存储过程

2025 年 5 月 9 日

一、主要任务：

针对 prereq 表, 实现一个 Procedure 来获取依赖于指定学院开设课程的所有课程

实现一个 Procedure

动态输入指定学院

递归获得所有依赖该指定学院课程的课程

客户端采用 Python 等来调用并显示结果

二、实现方法：

1. 用 create procedure（）begin.....end 可以创建存储过程，在其中可以使用临时表储存数据。

2. 在 java 中使用 mybatis 框架调用过程

3. 使用 with recursive 可以实现自我递归表。

三、成果展示：

1. 创建存储过程

```
DELIMITER //
CREATE PROCEDURE GetCourseByDepartment(IN deptName VARCHAR(50))
BEGIN
    with recursive rec_prereq(course_id, prereq_id) as (
    select course_id, prereq_id
    from prereq
    union
    select course_id rec_prereq.course_id,  prereq_id prereq.prereq_id from rec_prereq, prereq
    where rec_prereq.prereq_id = prereq.course_id
)

   select course_id,title,dept_name,credits from course
    where course.course_id in (
    select course.course_id from course  join rec_prereq on course.course_id = rec_prereq.course_id
                                        where rec_prereq.prereq_id in (select course_id from course
                                                    where dept_name=deptName));
END //
DELIMITER ;
```

使用 with recursive rec__prereq(course__id, prereq__id) as (select course__id, prereq__id from prereq union select rec__prereq.course__id, prereq.prereq__id

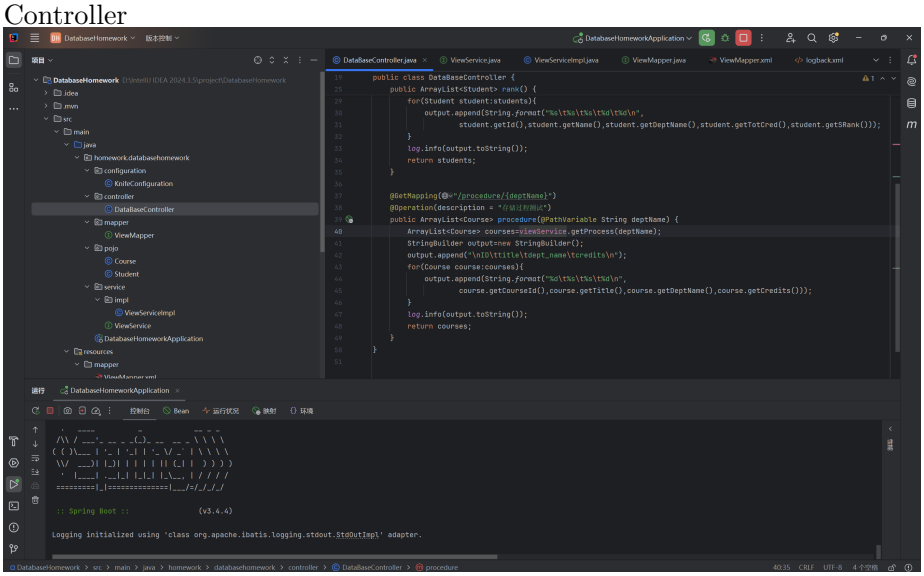
from rec_prereq, prereq where rec_prereq.prereq_id = prereq.course_id)
递归查询到所有课程与其前置课程的关系

select course_id,title,dept_name,credits from course where course.course_id in (select course.course_id from course join rec_prereq on course.course_id = rec_prereq.course_id where rec_prereq.prereq_id in (select course_id from course where dept_name=deptName));

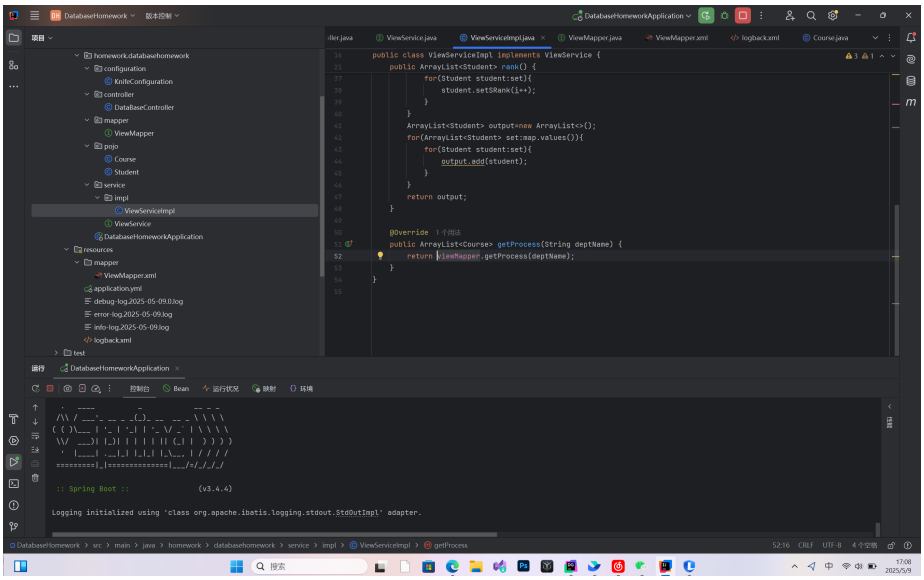
先在 course 表中查出所有输入的学院的课程 id，然后在课程与前置课程表中找出所有前置课程中有该学院的课课程 id，最后从 course 表中查出这些课程的信息

2.ssm 三层架构

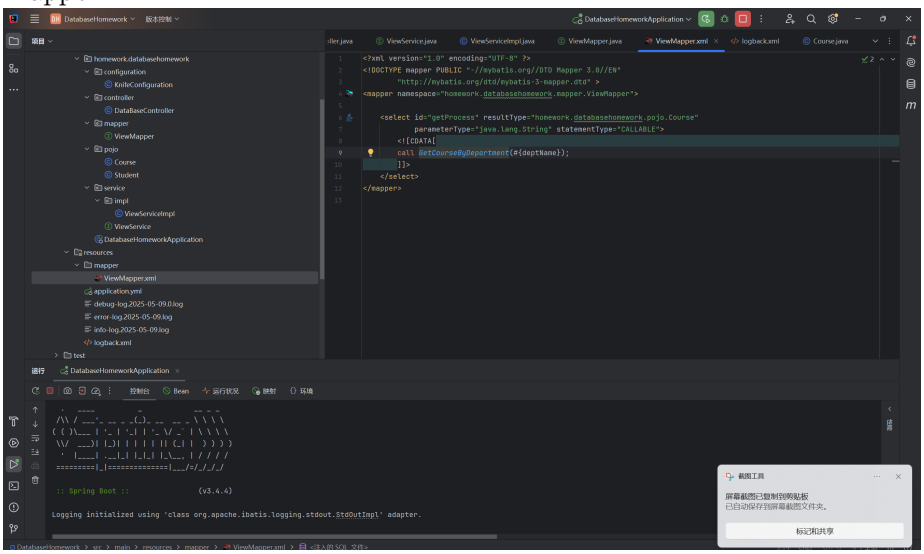
Controller



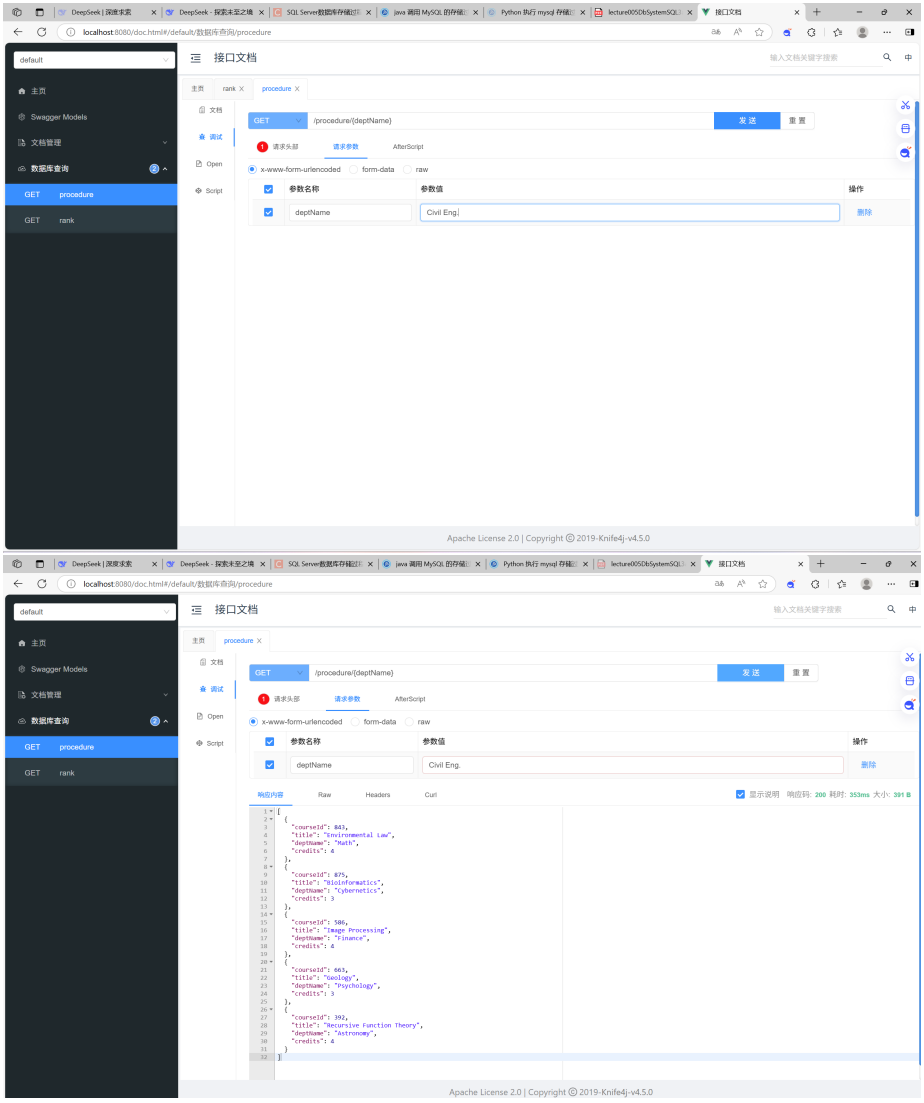
Service



Mapper

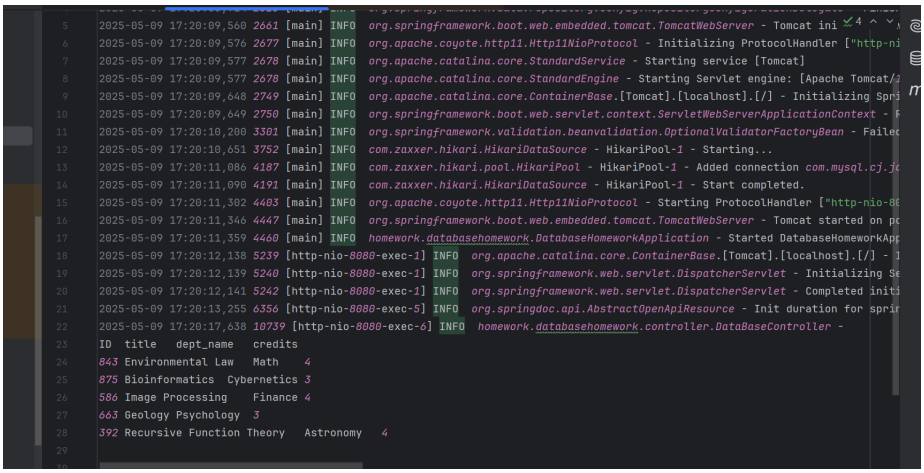


3. 使用 swagger 发送网络请求，查看返回信息



可见可正常返回数据

4. 日志中记录的数据



四、结论：

1. procedure 相比 function，可以没有返回值，只需要在内部做完需要做的事即可。

2. 递归表中先实现首次查找，再并上递归查找即可，无需使用多次迭代重复

3.mybatis 框架可以很容易地在 java 中操作数据库