package com.academy;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

public class QueryUtil {

public static boolean validateUserLogin(String loginId, String password) {

boolean success = false;

PreparedStatement pstmt = null;

ResultSet resultSet = null;

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("SELECT LOGIN\_ID FROM USERS WHERE LOGIN\_ID=? AND PASSWORD=?");

pstmt.setString(1, loginId);

pstmt.setString(2, password);

resultSet = pstmt.executeQuery();

while (resultSet != null && resultSet.next()) {

success = true;

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

resultSet.close();

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

return success;

}

public static List<String> getSubjects() {

PreparedStatement pstmt = null;

ResultSet resultSet = null;

List<String> subjects = new ArrayList<String>();

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("SELECT DISTINCT SUBJ\_NAME FROM SUBJECT ORDER BY SUBJ\_NAME");

resultSet = pstmt.executeQuery();

while (resultSet != null && resultSet.next()) {

subjects.add(resultSet.getString(1));

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

resultSet.close();

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

return subjects;

}

public static List<String> getTeachers() {

PreparedStatement pstmt = null;

ResultSet resultSet = null;

List<String> teachers = new ArrayList<String>();

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("SELECT DISTINCT TCHR\_NAME FROM TEACHER ORDER BY TCHR\_NAME");

resultSet = pstmt.executeQuery();

while (resultSet != null && resultSet.next()) {

teachers.add(resultSet.getString(1));

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

resultSet.close();

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

return teachers;

}

public static List<String> getClasses() {

PreparedStatement pstmt = null;

ResultSet resultSet = null;

List<String> classes = new ArrayList<String>();

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("SELECT DISTINCT CLASS\_NAME FROM CLASS ORDER BY CLASS\_NAME");

resultSet = pstmt.executeQuery();

while (resultSet != null && resultSet.next()) {

classes.add(resultSet.getString(1));

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

resultSet.close();

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

return classes;

}

public static void addSubject(String subjectName, String description) {

PreparedStatement pstmt = null;

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("INSERT INTO SUBJECT VALUES(?,?)");

pstmt.setString(1, subjectName);

pstmt.setString(2, description);

pstmt.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static void addTeacher(String teacherName, String description) {

PreparedStatement pstmt = null;

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("INSERT INTO TEACHER VALUES(?,?)");

pstmt.setString(1, teacherName);

pstmt.setString(2, description);

pstmt.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static void addClass(String className, String description) {

PreparedStatement pstmt = null;

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("INSERT INTO CLASS VALUES(?,?)");

pstmt.setString(1, className);

pstmt.setString(2, description);

pstmt.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static void addStudent(String studentId, String studentName, String className) {

PreparedStatement pstmt = null;

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("INSERT INTO STUDENT VALUES(?,?, ?)");

pstmt.setString(1, studentId);

pstmt.setString(2, studentId);

pstmt.setString(3, className);

pstmt.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static List<Student> getStudents() {

PreparedStatement pstmt = null;

ResultSet resultSet = null;

List<Student> students = new ArrayList<Student>();

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("SELECT STUDENT\_ID, NAME, CLASS\_NAME FROM STUDENT ORDER BY STUDENT\_ID");

resultSet = pstmt.executeQuery();

Student student = null;

while (resultSet != null && resultSet.next()) {

student = new Student(resultSet.getString(1), resultSet.getString(2), resultSet.getString(3));

students.add(student);

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

resultSet.close();

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

return students;

}

public static void assignClassSubject(String className, String subjName) {

PreparedStatement pstmt = null;

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("INSERT INTO CLASS\_SUBJECTS VALUES(?,?)");

pstmt.setString(1, className);

pstmt.setString(2, subjName);

pstmt.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static void assignClassTeacher(String className, String tchrName) {

PreparedStatement pstmt = null;

try {

Connection conn = DBConnectionUtil.getConnection();

pstmt = conn.prepareStatement("INSERT INTO CLASS\_TEACHERS VALUES(?,?)");

pstmt.setString(1, className);

pstmt.setString(2, tchrName);

pstmt.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

pstmt.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static List<ClassReport> getClassReport() {

PreparedStatement pstmt = null;

PreparedStatement pstmt1 = null;

PreparedStatement pstmt2 = null;

ResultSet resultSet = null;

ResultSet resultSet1 = null;

ResultSet resultSet2 = null;

List<ClassReport> reports = new ArrayList<ClassReport>();

try {

List<String> classNames = getClasses();

Connection conn = DBConnectionUtil.getConnection();

ClassReport report = null;

pstmt = conn.prepareStatement("SELECT DISTINCT SUBJ\_NAME FROM CLASS\_SUBJECTS where CLASS\_NAME=? order by SUBJ\_NAME ");

pstmt1 = conn.prepareStatement("SELECT DISTINCT TCHR\_NAME FROM CLASS\_TEACHERS where CLASS\_NAME=? order by TCHR\_NAME ");

pstmt2 = conn.prepareStatement("SELECT DISTINCT STUDENT\_ID FROM STUDENT where CLASS\_NAME=? order by STUDENT\_ID ");

for (String className :classNames ) {

report = new ClassReport();

report.setClassName(className);

pstmt.setString(1, className);

resultSet = pstmt.executeQuery();

List<String> subjects = new ArrayList<>();

while (resultSet != null && resultSet.next()) {

subjects.add(resultSet.getString(1));

}

report.setSubjects(subjects);

pstmt1.setString(1, className);

resultSet1 = pstmt1.executeQuery();

List<String> teachers = new ArrayList<>();

while (resultSet1 != null && resultSet1.next()) {

teachers.add(resultSet1.getString(1));

}

report.setTeachers(teachers);

pstmt2.setString(1, className);

resultSet2 = pstmt2.executeQuery();

List<String> students = new ArrayList<>();

while (resultSet2 != null && resultSet2.next()) {

students.add(resultSet2.getString(1));

}

report.setStudents(students);

reports.add(report);

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

resultSet.close();

pstmt.close();

resultSet1.close();

pstmt1.close();

resultSet2.close();

pstmt2.close();

} catch (Exception e) {

e.printStackTrace();

}

}

return reports;

}

}