

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таныгин Максим Олегович
Должность: и.о. декана факультета фундаментальной и прикладной информатики
Дата подписания: 21.09.2023 13:09:47
Уникальный программный ключ:
65ab2aa0d384efe8480e6a4c688eddbc475e411a

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования

«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ
Проректор по учебной работе

О.Г. Доктионова
« 21 » 12 2017



ИСПОЛЬЗОВАНИЕ БАЗЫ ДАННЫХ В КЛИЕНТ – СЕРВЕРНОМ ПРИЛОЖЕНИИ

Методические указания по выполнению лабораторной работы
по дисциплине «Конструирование программного обеспечения»
для студентов направления подготовки бакалавров 09.03.04
«Программная инженерия»

Курск 2017

УДК 004.65

Составители: В.Г. Белов, Т.М. Белова

Рецензент

Кандидат технических наук, доцент кафедры программной инженерии ЮЗГУ И.Н. Ефремова

Использование базы данных в клиент – серверном приложении: методические указания по выполнению лабораторной работы по дисциплине "Конструирование программного обеспечения" для студентов направления подготовки 09.03.04 "Программная инженерия" / Юго-Зап. гос. ун-т; сост.: В.Г. Белов, Т.М. Белова, – Курск, 2017. – 22 с.: ил. 16.

Изложена последовательность действий с программным обеспечением Java Eclipse по использованию базы данных в СУБД SQLite при клиент-серверном подключении.

Материал предназначен для студентов направления подготовки бакалавров 09.03.04 «Программная инженерия», а также будет полезен студентам всех направлений подготовки, изучающим технологии разработки баз данных.

Подписано в печать 27.12.17. Формат 60x84 1/16.
Усл. печ. л. 10. Уч.-изд. л. 29 Тираж 100 экз. Заказ 4388. Бесплатно.
Юго-Западный государственный университет

305040, Курск, ул.50 лет Октября, 94.

Содержание

1 Цель лабораторной работы	4
2 Порядок выполнения лабораторной работы	5
2.1 Необходимые условия для подключения	5
2.2 Установка соединения с базой данных SQLite	5
2.3. Использование базы данных в клиент-серверном приложении.....	14
3 Содержание отчета по лабораторной работе	21
4 Вопросы к защите лабораторной работе	22

1 Цель лабораторной работы

Целью лабораторной работы является получение знаний и умений создания в Java Eclipse базы данных с помощью СУБД SQLite.

Подключенную базу данных можно использовать в клиент-серверном приложении.

2 Порядок выполнения лабораторной работы

2.1 Необходимые условия для подключения

Для подключения SQLite к проекту требуется `sqlite-jdbc-connector.jar`.

2.2 Установка соединения с базой данных SQLite

Для установки соединения следуйте следующим шагам:

1. Откройте Data Source Explorer следующим путем [Window]/[Show View]/[Other...]/[Data Management]/[Data Source Explorer] (рисунок 1).

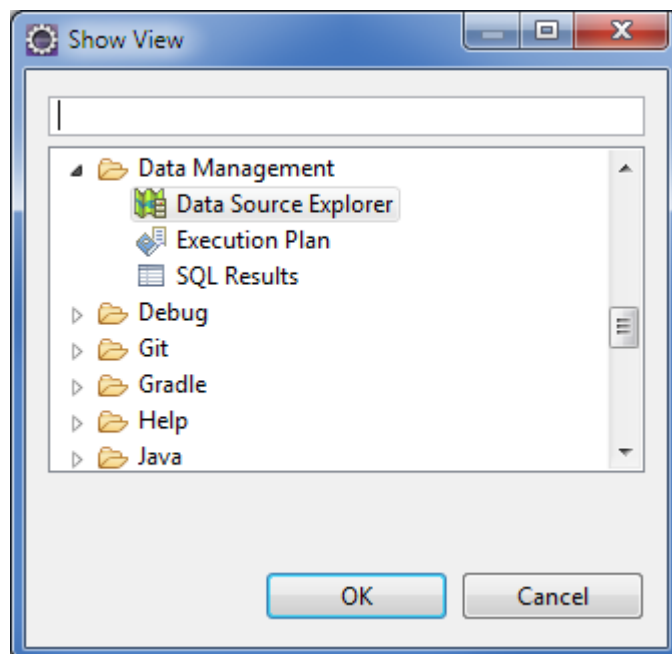


Рисунок 1 – Show View меню

2. Нажмите правой кнопкой мыши на папке Database Connections и выберите [New...] (рисунок 2).

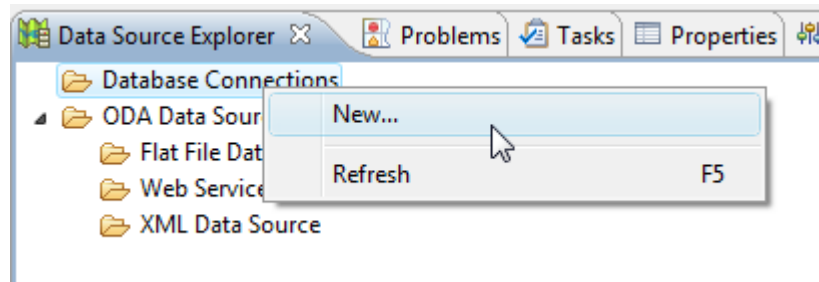


Рисунок 2 – Создание нового подключения

3. Выберите базу данных которую вы хотите подключить (SQLite, например), задайте имя и описание по желанию и нажмите Next (рисунок 3).

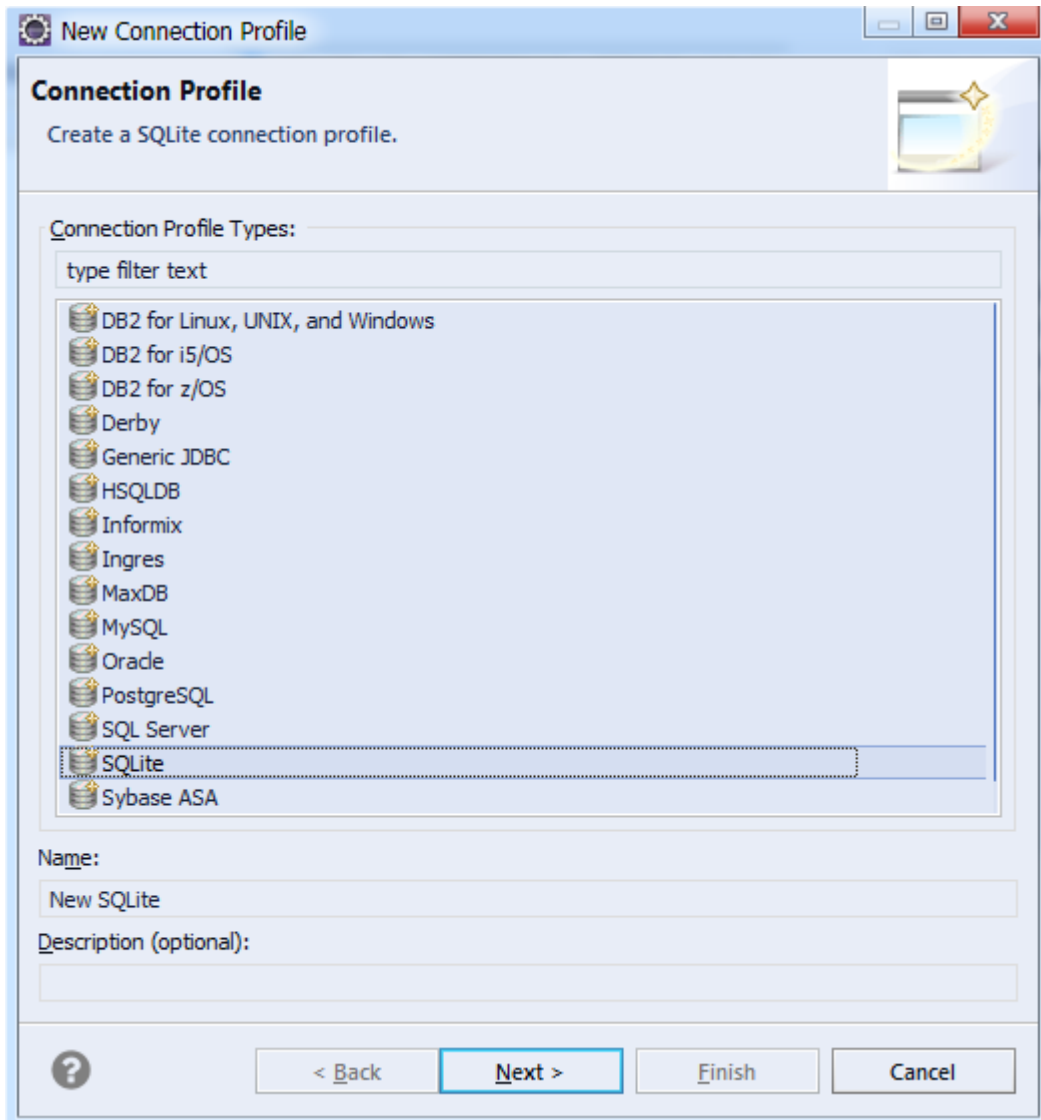


Рисунок 3 – Новый профиль подключения

4. На следующей странице нажмите на иконку с названием New Driver Definition рядом с выпадающим списком Drivers (рисунок 4).

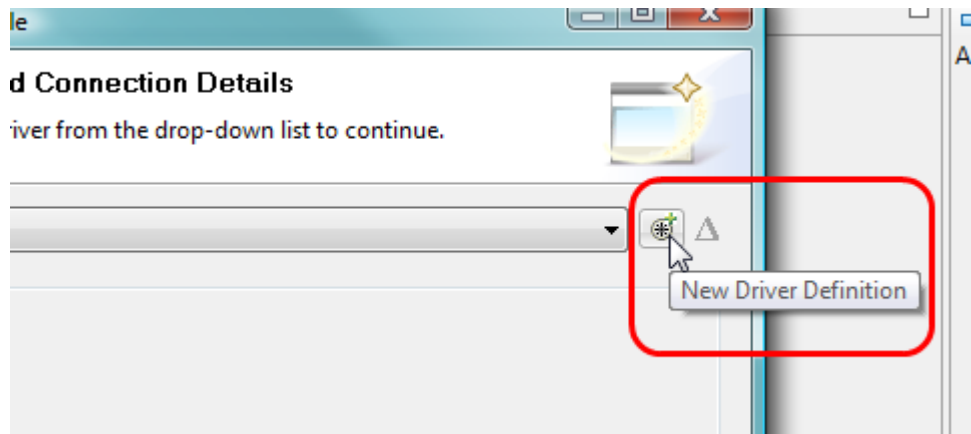


Рисунок 4 – Создание нового драйвера

5. Во вкладке Name/Type выберите версию драйвера (рисунок 5).

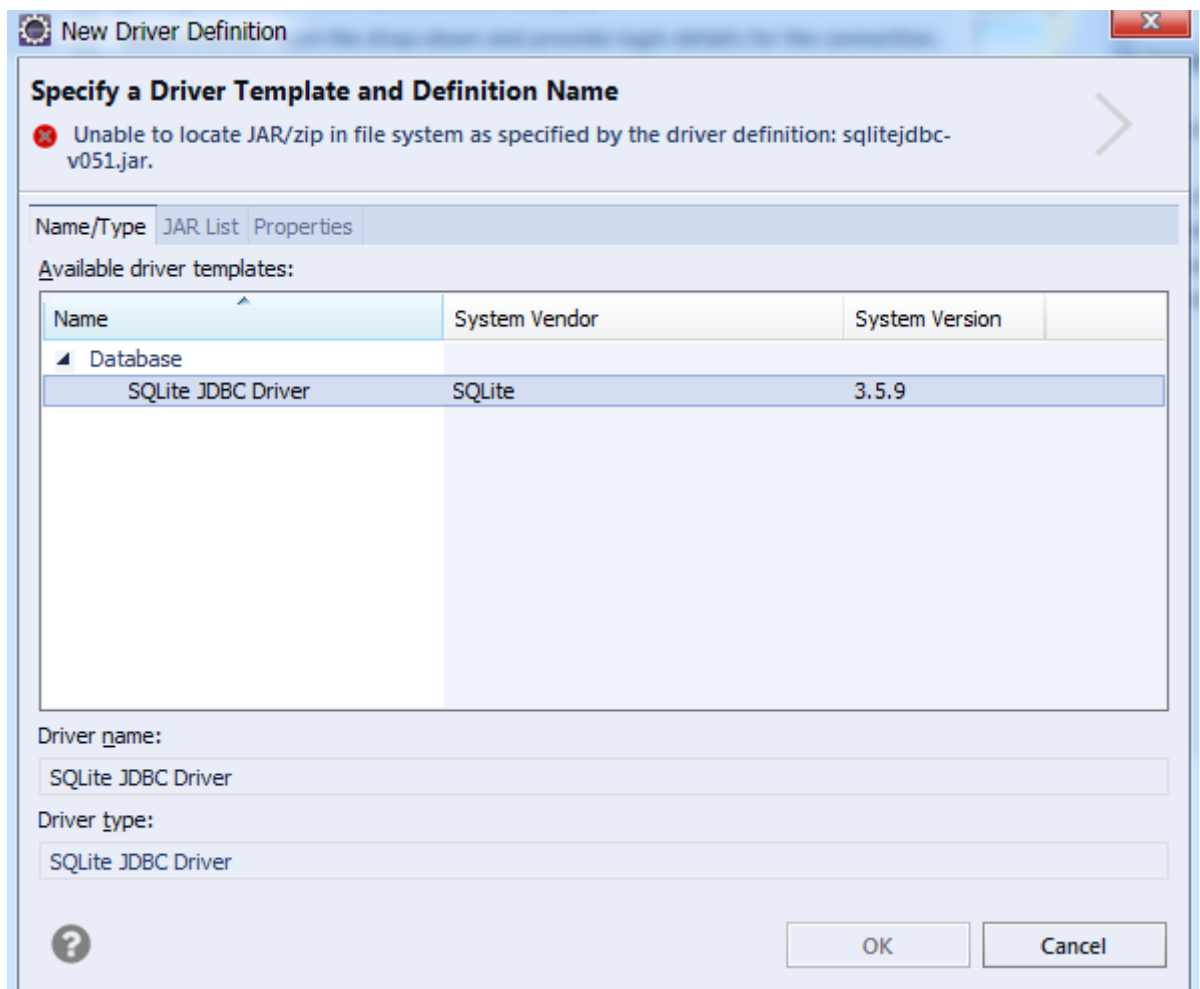


Рисунок 5 – SQLite JDBC драйвер

6. Перейдите на вкладку Jar List и нажмите кнопку Clear All для того чтобы сбросить все имеющиеся .jar подключения (рисунок 6).

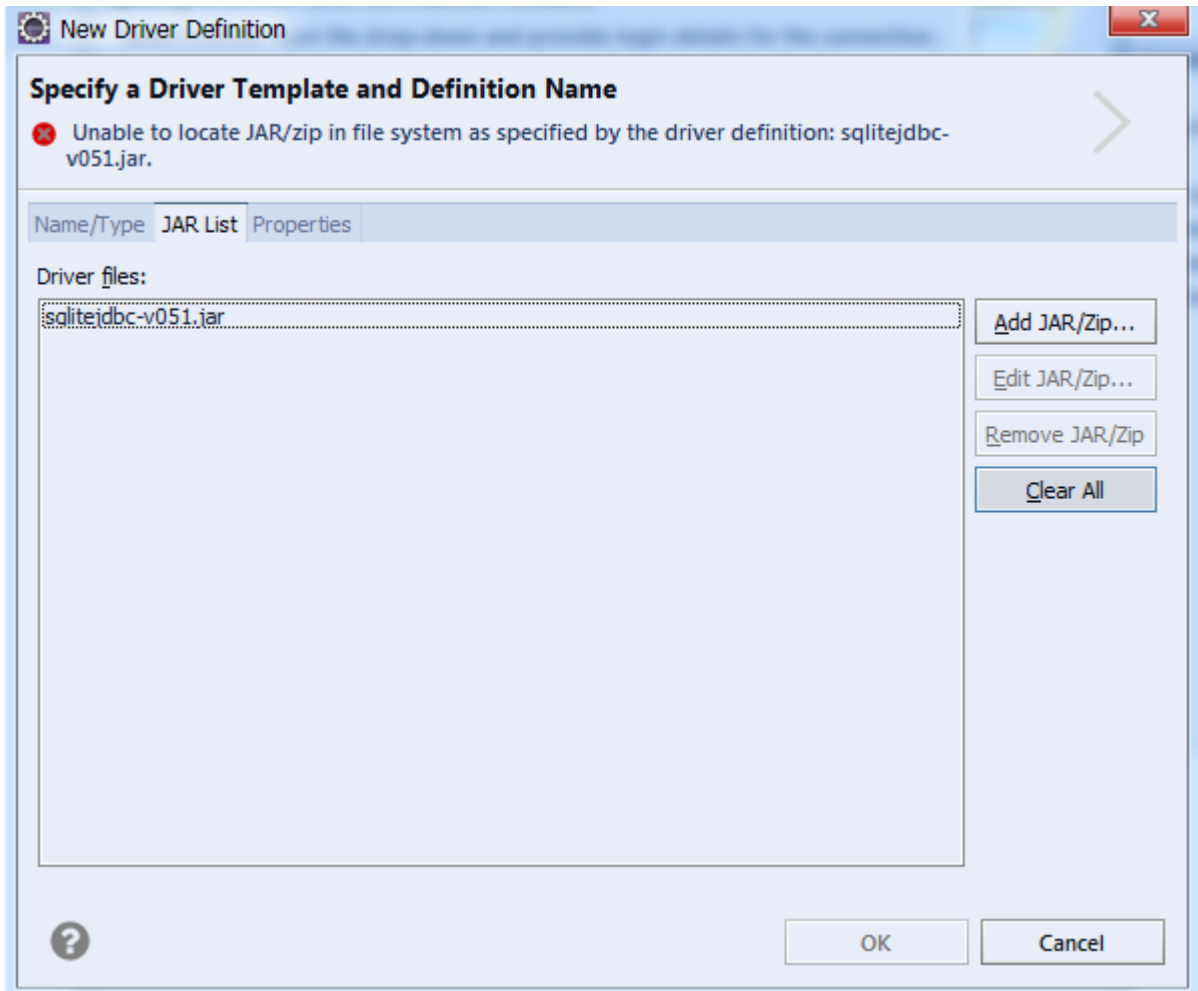


Рисунок 6 – Очистка jar списка

7. После необходимо нажать кнопку Add JAR/Zip... для добавления нового .jar файла (рисунок 7).



Рисунок 7 – Добавление Jar/Zip...

8. Укажите путь в котором хранится SQLite Connector/J zip файл (рисунок 8).

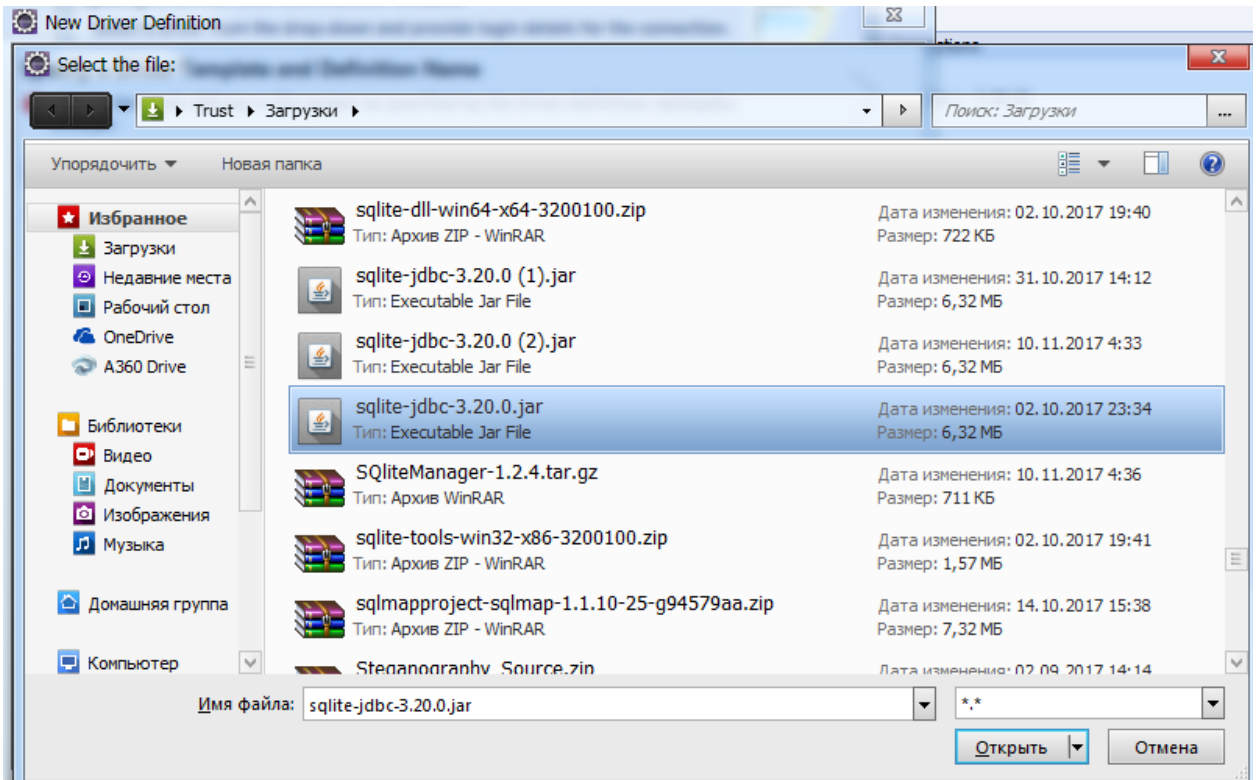


Рисунок 8 – Выбор .jar файла

9. Выберите вкладку «Свойства» и измените ваше SQL-подключение в соответствии с вашими параметрами после чего нажмите ОК. Примечание: The JDBC URL следует привести к следующей форме:

```
jdbc:sqlite://[host:port],[host:port].../[database][?propertyName1]=[propertyValue1][&propertyName2]=[propertyValue2]...
```

Если значение host name не определено, то по умолчанию оно равно 127.0.0.1. Если значение port не определено, то по умолчанию оно равно 9090, это стандартный порт для SQLite серверов. В коде ниже необходимо указать имя нашего подключению sqlite.db:

```
jdbc:sqlite://localhost:9090/sqlite.db или jdbc:sqlite:D:\sqlite.db
```

10. Проверьте строку connection URL во вкладке Properties (рисунок 9).

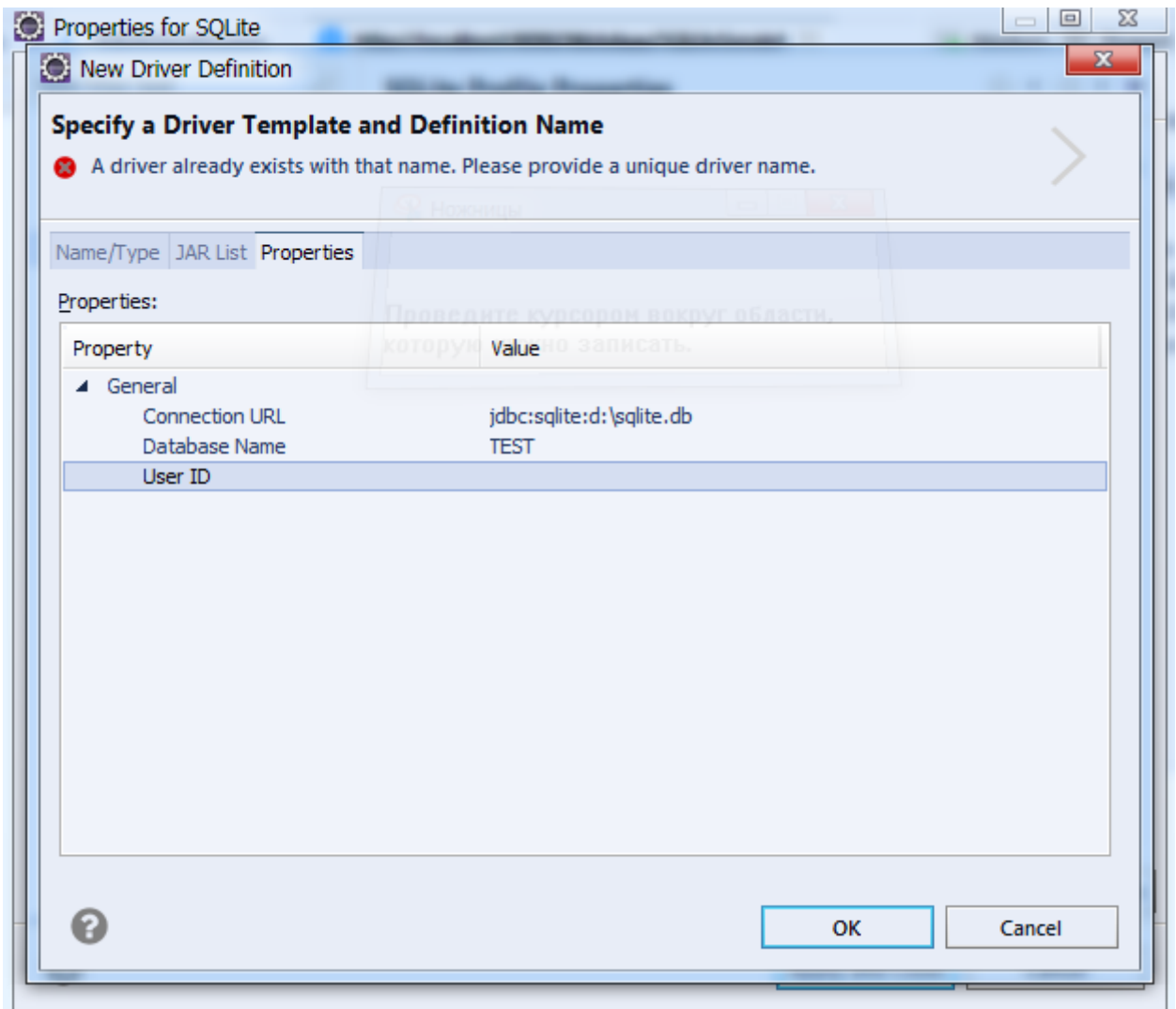


Рисунок 9 – Проверка Connection URL во вкладке Properties

11. Чтобы проверить правильность данных нажмите кнопку Test Connection... (рисунок 10).

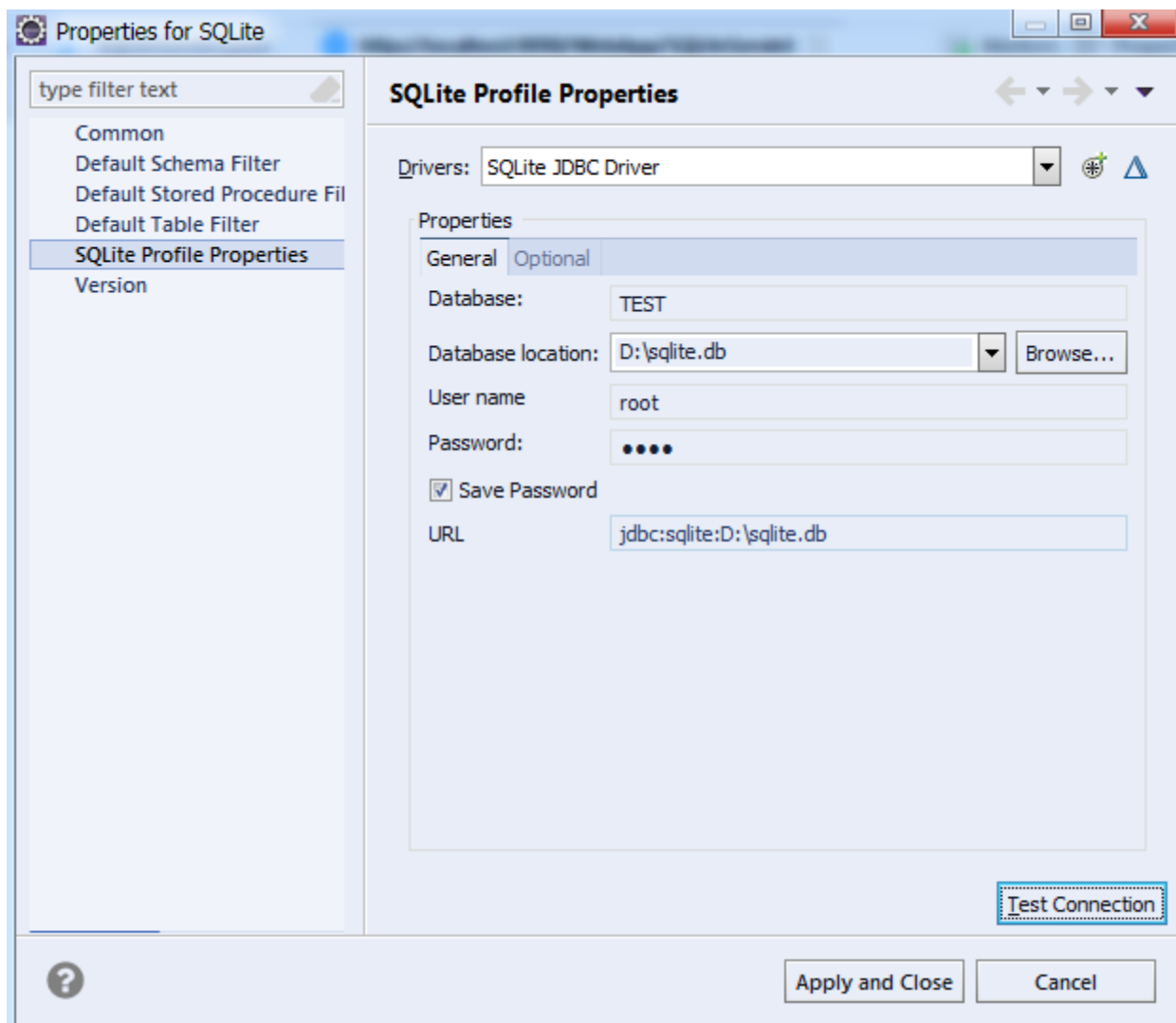


Рисунок 10 – Test Connection

12. В случае успеха появится окно с сообщением об успешном тестовом соединении, нажмите Ок чтобы продолжить (рисунок 11).

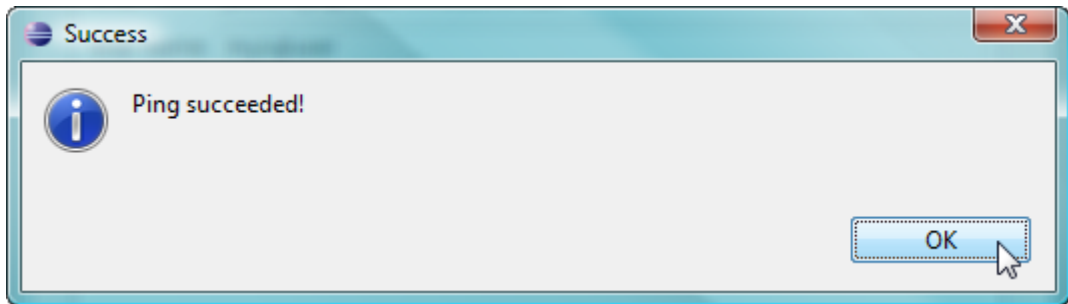


Рисунок 11 – Ping succeeded

13. Нажмите кнопку Finish чтобы закончить настройку и подключение к базе данных появится в папке Database Connections (рисунок 12).

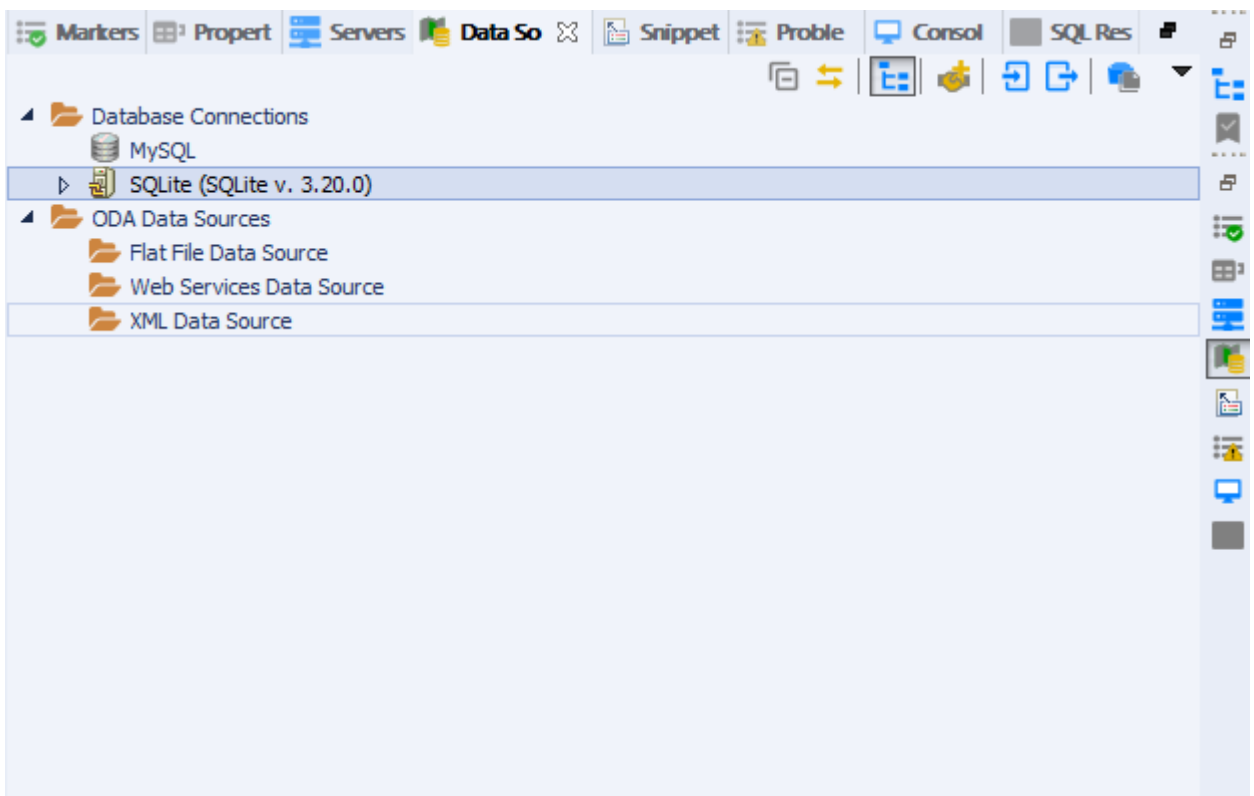


Рисунок 12 – Database Connections

2.3. Использование базы данных в клиент-серверном приложении

Для использования базы данных необходимо создать Servlet. Чтобы сделать это, выполните следующие шаги:

1. В Package Explorer проект в который необходимо добавить сервлет и нажмите правой кнопкой мыши на папке с ним, после перейдите в меню New-Servlet (рисунок 13).

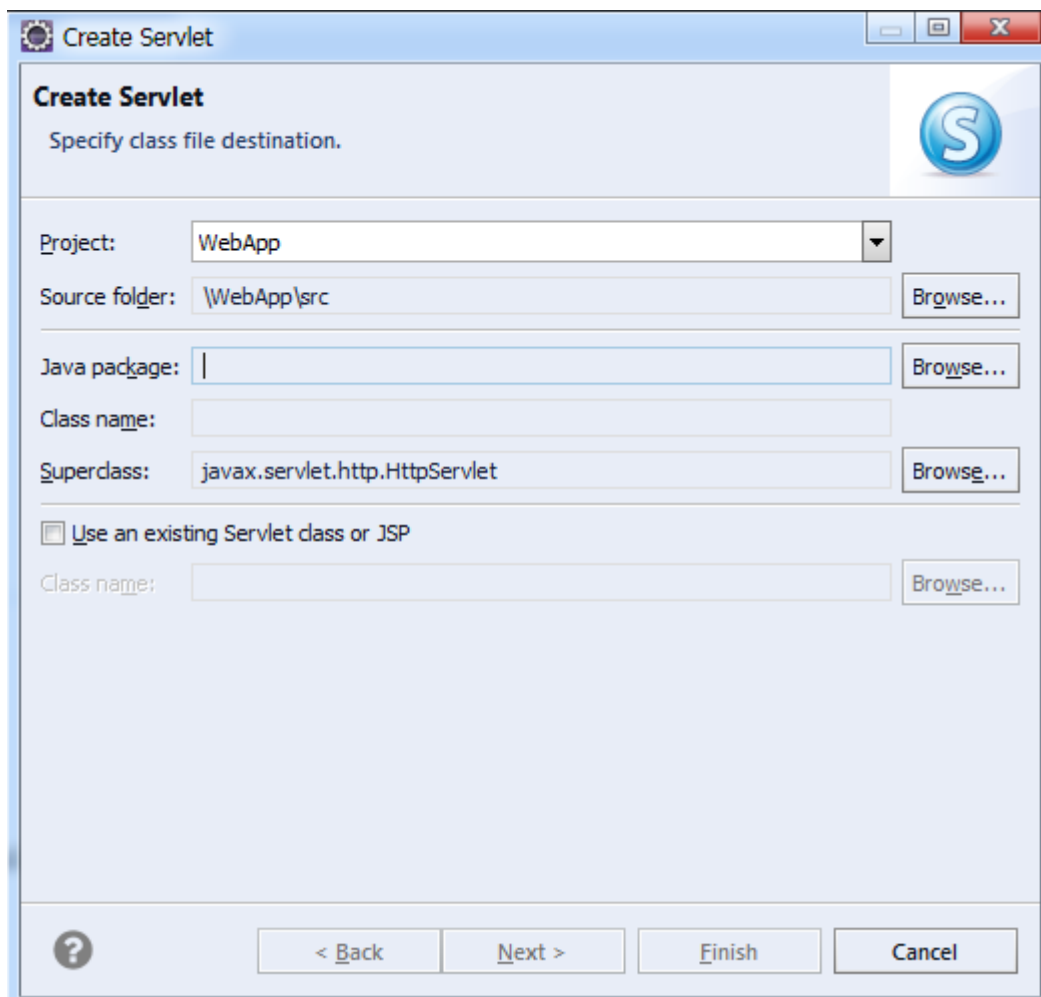


Рисунок 13 – Создание Servlet

2. Введите имя класса, например “MyServlet” и дважды нажмите на кнопку Next чтобы выбрать методы для сервлета. Снимите

галочки с “Constructors from superclass” и “doPost” методов и нажмите Finish (рисунок 14).

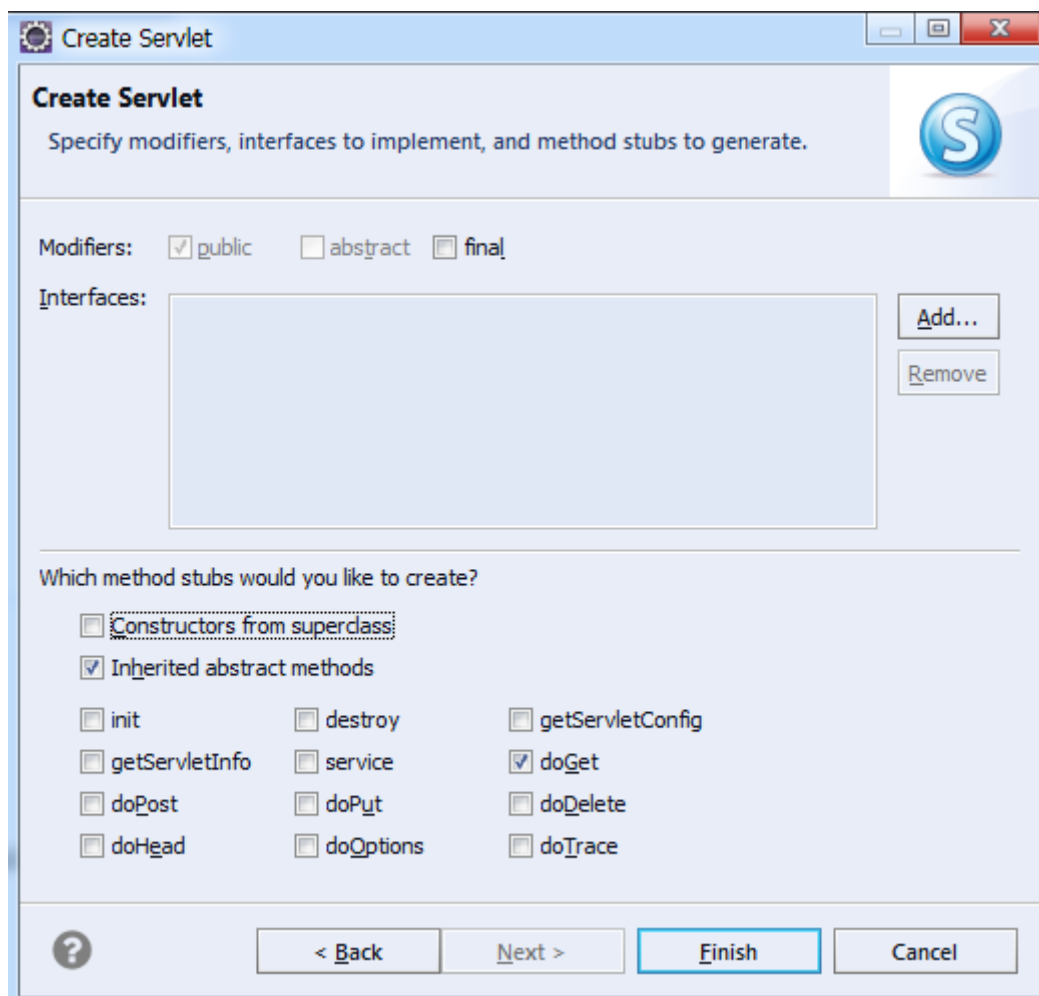


Рисунок 14 – Методы для Servlet

3. После выполнения предыдущих шагов приложение откроет “MyServlet.java” файл (рисунок 15).


```

import java.io.IOException;

/**
 * Servlet implementation class Servlet
 */
@WebServlet("/servlet")
public class Servlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }
}

```

Рисунок 15 – MyServlet.java файл

4. Первое что необходимо сделать это указать URL, username и password поля. И определить подключение. Пример указан в нижеприведенном коде:

```

private String url = "jdbc:sqlite:d:\\sqlite.db";
private String name = "root";
private String password = "root";

```

5. Второе что необходимо сделать это присвоить переменной подключения и переменной Statement значение null. Пример указан в нижеприведенном коде:

```

Connection con = null;
PreparedStatement stmt = null;

```

Далее необходимо загрузить драйвер. Пример указан в нижеприведенном коде:

```

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

    con = DriverManager.getConnection(url, name, pass);
}

```

6. Далее необходимо инициализировать SQL-запрос к базе данных. Пример указан в нижеприведенном коде:

```
String sql = "SELECT * from contacts";
stmt = con.prepareStatement(sql);
ResultSet rs = stmt.executeQuery();
```

7. Используйте `doGet` метод для вывода информации из базы данных в клиентское приложение. Обычно HTTP GET-запросы являются идемпотентными. То есть вы получаете точно такой же результат каждый раз, когда выполняете запрос (оставляя авторизацию / аутентификацию и чувствительный к времени характер результатов поиска страницы, последних новостей и т. д.). Мы можем говорить о bookmarkable запросе. Нажав на ссылку, щелкнув закладку, введя необработанный URL-адрес в адресной строке браузера, и т. д. все это будет запускать HTTP-запрос GET. Если сервлет прослушивает соответствующий URL-адрес, будет вызван его метод `doGet ()`. Обычно он используется для предварительной обработки запроса. То есть делая некоторые бизнес-материалы перед представлением вывода HTML из JSP, например, для сбора данных для отображения в таблице. Пример указан в нижеприведенном коде:

```
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    response.setContentType("text/html");
    response.setCharacterEncoding("UTF-8");
    PrintWriter out = response.getWriter();

    try
    {
        Class.forName("org.sqlite.JDBC");
        con = DriverManager.getConnection(url, name, pass);
        String sql = "SELECT * from contacts";
```

```

        stmt = con.prepareStatement(sql);
        ResultSet rs = stmt.executeQuery();
        while (rs.next()) {
            out.println(rs.getString("phone"));
        }
        out.close();
    }
    catch(SQLException | ClassNotFoundException ex)
    {

    }
    finally
    {
        try
        {
            if(con != null)
                con.close();
        }
        catch(SQLException ex)
        {

        }
    }
}
}

```

В строке:

```
response.setContentType("text/html");
```

Этим кодом устанавливается текстовый тип информации и html страница.

В строке:

```
response.setCharacterEncoding("UTF-8");
```

кодом задается тип кодировки UTF-8 для отображения кириллицы.

Этим кодом задается оператор и запрос на выборку всех данных из базы данных:

```

String sql = "SELECT * from contacts";
stmt = con.prepareStatement(sql);
ResultSet rs = stmt.executeQuery();

```

В нижеприведенном цикле выполняется вывод информации из базы данных “sqlite” и таблицы “contacts”, а конкретно строка “phone”:

```
        while (rs.next()) {  
            out.println(rs.getString("phone"));  
        }  
        out.close();
```

8. Запустите приложение на Tomcat Apache сервере (рисунок 16).

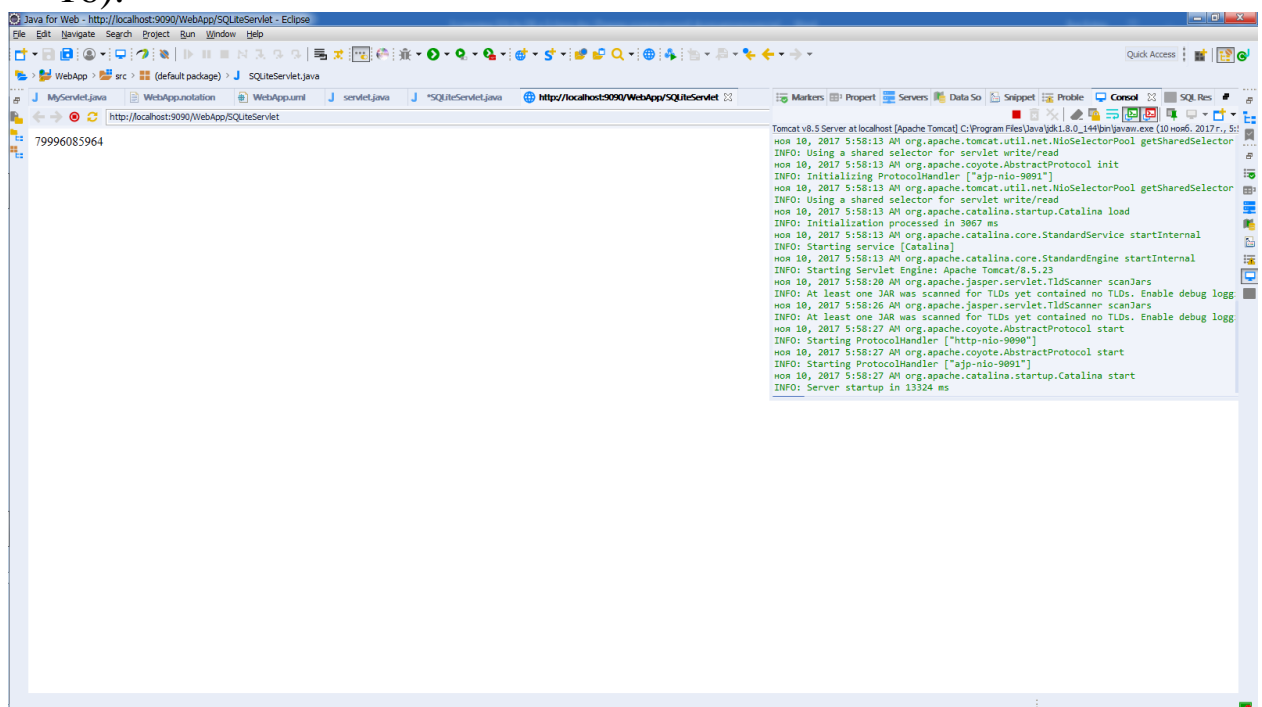


Рисунок 16 – Выборка из строки “phone” из базы данных

3 Содержание отчета по лабораторной работе

В сводный отчет по лабораторным работам в качестве одного из разделов или подразделов включаются скриншоты, показывающие подключение базы данных к Java приложению.

4 Вопросы к защите лабораторной работе

1. Что такое SQLite connector?
2. Какие строки из базы данных мы можем вывести?
3. Какова суть метода doGet?
4. Что помогает нормально отображать кириллицу?
5. Какой драйвер мы использовали?
6. Что такое localhost?
7. Что случится если при подключении мы укажем другой тип подключения (SQLite) с кодом приведенным в примерах?