

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет «Дніпровська політехніка»



ЗВІТ

Лабораторна робота №5

з дисципліни «Поглиблене програмування в середовищі Java»

Виконала: ст. гр. 122-21-1

Пивонос Ольга Олександрівна

Перевірів: доц. Мінесєв О.С.

Дніпро

2025

Тема: JDBC.

Завдання: Створити базу даних в будь-якому сервері баз даних. Створити таблицю з переліком студентів, вказати їх прізвище, ім'я, по батькові, день народження, номер залікової книжки та ID.

Створити програму, що буде дозволяти виводити на екран інформацію про студентів, які народилися в тому чи іншому місяці року. Програма повинна завдяки системі jdbc під'єднатися до бази даних та робити до неї запити. Вимог до розробки бази даних немає. Програма ж має бути написана за усіма стандартами ООП. Та може бути спроектована за двох принципів:

- при будь-якій ситуації буде забиратися весь перелік студентів, а вже на стороні java буде зроблено пошук необхідного;
- SQL-запит буде сформований згідно запиту, який зробив користувач і вже сервер управління баз даних буде вирішувати, які самі студенти народилися в тому чи іншому місяці.

У висновку обов'язково пояснити, чому вибрали той чи інший принцип, які в нього переваги та недоліки. Оцінка не залежить від того який сервер управління баз даних вибрали. Перелік студентів зробити не менше 20 людей. Місяць червень зробити місяцем, коли в жодного зі студентів немає дня народження.

SQL код створення бази даних розмістити проєкті 6 лабораторної роботи в файлі database в пекеджі resources. Для виконання цієї лабораторної роботи рекомендується активно використовувати знання отримані на дисципліні, що стосується розробки баз даних.

До паперового звіту обов'язково додати принтскрин з програми, в якій ви дивитесь інформацію вашого сервера управління баз даних, де показати створену таблицю, її ім'я та загальні відомості бази даних, наприклад назва, ім'я, назва користувача адміністратора, пароль тощо. Для роботи з сервером управління баз даних рекомендується використовувати програмне забезпечення компанії jetbrains datagrip. Або вбудовану панель користування

базами даних, що міститься у середовищі IntelliJ IDEA, яка на сьогоднішній день підтримує майже всі сервери управління баз даних.

Хід роботи

1. Для цієї роботи будемо використовувати базу даних SQLite. Для цього у файл pom.xml потрібно додати відповідні залежності, як показано на рисунку 1.

```
36      <dependency>
37          <groupId>org.xerial</groupId>
38          <artifactId>sqlite-jdbc</artifactId>
39          <version>3.45.2.0</version>
40      </dependency>
41      <dependency>
42          <groupId>org.slf4j</groupId>
43          <artifactId>slf4j-api</artifactId>
44          <version>2.0.9</version>
45      </dependency>
```

Рисунок 1 – Додавання залежностей для користування базою даних SQLite

2. Наступним кроком додаємо джерело даних (рисунок 2).

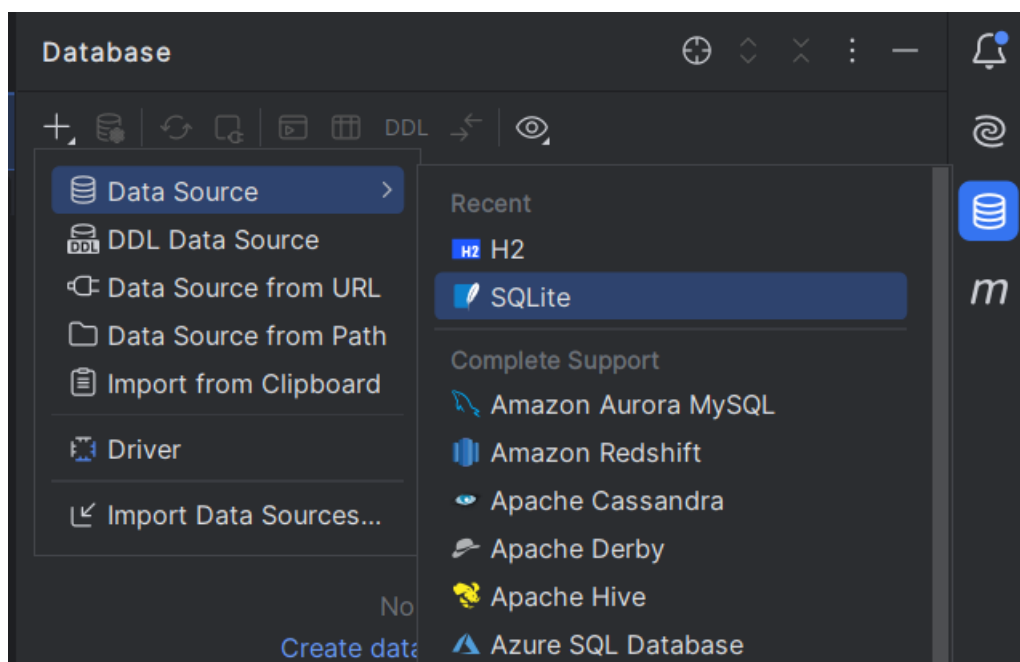


Рисунок 2 – Додавання джерела даних SQLite

3. Вказуємо назву файлу, де буде зберігатись база даних, та перевіряємо з'єднання (рисунок 3).

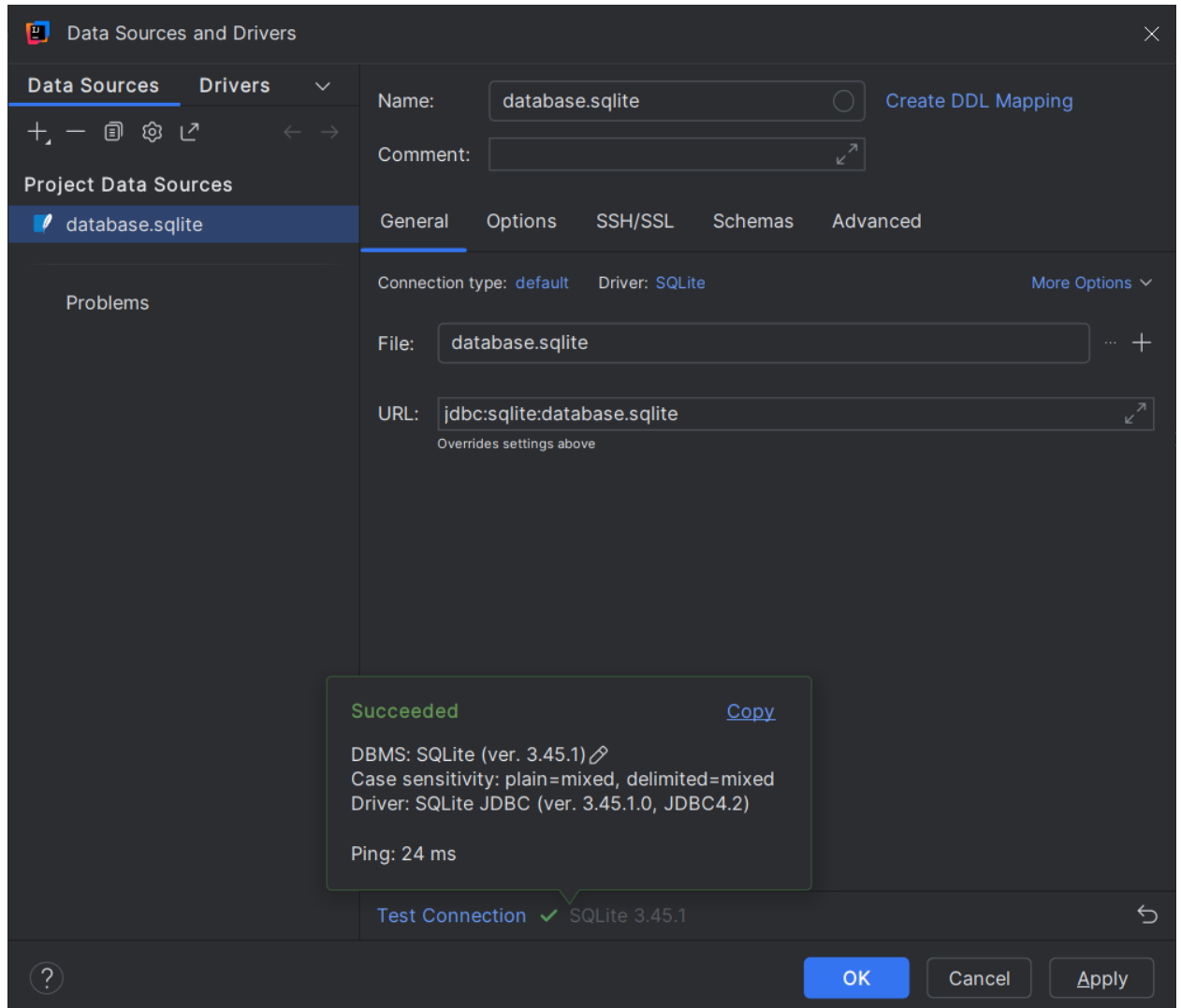


Рисунок 3 – Перевірка з'єднання з джерелом даних

4. У консолі вводимо запити для створення таблиці students та заповнення її даними про студентів (id, ПІБ, дата народження та номер залікової книжки). Запити для цієї таблиці наведено нижче. Ці запити містяться у файлі database.txt у пакеті resources.

```
CREATE TABLE students (  
    student_id INTEGER PRIMARY KEY AUTOINCREMENT,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    middle_name VARCHAR(50),  
    birth_date DATE NOT NULL,  
    record_book_num VARCHAR(10)  
);  
  
INSERT INTO students(first_name, last_name, middle_name, birth_date,
```

```
record_book_num) VALUES
('Іван', 'Руденко', 'Анатолійович', '2000-01-15', 'IR011'),
('Дмитро', 'Коваленко', 'Олександрович', '2000-02-20', 'DK005'),
('Анастасія', 'Ткаченко', 'Олегівна', '2000-03-10', 'AT003'),
('Ірина', 'Якибчук', 'Андріївна', '2000-04-05', 'IY007'),
('Василь', 'Мельник', 'Ігорович', '2000-05-12', 'VM023'),
('Ігор', 'Шевченко', 'Володимирович', '2000-07-18', 'IS010'),
('Андрій', 'Савченко', 'Сергійович', '2000-08-22', 'AS006'),
('Віктор', 'Бойко', 'Олегівич', '2000-09-30', 'VB002'),
('Ольга', 'Зімниця', 'Василівна', '2000-10-11', 'OZ030'),
('Валерія', 'Жук', 'Романівна', '2000-11-25', 'VZ017'),
('Марина', 'Захарченко', 'Михайлівна', '2000-12-03', 'MZ008'),
('Юлія', 'Бондар', 'Олексіївна', '2001-01-14', 'YB012'),
('Сергій', 'Марченко', 'Анатолійович', '2001-02-28', 'SM020'),
('Владислав', 'Павленко', 'Віталійович', '2001-03-17', 'VP015'),
('Віта', 'Волокова', 'Юріївна', '2001-04-09', 'VV009'),
('Гліб', 'Зубок', 'Іванович', '2001-05-21', 'HZ019'),
('Софія', 'Василенко', 'Петрівна', '2001-07-07', 'SV102'),
('Максим', 'Олійник', 'Денисович', '2001-08-19', 'MO032'),
('Катерина', 'Гончаренко', 'Олександрівна', '2001-09-23', 'KH027'),
('Людмила', 'Гончаренко', 'Олексіївна', '2001-10-31', 'LH001');
```

В результаті отримуємо таблицю із вказаними стовпцями та значеннями (рисунок 4).

student_id	first_name	last_name	middle_name	birth_date	record_book_num
1	Іван	Руденко	Анатолійович	2000-01-15	IR011
2	Дмитро	Коваленко	Олександрович	2000-02-20	DK005
3	Анастасія	Ткаченко	Олегівна	2000-03-10	AT003
4	Ірина	Якибчук	Андріївна	2000-04-05	IY007
5	Василь	Мельник	Ігорович	2000-05-12	VM023
6	Ігор	Шевченко	Володимирович	2000-07-18	IS010
7	Андрій	Савченко	Сергійович	2000-08-22	AS006
8	Віктор	Бойко	Олегівич	2000-09-30	VB002
9	Ольга	Зімниця	Василівна	2000-10-11	OZ030
10	Валерія	Жук	Романівна	2000-11-25	VZ017
11	Марина	Захарченко	Михайлівна	2000-12-03	MZ008
12	Юлія	Бондар	Олексіївна	2001-01-14	YB012
13	Сергій	Марченко	Анатолійович	2001-02-28	SM020
14	Владислав	Павленко	Віталійович	2001-03-17	VP015
15	Віта	Волокова	Юріївна	2001-04-09	VV009
16	Гліб	Зубок	Іванович	2001-05-21	HZ019
17	Софія	Василенко	Петрівна	2001-07-07	SV102
18	Максим	Олійник	Денисович	2001-08-19	MO032
19	Катерина	Гончаренко	Олександрівна	2001-09-23	KH027
20	Людмила	Гончаренко	Олексіївна	2001-10-31	LH001

Рисунок 4 – Створена таблиця students

5. В якості класу моделі створюємо клас Student.java, який буде використовуватись для створення студентів і збереження даних про них.

```
package edu.ntudp.pzks.lab5;

public class Student {
    private int studentId;
    private String lastName;
    private String firstName;
    private String middleName;
```

```

private String birthDate;
private String recordBookNumber;

public Student(int studentId, String firstName, String lastName, String
middleName, String birthDate, String recordBookNumber) {
    this.studentId = studentId;
    this.firstName = firstName;
    this.lastName = lastName;
    this.middleName = middleName;
    this.birthDate = birthDate;
    this.recordBookNumber = recordBookNumber;
}

public int getId() {
    return studentId;
}

public String getFirstName() {
    return firstName;
}

public String getLastName() {
    return lastName;
}

public String getMiddleName() {
    return middleName;
}

public String getBirthDate() {
    return birthDate;
}

public String getRecordBookNumber() {
    return recordBookNumber;
}

@Override
public String toString() {
    return lastName + " " + middleName + " " + firstName + ", дата
народження: " +
        birthDate + ", залікова книжка: " + recordBookNumber;
}
}

```

6. Відповідальним за взаємодію програми з базою даних буде клас DBManager.java. В ньому виконуємо підключення до бази даних та отримання списку студентів, що народились певного місяця. В даному випадку фільтрацію даних реалізовано на стороні бази даних, адже такий спосіб є більш ефективним з точки зору продуктивності – в програму надходить менша кількість даних, і, відповідно, навантаження є меншим, що особливо відчувається при роботі з великими обсягами даних.

```

package edu.ntudp.pzks.lab5;

import java.sql.*;

```

```

import java.util.ArrayList;
import java.util.List;

public class DBManager {
    private static final String URL = "jdbc:sqlite:database.sqlite";

    public List<Student> getStudentsByBirthMonth(int month) {
        List<Student> students = new ArrayList<>();

        String sql = "SELECT * FROM students WHERE substr(birth_date, 6, 2) =
?";

        try (Connection conn = DriverManager.getConnection(URL);
            PreparedStatement stmt = conn.prepareStatement(sql)) {

            String monthFormatted = String.format("%02d", month);
            stmt.setString(1, monthFormatted);

            ResultSet rs = stmt.executeQuery();

            while (rs.next()) {
                Student student = new Student(
                    rs.getInt("student_id"),
                    rs.getString("last_name"),
                    rs.getString("first_name"),
                    rs.getString("middle_name"),
                    rs.getString("birth_date"),
                    rs.getString("record_book_num")
                );
                students.add(student);
            }
        } catch (SQLException e) {
            throw new RuntimeException("Помилка при отриманні студентів", e);
        }
        return students;
    }
}

```

7. Для взаємодії з користувачем визначаємо клас Main.java. В ньому пропонуємо ввести номер місяця для пошуку студентів та виводимо отримані результати.

```

package edu.ntudp.pzks.lab5;

import java.util.List;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        DBManager dbManager = new DBManager();

        System.out.print("Введіть місяць (1-12): ");
        int month = scanner.nextInt();

        List<Student> students = dbManager.getStudentsByBirthMonth(month);

        if (students.isEmpty()) {
            System.out.println("Студентів, які народилися у цьому місяці, немає.");
        }
    }
}

```

```

    } else {
        System.out.println("Студенти, які народилися у місяці " + month +
":");
        students.forEach(System.out::println);
    }
}
}

```

Результати роботи програми для пошуку студентів, що народились у червні та жовтні, представлено на рисунках 5 та 6 відповідно.

```

Run Main x
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:E:\IntelliJ IDEA\IntelliJ IDEA 2024.3\lib\
Введіть місяць (1-12): 6
Студентів, які народилися у цьому місяці, немає.
Process finished with exit code 0

```

Рисунок 5 – Пошук студентів, що народились у червні

```

Run Main x
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:E:\IntelliJ IDEA\IntelliJ IDEA 2024.3\lib\
Введіть місяць (1-12): 10
Студенти, які народилися у місяці 10:
Ольга Василівна Зімниця, дата народження: 2000-10-11, залікова книжка: OZ030
Людмила Олексіївна Гончаренко, дата народження: 2001-10-31, залікова книжка: LH001
Process finished with exit code 0

```

Рисунок 6 – Пошук студентів, що народились у жовтні

Висновки: На даній лабораторній роботі було реалізовано програму, що здійснює підключення до бази даних SQLite, виконує запит до неї на пошук студентів за місяцем їх народження та виводить результати на екран.