**//ConnectionDB.java**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

public class ConnectionDB {

public static Connection connectDB() {

try {

return DriverManager.getConnection("jdbc:mysql://localhost:3306/university", "root", "1234");

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

}

**//MyWindow.java**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.List;

public class MyWindow extends JFrame {

private final StudentDAO studentDAO = new StudentDAO();

private JTable studentTable;

public MyWindow() {

setTitle("Student Information");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(800, 400);

setLayout(new BorderLayout());

studentDAO.loadStudents();

JButton showAllButton = new JButton("Show All");

JButton uniqueNamesButton = new JButton("Unique Names");

JButton underAgeButton = new JButton("Under Age 22");

showAllButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

updateTable(studentDAO.getAllStudents());

}

});

uniqueNamesButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

updateTable(studentDAO.filterByUniqueName());

}

});

underAgeButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

updateTable(studentDAO.filterByAgeLimit(22));

}

});

JPanel buttonPanel = new JPanel();

buttonPanel.add(showAllButton);

buttonPanel.add(uniqueNamesButton);

buttonPanel.add(underAgeButton);

studentTable = new JTable(new String[0][0], new String[]{"ID", "First Name", "Last Name", "Birth Date"});

JScrollPane scrollPane = new JScrollPane(studentTable);

add(scrollPane, BorderLayout.CENTER);

add(buttonPanel, BorderLayout.SOUTH);

setVisible(true);

}

private void updateTable(List<Student> students) {

String[][] data = new String[students.size()][4];

for (int i = 0; i < students.size(); i++) {

Student student = students.get(i);

data[i][0] = String.valueOf(student.getId());

data[i][1] = student.getFirstName();

data[i][2] = student.getLastName();

data[i][3] = student.getBirthDate().toString();

}

String[] columnNames = {"ID", "First Name", "Last Name", "Birth Date"};

studentTable.setModel(new javax.swing.table.DefaultTableModel(data, columnNames));

}

}

**//Student.java**

import java.sql.Date;

public class Student {

private int id;

private String firstName;

private String lastName;

private Date birthDate;

public Student(int id, String firstName, String lastName, Date birthDate) {

this.id = id;

this.firstName = firstName;

this.lastName = lastName;

this.birthDate = birthDate;

}

public int getId() {

return id;

}

public String getFirstName() {

return firstName;

}

public String getLastName() {

return lastName;

}

public Date getBirthDate() {

return birthDate;

}

@Override

public String toString() {

return id + " " + firstName + " " + lastName + " " + birthDate;

}

}

**//StudentDAO.java**

import java.sql.\*;

import java.time.LocalDate;

import java.time.Month;

import java.time.Period;

import java.util.ArrayList;

import java.util.List;

public class StudentDAO {

private final List<Student> studentList = new ArrayList<>();

public List<Student> getAllStudents() {

return studentList;

}

public void loadStudents() {

String query = "SELECT id, name, surname, birthdate FROM student";

try (Connection connection = ConnectionDB.connectDB(); Statement stmt = connection.createStatement(); ResultSet rs = stmt.executeQuery(query)) {

while (rs.next()) {

int id = rs.getInt("id");

String firstName = rs.getString("name");

String lastName = rs.getString("surname");

Date birthDate = rs.getDate("birthdate");

studentList.add(new Student(id, firstName, lastName, birthDate));

}

} catch (SQLException ex) {

throw new RuntimeException("Error fetching student data", ex);

}

}

public List<Student> filterByUniqueName() {

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

for (Student student : studentList) {

if (hasAllUniqueCharacters(student.getFirstName())) {

uniqueNameList.add(student);

}

}

return uniqueNameList;

}

private boolean hasAllUniqueCharacters(String name) {

int checker = 0;

for (char c : name.toLowerCase().toCharArray()) {

if (Character.isLetter(c)) {

int bitIndex = c - 'a';

if ((checker & (1 << bitIndex)) > 0) {

return false;

}

checker |= (1 << bitIndex);

}

}

return true;

}

public List<Student> filterByAgeLimit(int maxAge) {

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

for (Student student : studentList) {

if (computeAge(student.getBirthDate().toLocalDate()) < maxAge) {

ageFilteredList.add(student);

}

}

return ageFilteredList;

}

private int computeAge(LocalDate birthDate) {

LocalDate now = LocalDate.now();

int age = now.getYear() - birthDate.getYear();

if (birthDate.getMonthValue() > now.getMonthValue() || (birthDate.getMonthValue() == now.getMonthValue() && birthDate.getDayOfMonth() > now.getDayOfMonth())) {

age--;

}

return age;

}

}

**// Main.java**

public class Main {  
 public static void main(String[] args) {  
 new MyWindow();  
 }  
}

**How to Connect DB**

1) Open MySQL Command Line Client

2) input password

3) write the commands (use shift+enter to go to a new line)

CREATE DATABASE university;

USE university;

CREATE TABLE student (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(50),

surname VARCHAR(50),

birthdate DATE

);

INSERT INTO student (name, surname, birthdate) VALUES

('Liam', 'Taylor', '1998-11-30'),

('Sophia', 'Davis', '2002-03-14'),

('Ethan', 'Miller', '2000-06-22'),

('Olivia', 'Anderson', '2001-09-10'),

('Isaac', 'Martin', '1999-04-05'),

('Aurelia', 'Lopez', '2002-01-18');

4) check the database to be sure  
 SELECT \* FROM student;

**How to add MySQL Connector**

In IntelliJ, go to File -> Project Structure -> Libraries, select the “+” -> Java,

Find and select the mysql-connector-j-8.3.0.jar file -> OK -> Apply -> OK