**For ease of marking, I have highlighted all the SQL Syntax used in my assignment.**

**UI CLASS**

package ui1;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.Scanner;

import java.time.\*;

import java.sql.\*;

/\*\*

\* @author William Farrell.

\* @version 2.0, 1-05-2020

\*/

public class UI1 {

int StudentNumber = 0;

private final int CONTINUE = 0;

private final int TEST1 = 1;

private final int TEST2 = 2;

private final int TEST3 = 3;

private final int TEST4 = 4;

private final int TEST5 = 5;

private final int TEST6 = 6;

private final int TEST7 = 7;

private final int TEST8 = 8;

private final int TEST9 = 9;

private final int TEST10 = 10;

private final int TEST11 = 11;

private final int TEST12 = 12;

private final int TEST13 = 13;

private final int TEST14 = 14;

private final int EXIT = 15;

public UI1() {

try {

Class.forName("org.sqlite.JDBC");

Connection conn = DriverManager.getConnection("jdbc:sqlite:students.db");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT MAX(ID) FROM Students");

StudentNumber = ((Number) rs.getObject(1)).intValue();

System.out.print("Max Student ID is " + StudentNumber + ".\n");

stmt.close();

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

} catch (Exception ex) {

System.err.println("Unable to load MySQL driver.");

ex.printStackTrace();

}

}

public void addStudent() {

Scanner scan = new Scanner(System.in);

StudentNumber++;

int ID = StudentNumber;

System.out.print("Enter the Students First Name\n");

String first\_name = scan.next();

System.out.print("Enter the Students Last Name\n");

String last\_name = scan.next();

System.out.print("Enter the Students DOB (YYYY-MM-DD)\n");

LocalDate DOBL = LocalDate.parse(scan.next());

Date DOB = Date.valueOf(DOBL);

System.out.print("Enter the Students Group\n");

String group\_ = scan.next();;

try {

Class.forName("org.sqlite.JDBC");

Connection conn = DriverManager.getConnection("jdbc:sqlite:students.db");

Statement stmt = conn.createStatement();

String query = "INSERT INTO Students(ID, first\_name, last\_name, DOB, group\_) "

+ "VALUES (" + StudentNumber + ",'" + first\_name + "','" + last\_name

+ "','" + DOB + "','" + group\_ + "');";

stmt.executeUpdate(query);

stmt.close();

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

} catch (Exception ex) {

System.err.println("Unable to load SQLite driver.");

ex.printStackTrace();

}

}

public void recordResults(int studentID, int badgeID) {

int test1 = 0, test2 = 0, test3 = 0, test4 = 0, test5 = 0, test6 = 0,

test7 = 0, test8 = 0, test9 = 0, test10 = 0, test11 = 0,

test12 = 0, test13 = 0, test14 = 0;

try {

Class.forName("org.sqlite.JDBC");

Connection conn = DriverManager.getConnection("jdbc:sqlite:students.db");

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM ActivityRecord WHERE badgeID = " + badgeID + " AND StudentID = " + studentID);

while (rs.next()) {

test1 = rs.getInt("test1");

test2 = rs.getInt("test2");

test3 = rs.getInt("test3");

test4 = rs.getInt("test4");

test5 = rs.getInt("test5");

test6 = rs.getInt("test6");

test7 = rs.getInt("test7");

test8 = rs.getInt("test8");

test9 = rs.getInt("test9");

test10 = rs.getInt("test10");

test11 = rs.getInt("test11");

test12 = rs.getInt("test12");

test13 = rs.getInt("test13");

test14 = rs.getInt("test14");

}

} catch (SQLException ex) {

ex.printStackTrace();

} catch (Exception ex) {

System.err.println("Unable to load SQLite driver.");

ex.printStackTrace();

}

try {

int option = CONTINUE;

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

while (option != EXIT) {

System.out.println("Which test has the student completed?");

System.out.println("Please select an option:");

System.out.println("[1] Test1");

System.out.println("[2] Test2");

System.out.println("[3] Test3");

System.out.println("[4] Test4");

System.out.println("[5] Test5");

System.out.println("[6] Test6");

System.out.println("[7] Test7");

System.out.println("[8] Test8");

System.out.println("[9] Test9");

System.out.println("[10] Test10");

System.out.println("[11] Test11");

System.out.println("[12] Test12");

System.out.println("[13] Test13");

System.out.println("[14] Test14");

System.out.println("[15] Exit");

option = Integer.parseInt(in.readLine());

switch (option) {

case TEST1:

test1 = 1;

System.out.println("TEST1 added.");

break;

case TEST2:

test2 = 1;

System.out.println("TEST2 added.");

break;

case TEST3:

test3 = 1;

System.out.println("TEST3 added.");

break;

case TEST4:

test4 = 1;

System.out.println("TEST4 added.");

break;

case TEST5:

test5 = 1;

System.out.println("TEST5 added.");

break;

case TEST6:

test6 = 1;

System.out.println("TEST6 added.");

break;

case TEST7:

test7 = 1;

System.out.println("TEST7 added.");

break;

case TEST8:

test8 = 1;

System.out.println("TEST8 added.");

break;

case TEST9:

test9 = 1;

System.out.println("TEST9 added.");

break;

case TEST10:

test10 = 1;

System.out.println("TEST10 added.");

break;

case TEST11:

test11 = 1;

System.out.println("TEST11 added.");

break;

case TEST12:

test12 = 1;

System.out.println("TEST12 added.");

break;

case TEST13:

test13 = 1;

System.out.println("TEST13 added.");

break;

case TEST14:

test14 = 1;

System.out.println("TEST14 added.");

break;

case EXIT:

System.out.println("TEST session closed.");

}

}

} catch (IOException ioe) {

ioe.printStackTrace();

}

try {

Class.forName("org.sqlite.JDBC");

Connection conn = DriverManager.getConnection("jdbc:sqlite:students.db");

Statement stmt = conn.createStatement();

String query = "INSERT INTO ActivityRecord(StudentID, BadgeID, test1, test2, test3, test4, test5, test6, test7, test8, test9, test10, test11, test12, test13, test14)"

+ "VALUES(" + studentID + "," + badgeID //badge ID to identify which badge to add to

+ "," + test1 + "," + test2 + "," + test3 + "," + test4

+ "," + test5 + "," + test6 + "," + test7 + "," + test8

+ "," + test9 + "," + test10 + "," + test11 + "," + test12

+ "," + test13 + "," + test14 + ")";

stmt.close();

stmt.executeUpdate(query);

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

} catch (Exception ex) {

System.err.println("Unable to load SQLite driver.");

ex.printStackTrace();

}

int[] tests = new int[]{test1, test2, test3, test4, test5, test6, test7,

test8, test9, test10, test11, test12, test13, test14};

int count = 0;

for (int i = 8; i < 14; i++) {

if (tests[i] == 1) {

count++;

}

}

if (test1 == 1 && test2 == 1 && test3 == 1 && test4 == 1 && test5 == 1

&& test6 == 1 && test7 == 1 && count >= 3) {

System.out.println("Student " + studentID + " has earned badge number " + badgeID + ".");

}

}

public void recordAttendance(String day, int ID) {

try {

Class.forName("org.sqlite.JDBC");

Connection conn = DriverManager.getConnection("jdbc:sqlite:students.db");

Statement stmt = conn.createStatement();

String query = "INSERT INTO Dates('" + day + "',ID) VALUES (1," + ID + ")";

stmt.executeUpdate(query);

stmt.close();

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

} catch (Exception ex) {

System.err.println("Unable to load SQLite driver.");

ex.printStackTrace();

}

}

public void studentSearch() {

Scanner scan = new Scanner(System.in);

try {

Class.forName("org.sqlite.JDBC");

Connection conn = DriverManager.getConnection("jdbc:sqlite:students.db");

Statement stmt = conn.createStatement();

System.out.println("Enter the students last name: ");

String last\_name = scan.next();

ResultSet rs = stmt.executeQuery("SELECT ID FROM Students WHERE Students.last\_name='" + last\_name + "';");

int StudentID = ((Number) rs.getObject(1)).intValue();

ResultSet rs2 = stmt.executeQuery("SELECT Students.first\_name, Students.last\_name,"

+ "ActivityRecord.test1,"

+ "ActivityRecord.test2,"

+ "ActivityRecord.test3,"

+ "ActivityRecord.test4,"

+ "ActivityRecord.test5,"

+ "ActivityRecord.test6,"

+ "ActivityRecord.test7,"

+ "ActivityRecord.test8,"

+ "ActivityRecord.test9,"

+ "ActivityRecord.test10,"

+ "ActivityRecord.test11,"

+ "ActivityRecord.test12,"

+ "ActivityRecord.test13,"

+ "ActivityRecord.test14 "

+ "FROM Students "

+ "INNER JOIN ActivityRecord ON Students.ID=ActivityRecord.StudentID "

+ "WHERE ID = " + StudentID + ";");

ResultSetMetaData rsmd = rs2.getMetaData();

int columnsNumber = rsmd.getColumnCount();

while (rs2.next()) {

for (int i = 1; i <= columnsNumber; i++) {

if (i > 1) {

System.out.print(", ");

}

String columnValue = rs2.getString(i);

System.out.print(columnValue + " " + rsmd.getColumnName(i));

}

System.out.println("");

}

stmt.close();

conn.close();

} catch (SQLException ex) {

ex.printStackTrace();

} catch (Exception ex) {

System.err.println("Unable to load SQLite driver.");

ex.printStackTrace();

}

}

}

**MAIN CLASS**

package ui1;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.Scanner;

public class UImain {

private final int CONTINUE = 0;

private final int CONTINUE2 = 0;

private final int ADDSTUDENT = 1;

private final int RECORDATTENDANCE = 2;

private final int RECORDRESULTS = 3;

private final int SEARCH = 4;

private final int EXIT = 5;

private final int MONDAY = 1;

private final int TUESDAY = 2;

private final int WEDNESDAY = 3;

private final int THURSDAY = 4;

private final int FRIDAY = 5;

private final int SATURDAY = 6;

private final int SUNDAY = 7;

private final int EXIT2 = 8;

public void runUI() {

Connect c = new Connect();

c.connect();

UI1 sdr = new UI1();

try {

int option = CONTINUE;

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

while (option != EXIT) {

System.out.println("\t\tWELCOME TO THE SAD DATABASE");

System.out.println("Please select an option:");

System.out.println("[1] Add a student");

System.out.println("[2] Record weekly attendance");

System.out.println("[3] Record Results");

System.out.println("[4] Search for a student");

System.out.println("[5] Exit");

option = Integer.parseInt(in.readLine());

Scanner s = new Scanner(System.in);

switch (option) {

case ADDSTUDENT:

sdr.addStudent();

break;

case RECORDATTENDANCE:

rc();

break;

case RECORDRESULTS:

Scanner s2 = new Scanner(System.in);

System.out.print("Enter the Student ID:");

int x = s2.nextInt();

System.out.print("Enter the Badge ID:");

int y = s2.nextInt();

sdr.recordResults(x, y);

System.out.println();

break;

case SEARCH:

sdr.studentSearch();

break;

case EXIT:

System.out.println("Exiting Application.");

break;

default:

System.out.println("Please enter a valid option [1-5].");

break;

}

}

} catch (IOException ioe) {

ioe.printStackTrace();

}

}

public void rc() {

int option2 = CONTINUE2;

UI1 sdr = new UI1();

try {

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

Scanner scan = new Scanner(System.in);

System.out.print("Enter the Students ID:\n");

int ID = scan.nextInt();

System.out.println("Which days did they attend?");

System.out.println("[1] Monday");

System.out.println("[2] Tuesday");

System.out.println("[3] Wednesday");

System.out.println("[4] Thursday");

System.out.println("[5] Friday");

System.out.println("[6] Saturday");

System.out.println("[7] Sunday");

System.out.println("[8] Exit\n");

option2 = Integer.parseInt(in.readLine());

switch (option2) {

case MONDAY:

sdr.recordAttendance("mon", ID);

System.out.println("mon added");

break;

case TUESDAY:

sdr.recordAttendance("tue", ID);

System.out.println();

break;

case WEDNESDAY:

sdr.recordAttendance("wed", ID);

System.out.println();

break;

case THURSDAY:

sdr.recordAttendance("thu", ID);

System.out.println();

break;

case FRIDAY:

sdr.recordAttendance("fri", ID);

System.out.println();

break;

case SATURDAY:

sdr.recordAttendance("sat", ID);

System.out.println();

break;

case SUNDAY:

sdr.recordAttendance("sun", ID);

System.out.println();

break;

case EXIT2:

System.out.println("Exiting attendance record.");

break;

default:

System.out.println("Please enter a valid option [1-8].");

break;

}

} catch (IOException ioe) {

ioe.printStackTrace();

}

}

public static void main(String[] args) {

UImain m = new UImain();

m.runUI();

}

}

**CONNECT CLASS**

package ui1;

import java.io.File;

import java.sql.Connection;

import java.sql.DatabaseMetaData;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

public class Connect {

public static Connection conn = null;

public static String sqliteServer = "jdbc:sqlite:";

public static String resetPath = "";

public static boolean isDatabaseExist(String dbFilePath) {

File dbFile = new File(dbFilePath);

return dbFile.exists();

}

public static void connect() {

try {

createNewDatabase("students");

} catch (Exception ex) {

System.out.println("Error: " + ex);

}

}

public static void createNewDatabase(String fileName) {

try (Connection conn = DriverManager.getConnection(sqliteServer+"src\\ui1\\"+fileName+".db")) {

if (conn != null) {

DatabaseMetaData meta = conn.getMetaData();

Statement statement = conn.createStatement();

System.out.println("Database Has Been Connected!");

}

} catch (SQLException e) {

System.out.println(e.getMessage());

}

}

}