```java
import java.io.IOException;
import java.io.InputStream;
import java.sql.*;
import java.util.ArrayList;
import java.util.Properties;

public class DatabaseTools {
    public static Connection getConnected() {
        Connection conn = null;

        try {
            //Load MySQL database driver
            Class.forName("com.mysql.jdbc.Driver");

            // Connection string
            String dbURL = getConnectionString();

            // Credentials
            String username = "root";
            String password = "mysql";

            // Set connection to database to Connection object
            conn = DriverManager.getConnection(dbURL, username, password);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return conn;
    }

    public static void insert(User user) {
        // Connection to database;
        Connection myConn = DatabaseTools.getConnected();

        PreparedStatement ps = null;

        // Insert query
        String queryInsert = "INSERT INTO userdata (firstname, lastname, email) VALUES (?, ?, ?);";

        try {
            // Set PreparedStatement object to instance with query
            ps = myConn.prepareStatement(queryInsert);

            // Set values for parameters
            ps.setString(1, user.getFirstName());
            ps.setString(2, user.getLastName());
            ps.setString(3, user.getEmail());

            // Execute insert on database
            ps.execute();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DatabaseTools.closePreparedStatement(ps);
            DatabaseTools.closeConnection(myConn);
        }
    }

    public static ArrayList<User> selectAllUsers() {
        // Connection to database
        Connection conn = DatabaseTools.getConnected();

        PreparedStatement ps = null;
        ResultSet rs = null;
        ArrayList<User> myUserList = new ArrayList<>();

        // Select statement
        String selectQuery = "SELECT userid, firstname, lastname, email FROM userdata;";

        try {
            // Create prepared statement object
            ps = conn.prepareStatement(selectQuery);

            // Execute query and return result set
            rs = ps.executeQuery();

            // Loop through all rows in result set
            while (rs.next()) {
                // Create user object to add to list.
                User user = new User();

                // Add values for row of result set to properties of User object
                user.setId(rs.getInt("userid"));
                user.setFirstName(rs.getString("firstname"));
                user.setLastName(rs.getString("lastname"));
                user.setEmail(rs.getString("email"));

                // Add user to list
                myUserList.add(user);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DatabaseTools.closePreparedStatement(ps);
            DatabaseTools.closeResultSet(rs);
            DatabaseTools.closeConnection(conn);
        }

        return myUserList;
    }

    public static void update(User user) {
        // Connection to database;
        Connection myConn = DatabaseTools.getConnected();

        PreparedStatement ps = null;

        // Update query
        String queryUpdate = "UPDATE userdata SET firstname = ?, lastname = ?, email = ? WHERE firstname = ? AND lastname = ?;";

        try {
            // Set PreparedStatement object to instance with query
            ps = myConn.prepareStatement(queryUpdate);

            // Set values for parameters
            ps.setString(1, user.getFirstName());
            ps.setString(2, user.getLastName());
            ps.setString(3, user.getEmail());
            ps.setString(4, user.getFirstName());
            ps.setString(5, user.getLastName());

            // Execute update on database
            ps.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DatabaseTools.closePreparedStatement(ps);
            DatabaseTools.closeConnection(myConn);
        }
    }

    public static void delete(String firstName, String lastName) {
        // Connection to database;
        Connection myConn = DatabaseTools.getConnected();

        PreparedStatement ps = null;

        // Update query
        String queryUpdate = "DELETE FROM userdata WHERE firstname = ? AND lastname = ?;";

        try {
            // Set PreparedStatement object to instance with query
            ps = myConn.prepareStatement(queryUpdate);

            // Set values for parameters
            ps.setString(1, firstName);
            ps.setString(2, lastName);

            // Execute update on database
            ps.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            DatabaseTools.closePreparedStatement(ps);
            DatabaseTools.closeConnection(myConn);
        }
    }

    public static void closePreparedStatement(Statement ps) {
        try {
            if (ps != null) {
                ps.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public static void closeConnection(Connection conn) {
        try {
            if (conn != null) {
                conn.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public static void closeResultSet(ResultSet rs) {
        try {
            if (rs != null) {
                rs.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public static String getConnectionString() {
        Properties properties = new Properties();
        InputStream input = Thread.currentThread().getContextClassLoader().getResourceAsStream("settings.properties");
        try {
            properties.load(input);
        } catch (IOException e) {
            e.printStackTrace();
        }

        return properties.getProperty("dbURL").
toString();
    }
}

```