Labs

Lab 8.1: Accessing and Updating a Staff Table

What is the purpose?

In this lab, you will write a Java applet that helps view, insert, and update staff information stored in a database. See the sample output in Figure 8-1-3. The view button displays a record with a specified ID. The Staff table is created as follows:

```
create table Staff (
       id char(9) not null,
       lastName varchar(15),
       firstName varchar(15),
       mi char(1),
       address varchar(20),
       city varchar(20),
       state char(2),
       telephone char(10),
       email varchar(40),
       primary key (id)
);
```

Figure 8-1-1

What are the steps?

- **Task 1**

  Procedure:

  1. Create a Java applet class called Staff that helps view and update a database table called staff.
  2. Complete the following code or create your own code:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.*;
import java.sql.*;
import javax.swing.border.*;

public class Staff extends JApplet {
  boolean isStandalone = false;
  JPanel jpDisplay = new JPanel();
  JLabel jlblStatus = new JLabel();
  JPanel jpButtons = new JPanel();
  BorderLayout borderLayout1 = new BorderLayout();
  JPanel jpStaff = new JPanel();
  JButton jbtView = new JButton();
  JButton jbtInsert = new JButton();
  JButton jbtUpdate = new JButton();
  JButton jbtClear = new JButton();
  JPanel jPanel1 = new JPanel();
  JPanel jPanel2 = new JPanel();
  JPanel jPanel3 = new JPanel();
```

```
  JPanel jPanel4 = new JPanel();
  JPanel jPanel5 = new JPanel();
  JLabel jLabel2 = new JLabel();
  JLabel jLabel3 = new JLabel();
  JLabel jLabel4 = new JLabel();
  FlowLayout flowLayout1 = new FlowLayout();
  FlowLayout flowLayout2 = new FlowLayout();
  FlowLayout flowLayout3 = new FlowLayout();
  JTextField jtfID = new JTextField();
  JTextField jtfLastName = new JTextField();
  JLabel jLabel5 = new JLabel();
  JTextField jtfFirstName = new JTextField();
  JLabel jLabel6 = new JLabel();
  JTextField jtfmi = new JTextField();
  JTextField jtfAddress = new JTextField();
  JLabel jLabel7 = new JLabel();
  JTextField jtfCity = new JTextField();
  JLabel jLabel8 = new JLabel();
  JTextField jtfState = new JTextField();
  FlowLayout flowLayout4 = new FlowLayout();
  JLabel jLabel9 = new JLabel();
  JTextField jtfTelephone = new JTextField();
  FlowLayout flowLayout5 = new FlowLayout();
  JLabel jLabel10 = new JLabel();
  JTextField jtfEmail = new JTextField();
  FlowLayout flowLayout6 = new FlowLayout();

  // The Statement for processing queries
  Statement stmt;
  TitledBorder titledBorder1;
  GridLayout gridLayout1 = new GridLayout();

  /**Initialize the applet*/
  public void init() {
    try {
      jbInit();
    }
    catch (Exception e) {
      e.printStackTrace();
    }
  }

  private void jbInit() throws Exception {
    titledBorder1 = new TitledBorder("");
    this.setSize(450,350);
    jpDisplay.setLayout(borderLayout1);
    jpStaff.setLayout(gridLayout1);
    jbtView.setText("View");
    jbtView.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(ActionEvent e) {
        jbtView_actionPerformed(e);
      }
    });
    jbtInsert.setText("Insert");
    jbtInsert.addActionListener(new java.awt.event.ActionListener(){
      public void actionPerformed(ActionEvent e) {
        jbtInsert_actionPerformed(e);
```

```
      }
    });
    jbtUpdate.setText("Update");
    jbtUpdate.addActionListener(new java.awt.event.ActionListener(){
      public void actionPerformed(ActionEvent e) {
        jbtUpdate_actionPerformed(e);
      }
    });
    jbtClear.setText("Clear");
    jbtClear.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(ActionEvent e) {
        jbtClear_actionPerformed(e);
      }
    });
    jPanel5.setLayout(flowLayout5);
    jPanel4.setLayout(flowLayout4);
    jPanel3.setLayout(flowLayout3);
    jPanel2.setLayout(flowLayout2);
    jPanel1.setLayout(flowLayout1);
    jLabel2.setText("ID");
    jLabel3.setText("Last Name");
    jLabel4.setText("Address");
    flowLayout1.setAlignment(0);
    flowLayout2.setAlignment(0);
    flowLayout3.setAlignment(0);
    jtfID.setColumns(11);
    jtfID.setBackground(Color.yellow);
    jtfLastName.setColumns(10);
    jLabel5.setText("First Name");
    jtfFirstName.setColumns(10);
    jLabel6.setText("mi");
    jtfmi.setColumns(2);
    jtfAddress.setColumns(15);
    jLabel7.setText("City");
    jtfCity.setColumns(15);
    jLabel8.setText("State");
    jtfState.setColumns(2);
    flowLayout4.setAlignment(0);
    jLabel9.setText("Telephone");
    jtfTelephone.setColumns(12);
    flowLayout5.setAlignment(0);
    jLabel10.setText("Email");
    jtfEmail.setColumns(20);
    flowLayout6.setAlignment(0);
    jlblStatus.setBackground(Color.pink);
    jlblStatus.setText("Connecting ...");
    jpStaff.setBorder(titledBorder1);
    titledBorder1.setTitle("Staff Information");
    gridLayout1.setColumns(1);
    gridLayout1.setRows(5);
    this.getContentPane().add(jpDisplay, BorderLayout.CENTER);
    jpDisplay.add(jpButtons, BorderLayout.SOUTH);
    jpButtons.add(jbtView, null);
    jpButtons.add(jbtInsert, null);
    jpButtons.add(jbtUpdate, null);
    jpButtons.add(jbtClear, null);
    jpDisplay.add(jpStaff, BorderLayout.CENTER);
```

```
    jpStaff.add(jPanel1, null);
    jPanel1.add(jLabel2, null);
    jPanel1.add(jtfID, null);
    jpStaff.add(jPanel2, null);
    jPanel2.add(jLabel3, null);
    jPanel2.add(jtfLastName, null);
    jPanel2.add(jLabel5, null);
    jPanel2.add(jtfFirstName, null);
    jPanel2.add(jLabel6, null);
    jPanel2.add(jtfmi, null);
    jpStaff.add(jPanel3, null);
    jPanel3.add(jLabel4, null);
    jPanel3.add(jtfAddress, null);
    jpStaff.add(jPanel4, null);
    jPanel4.add(jLabel7, null);
    jPanel4.add(jtfCity, null);
    jPanel4.add(jLabel8, null);
    jPanel4.add(jtfState, null);
    jpStaff.add(jPanel5, null);
    jPanel5.add(jLabel9, null);
    jPanel5.add(jtfTelephone, null);
    jPanel5.add(jLabel10, null);
    jPanel5.add(jtfEmail, null);
    this.getContentPane().add(jlblStatus, BorderLayout.SOUTH);

    // Connect to the database
    initializeDB();
  }

  private void initializeDB() {
    try {
      // Load the driver


      // Connect to the ODBC - Access database


      // Create a statement

    }
    catch (Exception ex) {
      jlblStatus.setText("Connection failed: " + ex);
    }
  }

  void jbtInsert_actionPerformed(ActionEvent e) {
    insert();
  }

  void jbtView_actionPerformed(ActionEvent e) {
    view();
  }

  void jbtUpdate_actionPerformed(ActionEvent e) {
    update();
  }
```

```
  void jbtClear_actionPerformed(ActionEvent e) {
    clear();
  }

  /**View record by ID*/
  private void view() {
    // Build a SQL SELECT statement
    String query = "SELECT * FROM Staff WHERE ID = "
      + "'" + jtfID.getText().trim() + "'";

    try {
      // Execute query

      loadToTextField(rs);
      jlblStatus.setText("Querying: " + query);
    }
    catch(SQLException ex) {
      jlblStatus.setText("Select failed: " + ex);
    }
  }

  /**Load the record into text fields*/
  private void loadToTextField(ResultSet rs) throws SQLException {
    if (rs.next()) {
      jtfLastName.setText(rs.getString(2));
      jtfFirstName.setText(rs.getString(3));
      jtfmi.setText(rs.getString(4));
      jtfAddress.setText(rs.getString(5));
      jtfCity.setText(rs.getString(6));
      jtfState.setText(rs.getString(7));
      jtfTelephone.setText(rs.getString(8));
      jtfEmail.setText(rs.getString(9));
      jlblStatus.setText("Record found");
    }
    else
      jlblStatus.setText("Record not found");
  }

  /**Insert a new record*/
  private void insert() {
    // Build a SQL INSERT statement
    String insertStmt =
      "INSERT INTO Staff(ID, LastName, FirstName, mi, Address, " +
      " City, State, Telephone, Email) VALUES('" +
      jtfID.getText().trim() + "','" +
      jtfLastName.getText().trim() + "','" +
      jtfFirstName.getText().trim() + "','" +
      jtfmi.getText().trim() + "','" +
      jtfAddress.getText().trim() + "','" +
      jtfCity.getText().trim() + "','" +
      jtfState.getText().trim() + "','" +
      jtfTelephone.getText().trim() + "','" +
      jtfEmail.getText().trim() + "');";

    try {
      //System.out.println(insertStmt);
```

```
      jlblStatus.setText("record inserted");
    }
    catch (SQLException ex) {
      jlblStatus.setText("Insertion failed: " + ex);
    }
  }

  /**Update a record*/
  private void update() {
    // Build a SQL UPDATE statement
    String updateStmt = "UPDATE Staff " +
      "SET LastName = '" + jtfLastName.getText().trim() + "'," +
      "FirstName = '" + jtfFirstName.getText().trim() + "'," +
      "mi = '" + jtfmi.getText().trim() + "'," +
      "Address = '" + jtfAddress.getText().trim() + "'," +
      "City = '" + jtfCity.getText().trim() + "'," +
      "State = '" + jtfState.getText().trim() + "'," +
      "Telephone = '" + jtfTelephone.getText().trim() + "' " +
      "Email = '" + jtfEmail.getText().trim() + "' " +
      "WHERE ID = '" + jtfID.getText().trim() + "'";

    try {
      //System.out.println(updateStmt);

      jlblStatus.setText("Record updated");
    }
    catch(SQLException ex) {
      jlblStatus.setText("Update failed: " + updateStmt);
    }
  }

  /**Clear text fields*/
  private void clear() {
    jtfID.setText(null);
    jtfLastName.setText(null);
    jtfFirstName.setText(null);
    jtfmi.setText(null);
    jtfAddress.setText(null);
    jtfCity.setText(null);
    jtfState.setText(null);
    jtfTelephone.setText(null);
    jtfEmail.setText(null);
  }

  /**Get Applet information*/
  public String getAppletInfo() {
    return "Applet Information";
  }
  /**Get parameter info*/
  public String[][] getParameterInfo() {
    return null;
  }
  /**Main method*/
  public static void main(String[] args) {
    Staff applet = new Staff();
    applet.isStandalone = true;
    JFrame frame = new JFrame();
```

```
    //EXIT_ON_CLOSE == 3
    frame.setDefaultCloseOperation(3);
    frame.setTitle("Accessing Database Staff");
    frame.getContentPane().add(applet, BorderLayout.CENTER);
    applet.init();
    applet.start();
    frame.setSize(400,400);
    Dimension d = Toolkit.getDefaultToolkit().getScreenSize();
  frame.setLocation((d.width - frame.getSize().width) / 2, (d.height -
  frame.getSize().height) / 2);
    frame.setVisible(true);
  }
}
```

Figure 8-1-2

Here is a sample of output:

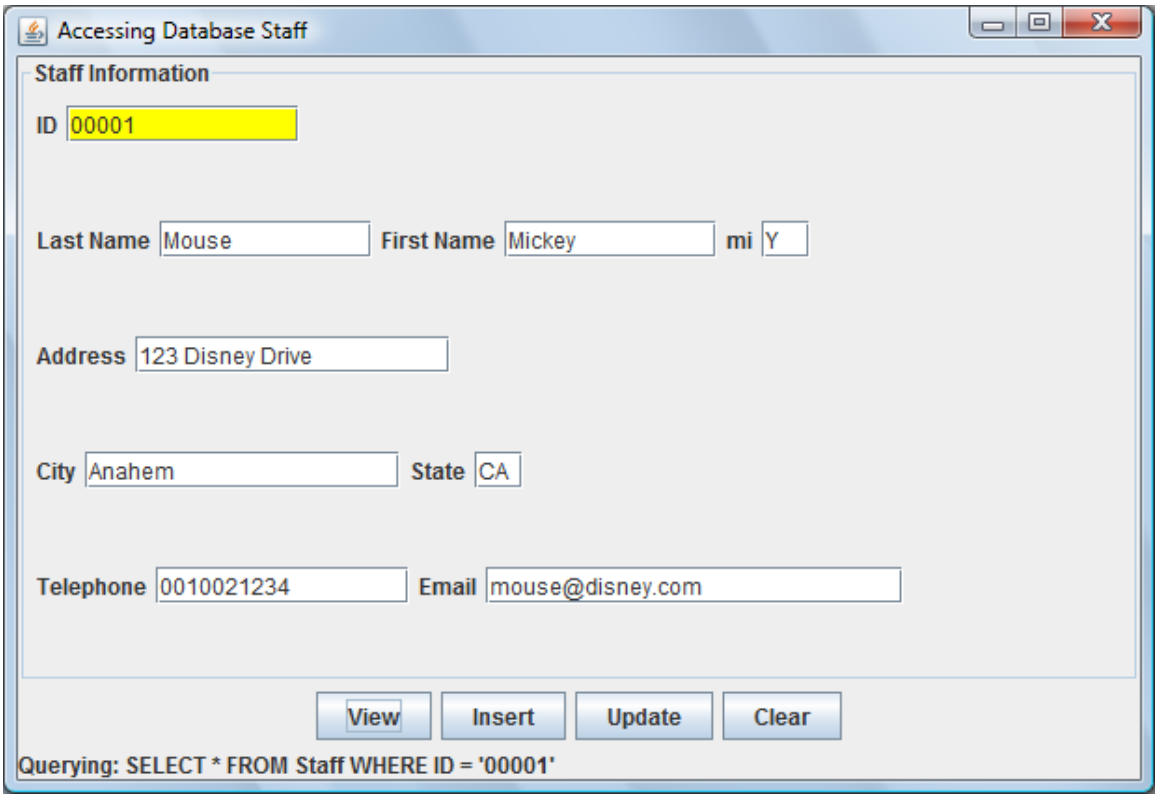

Figure 8-1-3: View

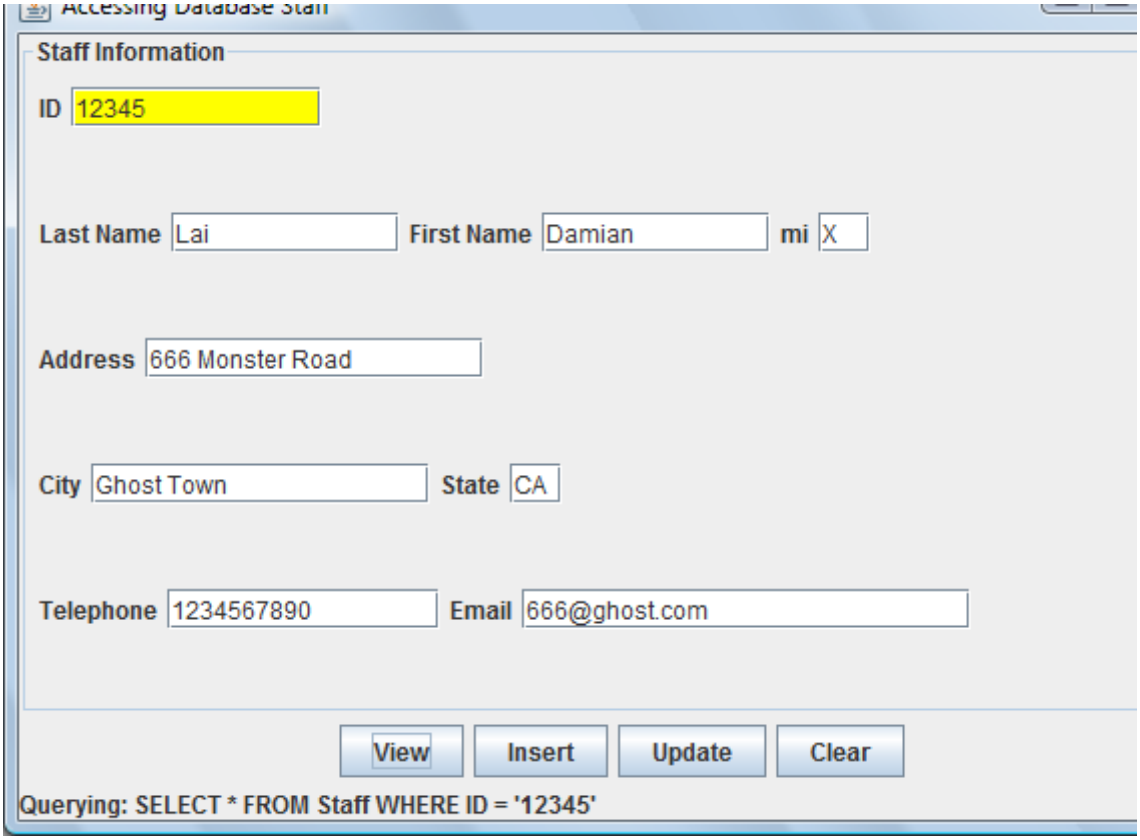

Figure 8-1-4: Update

3. Submit your Java code and sample output to your instructor.

Did it work?

Were you able to use the interface to:

- Show JDBC connection on the status bar?
- Display successfully executed SQL statements?
- Add a new record to the Staff table?
- Retrieve an existing record from Staff table by entering an ID?
- Update an existing record from Staff table?
- Clear all data entry?

Lab 8.2: Connection Dialog

What is the purpose?

In this lab, you will develop a JavaBeans component named DBConnectionPanel that enables the user to select a JDBC driver of either Microsoft Access or another database—in this example, Microsoft SQL Server—and to enter a user name and a password. When the OK button is clicked, a Connection object for the database is stored in the connection property. You can then use the getConnection() method to return the connection.

Note: You may have to install and configure a new JDBC driver. If you cannot find your preferred database driver, please check with your instructor.

What are the steps?

- **Task 1**

  Procedure

  1. Create a Java applet class called Connection that can connect to a database by two different JDBC drivers.
  2. Complete the following code or create your own code:

```
import javax.swing.*;
import javax.swing.border.*;
import java.sql.*;
import java.awt.*;
import java.awt.event.*;

public class Conection extends JApplet {
  // Connection to the database
  Connection connection;

  // Statement to execute SQL commands
  Statement statement;

  public static void main(String[] args) {
    DBConnectionPanel dbConnectPane = new DBConnectionPanel();
    JFrame frame = new JFrame("Connection Dialog");
    frame.setSize(400, 200);
    frame.getContentPane().add(dbConnectPane, BorderLayout.CENTER);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);
  }
}

class DBConnectionPanel extends JPanel {
  private Connection connection;
  private JLabel jlblConnectionStatus = new JLabel("No connection");

  private JButton jbtConnect = new JButton("Connect to DB");
  private JComboBox jcboDriver = new JComboBox(new String[] {
    "sun.jdbc.odbc.JdbcOdbcDriver",
    "com.microsoft.jdbc.sqlserver.SQLServerDriver"});
  private JComboBox jcboURL = new JComboBox(new String[] {
    "jdbc:odbc:Staff",
    "jdbc:microsoft:sqlserver://local:1433"});

  private JTextField jtfUsername = new JTextField();
  private JPasswordField jpfPassword = new JPasswordField();

  /** Creates new form DBConnectionPanel */
  public DBConnectionPanel() {
    jcboDriver.setEditable(true);
    jcboURL.setEditable(true);
```

```
    JPanel jPanel1 = new JPanel();
    jPanel1.setLayout(new BorderLayout());
    jPanel1.add(jlblConnectionStatus, BorderLayout.WEST);
    jPanel1.add(jbtConnect, BorderLayout.EAST);
    jbtConnect.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent evt) {
        connectDB();
      }
    });

    JPanel jPanel3 = new JPanel();
    jPanel3.setLayout(new GridLayout(4, 0));
    jPanel3.add(new JLabel("JDBC Driver"));
    jPanel3.add(new JLabel("Database URL"));
    jPanel3.add(new JLabel("Username"));
    jPanel3.add(new JLabel("Password"));

    JPanel jPanel4 = new JPanel();
    jPanel4.setLayout(new GridLayout(4, 0));
    jPanel4.add(jcboDriver);
    jPanel4.add(jcboURL);
    jPanel4.add(jtfUsername);
    jPanel4.add(jpfPassword);

    JPanel jPanel2 = new JPanel();
    jPanel2.setLayout(new BorderLayout());
    jPanel2.setBorder(new TitledBorder("Enter database information"));
    jPanel2.add(jPanel3, BorderLayout.WEST);
    jPanel2.add(jPanel4, BorderLayout.CENTER);

    this.setLayout(new BorderLayout());
    add(jPanel1, BorderLayout.SOUTH);
    add(jPanel2, BorderLayout.CENTER);
  }

  private void connectDB() {
    // Get database information from the user input
    String driver = (String)jcboDriver.getSelectedItem();
    String url = (String)jcboURL.getSelectedItem();
    String username = jtfUsername.getText().trim();
    String password = new String(jpfPassword.getPassword());

    // Connection to the database
    try {

      jlblConnectionStatus.setText("Connected to " + url);
    }
    catch (java.lang.Exception ex) {
      jlblConnectionStatus.setText("Failed connected to " + url);
      ex.printStackTrace();
    }
  }

  /** Return connection */
  public Connection getConnection() {
    return connection;
  }
```

```
}
```

Figure 8-2-1

Here is a sample of the output:

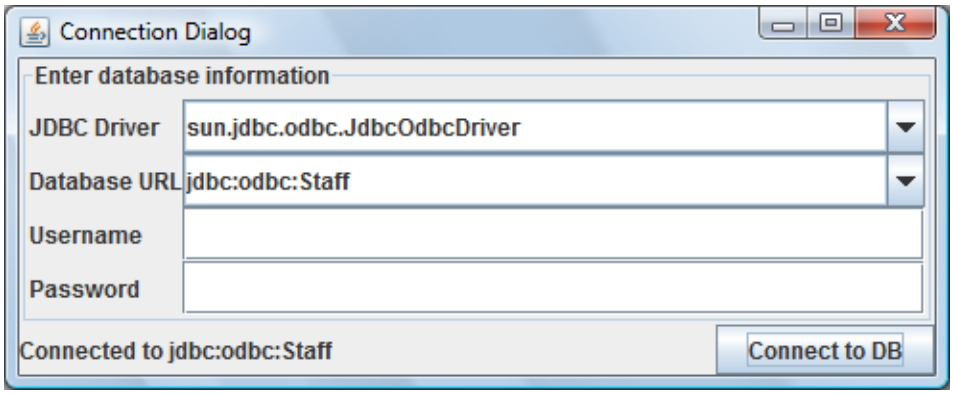

Figure 8-2-2: Sample Output 1

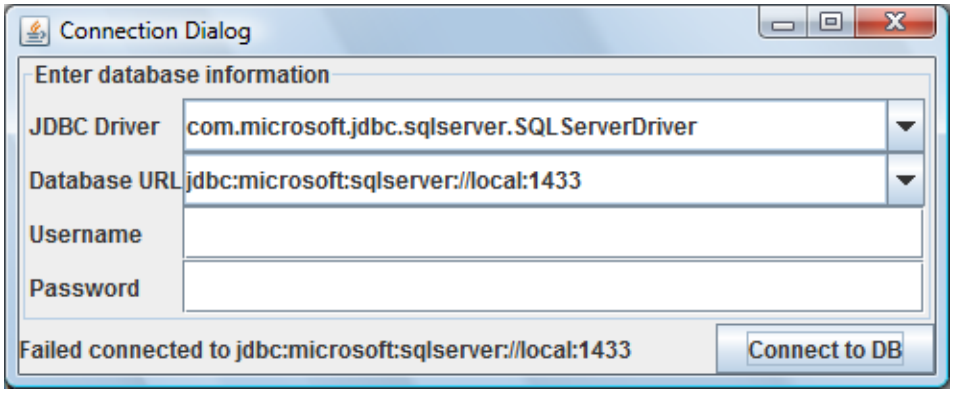

Figure 8-2-3: Sample Output 2

3. Submit your Java code and sample output to your instructor.

Did it work?

Were you able to:

- Show Access database connection exists if JDBC-ODBC driver is installed in the system?
- Show SQL Server database connection exists if JDBC-SQLDriver is installed in the system?