```java
package team.FixIt;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.Collection;
import java.util.LinkedList;
import java.util.List;


public class mysql
{
    Connection con;
    Statement st;

    public mysql()
    {

    }

    public boolean StartConnection(String username, String password)
    {
        boolean databaseAccessed = false;
        try
        {
            Connection con = DriverManager.getConnection(common.all, username,
                password);
            Statement st = con.createStatement();
            this.st = st;
            this.con = con;
            System.out.println("Starting Connection");
        }
        catch (SQLException ex)
        {
            System.out.println("SQL cannot be accessed, try again");
        }
        if(this.con != null && this.st != null)
        {
            databaseAccessed = true;
        }
        System.out.println("St is = " + this.st);
        System.out.println("Con is = " + this.con);
        return databaseAccessed;
    }

    public boolean CheckConnection()
    {
        boolean connectionEstablished = false;
        if(this.st != null)
        {
            connectionEstablished = true;
            System.out.println("This connection can be accessed");
        }
        else
        {
            connectionEstablished = false;
            System.out.println("This connection cannot be accessed");
        }
        return connectionEstablished;
    }

    private void CreateTable(String table) throws SQLException
    {
        if(CheckConnection())
        {
            String sql = "CREATE TABLE " + table +
                         "(id int," +
                         " account varchar(30)," +
                         " username varchar(30)," +
                         " password varchar(30)," +
```

```
73  ⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶" primary key (id));";
74  ⟶⟶⟶⟶⟶⟶⟶⟶this.st.executeUpdate(sql);
75  ⟶⟶⟶⟶⟶⟶⟶⟶System.out.println("Created table in given database...");
76  ⟶⟶⟶⟶}
77  ⟶⟶⟶⟶else
78  ⟶⟶⟶⟶{
79  ⟶⟶⟶⟶⟶⟶⟶⟶System.out.println("Error - Can't create table, connection fail!");
80  ⟶⟶⟶⟶}⟶⟶
81  ⟶⟶}
82  ⟶⟶
83  ⟶⟶private void DeleteTable(String table) throws SQLException
84  ⟶⟶{
85  ⟶⟶⟶⟶if(CheckConnection())
86  ⟶⟶⟶⟶{
87  ⟶⟶⟶⟶⟶⟶⟶⟶String sql = "drop table " + table;
88  ⟶⟶⟶⟶⟶⟶⟶⟶this.st.executeUpdate(sql);
89  ⟶⟶⟶⟶⟶⟶⟶⟶System.out.println("Table deleted in given database...");
90  ⟶⟶⟶⟶}
91  ⟶⟶⟶⟶else
92  ⟶⟶⟶⟶{
93  ⟶⟶⟶⟶⟶⟶⟶⟶System.out.println("Error - Can't delete table, connection fail!");
94  ⟶⟶⟶⟶}
95  ⟶⟶}
96  ⟶⟶
97  ⟶⟶private boolean checkIfIDExists(String table, int ID) throws SQLException
98  ⟶⟶{
99  ⟶⟶⟶⟶String strSelect = common.SELECT_VALUES + "from " + table + " where id = " +
        ID;
100 ⟶⟶⟶⟶⟶⟶⟶⟶System.out.println("The SQL query is: " + strSelect); // Echo for debugging
101 ⟶⟶⟶⟶⟶⟶⟶⟶ResultSet rset = this.st.executeQuery(strSelect);
102 ⟶⟶⟶⟶⟶⟶⟶⟶int idLocal = 0;
103 ⟶⟶⟶⟶⟶⟶⟶⟶while(rset.next())
104 ⟶⟶⟶⟶⟶⟶⟶⟶{
105 ⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶String idString = rset.getString("id");
106 ⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶idLocal = Integer.parseInt(idString);
107 ⟶⟶⟶⟶⟶⟶⟶⟶}
108 ⟶⟶⟶⟶⟶⟶⟶⟶if(idLocal == ID)
109 ⟶⟶⟶⟶⟶⟶⟶⟶{
110 ⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶return true;
111 ⟶⟶⟶⟶⟶⟶⟶⟶}
112 ⟶⟶⟶⟶⟶⟶⟶⟶else
113 ⟶⟶⟶⟶⟶⟶⟶⟶{
114 ⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶return false;
115 ⟶⟶⟶⟶⟶⟶⟶⟶}
116 ⟶⟶}
117 ⟶⟶
118 ⟶⟶public boolean checkIfAccountExists(String table, String Account) throws
        SQLException
119 ⟶⟶{
120 ⟶⟶⟶⟶String AccountName = "'" + Account + "'";
121 ⟶⟶⟶⟶String strSelect = common.SELECT_VALUES + "from " + table + " where account
        = " + AccountName;
122 ⟶⟶⟶⟶⟶⟶⟶⟶System.out.println("The SQL query is: " + strSelect); // Echo for debugging
123 ⟶⟶⟶⟶⟶⟶⟶⟶ResultSet rset = this.st.executeQuery(strSelect);
124 ⟶⟶⟶⟶⟶⟶⟶⟶String accountString = "";
125 ⟶⟶⟶⟶⟶⟶⟶⟶while(rset.next())
126 ⟶⟶⟶⟶⟶⟶⟶⟶{
127 ⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶accountString = rset.getString("account");
128 ⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶//System.out.println("Printing record Account name = " + accountString);
129 ⟶⟶⟶⟶⟶⟶⟶⟶}
130 ⟶⟶⟶⟶⟶⟶⟶⟶System.out.println(accountString);
131 ⟶⟶⟶⟶if(accountString.equals(Account))
132 ⟶⟶⟶⟶⟶⟶⟶⟶{
133 ⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶System.out.println("True");
134 ⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶return true;
135 ⟶⟶⟶⟶⟶⟶⟶⟶}
136 ⟶⟶⟶⟶⟶⟶⟶⟶else
137 ⟶⟶⟶⟶⟶⟶⟶⟶{
138 ⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶⟶return false;
139 ⟶⟶⟶⟶⟶⟶⟶⟶}
140 ⟶⟶}
141 ⟶⟶
142 ⟶⟶public String getUsername(String table, String Account) throws SQLException
```

```
143     {
144         String Input = "'" + Account + "'";
145         String strSelect = common.SELECT_VALUES + "from " + table + " where account
                = " + Input;
146         System.out.println("The SQL query is: " + strSelect);  // Echo for debugging
147         ResultSet rset = this.st.executeQuery(strSelect);
148         int idLocal = 0;
149         String username = "";
150         while(rset.next())
151         {
152             username = rset.getString("username");
153         }
154         return username;
155     }
156
157     public String getPassword(String table, String Account) throws SQLException
158     {
159         String Input = "'" + Account + "'";
160         String strSelect = common.SELECT_VALUES + "from " + table + " where account
                = " + Input;
161         System.out.println("The SQL query is: " + strSelect);  // Echo for debugging
162         ResultSet rset = this.st.executeQuery(strSelect);
163         int idLocal = 0;
164         String password = "";
165         while(rset.next())
166         {
167             password = rset.getString("password");
168         }
169         return password;
170     }
171
172     public void CreateNewLogin(String table, String account, String username, String
            password) throws SQLException
173     {
174         if(CheckConnection())
175         {
176             int ID = 0;
177             boolean idTrue = true;
178             int idLoop = 101;
179             boolean exit = false;
180             while(idLoop <= common.MAX_ID_NUMBER && exit == false)
181             {
182                 idTrue = checkIfIDExists(table, idLoop);
183                 if(!idTrue)
184                 {
185                     ID = idLoop;
186                     exit = true;
187                 }
188                 idLoop++;
189             }
190             if(ID != 0)
191             {
192                 String id = String.valueOf(ID);
193                 System.out.println("Available ID" + id);
194                 String sql = common.INSERT_VALUES + table + " values ('" + id +
                        "','" + account + "','" + username + "','" + password + "');";
195                 System.out.println(sql);
196                 this.st.executeUpdate(sql);
197             }
198         }
199         else
200         {
201             System.out.println("Error - Can't make a new login, connection fail!");
202         }
203     }
204
205     public boolean DeleteLogin(String table, String account) throws SQLException
206     {
207         boolean deleted = false;
208         if(CheckConnection())
209         {
210             String Input = "'" + account + "'";
211             String sql = common.REMOVE_VALUES + table + " where account = " + Input;
```

```
212 		System.out.println(sql);
213 		this.st.executeUpdate(sql);
214 		if(!checkIfAccountExists(table, account))
215 		{
216 			deleted = true;
217 		};
218 	}
219 	else
220 	{
221 		System.out.println("Error - Can't delete login, connection fail!");
222 	}
223 	return deleted;
224 }
225 
226 
227 public LinkedList<String> PrintDetails(String table) throws SQLException
228 {
229 	List<String> TypeList = new LinkedList<String>();
230 	if(CheckConnection())
231 	{
232 		String strSelect = common.SELECT_VALUES + "from " + table;
233 		System.out.println("The SQL query is: " + strSelect); // Echo for
		debugging
234 		ResultSet rset = this.st.executeQuery(strSelect);
235 		while(rset.next())
236 		{
237 			String typeString = rset.getString("account");
238 			TypeList.add(typeString);
239 		}
240 	}
241 	return (LinkedList<String>) TypeList;
242 }
243 public LinkedList<String> PrintDetailsOne(String table, String Type, String
    TypeName) throws SQLException
244 {
245 	List<String> TypeList = new LinkedList<String>();
246 	if(CheckConnection())
247 	{
248 		String strSelect = common.SELECT_VALUES + "from " + table + " where " +
		Type + " = " + TypeName;
249 		System.out.println("The SQL query is: " + strSelect); // Echo for
		debugging
250 		ResultSet rset = this.st.executeQuery(strSelect);
251 		while(rset.next())
252 		{
253 			String typeString = rset.getString(Type);
254 			TypeList.add(typeString);
255 		}
256 	}
257 	return (LinkedList<String>) TypeList;
258 }
259 
260 public void UpdateUsername(String table, String account, String username) throws
    SQLException
261 {
262 	if(CheckConnection())
263 	{
264 		PrintDetailsOne(table, account, username);
265 		String strUpdate = common.UPDATE_VALUES + table + " set username = "
266 				+ username + " where account = " + account;
267 		System.out.println("The SQL query is: " + strUpdate); // Echo for
		debugging
268 		this.st.executeUpdate(strUpdate);
269 	}
270 	else
271 	{
272 		System.out.println("Error - Can't update account username, connection
		fail!");
273 	}
274 }
275 
276 public void UpdatePassword(String table, String account, String password) throws
    SQLException
```

```
277     {
278         if(CheckConnection())
279         {
280             PrintDetailsOne(table, account, password);
281             String strUpdate = common.UPDATE_VALUES + table + " set username = "
282                                + password + " where account = " + account;
283             System.out.println("The SQL query is: " + strUpdate); // Echo for
                debugging
284             this.st.executeUpdate(strUpdate);
285         }
286         else
287         {
288             System.out.println("Error - Can't update account password, connection
                fail!");
289         }
290     }
291     
292 }
293 
```