

**构件与中间件技术**

课程实验报告

实验名称：学生信息查询系统

实验次数：第三次

任课教师：徐悦甡

课程班级：软件工程方向

学号姓名：21030540006 张平

提交日期：2022年04月20日

**一、 实验名称**

第 3 次实验：学生信息查询系统

**二、 实验日期**

2022 年 04 月 16 日

**三、 实验学生**

21030540006 张平

1. **实验目的**

本次实验通过开发一个简单的Java Web项目，在前端页面查询学生信息、后端程序访问数据库并返回结果，体验一次完整的全栈开发流程，从而理解前后端的对接和信息交换。

**五、 实验内容**

本次实验中，使用JSP+Servlet+Tomcat+JDBC+MySQL构建一个Java Web工程，在本地实现一个简单的网站，用来查询学生信息。具体来说，是在MySQL中构建一个简单的学生信息数据库；再用JSP实现一个简单的网页，客户可通过浏览器访问网站，在输入框中输入学生学号；Web服务器Tomcat则从JSP页面获取用户请求，然后将请求信息转发给Servlet；Servlet执行数据库连接、查询操作，并将该学生的信息返回到网页上，显示其学号、姓名、班级等。

**六、 实验过程**

本次实验过程中，我们要做的包括配置实验环境、建立数据库和关系表、编写JSP页面、编写Servlet、实际运行测试等。

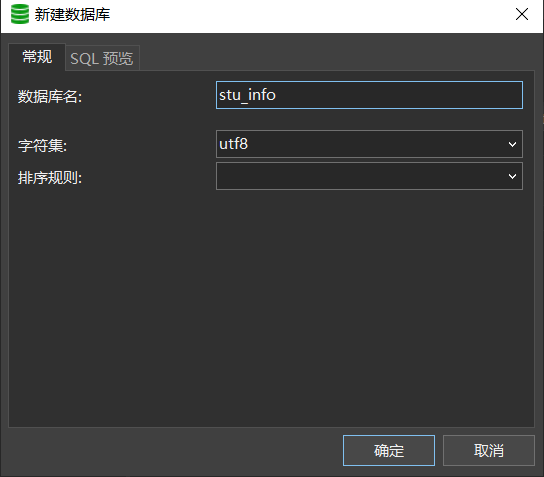
1. 配置实验环境：首先安装MySQL和Navicat for MySQL，参考网络上的安装教程即可。为了验证安装配置是否成功，我们运行mysql --version查看MySQL的版本：



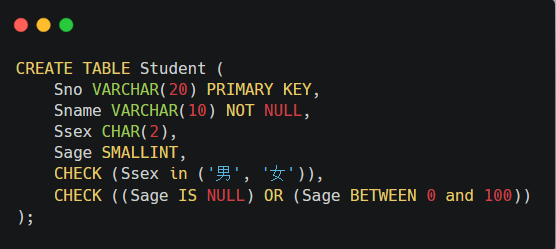
为了使用MySQL数据库，必须先启动MySQL服务。在命令行中，（以管理员身份）运行net start mysql命令来启动MySQL服务。实验结束后，需要输入net stop mysql命令、停止MySQL服务。



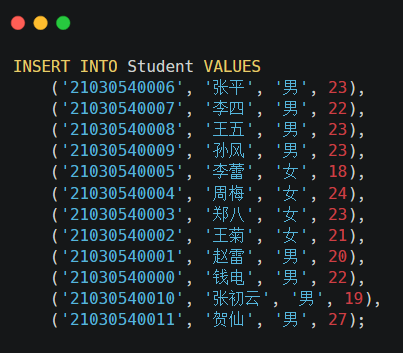
1. 建立数据库和关系表：随后，运行Navicat for MySQL，新建一个名为myconn的数据库连接。打开连接后，建立学生信息数据库如下：



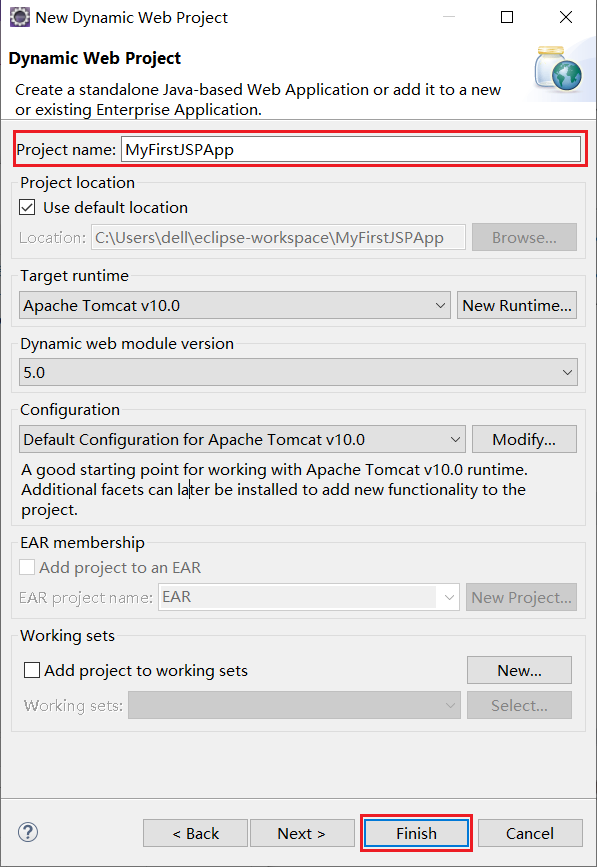
在stu\_info数据库中，运行如下语句建立学生信息表：



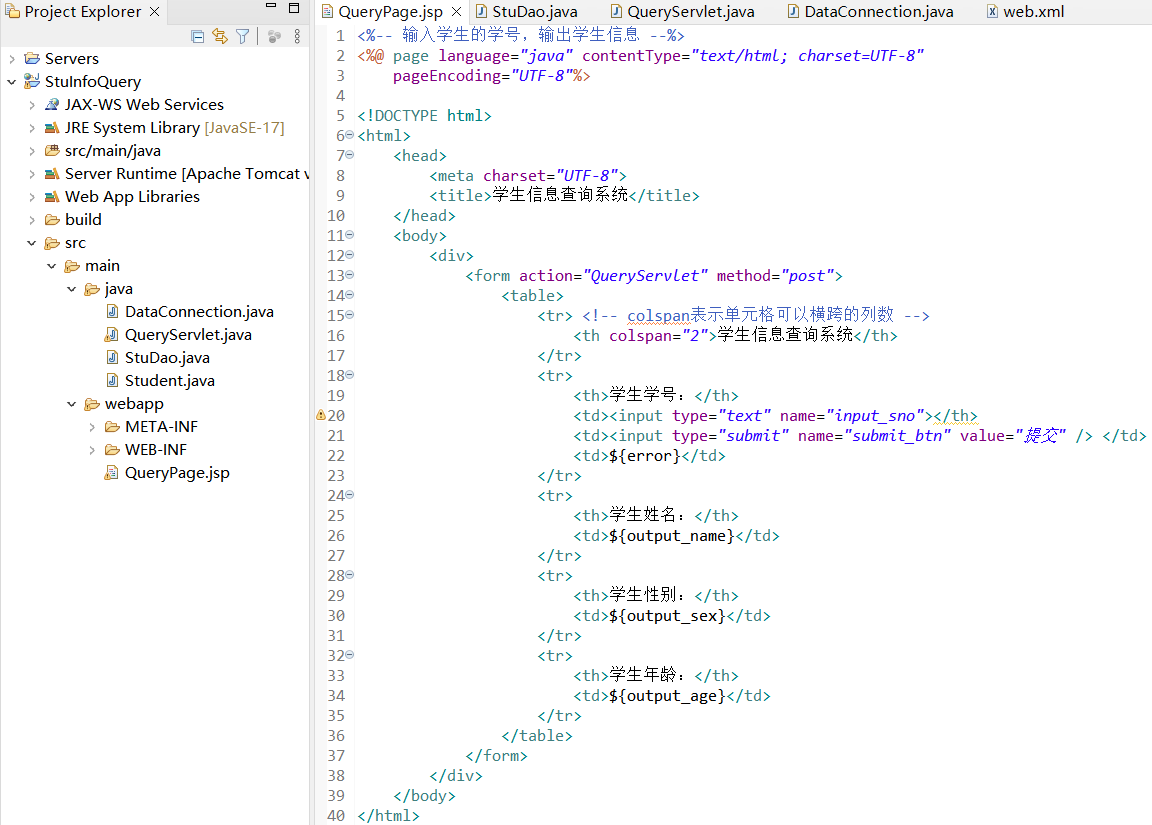
用INSERT INTO语句，插入如下数据到学生表中：



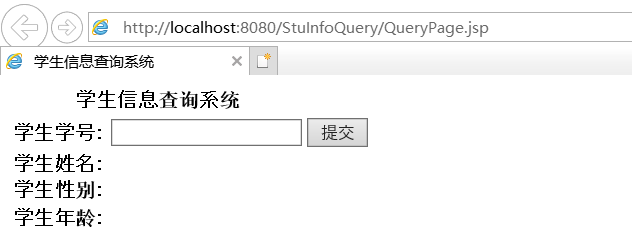
1. 编写JSP界面。在Eclipse菜单栏，选择 "File->New->Dynamic Web Project"，创建 StuInfoQuery项目：



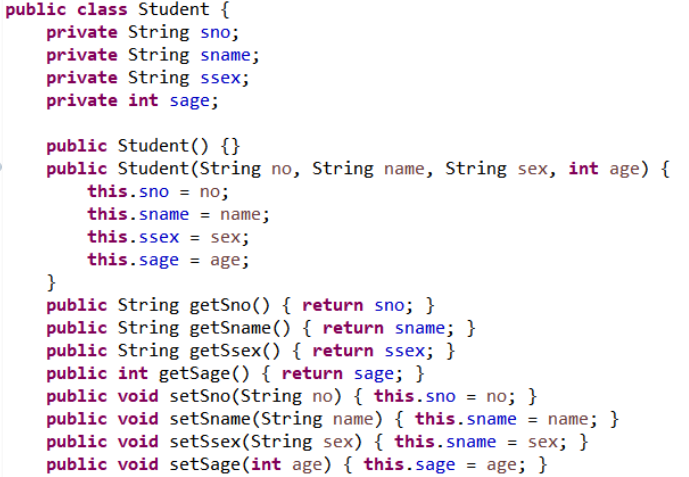
右键点击新建QueryPage.jsp文件，输入下图的代码。



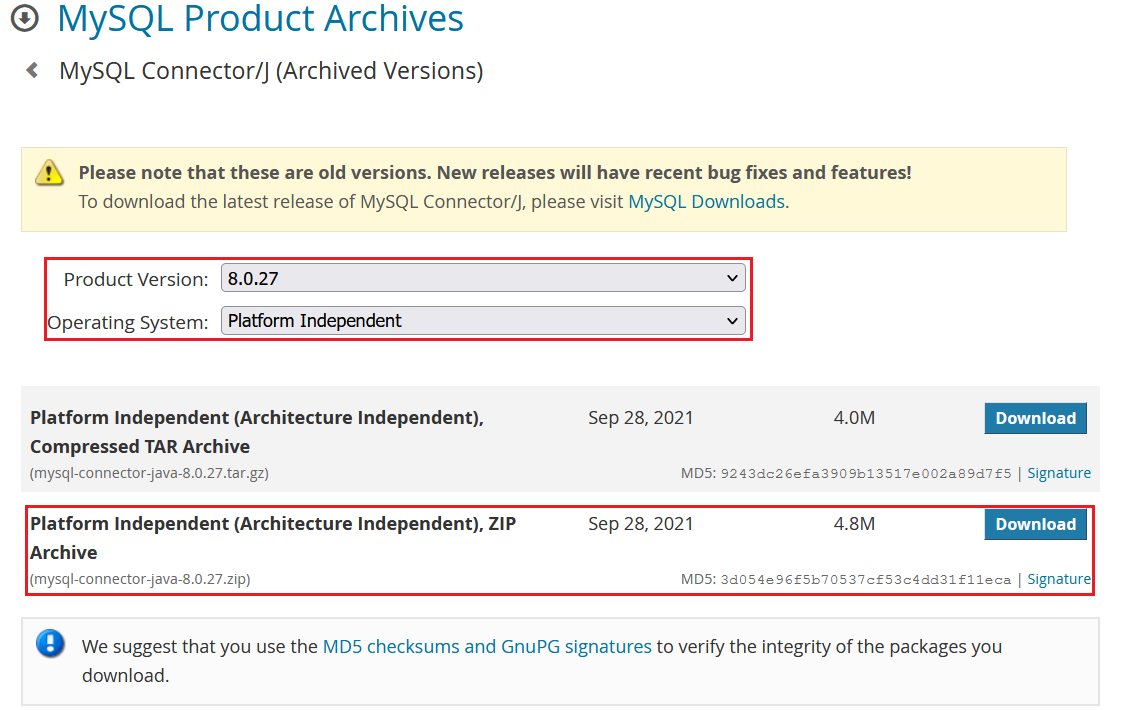
这一JSP界面显示如下：



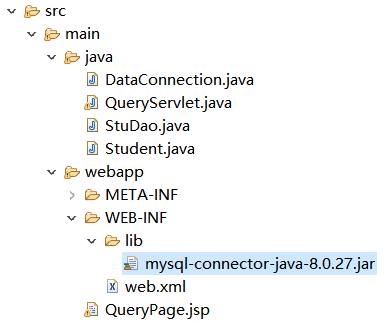
1. 在编写Servlet之前，编写一些辅助类，用于数据库连接、查询等。具体地，首先创建一个实体类Student，用来表示数据库中的学生信息，包含学号、姓名、性别、年龄这四个属性。



需要注意的是，为了通过JDBC连接到MySQL数据库，我们需要下载MySQL Connector/J驱动程序，帮助Java程序操作MySQL。MySQL官方给出的网址为[https://downloads.mysql.com/archives/c-j/，页面如下所示。我们要在Product](https://downloads.mysql.com/archives/c-j/，官网页面如下所示，在Product) Version一栏中选择与mysql --version匹配的版本，在Operating System一栏中选择Platform Independent，并下载Platform Independent (Architecture Independent), ZIP Archive这个文件包：



解压后，（对于我下载的版本来说）将mysql-connector-java-8.0.27.jar放到项目的lib文件夹中：



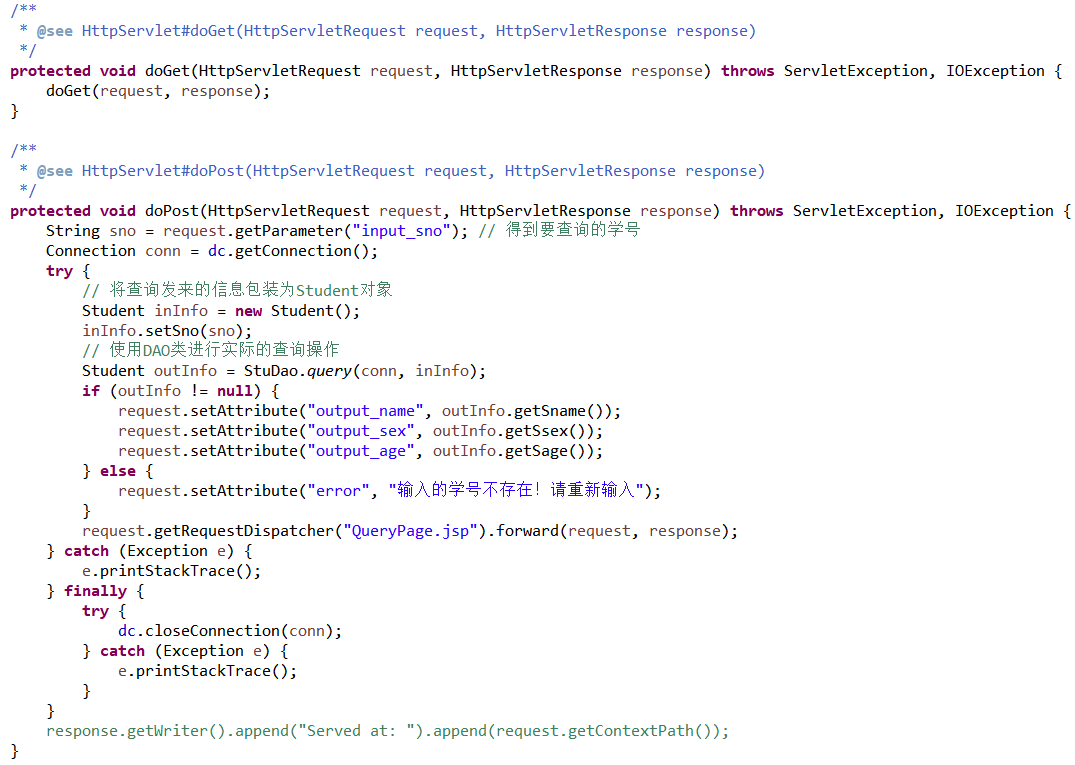
接着，我们编写代表数据库连接的类DataConnection，由于代码较多，这里仅展示部分内容。可见，类Connection中包含JDBC驱动名、数据库路径、数据库用户名、数据库密码这四个属性。在获取数据库连接时，我们先通过反射加载JDBC的Driver类，再向DriverManager的getConnection方法传入其他三个属性，获取数据库连接Connection对象。如果出现异常，则需要捕获并打印错误信息。特别地，在使用完成后，需要调用DataConnection类的closeConnection方法，关闭已经创建的连接。



之后再编写一个DAO类（Data Access Object），用于对数据库记录实体进行操作。由于这里就一个Student实体，所以只创建了一个StuDao类，将对学生信息的一些数据库相关的操作放在里面。这里只有一个查询query操作，它从已经创建的连接conn，根据用户输入的学生信息stu（即学号）从数据库中查询数据，得到的数据又被包装为一个Student对象并返回。



1. 之后编写Servlet。JSP代码中form表单的action属性，表示点击提交按钮后Get/Post请求要访问的程序，这一程序实际上就是某个Servlet类，它处理前台的请求、完成前后端的交互。

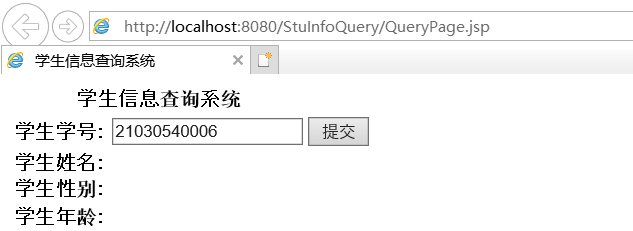


上图展示了QueryServlet类的核心操作doGet方法和doPost方法。这两个操作分别处理客户端发来的Get请求或Post请求，代码中以同一方式处理这两种请求。我们先将查询发过来的信息包装为一个Student对象inInfo，利用StuDao类进行实际的查询操作，如果查询结果outInfo不为空，则将对应学生的姓名、性别、年龄等信息，利用setAttribute方法存储在request对象中；否则返回错误提示。最后，用 request. getRequestDispatcher这一请求分发器对象包装QueryPage.jsp，并将当前的request传递给该JSP页面。

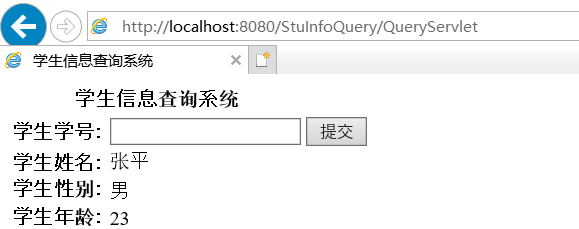
1. **实验结果**

完成上述编码工作后，检查实验结果如下：

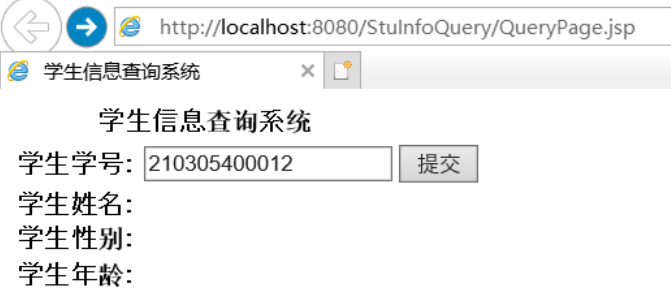
1. 客户端输出： 输入本人的学号21030540006，点击提交按钮：



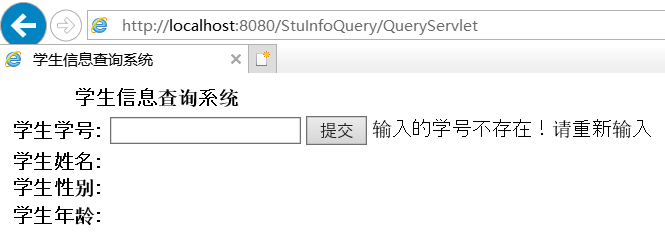
输出结果如下：



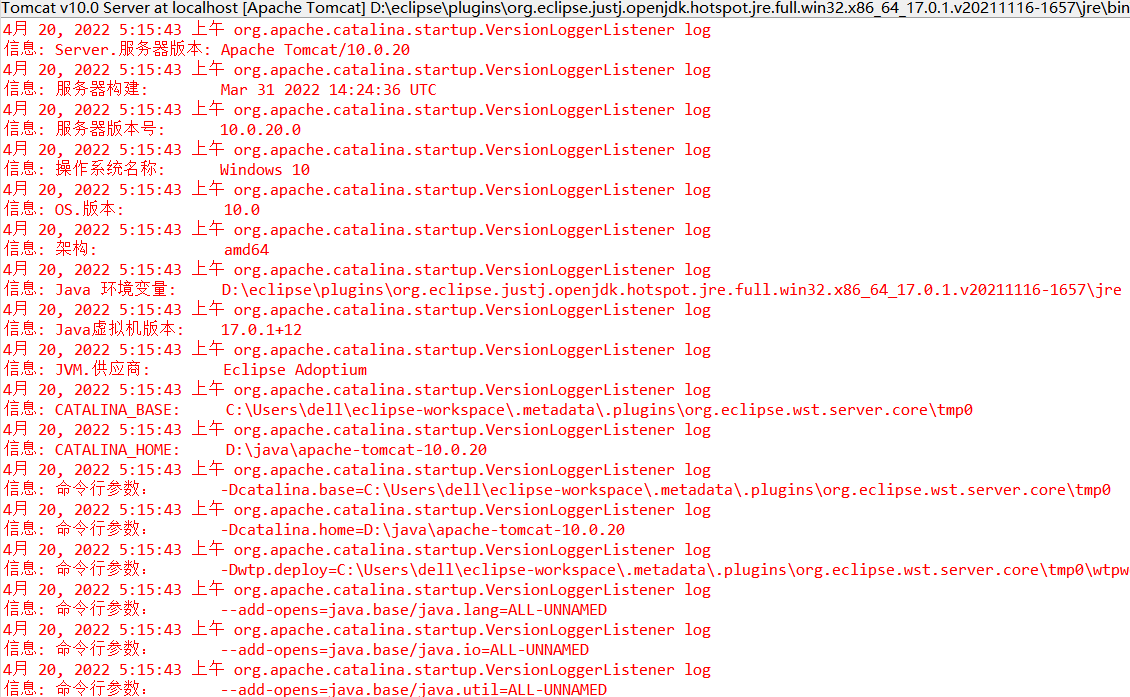
如果故意输入一个不存在的学号如21030540012：



页面中显示错误提示信息：



2. 服务端输出：



**八、 总结建议**

本次实验中，基本没有遇到什么问题。如果说有问题的话，还是由于不熟悉HTMP、CSS、JavaScript，实现不了什么好看的界面，有空要学一下前端界面设计。

本次实验中，采用了MVC开发模式——使用JSP实现表示层，让Servlet完成更深层次的用户请求处理工作、实现应用和控制层，用其他辅助Java类连接数据库、完成实际的数据查询。总体来说，这一技术方案的优点在于：在JSP页面中没有处理逻辑，它仅负责将请求转发给Servlet，并从Servlet中获取动态内容、插入到静态模板中。从而清楚地分离了表达和内容，更有利于复杂项目的开发。

JSP代码如下：

<%-- 输入学生的学号，输出学生信息 --%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>学生信息查询系统</title>

</head>

<body>

<div>

<form action=*"QueryServlet"* method=*"post"*>

<table>

<tr> <!-- colspan表示单元格可以横跨的列数 -->

<th colspan=*"2"*>学生信息查询系统</th>

</tr>

<tr>

<th>学生学号：</th>

<td><input type=*"text"* name=*"input\_sno"*></th>

<td><input type=*"submit"* name=*"submit\_btn"* value=*"提交"* /> </td>

<td>${error}</td>

</tr>

<tr>

<th>学生姓名：</th>

<td>${output\_name}</td>

</tr>

<tr>

<th>学生性别：</th>

<td>${output\_sex}</td>

</tr>

<tr>

<th>学生年龄：</th>

<td>${output\_age}</td>

</tr>

</table>

</form>

</div>

</body>

</html>

Student类如下：

**public** **class** Student {

**private** String sno;

**private** String sname;

**private** String ssex;

**private** **int** sage;

**public** Student() {}

**public** Student(String no, String name, String sex, **int** age) {

**this**.sno = no;

**this**.sname = name;

**this**.ssex = sex;

**this**.sage = age;

}

**public** String getSno() { **return** sno; }

**public** String getSname() { **return** sname; }

**public** String getSsex() { **return** ssex; }

**public** **int** getSage() { **return** sage; }

**public** **void** setSno(String no) { **this**.sno = no; }

**public** **void** setSname(String name) { **this**.sname = name; }

**public** **void** setSsex(String sex) { **this**.sname = sex; }

**public** **void** setSage(**int** age) { **this**.sage = age; }

}

DataConnection类如下：

**import** java.sql.DriverManager;

**import** java.sql.Connection;

**import** java.sql.SQLException;

**public** **class** DataConnection {

// JDBC驱动名

**private** String driver;

// 数据库路径

**private** String url;

// 数据库用户名

**private** String username;

// 数据库密码

**private** String password;

// 构造函数

**public** DataConnection(String driver, String url, String username, String password) {

**this**.driver = driver;

**this**.url = url;

**this**.username = username;

**this**.password = password;

}

// 获取数据库的连接

**public** Connection getConnection() {

Connection conn = **null**;

**try** {

// 反射加载JDBC的Driver类

Class.*forName*(driver);

// 通过DriverManager获取Connection对象

conn = DriverManager.*getConnection*(url, username, password);

} **catch** (ClassNotFoundException e) {

System.***out***.print("ClassNotFoundException: " + e);

} **catch** (SQLException e) {

System.***out***.print("SQLException: " + e);

} **catch** (Exception e) {

e.printStackTrace();

}

**return** conn;

}

// 关闭数据库的连接

**public** **void** closeConnection(Connection conn) **throws** SQLException {

**if** (conn != **null**) conn.close();

}

// 测试连接

**public** **static** **void** main(String[] args) {

DataConnection dc = **new** DataConnection("com.mysql.cj.jdbc.Driver",

"jdbc:mysql://localhost:3306/stu\_info?useSSL=false&useUnicode=true&characterEncoding=utf-8",

"root",

"wdmysqlmm123");

**try** {

**if** (dc.getConnection() != **null**)

System.***out***.println("数据库连接成功!");

} **catch** (Exception e) {

e.printStackTrace();

System.***out***.println("数据库连接失败!");

}

}

}

StuDao类如下：

**import** java.sql.Connection;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**public** **class** StuDao { // DAO类是处理数据库相关操作的类

**private** **static** String *sqlStr* = "SELECT \* FROM STUDENT WHERE Sno = ?;"; // Sno是学号

**public** **static** Student query(Connection conn, Student stu) **throws** SQLException {

// 准备SQL语句

PreparedStatement stmt = conn.prepareStatement(*sqlStr*);

stmt.setString(1, stu.getSno()); // 填充?区域

// 执行并获取结果集

ResultSet rs = stmt.executeQuery(); // 返回一个二维的结果集

Student info = **null**;

**if** (rs.next()) { // 查询数据

info = **new** Student(rs.getString("Sno"),

rs.getString("Sname"),

rs.getString("Ssex"),

Integer.*valueOf*(rs.getString("Sage"))

);

}

**if** (rs != **null**) rs.close();

**if** (stmt != **null**) stmt.close();

**return** info;

}

}

QueryServlet类如下：

**import** java.sql.Connection;

**import** jakarta.servlet.ServletException;

**import** jakarta.servlet.annotation.WebServlet;

**import** jakarta.servlet.http.HttpServlet;

**import** jakarta.servlet.http.HttpServletRequest;

**import** jakarta.servlet.http.HttpServletResponse;

**import** java.io.IOException;

/\*\*

\* 处理QueryPage.jsp发送的查询请求

\*/

**public** **class** QueryServlet **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** DataConnection dc;

/\*\*

\* **@see** HttpServlet#HttpServlet()

\*/

**public** QueryServlet() {

**super**();

dc = **new** DataConnection("com.mysql.cj.jdbc.Driver", "jdbc:mysql://localhost:3306/stu\_info?useSSL=false&useUnicode=true&characterEncoding=utf-8",

"root",

"wdmysqlmm123");

}

/\*\*

\* **@see** HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

doGet(request, response);

}

/\*\*

\* **@see** HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

String sno = request.getParameter("input\_sno"); // 得到要查询的学号

Connection conn = dc.getConnection();

**try** {

// 将查询发来的信息包装为Student对象

Student inInfo = **new** Student();

inInfo.setSno(sno);

// 使用DAO类进行实际的查询操作

Student outInfo = StuDao.*query*(conn, inInfo);

**if** (outInfo != **null**) {

request.setAttribute("output\_name", outInfo.getSname());

request.setAttribute("output\_sex", outInfo.getSsex());

request.setAttribute("output\_age", outInfo.getSage());

} **else** {

request.setAttribute("error", "输入的学号不存在！请重新输入");

}

request.getRequestDispatcher("QueryPage.jsp").forward(request, response);

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

**try** {

dc.closeConnection(conn);

} **catch** (Exception e) {

e.printStackTrace();

}

}

// response.getWriter().append("Served at: ").append(request.getContextPath());

}

}