目录

[1、 增加老师资料接口，不能增加重复的老师。 1](#_Toc23401)

[2、 增加学生资料接口，不能增加重复的学生，不能增加无老师的学生 2](#_Toc8670)

[3、 删除老师资料接口，不能删除有学生的老师。 3](#_Toc17262)

[4、 学生查询接口需要做数据库分页加载，可按老师名称进行查询。 5](#_Toc4320)

[访问地址1：查询所有,每页五条数据 5](#_Toc8024)

[访问地址2：按老师名称进行查询,每页五条数据 6](#_Toc10238)

[数据库：三张表分别为老师（teacher），学生（student），课程表（class） 7](#_Toc26770)

[Github： 7](#_Toc29491)

[Test 8](#_Toc6759)

###### 引入：

**该项目使用的是Maven+SpringBoot+Mybatis-Plus+MySQL（索引+触发器）**

**Pom.xml**

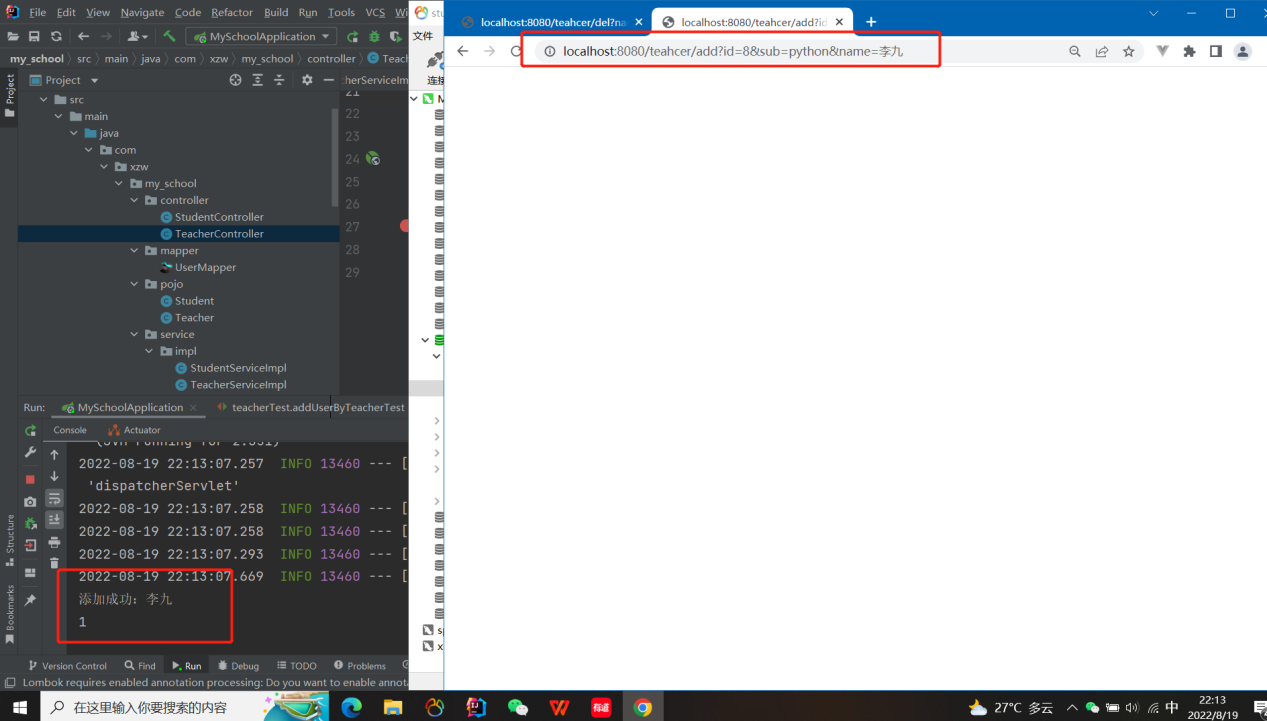
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
</dependency>  
<dependency>  
 <groupId>org.mybatis.spring.boot</groupId>  
 <artifactId>mybatis-spring-boot-starter</artifactId>  
 <version>2.2.2</version>  
</dependency>  
  
<dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <scope>runtime</scope>  
</dependency>  
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
</dependency>  
<dependency>  
 <groupId>com.baomidou</groupId>  
 <artifactId>mybatis-plus-boot-starter</artifactId>  
 <version>3.5.1</version>  
</dependency>  
<dependency>  
 <groupId>org.projectlombok</groupId>  
 <artifactId>lombok</artifactId>  
</dependency>

###### 增加老师资料接口，不能增加重复的老师。

访问地址：[http://localhost:8080/teahcer/add?id=4&sub=python&name=李九](http://localhost:8080/teahcer/add?id=8&sub=python&name=李九)

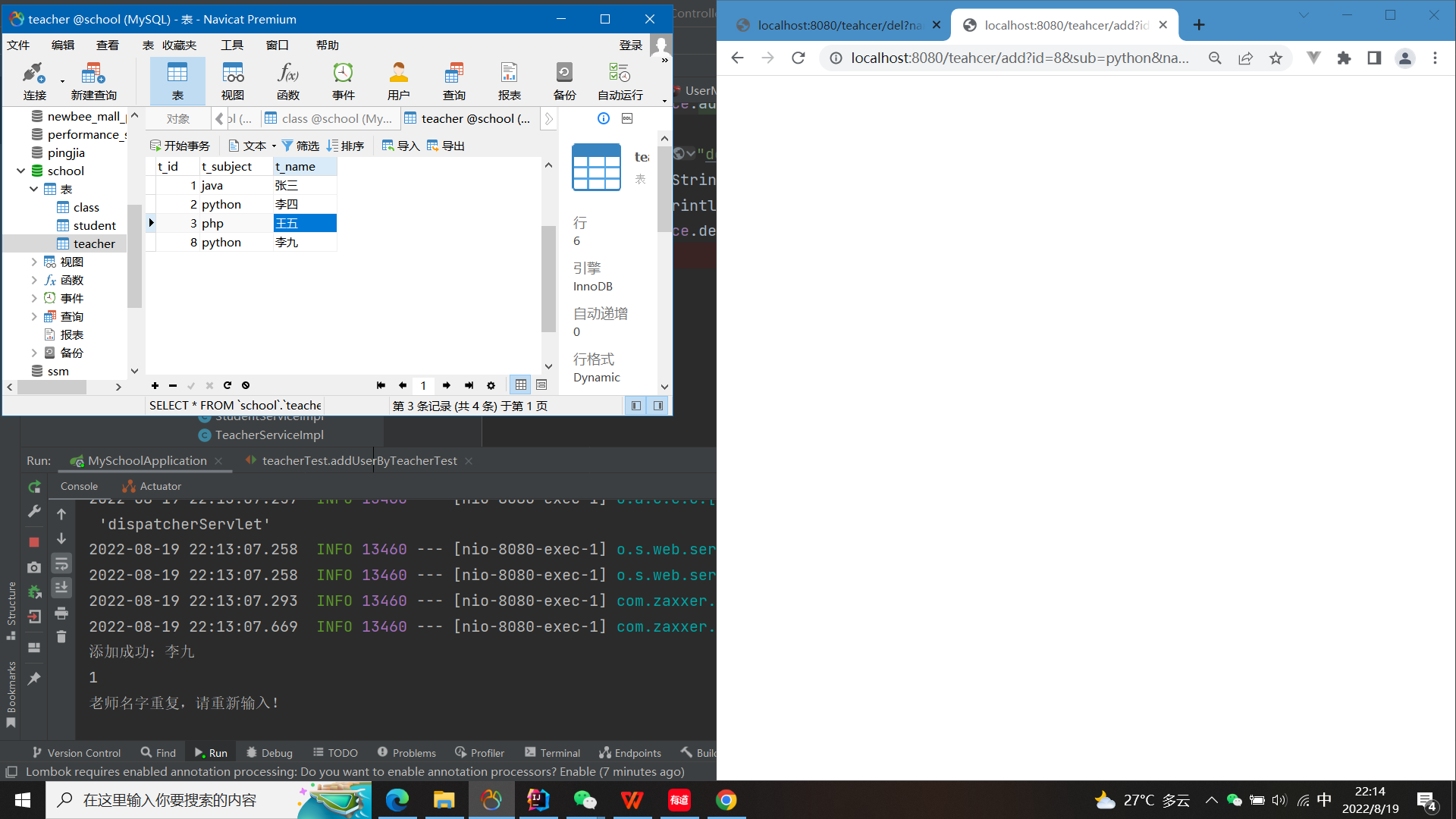
**第一次添加：**

**因为给老师名字（t\_nmae）添加了唯一索引，如果重复则抛出异常显示（老师名字重复，请重新输入！）**



**第二次添加重复数据：**

**（老师名字重复，请重新输入！）**



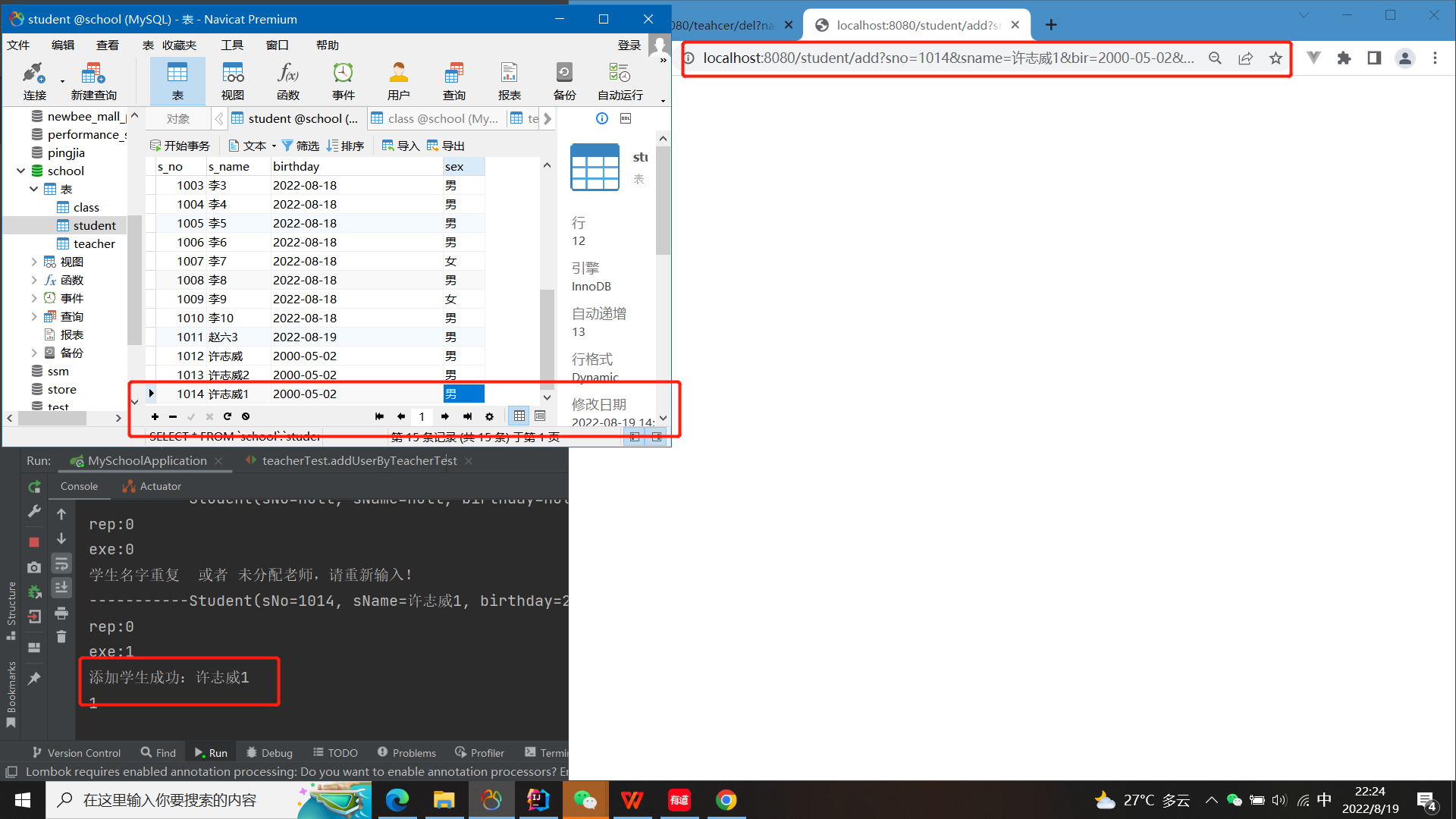
###### 增加学生资料接口，不能增加重复的学生，不能增加无老师的学生

访问地址：**localhost:8080/student/add?sno=1015&sname=许志威1&bir=2000-05-02&sex=男&tid=2**

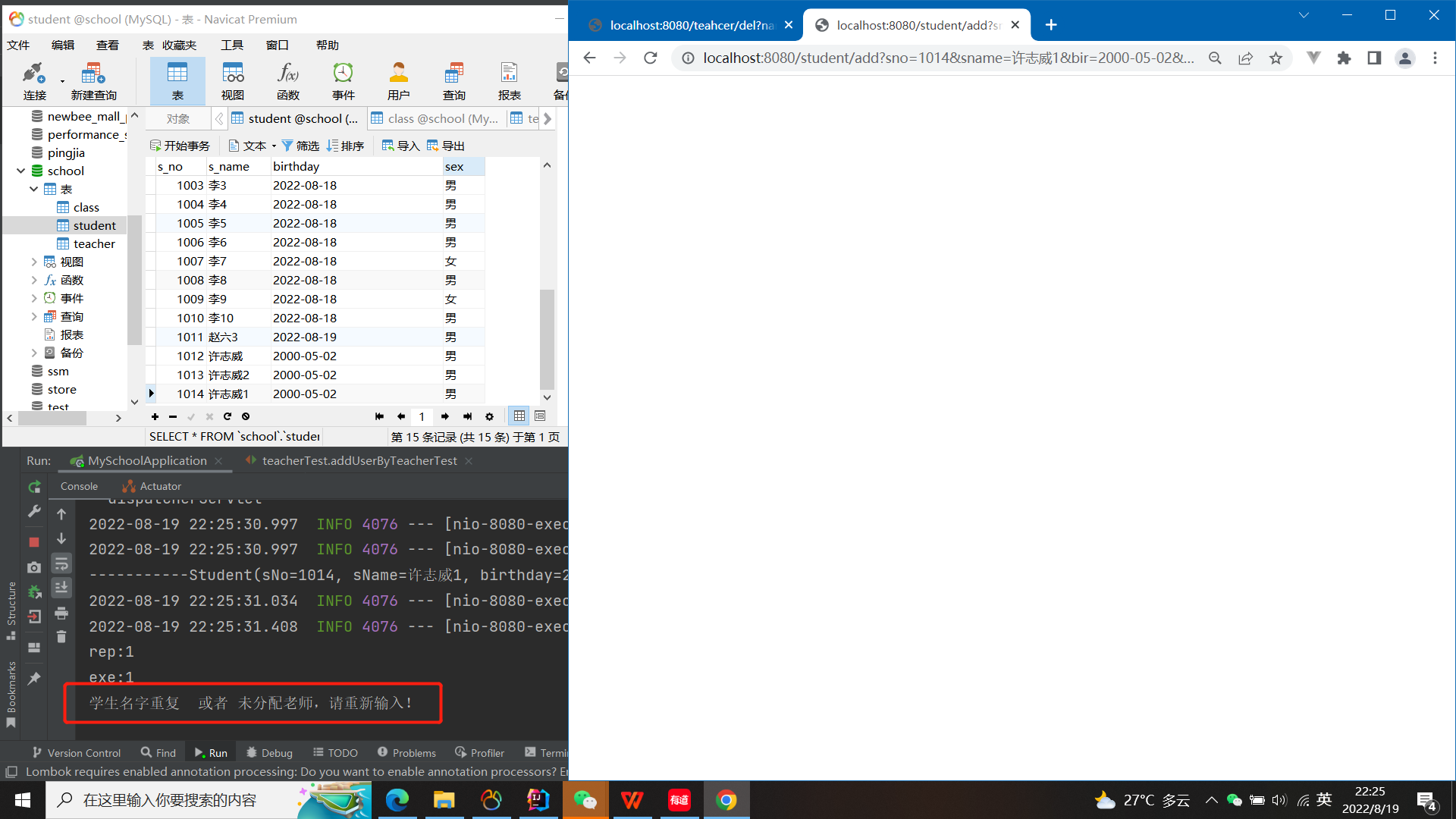
**原理：**

**访问参数为（学生对象，老师id），此处未使用索引来限制学生唯一索引。**

**先在class课程表中添加老师与学生的信息，然后再在student表中添加学生的信息**



**再运行一次：抛出异常：添加学生名字重复 或者 未分配老师，请重新输入！**



**关键代码：**

//添加学生  
@Override  
public void addStudent(Student student,Integer tid) {  
 try{  
 //第一步，添加课程（即先添加老师），此处为自定义sql语句  
 teacherMapper.addUserByClass(tid,student.getSNo());  
 studentMapper.insert (student);  
 }catch (Exception e){  
 System.*out*.println("学生名字重复 或者 未分配老师 或者老师id不存在，请重新输入！");  
 }  
}

###### 删除老师资料接口，不能删除有学生的老师。

访问网址：**<http://localhost:8080/teahcer/del?name=李九>**

**原理：**

**mysql有触发器，当执行delete时触发删除class课程中的与老师匹配的id,因此当删除老师的前，课程表中已经没有与该老师相关的信息**

**创建触发器：**

**CREATE TRIGGER delteacher**

**BEFORE DELETE ON teacher**

**FOR EACH ROW**

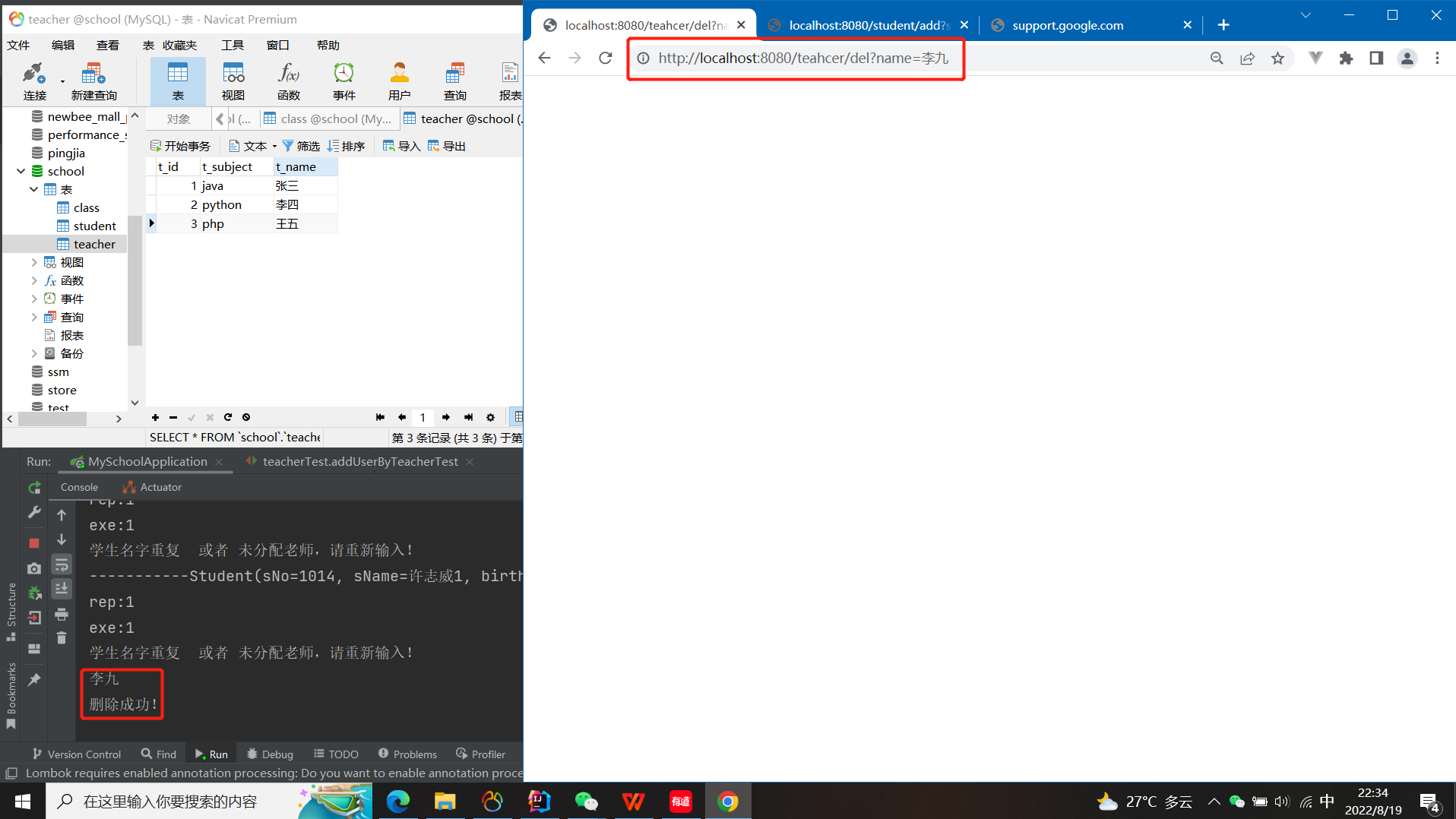
**BEGIN**

**DELETE FROM class WHERE class.t\_id=old.t\_id;**

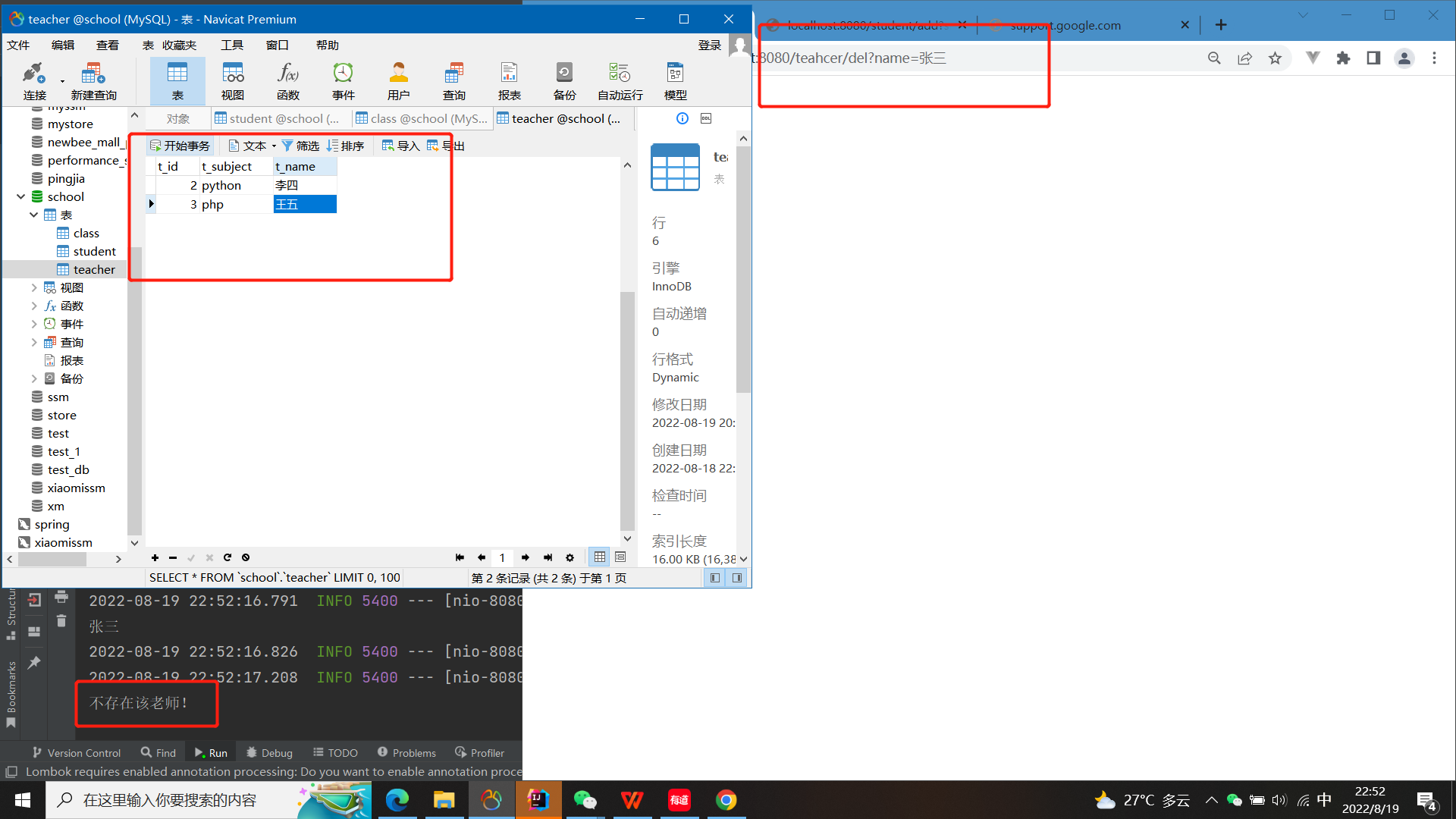
**END**

**SHOW TRIGGERS**

**DROP TRIGGER delteacher**



**再删一次同一个老师：不存在该老师！**



**关键代码：**

//删除老师  
@Override  
public void delTeacher(String name) {  
 try{  
 int tid=teacherMapper.selUserByTeacherName(name);  
 //mysql有触发器，当执行delete时触发删除class课程中的与老师匹配的id  
 teacherMapper.delUserByTeacher(tid);  
 }catch (Exception e){  
 System.*out*.println("老师工号错误或者不存在，请重新输入！");  
 }  
}

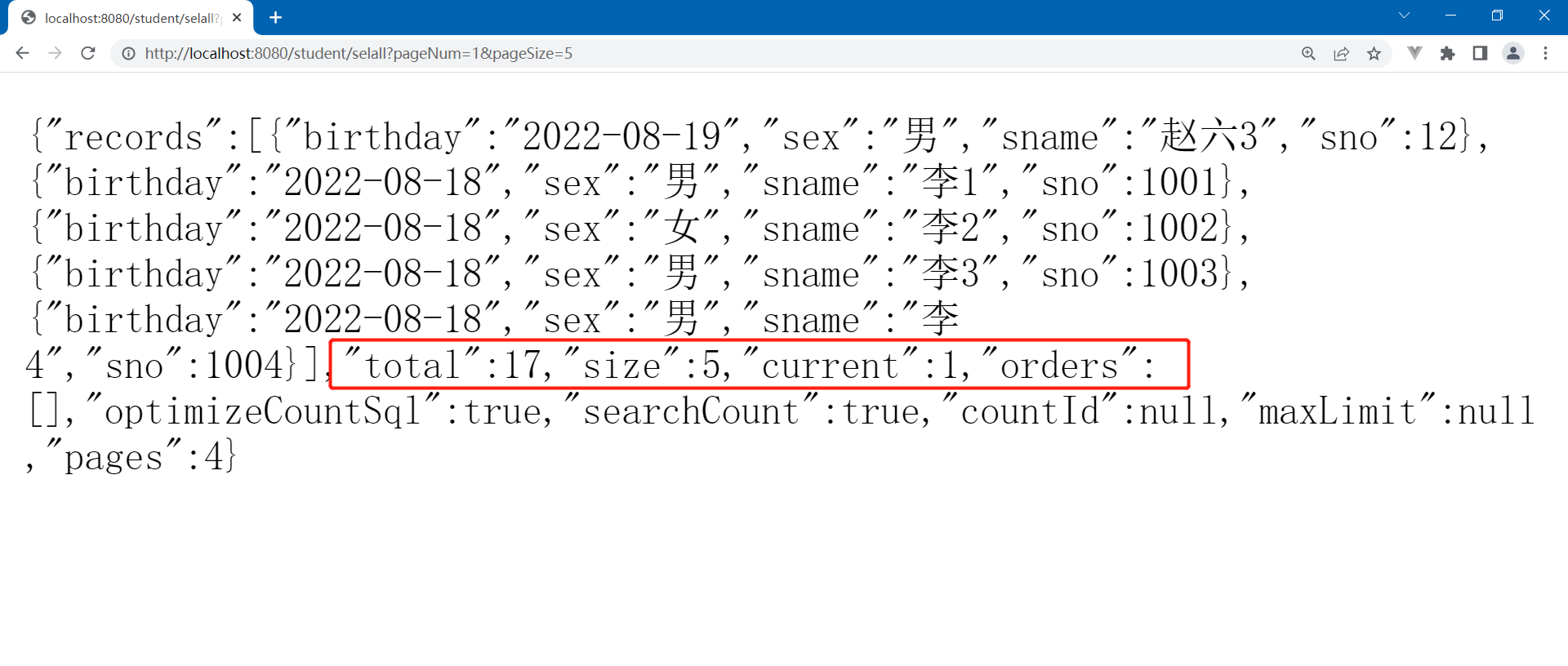
###### 学生查询接口需要做数据库分页加载，可按老师名称进行查询。

访问地址1：查询所有,每页五条数据

[localhost:8080/student/selall?pageNum=3&pageSize=5](http://localhost:8080/student/sel?pageNum=3&pageSize=5)

**原理：**

1. **添加分页插件依赖，配置yml**
2. **以（所在页数，每页最大值）为参数传入**
3. **返回PageInfo**



**关键代码：**

@Override  
public Page<Student> selStudent(Integer pageNum, Integer pageSize) {  
 //设置分页参数  
 Page<Student> page = new Page<>(pageNum, pageSize);  
 studentMapper.selectPage(page, null);  
 //获取分页数据  
 List<Student> list = page.getRecords();  
  
 list.forEach(System.*out*::println);  
 System.*out*.println("当前页：" + page.getCurrent());  
 System.*out*.println("每页显示的条数：" + page.getSize());  
 System.*out*.println("总记录数：" + page.getTotal());  
 System.*out*.println("总页数：" + page.getPages());  
 System.*out*.println("是否有上一页：" + page.hasPrevious());  
 System.*out*.println("是否有下一页：" + page.hasNext());  
 return page;  
}

访问地址2：按老师名称进行查询,每页五条数据

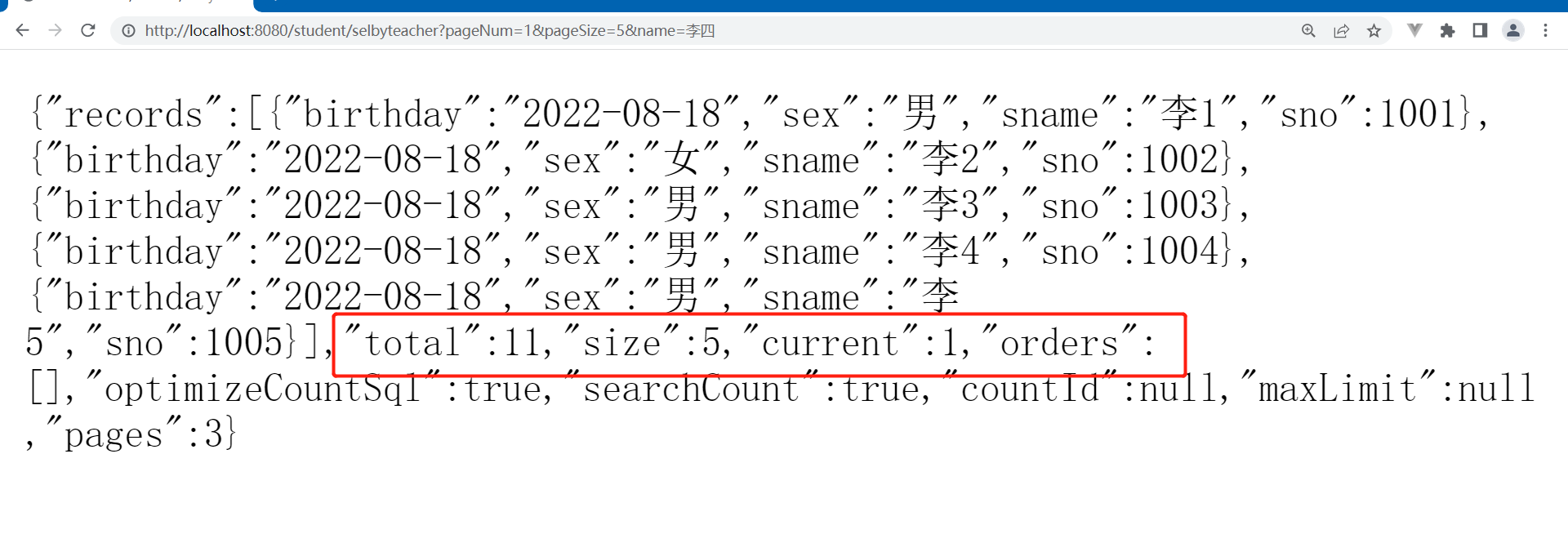
<http://localhost:8080/student/selbyteacher?pageNum=1&pageSize=5&name=李四>

**原理：**

**1、添加分页插件依赖，配置yml**

**2、以（所在页数，每页最大值，老师对应的tid）为参数传入**

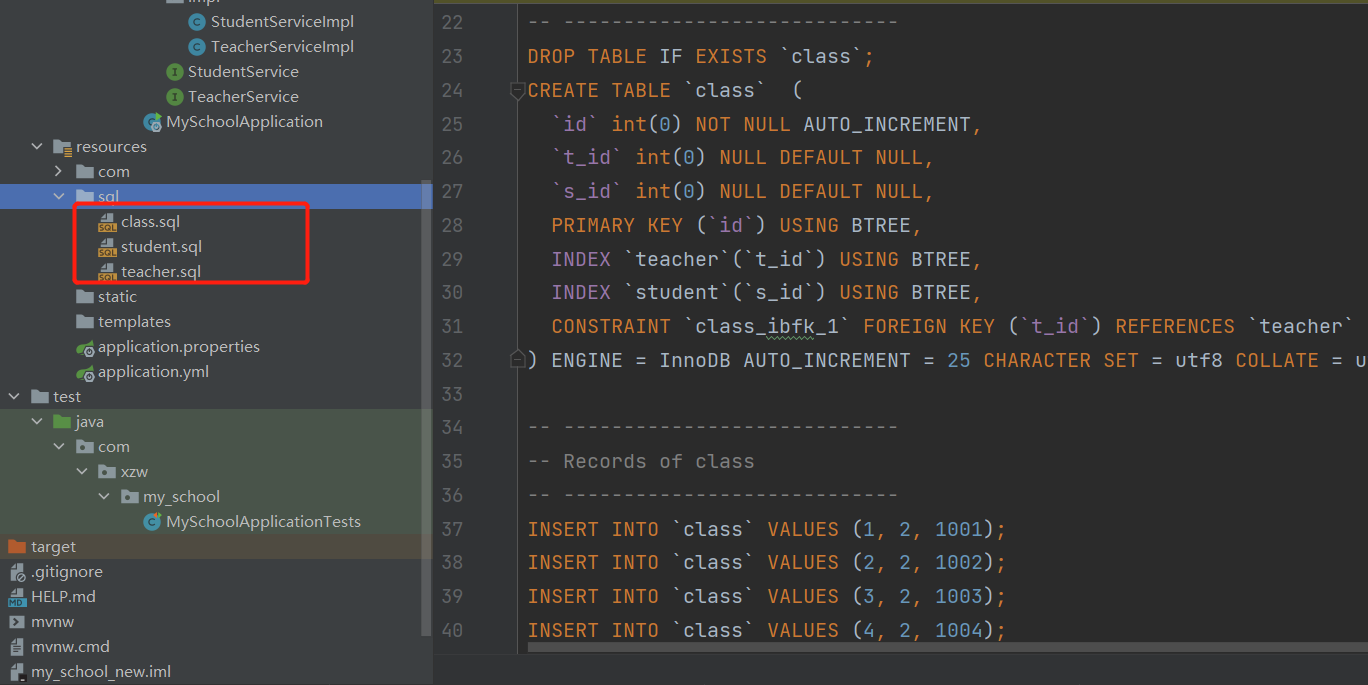
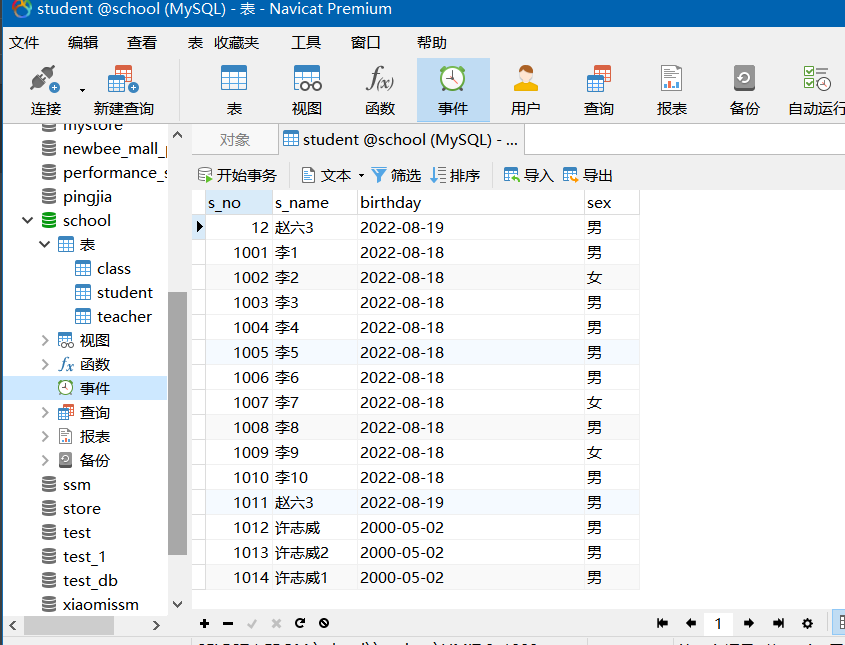
**3、返回PageInfo**



核心代码：

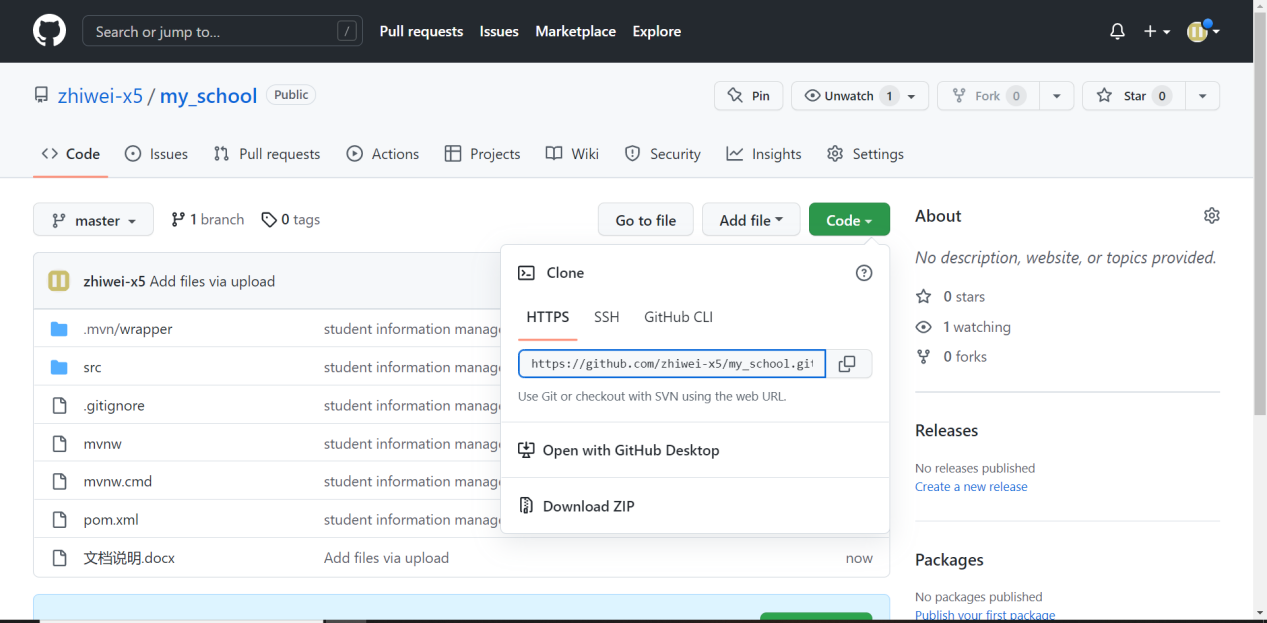
@Override  
 public Page<Student> selStudentByTeacher(Integer pageNum, Integer pageSize, String name) {  
 //设置分页参数  
 Page<Student> page = new Page<>(pageNum, pageSize);  
 int tid = teacherMapper.selUserByTeacherName(name);  
 System.*out*.println(tid);  
// IPage<Student> list= studentMapper.selUserByAllstudent\_teacher(tid);  
 studentMapper.selUserByAllstudent\_teacher(page, tid);  
  
 //获取分页数据  
 List<Student> list = page.getRecords();  
 list.forEach(System.*out*::println);  
 System.*out*.println("当前页：" + page.getCurrent());  
 System.*out*.println("每页显示的条数：" + page.getSize());  
 System.*out*.println("总记录数：" + page.getTotal());  
 System.*out*.println("总页数：" + page.getPages());  
 System.*out*.println("是否有上一页：" + page.hasPrevious());  
 System.*out*.println("是否有下一页：" + page.hasNext());  
  
 return page;  
 }

###### 数据库：三张表分别为老师（teacher），学生（student），课程表（class）



# Github：

https://github.com/zhiwei-x5/my\_school.git



# Test

