
out.println("<br>");
out.println("OS Type:  ");
out.println("<input type=text size=40 name=ostype>");
out.println("<br>");
out.println("Memory:  ");
out.println("<input type=text size=40 name=memory>");
out.println("<br>");
out.println("Free Memory:  ");
out.println("<input type=text size=40 name=freememory>");
out.println("<br>");
out.println("#CPU's:  ");
out.println("<input type=text size=40 name=numcpu>");
out.println("<br>");
out.println("Actual #CPU's:  ");
out.println("<input type=text size=40 name=actualcpu>");
out.println("<br>");
out.println("CPU Utilization:  ");
out.println("<input type=text size=40 name=cpuutil>");
out.println("<br>");
out.println("Disk:  ");
out.println("<input type=text size=40 name=disk>");
out.println("<br>");
out.println("Available Disk:  ");
out.println("<input type=text size=40 name=availdisk>");
out.println("<br>");
out.println("Inter Network Device:  ");
out.println("<input type=text size=40 name=interdevice>");
out.println("<br>");
out.println("Inter Network Bandwidth:  ");
out.println("<input type=text size=40 name=interband>");
out.println("<br>");
out.println("Actual Inter Network Bandwidth:  ");
out.println("<input type=text size=40 name=actualinter>");
out.println("<br>");
out.println("Intra Network Device:  ");
out.println("<input type=text size=40 name=intradevice>");
out.println("<br>");
out.println("Intra Network Bandwidth:  ");
out.println("<input type=text size=40 name=intraband>");
```

```
                out.println("<br>");
                out.println("Actual Intra Network Bandwidth:  ");
                out.println("<input type=text size=40 name=actualintra>");
                out.println("<br>");
                out.println("Libraries:  ");
                out.println("<input type=text size=40 name=libraries>");
                out.println("<br>");
                out.println("Application:  ");
                out.println("<input type=text size=40 name=application>");
                out.println("<br>");

                out.println("<input type=submit>");

                out.println("</form>");

                out.println("</body>");
                out.println("</html>");
            }
    }

    public void doPost(HttpServletRequest request,
                       HttpServletResponse response)
        throws IOException, ServletException
    {
System.out.println("~/system_new/tomcat/webapps/metacomputer/WEB-
INF/classes/Insert.java ... doPost()\n");
        doGet(request, response);
    }

}
```

Delete.java-

```
/*
 * Name : Delete.java
 * Date : 4/20/2003
 * Author : Jon Hagen & Eric Nelson
 *
 * Description:  This is the servlet used to delete information from the
 * resource database.
 *
 */

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

```
import javax.servlet.*;
import javax.servlet.http.*;


public class Delete extends HttpServlet {


   //ResourceBundle rb = ResourceBundle.getBundle("LocalStrings");

   public void doGet(HttpServletRequest request,
                     HttpServletResponse response)
      throws IOException, ServletException
   {
System.out.println("~/system_new/tomcat/webapps/metacomputer/WEB-
INF/classes/Delete.java ... doGet()\n");
      response.setContentType("text/html");

      PrintWriter out = response.getWriter();
      out.println("<html>");
      out.println("<body>");
      out.println("<head>");

      String title = new String("Delete from Resource Database");
      out.println("<title>" + title + "</title>");
      out.println("</head>");
      out.println("<body bgcolor=\"white\">");


                  //Print out the title to the screen
      out.println("<h3>" + title + "</h3>");

      //Assign names to string variables
      String c = request.getParameter("corpse");

      //If String variables received values, then insert completed
      //****USED TO TEST****
      if (c != null) {

            String jdbcDriver = "com.mysql.jdbc.Driver";
            String dbURL ="jdbc:mysql:///grid";
            Connection dbConn = null;
            Statement stmt = null;
            int rowId = Integer.parseInt(c);

            try
            {
```

```
            Class.forName(jdbcDriver) .newInstance() ;
            dbConn = DriverManager.getConnection(dbURL, "jhagen", "jhagen1");
            stmt = dbConn.createStatement();

            String sql = "DELETE From GRIDRCS WHERE id = " + rowId  + ";";

            int numRows = stmt.executeUpdate(sql);

            System.out.println(numRows + " rows deleted.");

            stmt.close();
        }
        catch (ClassNotFoundException e)
        {
            System.out.println("JDBC driver not found: " + jdbcDriver);
        }
        catch (SQLException e)
        {
            System.out.println("Error retrieving data from: " + dbURL);
        }
        catch (Exception e)
        {
            System.out.println("Error: " + e);
        }
        finally
        {
            try
            {
               if (dbConn != null)
               {
                    dbConn.close();
               }
            }
            catch (SQLException ignored) {}
        }

    out.println("Delete was completed successfully!");
                    out.println("<br><br><a href=\"/metacomputer/servlet/Index\">");
                    out.println(" Back to Main Menu </a>");
                    c = null;
}
else{
        out.println("<P>");
        out.print("<form action=\"");
        out.print("Delete\" ");
        out.println("method=POST>");
```

```
                out.println("Row to Delete:  ");
                out.println("<input type=text size=10 name=corpse>");
                out.println("<br>");

                out.println("<input type=submit>");

                out.println("</form>");

                out.println("</body>");
                out.println("</html>");
            }
    }

    public void doPost(HttpServletRequest request,
                       HttpServletResponse response)
        throws IOException, ServletException
    {
System.out.println("~/system_new/tomcat/webapps/metacomputer/WEB-
INF/classes/Delete.java ... doPost()\n");
        doGet(request, response);
    }

}
```

Display.java-

```
/*
 * Name : Display.java
 * Date : 4/20/2003
 * Author : Jon Hagen & Eric Nelson
 *
 * Description:  This is the servlet used to display information from the
 * resource database.
 *
 */

import java.io.*;
import java.sql.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.lang.Integer;


public class Display extends HttpServlet {
```

```
   //ResourceBundle rb = ResourceBundle.getBundle("LocalStrings");

   public void doGet(HttpServletRequest request,
                     HttpServletResponse response)
      throws IOException, ServletException
   {
System.out.println("~/system_new/tomcat/webapps/metacomputer/WEB-
INF/classes/Display.java ... doGet()\n");
      response.setContentType("text/html");

      PrintWriter out = response.getWriter();
      out.println("<html>");
      out.println("<body>");
      out.println("<head>");

      String title = new String("Display Resource Database Entries");
      out.println("<title>" + title + "</title>");
      out.println("</head>");
      out.println("<body bgcolor=\"white\">");


                  //Print out the title to the screen
      out.println("<h3>" + title + "</h3>");

      //Assign names to string variables
      String fromRow = request.getParameter("fromrow");
      String toRow = request.getParameter("torow");
      String checkBox = request.getParameter("box");
      String sql = null;

      int fromRow_i = 0;
         int toRow_i = 0;

         if(checkBox != null) {
            fromRow = null;
            toRow = null;
         }

         if(fromRow != null) fromRow_i = Integer.parseInt(fromRow);
         if(toRow != null) toRow_i = Integer.parseInt(toRow);
         int increment = toRow_i - fromRow_i + 1;

      //If String variables received values, then insert completed
      //****USED TO TEST****
```

```
if((checkBox != null) && (increment>=0))
{
	String jdbcDriver = "com.mysql.jdbc.Driver";
			String dbURL ="jdbc:mysql:///grid";
			Connection dbConn = null;
			Statement stmt = null;

			try
			{
				Class.forName(jdbcDriver).newInstance() ; //load driver
							dbConn =
				DriverManager.getConnection(dbURL, "jhagen",
				"jhagen1"); //connect
			}
			catch(ClassNotFoundException e)
			{
				throw new UnavailableException("JDBC driver not
				found:" +
				jdbcDriver);
			}
			catch (SQLException e)
			{
				throw new UnavailableException("Unable to connect to: "
				+
				dbURL);
			}
			catch (Exception e)
			{
				throw new UnavailableException("Error: " + e);
			}

			response.setContentType("text/html");

			out = response.getWriter();

			try
			{
			//Display all data from the database

			sql = "select * FROM GRIDRCS;";
			stmt = dbConn.createStatement();
			ResultSet rs = stmt.executeQuery(sql);

			out.println("<HTML>");
			out.println("<HEAD><TITLE>Grid Resource
Database</TITLE></HEAD>");
```

```
out.println("<BODY>");
out.println("<TABLE BORDER=\"1\" CELLPADDING=\"3\">");
out.println("<TR>");

out.println("<TH>IPName</TH>");
out.println("<TH>IPAddress</TH>");
out.println("<TH>Current IP Address</TH>");
out.println("<TH>Owner</TH>");
out.println("<TH>Phone</TH>");
out.println("<TH>Email</TH>");
out.println("<TH>Time Zone</TH>");
out.println("<TH>Offered Hours 1</TH>");
out.println("<TH>Offered Hours 2</TH>");
out.println("<TH>Offered Hours 3</TH>");
out.println("<TH>CPU Architecture</TH>");
out.println("<TH>CPU Performance</TH>");
out.println("<TH>OS Type</TH>");
out.println("<TH>Memory</TH>");
out.println("<TH>Free Memory</TH>");
out.println("<TH>#CPUs</TH>");
out.println("<TH>Actual CPUs</TH>");
out.println("<TH>CPU Utilization</TH>");
out.println("<TH>Disk</TH>");
out.println("<TH>Available Disk</TH>");
out.println("<TH>Inter Network Device</TH>");
out.println("<TH>Inter Network Bandwidth</TH>");
out.println("<TH>Actual Inter Network Bandwidth</TH>");
out.println("<TH>Intra Network Device</TH>");
out.println("<TH>Intra Network Bandwidth</TH>");
out.println("<TH>Actual Intra Network Bandwidth</TH>");
out.println("<TH>Libraries</TH>");
out.println("<TH>Application</TH>");

out.println("</TR>");

        while (rs.next())
        {
                out.println("<TR>");

                        out.println("<TD>" +
                rs.getString("IPName") + "</TD>");
                out.println("<TD>" + rs.getString("IPAddress") +
                "</TD>");
                out.println("<TD>" +
                rs.getString("CurrentIPAddress") + "</TD>");
```

```
out.println("<TD>" + rs.getString("Owner") +
"</TD>");
out.println("<TD>" + rs.getString("Phone") +
"</TD>");
out.println("<TD>" + rs.getString("Email") +
"</TD>");
out.println("<TD>" + rs.getInt("TimeZone") +
"</TD>");
out.println("<TD>" +
rs.getString("OfferedHours1") + "</TD>");
out.println("<TD>" +
rs.getString("OfferedHours2") + "</TD>");
out.println("<TD>" +
rs.getString("OfferedHours3") + "</TD>");
out.println("<TD>" +
rs.getString("CPUArchitecture") + "</TD>");
out.println("<TD>" + rs.getInt("CPUPerformance")
+ "</TD>");
out.println("<TD>" + rs.getString("OSType") +
"</TD>");
out.println("<TD>" + rs.getInt("Memory") +
"</TD>");
out.println("<TD>" + rs.getInt("FreeMemory") +
"</TD>");
out.println("<TD>" + rs.getInt("NumberOfCPUs")
+ "</TD>");
out.println("<TD>" +
rs.getInt("ActualNumberOfCPUs") + "</TD>");
out.println("<TD>" + rs.getInt("CPUUtilization") +
"</TD>");
out.println("<TD>" + rs.getDouble("Disk") +
"</TD>");
out.println("<TD>" + rs.getInt("AvailableDisk") +
"</TD>");
out.println("<TD>" +
rs.getString("InterNetworkDevice") + "</TD>");
out.println("<TD>" +
rs.getInt("InterNetworkBandwidth") + "</TD>");
out.println("<TD>" +
rs.getInt("ActualInterNetworkBandwidth") +
"</TD>");
out.println("<TD>" +
rs.getString("IntraNetworkDevice") + "</TD>");
out.println("<TD>" +
rs.getInt("IntraNetworkBandwidth") + "</TD>");
```

```
                                out.println("<TD>" +
                                rs.getInt("ActualIntraNetworkBandwidth") +
                                "</TD>");
                                out.println("<TD>" + rs.getString("Libraries") +
                                "</TD>");
                                out.println("<TD>" + rs.getString("Application") +
                                "</TD>");

                                out.println("</TR>");
                        }

                        out.println("</TABLE>");
                        out.println("</BODY></HTML>");

                        rs.close();
                        stmt.close();
                }
                catch (SQLException e)
                {
                        out.println("<H2>Database currently unavailable.</H2>");
                }
                out.close();




   out.println("Display All was completed successfully!");
   checkBox = null;
                out.println("<br><br><a href=\"/metacomputer/servlet/Index\">");
                out.println(" Back to Main Menu </a>");
        }

else if ((fromRow!= null || toRow != null)&& (increment>=0)) {

        String jdbcDriver = "com.mysql.jdbc.Driver";
                String dbURL ="jdbc:mysql:///grid";
                Connection dbConn = null;
                Statement stmt = null;

                try
                {
                        Class.forName(jdbcDriver) .newInstance() ; //load driver
                                                        dbConn =
                        DriverManager.getConnection(dbURL, "jhagen",
                        "jhagen1"); //connect
                }
```

```
catch(ClassNotFoundException e)
{
        throw new UnavailableException("JDBC driver not
        found:" +
        jdbcDriver);
}
catch (SQLException e)
{
        throw new UnavailableException("Unable to connect to: "
        +
        dbURL);
}
catch (Exception e)
{
        throw new UnavailableException("Error: " + e);
}

response.setContentType("text/html");

try
{
//Display all data from the database

//Convert to int and place in statement, check to see if statement
//will work!!!!

//SELECT * FROM table LIMIT 5,10;  # Retrieve rows 6-15
//Make fromRow and increment strings!!!!
sql = "select * FROM GRIDRCS LIMIT " + fromRow + "," +
increment;

stmt = dbConn.createStatement();
ResultSet rs = stmt.executeQuery(sql);

out.println("<HTML>");
                        out.println("<HEAD><TITLE>Grid
Resource Database</TITLE></HEAD>");
out.println("<BODY>");
out.println("<TABLE BORDER=\"1\" CELLPADDING=\"3\">");
out.println("<TR>");

out.println("<TH>IPName</TH>");
out.println("<TH>IPAddress</TH>");
out.println("<TH>Current IP Address</TH>");
out.println("<TH>Owner</TH>");
out.println("<TH>Time Zone</TH>");
```

```
out.println("<TH>Offered Hours 1</TH>");
out.println("<TH>Offered Hours 2</TH>");
out.println("<TH>Offered Hours 3</TH>");
out.println("<TH>CPU Architecture</TH>");
out.println("<TH>CPU Performance</TH>");
out.println("<TH>OS Type</TH>");
out.println("<TH>Memory</TH>");
out.println("<TH>Free Memory</TH>");
out.println("<TH>#CPUs</TH>");
out.println("<TH>Actual CPUs</TH>");
out.println("<TH>CPU Utilization</TH>");
out.println("<TH>Disk</TH>");
out.println("<TH>Available Disk</TH>");
out.println("<TH>Inter Network Device</TH>");
out.println("<TH>Inter Network Bandwidth</TH>");
out.println("<TH>Actual Inter Network Bandwidth</TH>");
out.println("<TH>Intra Network Device</TH>");
out.println("<TH>Intra Network Bandwidth</TH>");
out.println("<TH>Actual Intra Network Bandwidth</TH>");
out.println("<TH>Libraries</TH>");
out.println("<TH>Application</TH>");

out.println("</TR>");

        while (rs.next())
        {
                out.println("<TR>");

                        out.println("<TD>" +
                rs.getString("IPName") + "</TD>");
                out.println("<TD>" + rs.getString("IPAddress") +
                "</TD>");
                out.println("<TD>" +
                rs.getString("CurrentIPAddress") + "</TD>");
                out.println("<TD>" + rs.getString("Owner") +
                "</TD>");
                out.println("<TD>" + rs.getString("Phone") +
                "</TD>");
                out.println("<TD>" + rs.getString("Email") +
                "</TD>");
                out.println("<TD>" + rs.getInt("TimeZone") +
                "</TD>");
                out.println("<TD>" +
                rs.getString("OfferedHours1") + "</TD>");
                out.println("<TD>" +
                rs.getString("OfferedHours2") + "</TD>");
```

```
                out.println("<TD>" +
                rs.getString("OfferedHours3") + "</TD>");
                out.println("<TD>" +
                rs.getString("CPUArchitecture") + "</TD>");
                out.println("<TD>" + rs.getInt("CPUPerformance")
                + "</TD>");
                out.println("<TD>" + rs.getString("OSType") +
                "</TD>");
                out.println("<TD>" + rs.getInt("Memory") +
                "</TD>");
                out.println("<TD>" + rs.getInt("FreeMemory") +
                "</TD>");
                out.println("<TD>" + rs.getInt("NumberOfCPUs")
                + "</TD>");
                out.println("<TD>" +
                rs.getInt("ActualNumberOfCPUs") + "</TD>");
                out.println("<TD>" + rs.getInt("CPUUtilization") +
                "</TD>");
                out.println("<TD>" + rs.getDouble("Disk") +
                "</TD>");
                out.println("<TD>" + rs.getInt("AvailableDisk") +
                "</TD>");
                out.println("<TD>" +
                rs.getString("InterNetworkDevice") + "</TD>");
                out.println("<TD>" +
                rs.getInt("InterNetworkBandwidth") + "</TD>");
                out.println("<TD>" +
                rs.getInt("ActualInterNetworkBandwidth") +
                "</TD>");
                out.println("<TD>" +
                rs.getString("IntraNetworkDevice") + "</TD>");
                out.println("<TD>" +
                rs.getInt("IntraNetworkBandwidth") + "</TD>");
                out.println("<TD>" +
                rs.getInt("ActualIntraNetworkBandwidth") +
                "</TD>");
                out.println("<TD>" + rs.getString("Libraries") +
                "</TD>");
                out.println("<TD>" + rs.getString("Application") +
                "</TD>");

                out.println("</TR>");
        }

        out.println("</TABLE>");
        out.println("</BODY></HTML>");
```

```
                        rs.close();
                        stmt.close();
                    }
                    catch (SQLException e)
                    {
                        out.println("<H2>Database currently unavailable.</H2>");
                        e.printStackTrace();
                    }
                    out.close();


        out.println("Display was completed successfully!");

                        out.println("<br><br><a href=\"/metacomputer/servlet/Index\">");
                        out.println(" Back to Main Menu </a>");
                fromRow = null;
        toRow = null;
    }
    else{
            out.println("<P>");
            out.print("<form action=\"");
            out.print("Display\" ");
            out.println("method=POST>");
            out.println("From Row:  ");
            out.println("<input type=text size=10 name=fromrow>");
            out.println("  To Row:  ");
            out.println("<input type=text size=10 name=torow>");
            out.println("<br><br>");
            out.println("Display All?");
            out.println("<input type=checkbox size=2 name=box>");
            out.println("<br><br>");


            out.println("<input type=submit>");

            out.println("</form>");

            out.println("</body>");
            out.println("</html>");
        }
}

public void doPost(HttpServletRequest request,
            HttpServletResponse response)
    throws IOException, ServletException
```

```
    {
System.out.println("~/system_new/tomcat/webapps/metacomputer/WEB-
INF/classes/Display.java ... doPost()\n");
        doGet(request, response);
    }

}
```

Update.java-

```
/*
 * Name : Update.java
 * Date : 4/20/2003
 * Author : Jon Hagen & Eric Nelson
 *
 * Description:  This is the servlet used to update information into the
 * resource database.
 *
 */

import java.io.*;
import java.sql.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.lang.Integer;
import java.lang.Double;


public class Update extends HttpServlet {


    //ResourceBundle rb = ResourceBundle.getBundle("LocalStrings");

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {
System.out.println("~/system_new/tomcat/webapps/metacomputer/WEB-
INF/classes/Update.java ... doGet()\n");
        response.setContentType("text/html");
```

```
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<body>");
        out.println("<head>");

        String title = new String("Update the Resource Database");
        out.println("<title>" + title + "</title>");
        out.println("</head>");
        out.println("<body bgcolor=\"white\">");


           //Print out the title to the screen
        out.println("<h3>" + title + "</h3>");

        String firstName = request.getParameter("firstname");
        String lastName = request.getParameter("lastname");
        String checkBox = request.getParameter("checkbox");

           String sql = "UPDATE GRIDRCS SET ";
           /*        "IPName=" + ipName + ",IPAddress=" + ipAddress +
",CurrentIPAddress=" + curIp + "," +
                     "Owner=" + eOwner + ",Phone=" + ph + ",Email=" + mail +
",TimeZone=" + timeZone_i + "," +
                     "OfferedHours1=" + offerOne + ",OfferedHours2=" + offerTwo + "," +
                     "OfferedHours3=" + offerThree + ",CPUArchitecture=" + cpuArch + "," +
                     "CPUPerformance=" + cpuPerf_i + ",OSType=" + osType + "," +
                     "Memory=" + memory_i + ",FreeMemory=" + freeMemory_i +
",NumberOfCPUs=" + numCpu_i + "," +
                     "ActualNumberOfCPUs=" + actualCpu_i + ",CPUUtilization=" +
cpuUtil_i + "," +
                     "Disk=" + disk_d + ",AvailableDisk=" + availDisk_i +
",InternetworkDevice=" + interDevice + "," +
                     "InternetworkBandwidth=" + interBand_i +
",ActualInternetworkBandwidth=" + actualInter_i + "," +
                     "IntraNetworkDevice=" + intraDevice + ",IntraNetworkBandwidth=" +
intraBand_i + "," +
                     "ActualIntraNetworkBandwidth=" + actualIntra_i + ",Libraries=" + lib +
",Application=" + appl +
                  " WHERE id= " + rowId + ";"; */


           // Capture text field data to update
        String ipName = request.getParameter("ipname");
           if(ipName != null) sql += ("IPName=" + ipName);
        String ipAddress = request.getParameter("ipaddress");
           if(ipAddress != null) sql += (",IPAddress=" + ipAddress);
```

```
    String curIp = request.getParameter("curip");
    if(curIp != null) sql += (",CurrentIPAddress=" + curIp );
String eOwner = request.getParameter("eowner");
    if(eOwner != null) sql += (",Owner=" + eOwner);
String ph = request.getParameter("phone");
    if(ph != null) sql += (",Phone=" + ph);
String mail = request.getParameter("email");
    if(mail != null) sql += (",Email=" + mail);
String timeZone = request.getParameter("timezone");
    int timeZone_i = -1;
    if(timeZone != null) timeZone_i = Integer.parseInt(timeZone);
    if(timeZone != null) sql += (",TimeZone=" + timeZone_i);
String offerOne = request.getParameter("offerone");
    if(offerOne != null) sql += (",OfferedHours1=" + offerOne);
String offerTwo = request.getParameter("offertwo");
    if(offerTwo != null) sql += (",OfferedHours2=" + offerTwo);
String offerThree = request.getParameter("offerthree");
    if(offerThree != null) sql += (",OfferedHours3=" + offerThree);
String cpuArch = request.getParameter("cpuarch");
    if(cpuArch != null) sql += (",CPUArchitecture=" + cpuArch);
String cpuPerf = request.getParameter("cpuperf");
    int cpuPerf_i = -1;
    if(cpuPerf != null) cpuPerf_i = Integer.parseInt(cpuPerf);
    if(cpuPerf != null) sql += (",CPUPerformance=" + cpuPerf_i);
String osType = request.getParameter("ostype");
    if(osType != null) sql += (",OSType=" + osType);
String mem = request.getParameter("memory");
    int memory_i = -1;
    if(mem != null) memory_i = Integer.parseInt(mem);
    if(mem != null) sql += (",Memory=" + memory_i);
String freeMemory = request.getParameter("freememory");
    int freeMemory_i = -1;
    if(freeMemory != null) freeMemory_i = Integer.parseInt(freeMemory);
    if(freeMemory != null) sql += (",FreeMemory=" + freeMemory_i);
String numCpu = request.getParameter("numcpu");
    int numCpu_i = -1;
    if(numCpu != null) numCpu_i = Integer.parseInt(numCpu);
    if(numCpu != null) sql += (",NumberOfCPUs=" + numCpu_i);
String actualCpu = request.getParameter("actualcpu");
    int actualCpu_i = -1;
    if(actualCpu != null) actualCpu_i = Integer.parseInt(actualCpu);
    if(actualCpu != null) sql += (",ActualNumberOfCPUs=" + actualCpu_i);
String cpuUtil = request.getParameter("cpuutil");
    int cpuUtil_i = -1;
    if(cpuUtil != null) cpuUtil_i = Integer.parseInt(cpuUtil);
    if(cpuUtil != null) sql += (",CPUUtilization=" + cpuUtil_i);
```

```
String dsk = request.getParameter("disk");
   double disk_d = -1;
   if(dsk != null) disk_d = Double.parseDouble(dsk);
   if(dsk != null) sql += (",Disk=" + disk_d);
String availDisk = request.getParameter("availdisk");
   int availDisk_i = -1;
   if(availDisk != null) availDisk_i = Integer.parseInt(availDisk);
   if(availDisk != null) sql += (",AvailableDisk=" + availDisk_i);
String interDevice = request.getParameter("interdevice");
   if(interDevice != null) sql += (",InternetworkDevice=" + interDevice);
String interBand = request.getParameter("interband");
   int interBand_i = -1;
   if(interBand != null) interBand_i = Integer.parseInt(interBand);
   if(interBand != null) sql += (",InternetworkBandwidth=" + interBand_i );
String actualInter = request.getParameter("actualinter");
   int actualInter_i = -1;
   if(actualInter != null) actualInter_i = Integer.parseInt(actualInter);
   if(actualInter != null) sql += (",ActualInternetworkBandwidth=" + actualInter_i);
String intraDevice = request.getParameter("intradevice");
   if(intraDevice != null) sql += (",IntraNetworkDevice=" + intraDevice);
String intraBand = request.getParameter("intraband");
   int intraBand_i = -1;
   if(intraBand != null) intraBand_i = Integer.parseInt(intraBand);
   if(intraBand != null) sql += (",IntraNetworkBandwidth=" + intraBand_i);
String actualIntra = request.getParameter("actualintra");
   int actualIntra_i = -1;
   if(actualIntra != null) actualIntra_i = Integer.parseInt(actualIntra);
   if(actualIntra != null) sql += (",ActualIntraNetworkBandwidth=" + actualIntra_i);
String lib = request.getParameter("libraries");
   if(lib != null) sql += (",Libraries=" + lib);
String appl = request.getParameter("application");
   if(appl != null) sql += (",Application=" + appl);

   // Capture the id of the row to be updated
   String updateRow = request.getParameter("updaterow");
   int rowId = -1;
   if(updateRow != null) rowId = Integer.parseInt(updateRow);
   if(updateRow != null) sql += (" WHERE id=" + rowId + ";");


   String jdbcDriver = "com.mysql.jdbc.Driver";
   String dbURL ="jdbc:mysql:///grid";
   Connection dbConn = null;
   Statement stmt = null;
   try
   {
```

```
      Class.forName(jdbcDriver) .newInstance() ;
      dbConn = DriverManager.getConnection(dbURL, "jhagen", "jhagen1");
      stmt = dbConn.createStatement();
      boolean results = stmt.execute(sql);
      if(results) System.out.println("results = TRUE\n");
      else System.out.println("results = FALSE\n");
      stmt.close();
   }
   catch (ClassNotFoundException e)
   {
      System.out.println("JDBC driver not found: " + jdbcDriver);
      System.out.println(e.getMessage());
   }
   catch    (SQLException e)
   {
      System.out.println("Error retrieving data from: " + dbURL);
      System.out.println(e.getMessage());
   }
   catch (Exception e)
   {
      System.out.println("Error: " + e);
      System.out.println(e.getMessage());
   }
   finally
   {
      try
      {
            if (dbConn != null)
            {
               dbConn.close();
            }
      }
      catch (SQLException ignored) {}
   }

//If String variables received values, then insert completed
//****USED TO TEST****
if (ipName!= null || ipAddress != null) {
   out.println("Update was completed successfully!");

                  out.println("<br><br><a href=\"/metacomputer/servlet/Index\">");
                  out.println(" Back to Main Menu </a>");
        ipName = null;
   ipAddress = null;
}
else{
```

```
out.println("<P>");
out.print("<form action=\"");
out.print("Update\" ");
out.println("method=POST>");
out.println("Row to Update:  ");
out.println("<input type=text size=10 name=updaterow>");
out.println("<br>");
out.println("IP Name:  ");
out.println("<input type=text size=40 name=ipname>");
out.println("<br>");
out.println("IP Address:  ");
out.println("<input type=text size=40 name=ipaddress>");
out.println("<br>");
out.println("Current IP Address:  ");
out.println("<input type=text size=40 name=curip>");
out.println("<br>");
out.println("Owner:  ");
out.println("<input type=text size=40 name=eowner>");
out.println("<br>");
out.println("Phone:  ");
out.println("<input type=text size=40 name=phone>");
out.println("<br>");
out.println("Email:  ");
out.println("<input type=text size=40 name=email>");
out.println("<br>");
out.println("Time Zone:  ");
out.println("<input type=text size=40 name=timezone>");
out.println("<br>");
out.println("Offered Hours 1:  ");
out.println("<input type=text size=40 name=offerone>");
out.println("<br>");
out.println("Offered Hours 2:  ");
out.println("<input type=text size=40 name=offertwo>");
out.println("<br>");
out.println("Offered Hours 3:  ");
out.println("<input type=text size=40 name=offerthree>");
out.println("<br>");
out.println("CPU Architecture:  ");
out.println("<input type=text size=40 name=cpuarch>");
out.println("<br>");
out.println("CPU Performance:  ");
out.println("<input type=text size=40 name=cpuperf>");
out.println("<br>");
out.println("OS Type:  ");
out.println("<input type=text size=40 name=ostype>");
out.println("<br>");
```

```
out.println("Memory:  ");
out.println("<input type=text size=40 name=memory>");
out.println("<br>");
out.println("Free Memory:  ");
out.println("<input type=text size=40 name=freememory>");
out.println("<br>");
out.println("#CPU's:  ");
out.println("<input type=text size=40 name=numcpu>");
out.println("<br>");
out.println("Actual #CPU's:  ");
out.println("<input type=text size=40 name=actualcpu>");
out.println("<br>");
out.println("CPU Utilization:  ");
out.println("<input type=text size=40 name=cpuutil>");
out.println("<br>");
out.println("Disk:  ");
out.println("<input type=text size=40 name=disk>");
out.println("<br>");
out.println("Available Disk:  ");
out.println("<input type=text size=40 name=availdisk>");
out.println("<br>");
out.println("Inter Network Device:  ");
out.println("<input type=text size=40 name=interdevice>");
out.println("<br>");
out.println("Inter Network Bandwidth:  ");
out.println("<input type=text size=40 name=interband>");
out.println("<br>");
out.println("Actual Inter Network Bandwidth:  ");
out.println("<input type=text size=40 name=actualinter>");
out.println("<br>");
out.println("Intra Network Device:  ");
out.println("<input type=text size=40 name=intradevice>");
out.println("<br>");
out.println("Intra Network Bandwidth:  ");
out.println("<input type=text size=40 name=intraband>");
out.println("<br>");
out.println("Actual Intra Network Bandwidth:  ");
out.println("<input type=text size=40 name=actualintra>");
out.println("<br>");
out.println("Libraries:  ");
out.println("<input type=text size=40 name=libraries>");
out.println("<br>");
out.println("Application:  ");
out.println("<input type=text size=40 name=application>");
out.println("<br>");
```

```
                out.println("<input type=submit>");

                out.println("</form>");

                out.println("</body>");
                out.println("</html>");
            }
        }

        public void doPost(HttpServletRequest request,
                           HttpServletResponse response)
            throws IOException, ServletException
        {
System.out.println("~/system_new/tomcat/webapps/metacomputer/WEB-
INF/classes/Update.java ... doPost()\n");
            doGet(request, response);
        }

}
```