针对prerea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

MySQL：

drop procedure if exists course\_cnt\_dept;

delimiter $$

create procedure course\_cnt\_dept(

in dept\_name varchar(20),out cnt integer)

deterministic

reads sql data

begin

select count(course\_id) into cnt

from course

where dept\_name=course.dept\_name;

select course\_id,dept\_name from course where dept\_name=course.dept\_name;

end $$

delimiter ;

set @dept\_name="Mech. Eng.";

call course\_cnt\_dept(@dept\_name,@cnt);

select @cnt;

set @dept\_name="Astronomy";

call course\_cnt\_dept(@dept\_name,@cnt);

select @cnt;

drop procedure if exists course\_pre\_dept;

delimiter $$

create procedure course\_pre\_dept(

in dept\_name varchar(20),out cnt integer)

deterministic

reads sql data

begin

select count(course\_id) into cnt

from course

where dept\_name=course.dept\_name;

select dept\_name,prereq.course\_id

from course,prereq

where dept\_name=course.dept\_name and course.course\_id=prereq.course\_id;

end $$

delimiter ;

set @dept\_name="Mech. Eng.";

call course\_pre\_dept(@dept\_name,@cnt);

select @cnt;

set @dept\_name="Astronomy";

call course\_pre\_dept(@dept\_name,@cnt);

select @cnt;

递归查询所有学院课程的所有前导课程

with recursive rec\_prereq(dept\_name,course\_id,prereq\_id) as(

select course.dept\_name,course.course\_id,prereq.prereq\_id

from course,prereq

where dept\_name=course.dept\_name and course.course\_id=prereq.course\_id

union

select rec\_prereq.dept\_name,rec\_prereq.course\_id,prereq.prereq\_id

from rec\_prereq,prereq

where rec\_prereq.course\_id=prereq.course\_id

)

select \* from rec\_prereq;

Python：

**import** pymysql

*# 连接数据库*

conn = pymysql.connect(host=**'127.0.0.1'**, user=**'myuser'**, password=**'Lp200211'**, database=**'dbsclab2018'**, charset=**'utf8'**)

cursor = conn.cursor()

*# 获取用户输入的课程 ID*

Dept\_name = input(**'请输入学院名 dept\_name：'**)

*# 调用递归查询前导课程的procedure*

*# 查询直接前导课程*

cursor.callproc(**'pre\_course\_proc'**, (dept\_name ,))

retdata = cursor.fetchall()

*# 关闭游标和连接*

cursor.close()

conn.close()

*# 输出结果*

**for** row **in** retdata:

Print ( row )