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)

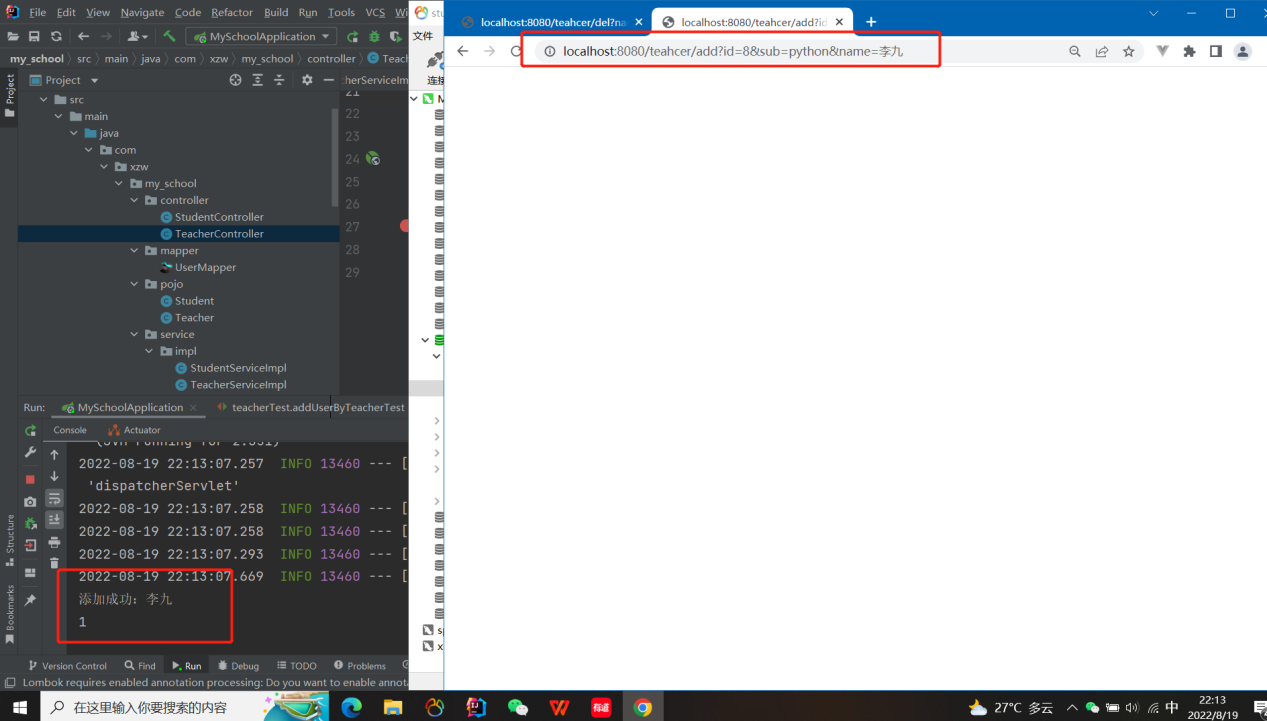
[Test 8](#_Toc6759)

###### 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=8&sub=python&name=李九>

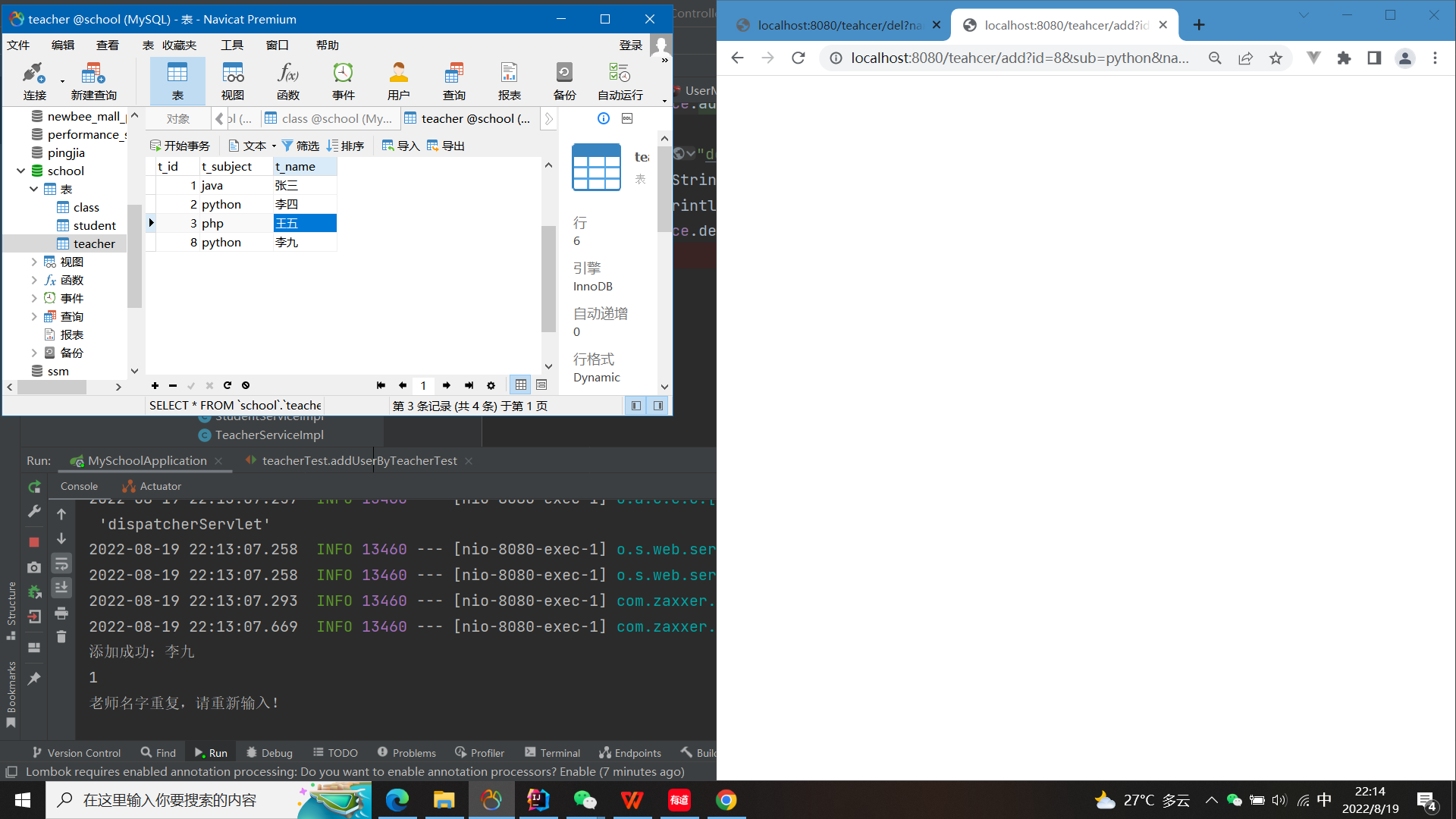
**第一次添加：**

**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！）**

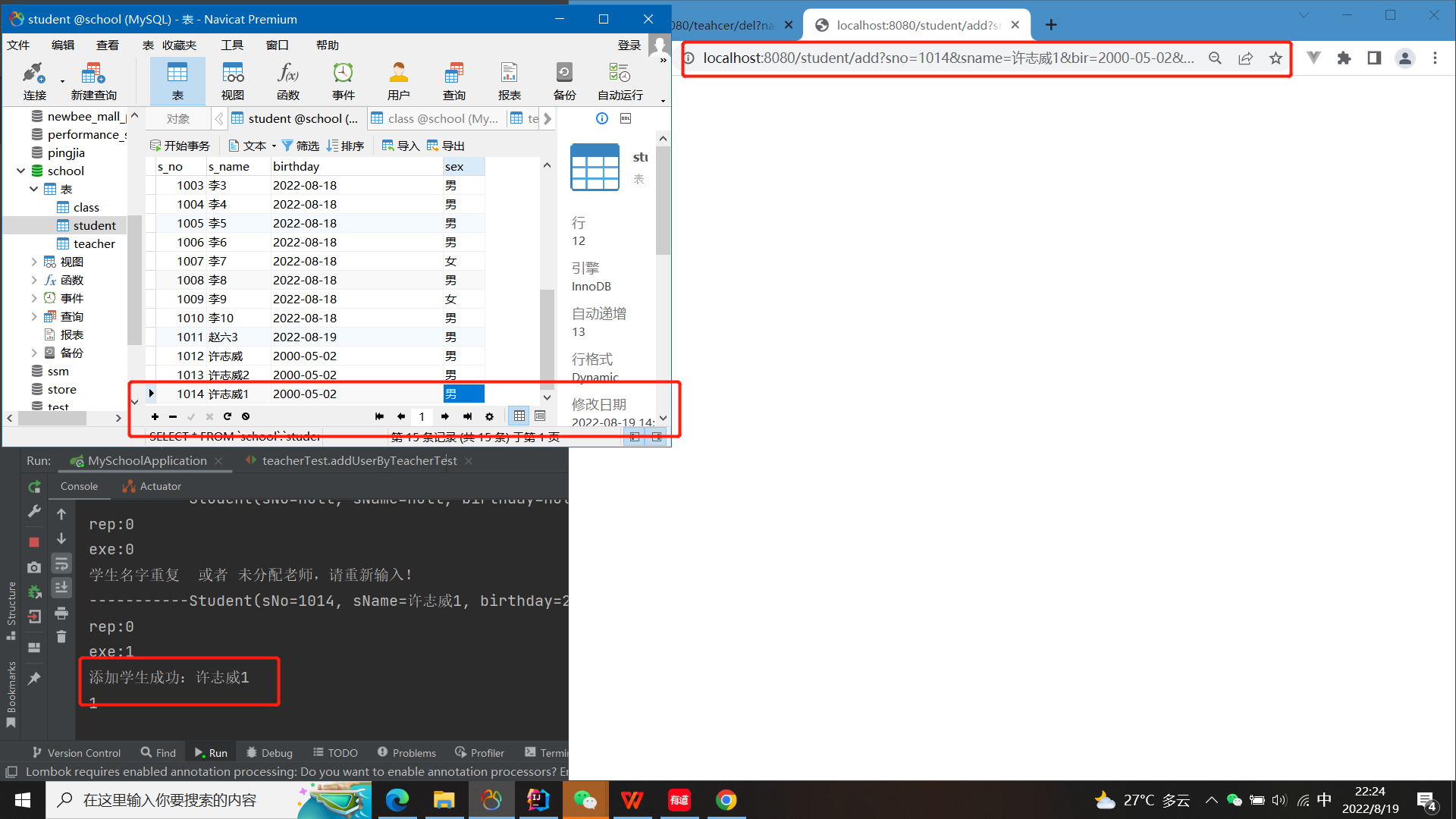


###### 增加学生资料接口，不能增加重复的学生，不能增加无老师的学生

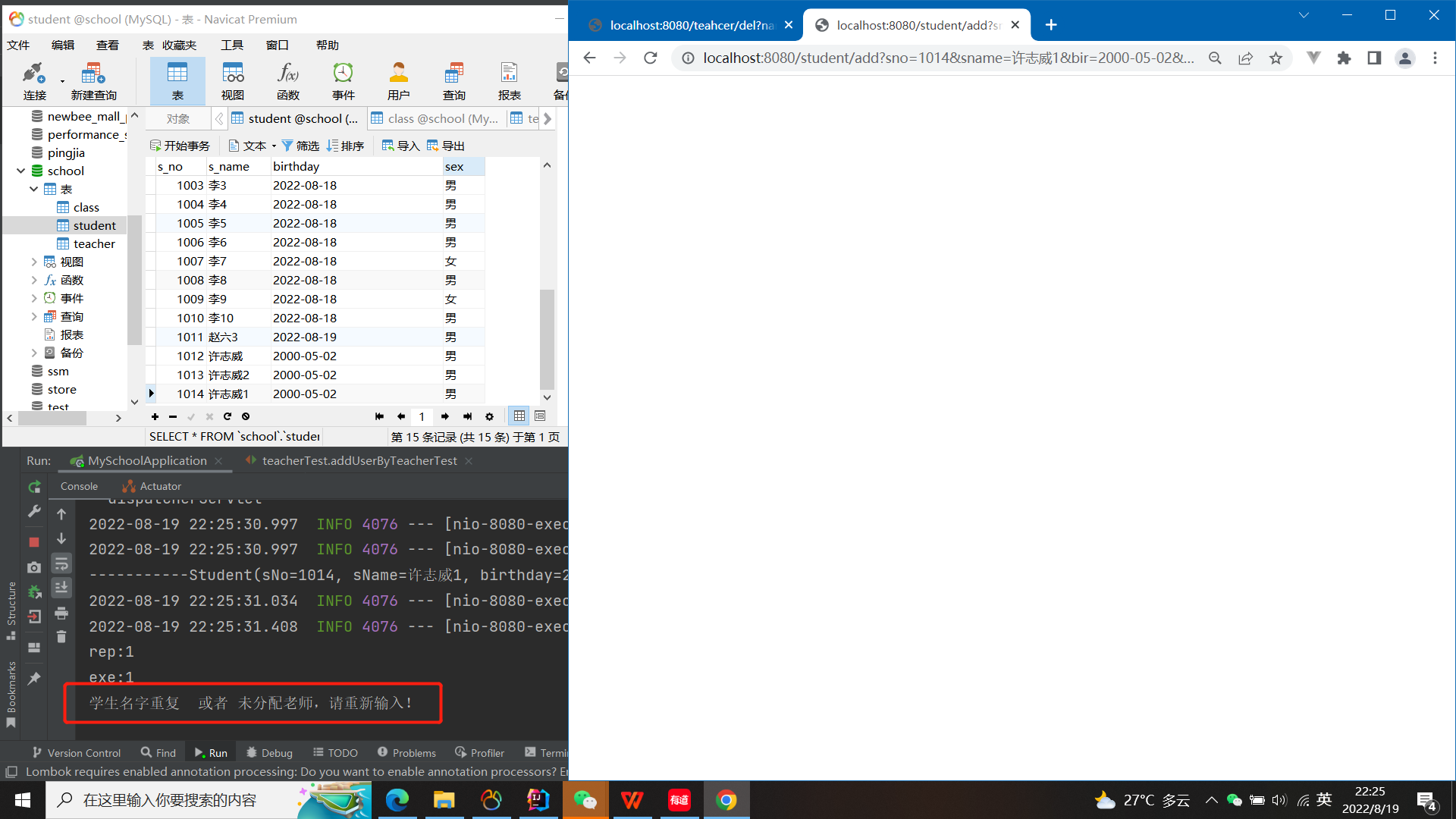
访问地址：**localhost:8080/student/add?sno=1014&sname=许志威1&bir=2000-05-02&sex=男&tid=1**

**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。**

1. **获取（student对象）学生名字（s\_name)并查询判断是否有重名**
2. **如果不重名即返回0，然后class课程表为该学生添加老师**
3. **查询class课程表中是否已经配有老师，有则查到的exe值不为0**
4. **判断exe！=0则添加学生**



**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！**

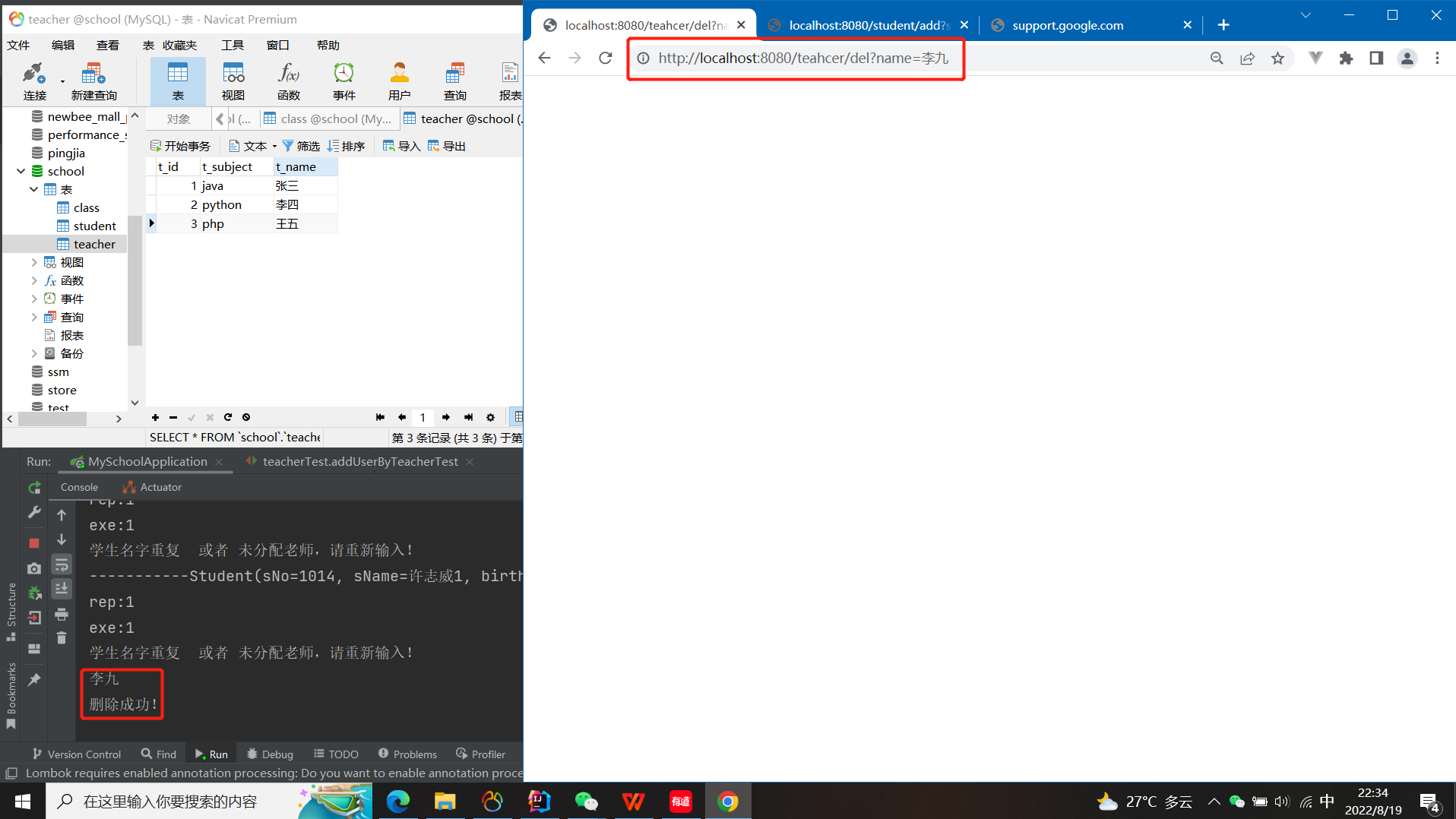


###### 删除老师资料接口，不能删除有学生的老师。

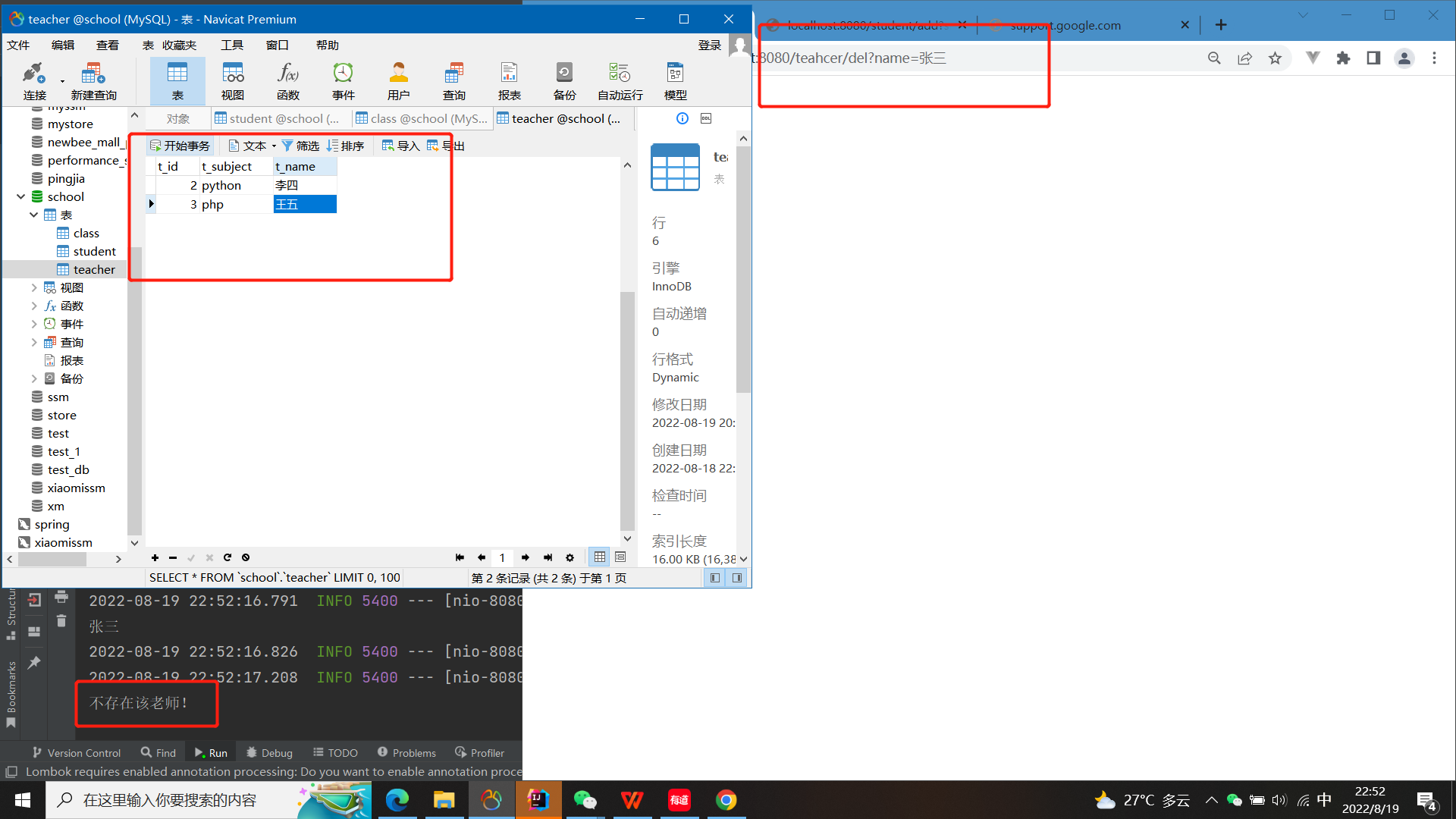
访问网址：**<http://localhost:8080/teahcer/del?name=李九>**

**原理：**

1. **输入需要删除老师的名字（t\_name),查询其tid**
2. **删除class课程表中老师的信息，以确保学生没有该老师**
3. **查询class中是否还存有老师的id，如果有返回的是总数==0则删除teacher的信息**



**再删一次同一个老师：不存在该老师！**



**关键代码：**

@Override  
public void delTeacher(String name) {  
 try{  
 //搜索需要删除老师的id  
 int res1 = userMapper.selUserByTeacherName(name);  
 //删除class中老师的信息，以确保学生没有该老师  
 userMapper.delUserByClassTeacher(res1);  
 //查询class中是否还存有老师的id，如果有返回的是总数  
 int res3 = userMapper.selUserByClassTeacherId(res1);  
 if(res3 == 0){  
 userMapper.delUserByTeacher(name);  
 System.*out*.println("删除成功！");  
 }  
 }catch (Exception e){  
 System.*out*.println("老师还有学生，不能删除！");  
 }  
}

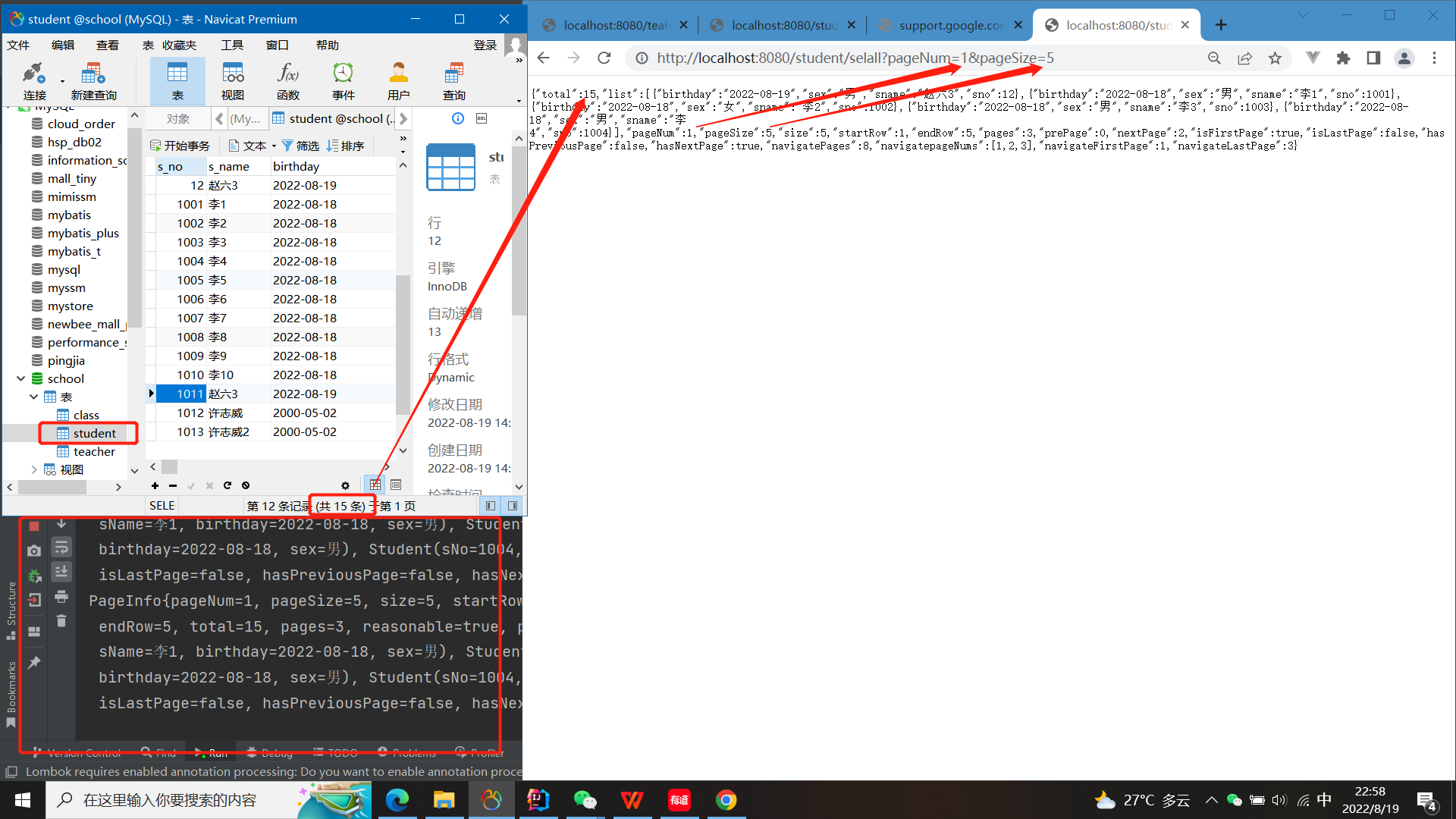
###### 学生查询接口需要做数据库分页加载，可按老师名称进行查询。

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 PageInfo selStudent(Integer pageNum, Integer pageSize) {  
 //分页工具，访问传入该页数，每页最大数  
 PageHelper.*startPage*(pageNum,pageSize);  
 List<Student> list= userMapper.selUserByAllstudent();  
 //用PageInfo对结果进行包装  
 PageInfo page = new PageInfo(list);  
 System.*out*.println(page);  
 return page;  
}

访问地址2：按老师名称进行查询,每页五条数据

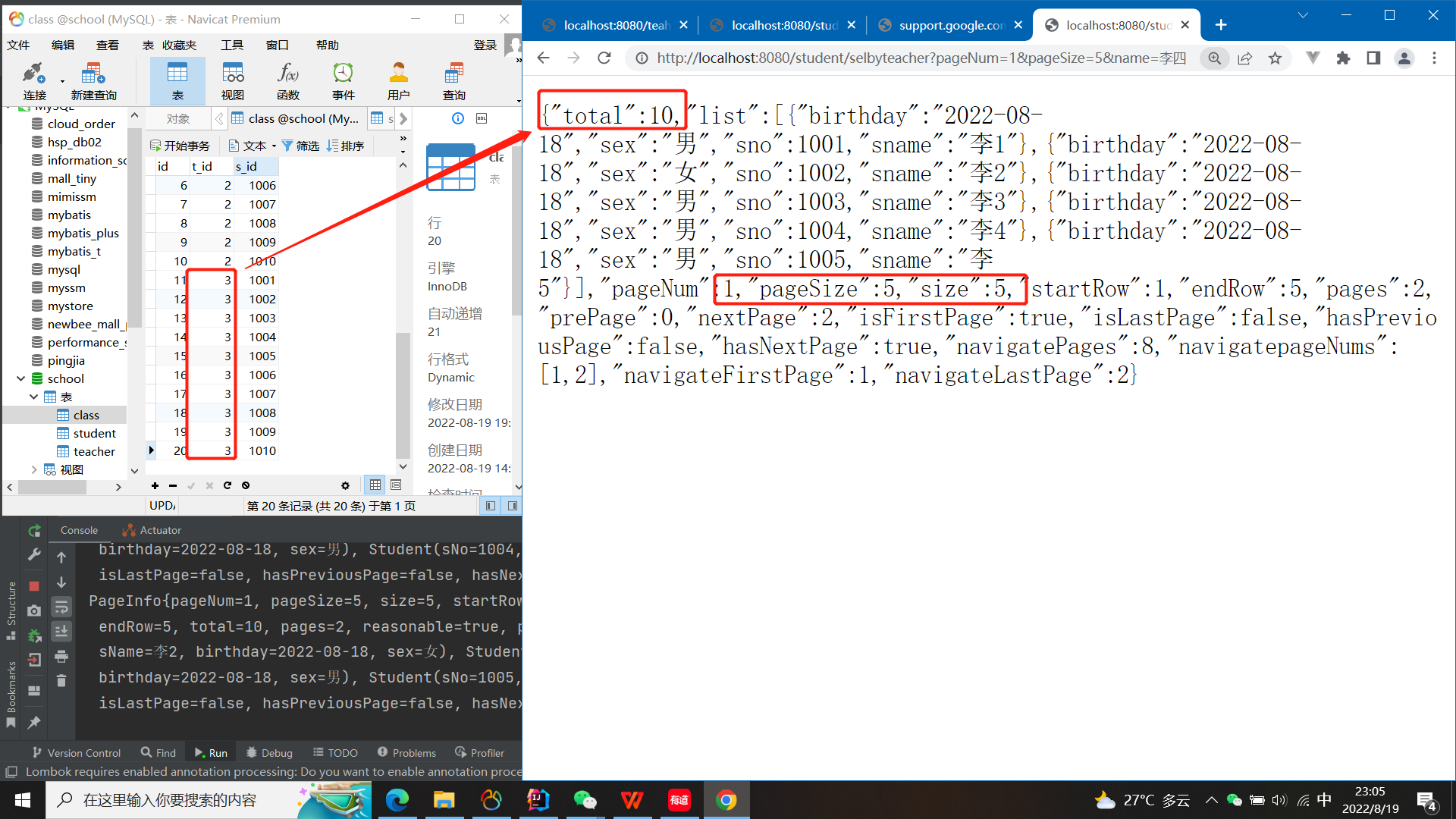
<http://localhost:8080/student/selbyteacher?pageNum=1&pageSize=5&name=李四>

**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**

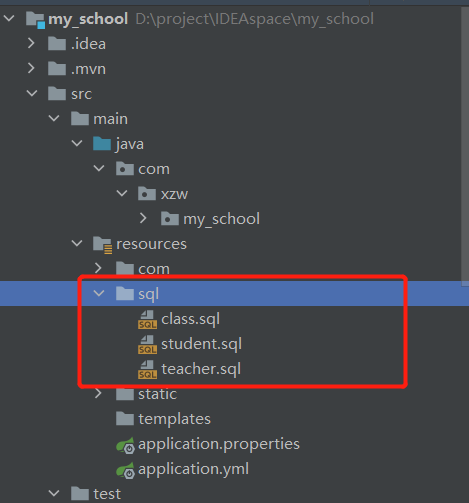
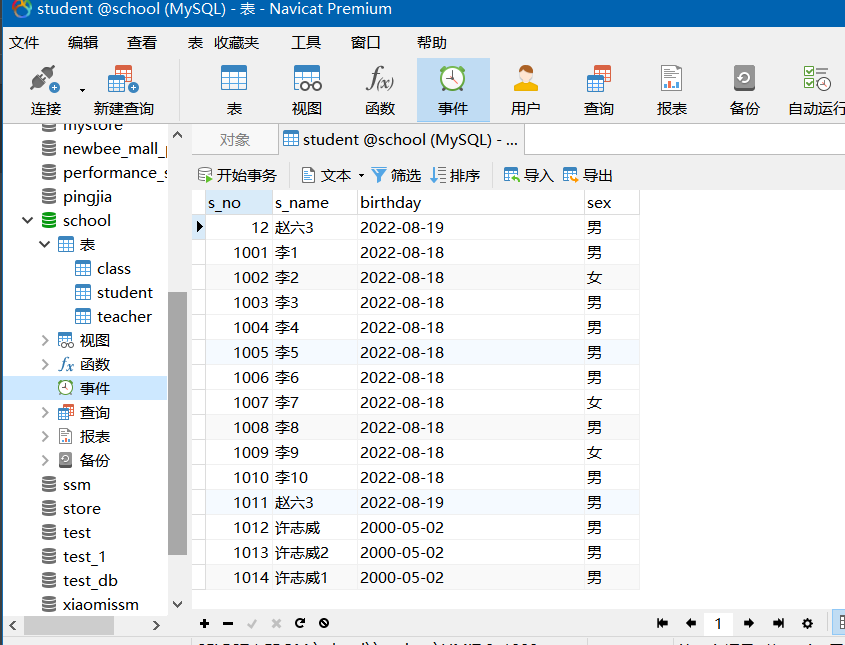
**3、返回PageInfo**



核心代码：

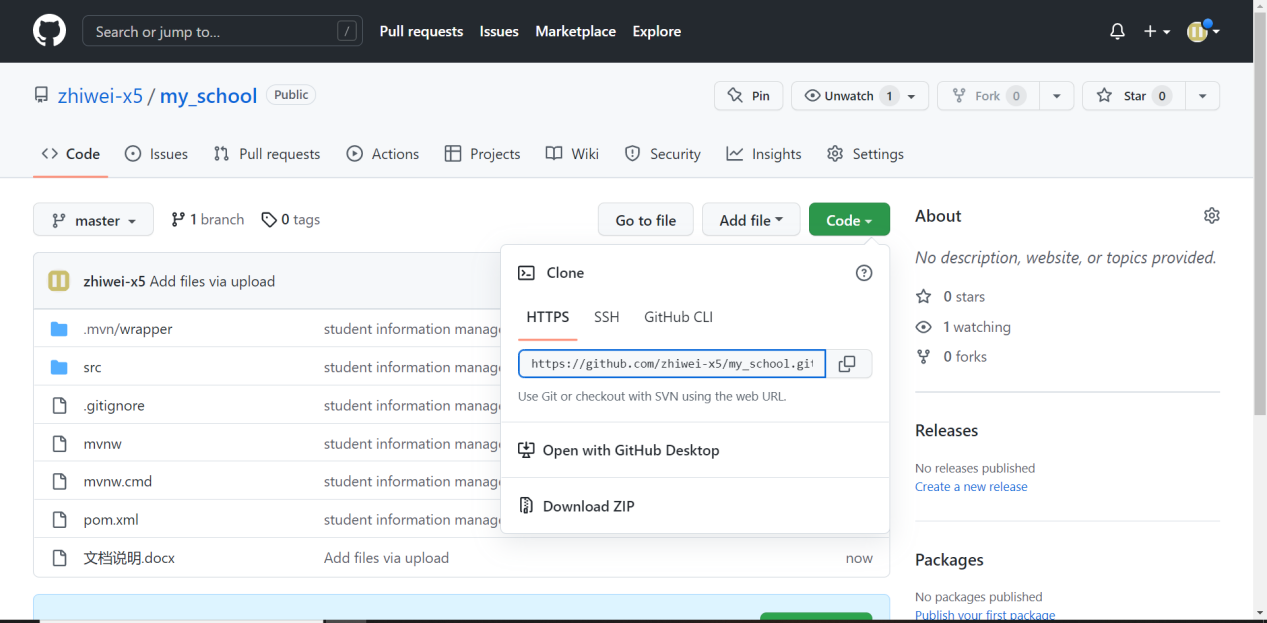
@Override  
public PageInfo selStudentByTeacher(Integer pageNum, Integer pageSize, String name) {  
 int tid=userMapper.selUserByTeacherName(name);  
 PageHelper.*startPage*(pageNum,pageSize);  
 List<Student> list= userMapper.selUserByAllstudent\_teacher(tid);  
 PageInfo page = new PageInfo(list);  
 System.*out*.println(page);  
 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

