

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 //
1 CREATE PROCEDURE GetCourseByDepartment(IN deptName VARCHAR(50))
2 BEGIN
3     with recursive rec_prereq(course_id, prereq_id) as (
4         select course_id, prereq_id
5         from prereq
6         union
7         select course_id rec_prereq.course_id, prereq_id prereq.prereq_id from rec_prereq, prereq
8         where rec_prereq.prereq_id = prereq.course_id
9     )
10
11     select course_id,title,dept_name,credits from course
12     where course.course_id in (
13         select course_id from course join rec_prereq on course.course_id = rec_prereq.course_id
14         where rec_prereq.prereq_id in (select course_id from course
15                                     where dept_name=deptName));
16 END //
17 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

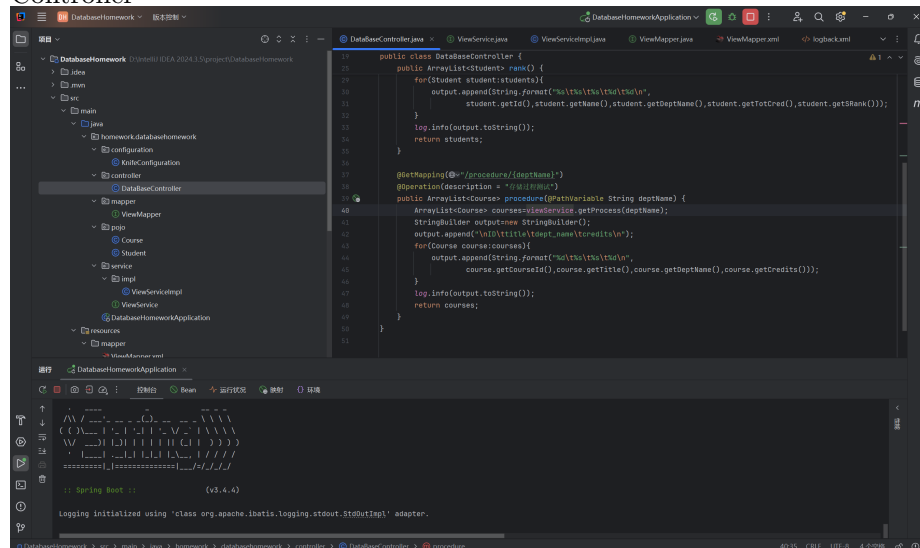
```

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));

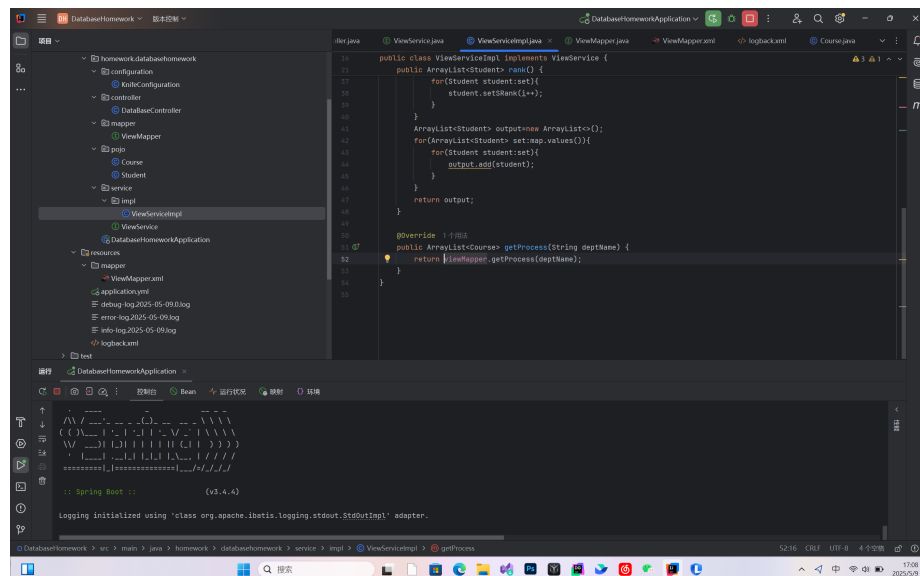
```

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

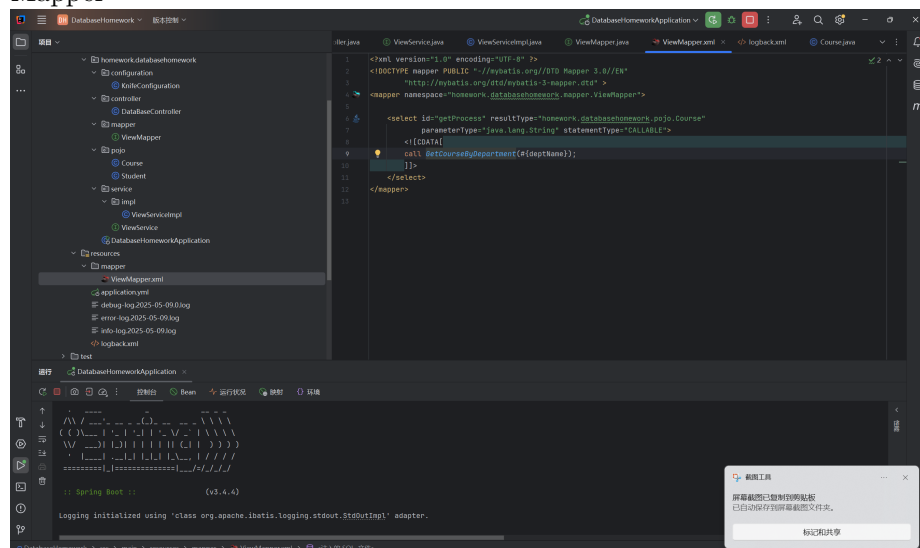
2.ssm 三层架构

Controller

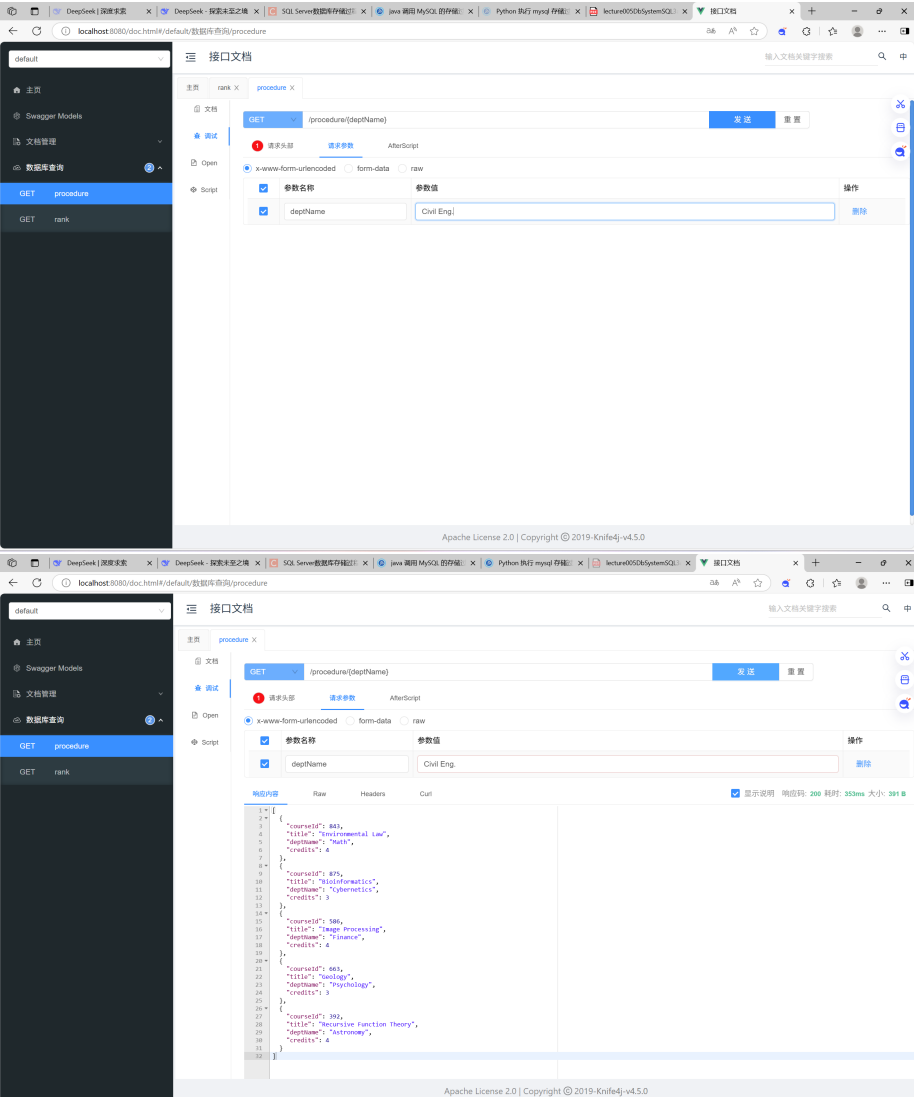
Service



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

```
5 2025-05-09 17:20:09,560 2661 [main] INFO org.springframework.boot.web.embedded.tomcat.TomcatWebServer - Tomcat ini
6 2025-05-09 17:20:09,576 2677 [main] INFO org.apache.coyote.http11.Http11NioProtocol - Initializing ProtocolHandler ["http-ni
7 2025-05-09 17:20:09,577 2678 [main] INFO org.apache.catalina.core.StandardService - Starting service [Tomcat]
8 2025-05-09 17:20:09,577 2678 [main] INFO org.apache.catalina.core.StandardEngine - Starting Servlet engine: [Apache Tomcat/9
9 2025-05-09 17:20:09,648 2749 [main] INFO org.apache.catalina.core.ContainerBase.[Tomcat].[localhost].[/] - Initializing Sprin
10 2025-05-09 17:20:09,649 2750 [main] INFO org.springframework.boot.web.servlet.context.ServletWebServerApplicationContext - F
11 2025-05-09 17:20:10,200 3301 [main] INFO org.springframework.validation.beanvalidation.OptionalValidatorFactoryBean - Fail
12 2025-05-09 17:20:10,651 3752 [main] INFO com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Starting...
13 2025-05-09 17:20:11,086 4187 [main] INFO com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection com.mysql.cj.jc
14 2025-05-09 17:20:11,090 4191 [main] INFO com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Start completed.
15 2025-05-09 17:20:11,302 4403 [main] INFO org.apache.coyote.http11.Http11NioProtocol - Starting ProtocolHandler ["http-nio-80
16 2025-05-09 17:20:11,346 4447 [main] INFO org.springframework.boot.web.embedded.tomcat.TomcatWebServer - Tomcat started on po
17 2025-05-09 17:20:11,359 4460 [main] INFO homework.databasehomework.DatabaseHomeworkApplication - Started DatabaseHomeworkApp
18 2025-05-09 17:20:12,138 5239 [http-nio-8080-exec-1] INFO org.apache.catalina.core.ContainerBase.[Tomcat].[localhost].[/] - I
19 2025-05-09 17:20:12,139 5240 [http-nio-8080-exec-1] INFO org.springframework.web.servlet.DispatcherServlet - Initializing Se
20 2025-05-09 17:20:12,141 5242 [http-nio-8080-exec-1] INFO org.springframework.web.servlet.DispatcherServlet - Completed initi
21 2025-05-09 17:20:13,255 6356 [http-nio-8080-exec-5] INFO org.springframeworkdoc.api.AbstractOpenApiResource - Init duration for sprin
22 2025-05-09 17:20:17,638 10739 [http-nio-8080-exec-6] INFO homework.databasehomework.controller.DataBaseController -
23 ID title dept_name credits
24 843 Environmental Law Math 4
25 875 Bioinformatics Cybernetics 3
26 586 Image Processing Finance 4
27 663 Geology Psychology 3
28 392 Recursive Function Theory Astronomy 4
29
```

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