

《数据库系统实验》

实验报告

题目	(实验 4)
姓名	TRY
学号	
班级	计科 X班

一. 实验环境:

操作系统: windows

图形界面: mysql3.7.31, mysql workbench

二. 实验内容与完成情况:

2.1 创建数据库&表, 插入数据

创建一个名为 jxgl 的数据库, 包含表 student\course\sc, 步骤同实验 3。(且这里添加了实验 5 的部分数据, 代码省略)

代码如下:

```
create database jxgl character set utf8 collate utf8_general_ci; #这样才可以输入中文
```

```
create table student
(sno char(7) primary key,
sname varchar(20),
sage int,
ssex varchar(2),
sdept varchar(2));
```

```
create table course
(cno char(2) primary key,
cname varchar(20),
cpno char(2) default null,
ccredit int);
```

```
create table sc
(sno char(7),
cno char(2),
grade int,
primary key(sno,cno),
foreign key(sno) references student(sno),
foreign key(cno) references course(cno)); #这里不同于书上! 要在()内写上参考的变量名
```

```

insert into student
values('2005001','钱横',18,'男','Cs'),
('2005002','王林',19,'女','Cs'),
('2005003','李民',20,'男','Is'),
('2005004','赵欣然',16,'女','Ma'),
('2005005','杨洋',25,'男','Cs');    #自己补充添加的!

```

```

insert into course
values('1','数据库系统','5',4),
('2','数学分析',null,2),
('3','信息系统导论','1',3),
('4','操作系统原理','6',3),
('5','数据结构','7',4),
('6','数据处理基础',null,4),
('7','C 语言','6',3);

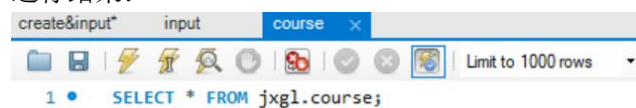
```

```

insert into sc
values('2005001','1',87),
('2005001','2',67),
('2005001','3',90),
('2005001','4',80),    #自己添加的
('2005001','7',98),    #自己添加的
('2005002','2',95),
('2005003','3',88),
('2005004','3',90),
('2005005','3',80),
('2005013','3',90),
('2005014','3',95);

```

运行结果:



Result Grid				
	cno	cname	cpno	ccredit
1	1	数据库系统	5	4
2	2	数学分析	NULL	2
3	3	信息系统导论	1	3
4	4	操作系统原理	6	3
5	5	数据结构	7	4
6	6	数据处理基础	NULL	4
7	7	C语言	6	3
*	NULL	NULL	NULL	NULL

create&input* input course sc

Limit to 1000 rows

1 • SELECT * FROM jxgl.sc;

Result Grid Filter Rows: Edit: Export

	sno	cno	grade
▶	2005001	1	87
	2005001	2	67
	2005001	3	90
	2005002	2	95
	2005003	3	88
*	NULL	NULL	NULL

Result Grid Filter Rows: Edit:

	sno	sname	sage	ssex	sdept
▶	2005001	钱横	18	男	Cs
	2005002	王林	19	女	Cs
	2005003	李民	20	男	Is
	2005004	赵欣然	16	女	Ma
	2005005	杨洋	25	男	Cs
	2005010	赵青江	18	男	Cs
	2005011	张丽萍	19	女	Ch
	2005012	陈景欢	20	男	Is
	2005013	陈婷婷	16	女	Ph
	2005014	李军	16	女	Ph
*	NULL	NULL	NULL	NULL	NULL

2.2 使用 SQL 语句进行下面的查询：

2.2.1 检索年龄大于 23 岁的男学生的学号和姓名

代码：

```
select sno,sname
from student
where sage>23 and ssex='男';
```

运行结果：

create&input* select student sc

Limit to 10

1 • select sno,sname
2 from student
3 where sage>23 and ssex='男';
4
5 • select sname
6 from student natural join sc
7 where ssex='女';

Result Grid Filter Rows: Edit:

	sno	sname
▶	2005005	杨洋
*	NULL	NULL

2.2.2 检索至少选修一门课的女学生姓名

代码:

```
select distinct sname
from student natural join sc
where ssex='女';
```

运行结果:

The screenshot shows a database query editor with a tab labeled 'select'. The SQL query is entered in the editor:

```
4
5 • select distinct sname
6   from student natural join sc
7   where ssex='女';
8
9 • select cno
10  from course
```

Below the editor, the 'Result Grid' is displayed, showing the results of the query. The results are as follows:

sname
王林
赵欣然
张丽萍
陈婷婷
李军

2.2.3 检索王林不学的课程的课程号

代码:

```
select cno
from course
where cno not in
(select cno
from student natural join sc
where sname='王林');
```

运行结果:

The screenshot shows the 'Result Grid' of a database query. The results are as follows:

cno
1
4
5
6
7
NULL

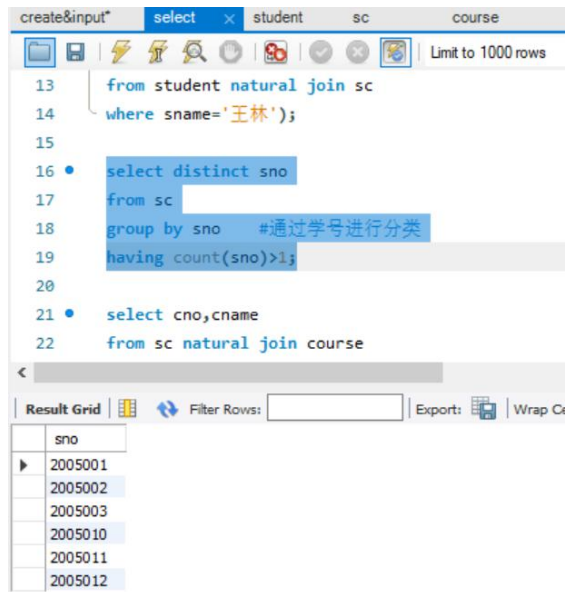
2.2.4 检索至少选修两门课的学生学号

代码:

```
select distinct sno
from sc
```

group by sno #通过学号进行分类
having count(sno)>1;

运行结果:



The screenshot shows a SQL IDE with a query editor and a results grid. The query editor contains the following SQL code:

```
13 from student natural join sc
14 where sname='王林';
15
16 • select distinct sno
17 from sc
18 group by sno #通过学号进行分类
19 having count(sno)>1;
20
21 • select cno,cname
22 from sc natural join course
```

The results grid shows the output of the query, with columns 'sno' and 'cname'.

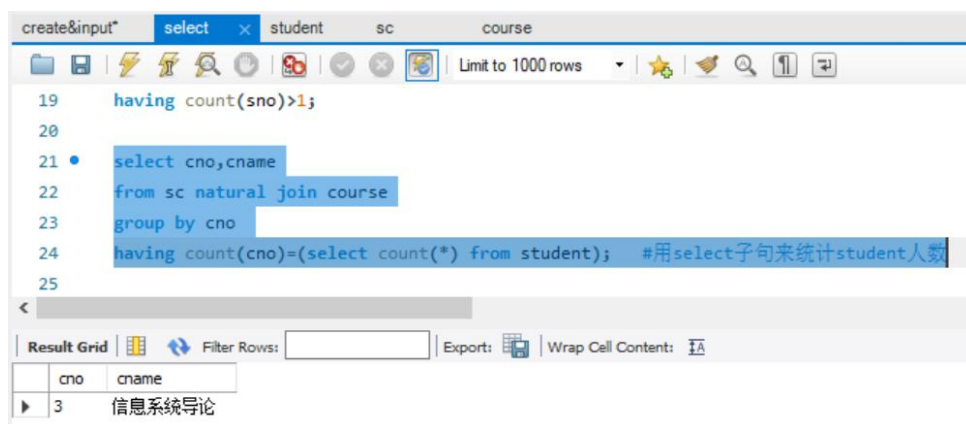
sno	cname
2005001	
2005002	
2005003	
2005010	
2005011	
2005012	

2.2.5 检索全部学生都选修的课程的课程号和课程名

代码:

```
select cno,cname
from sc natural join course
group by cno
having count(cno)=(select count(*) from student); #用 select 子句来统计 student 人数
```

运行结果:



The screenshot shows a SQL IDE with a query editor and a results grid. The query editor contains the following SQL code:

```
19 having count(sno)>1;
20
21 • select cno,cname
22 from sc natural join course
23 group by cno
24 having count(cno)=(select count(*) from student); #用select子句来统计student人数
25
```

The results grid shows the output of the query, with columns 'cno' and 'cname'.

cno	cname
3	信息系统导论

2.2.6 检索选修了所有 3 学分课程的学生平均成绩

代码:

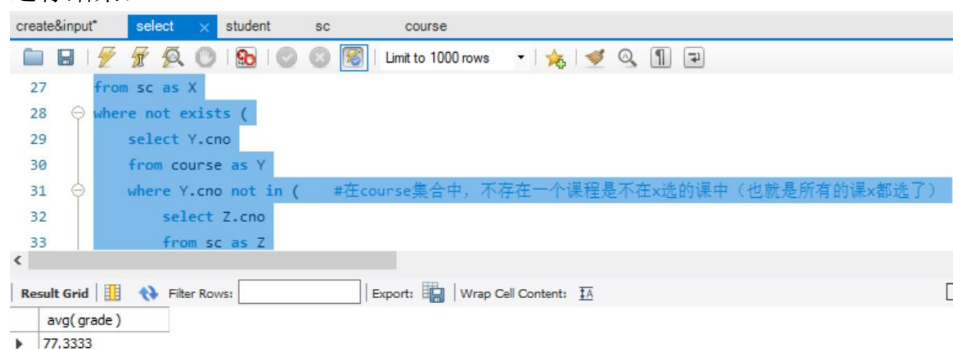
```
select avg( grade )
from sc as X
```

```

where not exists (
    select Y.cno
    from course as Y
    where Y.cno not in (    #在 course 集合中，不存在一个课程是不在 x 选的课中（也就是所有的课 x 都选了）
        select Z.cno
        from sc as Z
        where X.sno = Z.sno    #选出 X 同学所选的课
    )
    and ccredit = 3)
group by X.sno ;

```

运行结果：



2.3 基于 jxgl 数据库，使用 SQL 语句表达以下查询

2.3.1 统计有学生选修的课程门数

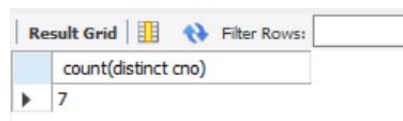
代码：

```

select count(distinct cno)
from sc;

```

运行结果：



2.3.2 求选修 4 号课程的学生的平均年龄

代码：

```

select avg(sage)
from sc natural join student
where cno='4';

```

运行结果：

create&input*

selectstudentsccourseselect2

2.3.3 求学分为 3 的每门课程的学生平均成绩

代码:

```
select avg(grade)
from sc natural join course
where ccredit=3
group by cno;
```

运行结果:

create&input*

select

student

sc

course

select2*

Limit to 1000 rows

7

8 •

```
select avg(grade)
```

9

```
from sc natural join course
```

10

```
where ccredit=3
```

11

```
group by cno;
```

12

13 •

```
select cno ,count(cno)
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	avg(grade)
▶	83.6250
	77.0000
	NULL

2.3.4 统计每门课程的学生选修人数，要求超过三人的课程才统计，要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列

代码:

```
select cno ,count(cno)
from sc
group by cno #按照 cno 课程号分组
having count(cno)>3
order by count(cno) desc,cno asc; #升序降序的表示!!
```

运行结果:

The screenshot shows a database query editor with a toolbar at the top. The query is as follows:

```

13 • select cno ,count(cno)
14   from sc
15  group by cno  #按照cno课程号分组
16  having count(cno)>3
17  order by count(cno) desc,cno asc;  #升序降序的表示!!
18
19 • select X.sname

```

Below the query, the 'Result Grid' is displayed with the following data:

	cno	count(cno)
▶	3	10
	1	4
	2	4

2.3.5 检索学号比王林同学大而年龄比王林同学小的学生姓名

代码:

```

select X.sname
from student as X, student as Y
where Y.sname='王林' and X.sno>Y.sno and X.sage<Y.sage;

```

运行结果:

The screenshot shows a database query editor with a toolbar at the top. The query is as follows:

```

16   having count(cno)>3
17   order by count(cno) desc,cno asc;  #升序降序的表示!!
18
19 • select X.sname
20   from student as X, student as Y
21  where Y.sname='王林' and X.sno>Y.sno and X.sage<Y.sage;
22

```

Below the query, the 'Result Grid' is displayed with the following data:

	sname
▶	赵欣然
	赵青江
	陈婷婷
	李军

2.3.6 检索姓名以'王'开头的学生的姓名和年龄

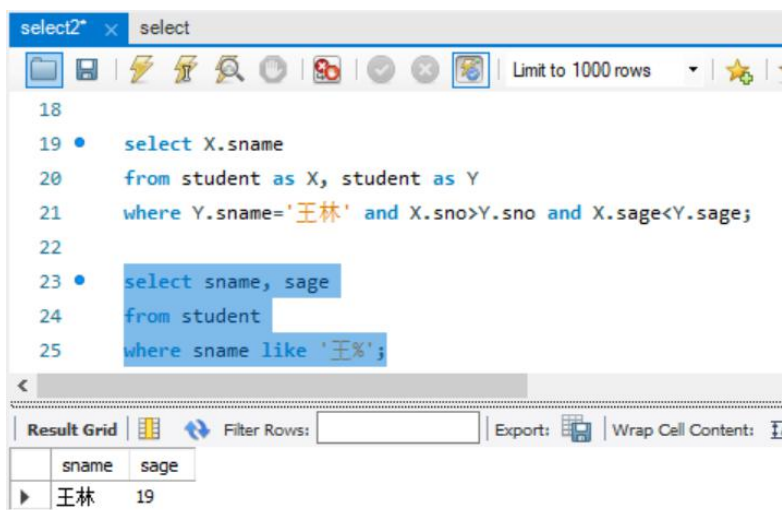
代码:

```

select sname, sage
from student
where sname like '王%';

```

运行结果:



2.3.7 在 sc 表中检索成绩为空值的学生的学号和课程号

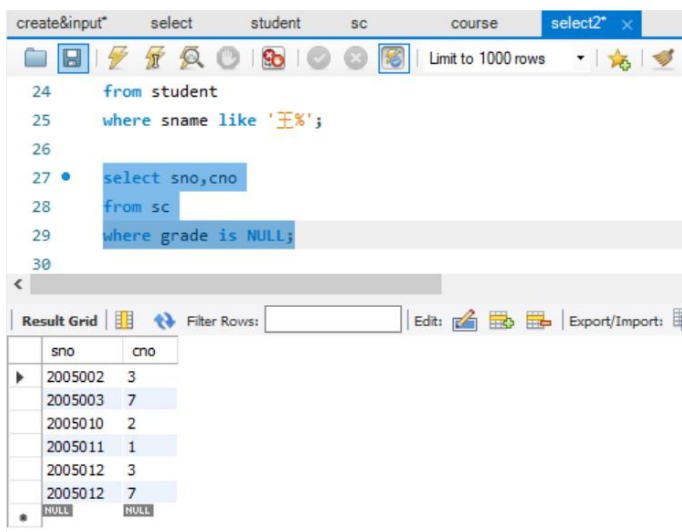
代码:

```
select sno,cno
```

```
from sc
```

```
where grade is null;    #这里一定要用 is (而不是用=) !!
```

运行结果:



2.3.8 求年龄大于女学生平均年龄的男同学的姓名和年龄

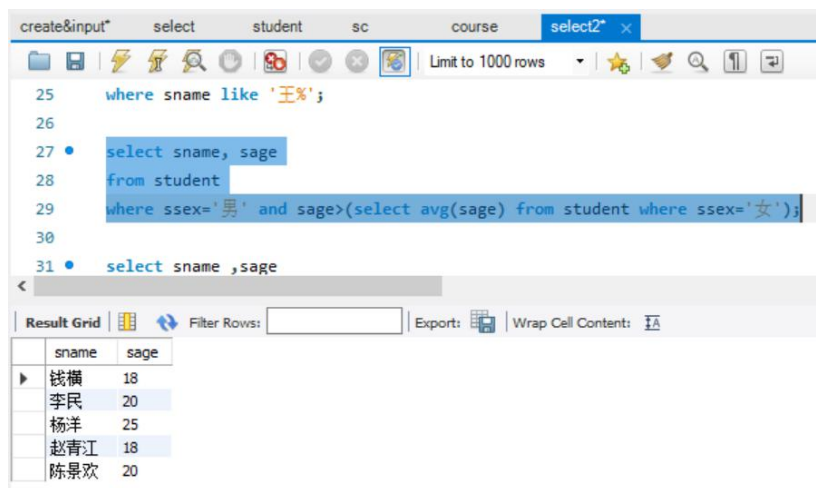
代码:

```
select sname, sage
```

```
from student
```

```
where ssex='男' and sage>(select avg(sage) from student where ssex='女');
```

运行结果:



```

25 where sname like '王%';
26
27 • select sname, sage
28 from student
29 where ssex='男' and sage>(select avg(sage) from student where ssex='女');
30
31 • select sname ,sage

```

sname	sage
钱横	18
李民	20
杨洋	25
赵青江	18
陈景欢	20

2.3.9 求年龄大于所有女学生年龄的男学生的姓名和年龄

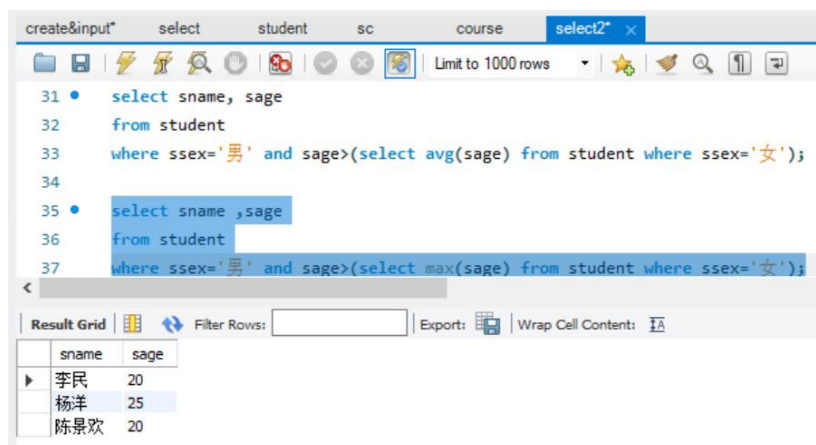
代码:

```
select sname ,sage
```

```
from student
```

```
where ssex='男' and sage>(select max(sage) from student where ssex='女');
```

运行结果:



```

31 • select sname, sage
32 from student
33 where ssex='男' and sage>(select avg(sage) from student where ssex='女');
34
35 • select sname ,sage
36 from student
37 where ssex='男' and sage>(select max(sage) from student where ssex='女');

```

sname	sage
李民	20
杨洋	25
陈景欢	20

2.3.10 检索选修了 4 门课程的学生总成绩(不统计不及格课程), 并要求按总成绩的降序排列出来。

代码:

```
select sum(grade)
```

```
from sc
```

```
where grade>=60      #在这里限制 grade>=60
```



```
group by sno
```

```
having count(*)>4
```

```
order by sum(grade) desc;
```




运行结果:

create&input* select student sc course select2* x

 Limit to 1000 rows 

```
43 • select sum(grade)
44 from sc
45 where grade>=60 #在这里限制grade>=60
46 group by sno
47 having count(*)>4
48 order by sum(grade) desc;
```

<

Result Grid  Filter Rows: Export:  Wrap Cell Content: 

	sum(grade)
▶	422