

《数据库原理》实验报告

实验名称 数据查询实验

班 级 2014211304

学 号 2014211218

姓 名 史文翰

实验三 数据查询实验

一、实验目的

通过对实验二中建立的学生数据库关系表和视图的各种查询的操作，加深对 SQL 查询语言的了解，掌握相关查询语句的语法和使用方法。

二、实验平台及环境

MySQL 5.7 Command Line x64 on windows 10。

三、实验内容

1、数据库关系表查询

- (1) 简单的查询操作，包括单表的查询、选择条件、结果排序等的练习；
- (2) 多表的连接查询，包括等值连接、自然连接等；
- (3) 复杂的查询操作，包括使用分组函数等库函数的查询操作；
- (4) 练习带有 IN、比较符的嵌套查询。

2、视图查询

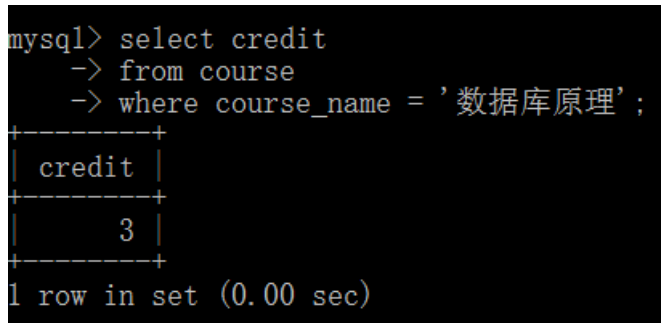
对实验二建立的视图进行相关的查询操作。

四、实验步骤及结果分析

A、数据库关系查询部分：

1、查询“数据库原理”课程的学分

```
select credit  
from course  
where course_name = '数据库原理';
```



```
mysql> select credit  
-> from course  
-> where course_name = '数据库原理';  
+-----+  
| credit |  
+-----+  
|      3 |  
+-----+  
1 row in set (0.00 sec)
```

2、查询选修了课程编号为“C01”的学生的学号和成绩，并将成绩按降序输出

```
select stu_no, grade  
from sel  
where course_no = 'C01' order by grade desc;
```

```
mysql> select stu_no, grade
-> from sel
-> where course_no = 'C01' order by grade desc;
```

stu_no	grade
31411	97
31417	97
31401	94
31413	91
31419	86
31418	55
31402	40
31403	40

3、查询学号为“31401”的学生选修的课程编号和成绩

```
select course_no, grade
from sel
where stu_no = '31401';
```

```
mysql> select course_no, grade
-> from sel
-> where stu_no = '31401';
```

course_no	grade
C01	94
C02	65
C03	NULL

3 rows in set (0.00 sec)

4、查询选修了课程编号为“C01”且成绩高于 85 分的学生的学号和成绩

```
select stu_no, grade
from sel
where course_no = 'C01' and grade > 85;
```

```
mysql> select stu_no, grade
-> from sel
-> where course_no = 'C01' and grade > 85;
```

stu_no	grade
31401	94
31411	97
31413	91
31417	97
31419	86

5 rows in set (0.00 sec)

5、查询选修了课程编号为“C01”且成绩高于 85 分的学生的学号、姓名和成

绩

```
select stu_no, stu_name, grade
from student natural join sel
where course_no = 'C01' and
grade > 85;
```

```
mysql> select stu_no, stu_name, grade
-> from student natural join sel
-> where course_no = 'C01' and
-> grade > 85
-> ;
```

stu_no	stu_name	grade
31401	郭子敬	94
31411	毛文亮	97
31413	李承彧	91
31417	张超	97
31419	崔江平	86

5 rows in set (0.02 sec)

6、查询所有学生的学号、姓名、选修的课程名称和成绩

```
select stu_no, stu_name, course_name, grade
from student natural join course natural join sel;
```

```
mysql> select stu_no, stu_name, course_name, grade
-> from student natural join course natural join sel
-> ;
```

stu_no	stu_name	course_name	grade
30201	吴磊	操作系统	40
30201	吴磊	JAVA 程序设计	88
30201	吴磊	计算机组成原理	93
30202	袁青春	操作系统	40
30202	袁青春	JAVA 程序设计	40
30203	唐雷	操作系统	57
30203	唐雷	JAVA 程序设计	50
30203	唐雷	计算机组成原理	40
30204	吴霏	操作系统	54
30204	吴霏	JAVA 程序设计	50
30204	吴霏	计算机组成原理	40
30206	连洪炽	操作系统	40
30206	连洪炽	JAVA 程序设计	40
30206	连洪炽	计算机组成原理	50
30207	王金柱	操作系统	82
30207	王金柱	JAVA 程序设计	40
30207	王金柱	计算机组成原理	40

7、查询至少选修了三门课程的学生学号和姓名

```
select stu_no, stu_name, count(course_no)
from student natural join sel
group by stu_no
having count(course_no) >= 3;
```

```
mysql> select stu_no, stu_name, count(course_no)
-> from student natural join sel
-> group by stu_no
-> having count(course_no) >= 3;
```

stu_no	stu_name	count(course_no)
30201	吴磊	3
30203	唐雷	3
30204	吴霏	3
30206	连洪炽	3
30207	王金柱	3
30208	苏广学	3
30212	张伟	3
30214	车平跃	3
30215	张鑫	3
30216	王凯	3
30218	王佳楠	3
30219	马浩	3
30220	邵苏杰	3
30221	郑建华	3
30222	郑侃彦	3
30223	宁剑霞	3
30225	陈菊英	3
30226	李新元	3
30228	王倩	3

8、查询所有学生的学号和他选修课程的最高成绩，要求他的选修课程中没有成绩为空的

有成绩为空的

```
select stu_no, course_no, grade, max(grade)
from student natural join sel
where stu_no not in (
select stu_no
from sel
where grade is null )
group by stu_no;
```

```
mysql> select stu_no, course_no, grade, max(grade)
-> from student natural join sel
-> where stu_no not in (
-> select stu_no
-> from sel
-> where grade is null )
-> group by stu_no;
```

stu_no	course_no	grade	max(grade)
30201	C03	40	93
30202	C03	40	40
30203	C03	57	57
30204	C03	54	54
30206	C03	40	50
30207	C03	82	82
30208	C03	40	97
30209	C03	40	88
30210	C05	40	40
30211	C04	40	40
30212	C03	50	91
30213	C03	88	88
30214	C03	50	50
30215	C03	40	61

31409	C01	40	65
31410	C03	40	40
31411	C01	97	97
31412	C01	40	56
31413	C01	91	91
31414	C01	40	94
31415	C01	40	50
31416	C01	40	63
31417	C01	97	97
31418	C01	55	93
31419	C01	86	88
31420	C01	40	57
31422	C01	40	85
31423	C01	40	40
31424	C01	40	92
31425	C02	40	40
31426	C01	40	96
31427	C01	40	88
31428	C01	40	40

```
52 rows in set (0.01 sec)
```

需要说明的是，这里使用了嵌套查询，这是因为在分组之前，必须剔除掉带有 NULL 成绩的同学，单纯的 where 子句无法满足这个要求，因此使用嵌套的方法，先查询出带有 NULL 成绩的同学列表，再根据这个列表筛选出结果。在此之后，才可以进行分组。值得注意的是，如果只用 where grade is not null 来筛选，那带有 NULL 成绩的学生仍然可能出现在 group by 结果中，这是因为他可能有其他的课程成绩是非 NULL 的。

9、查询选修了数据库原理的学生的学号和姓名

```
select stu_no, stu_name
```

```

from student
where stu_no in (
select stu_no
from sel natural join course
where course_name = '数据库原理');

```

```

mysql> select stu_no, stu_name
-> from sel natural join student
-> where course_no in (
-> select course_no
-> from course
-> where course_name = '数据库原理' );

```

stu_no	stu_name
31401	郭子敬
31403	杨磊
31404	冯宇宁
31406	宋月
31407	赵卫京
31408	刘树魁
31409	牛海锋
31411	毛文亮
31412	刘俊亮
31414	王朝远
31415	刘琛
31416	徐亮亮
31417	张超
31418	苏旭浩

31419	崔江平
31420	唐飞
31422	林昕
31423	操牡丹
31424	张乐园
31425	雷爽
31426	秦灵伶
31427	黄金花
31428	张敏

23 rows in set (0.00 sec)

10、查询没有选修数据库原理的学生的学号和姓名

```

select stu_no, stu_name
from student
where stu_no not in (
select stu_no
from sel natural join course
where course_name = '数据库原理');

```

```
mysql> select stu_no, stu_name
-> from student
-> where stu_no not in (
-> select stu_no
-> from sel natural join course
-> where course_name = '数据库原理' );
```

stu_no	stu_name
30201	吴磊
30202	袁青春
30203	唐雷
30204	吴霏
30206	连洪焱
30207	王金柱
30208	苏广学
30209	唐元亮
30210	葛艳杰
30211	张永超
30212	张伟
30213	孙刚

30214	车平跃
30215	张鑫
30216	王凯
30217	马涛
30218	王佳楠
30219	马浩
30220	邵苏杰
30221	郑建华
30222	郑侃彦
30223	宁剑霞
30224	龙海燕
30225	陈菊英
30226	李新元
30227	郭小莉
30228	王倩
31402	胡雅杰
31405	黄荣
31410	吴宁
31413	李承斌

31 rows in set (0.00 sec)

11、查询至少选修了学号为“31401”的学生所选修的所有课程的学生学号和姓名

```
select stu_no, stu_name
from student natural join sel
where course_no in
( select course_no
from sel
where stu_no = '31401')
```



```

group by stu_no
having count(course_no) >=
( select count(*)
from
( select course_no
from sel
where stu_no = '31401') as temp );

```

```

mysql> select stu_no, stu_name
-> from student natural join sel
-> where course_no in
-> ( select course_no
-> from sel
-> where stu_no = '31401')
-> group by stu_no
-> having count(course_no) >=
-> ( select count(*)
-> from
-> ( select course_no
-> from sel
-> where stu_no = '31401') as temp );

```

stu_no	stu_name
31401	郭子敬
31403	杨磊
31404	冯宇宁
31406	宋月
31407	赵卫京
31408	刘树魁
31409	牛海锋
31411	毛文亮
31412	刘俊亮
31414	王朝远

31415	刘焱
31417	张超
31418	苏旭浩
31419	崔江平
31420	唐飞
31422	林昕
31423	操牡丹
31424	张乐园
31426	秦灵伶
31428	张敏

20 rows in set (0.00 sec)

说明：这里子查询构造学号为 X 的 course_no 表，这个表有两个作用：其一，作为 where 子句的筛选对象，即在分组之前筛选选修了不在此表的课的元组；其二，作为标量子查询返回这个表的选课数。我的思路是，先排除不在此表里的课，将剩余的选课情况进行分组，分组后的选课应都在此表中，此时再根据数量比对，如果一个学生分组之后的选课数仍大于等于这个表的行数，那么他一定选修了 X 的所有课程。

B、视图查询部分：

1、查询选修了课程编号为“C01”的学生的学号和成绩

```
select stu_no, grade
from info_sel
where course_no = 'C01';
```

```
mysql> select stu_no, grade
-> from info_sel
-> where course_no = 'C01';
```

stu_no	grade
31401	94
31402	40
31403	40
31404	40
31405	40
31406	40
31407	40
31408	40
31409	40
31411	97
31412	40
31413	91
31414	40

31414	40
31415	40
31416	40
31417	97
31418	55
31419	86
31420	40
31422	40
31423	40
31424	40
31426	40
31427	40
31428	40

```
25 rows in set (0.00 sec)
```

2、查询所有学生的学号、姓名、选修的课程名称和成绩

```
select stu_no, stu_name, course_name, grade
from info_sel;
```

```
mysql> select stu_no, stu_name, course_name, grade
-> from info_sel;
```

stu_no	stu_name	course_name	grade
30201	吴磊	操作系统	40
30201	吴磊	JAVA 程序设计	88
30201	吴磊	计算机组成原理	93
30202	袁青春	操作系统	40
30202	袁青春	JAVA 程序设计	40
30203	唐雷	操作系统	57
30203	唐雷	JAVA 程序设计	50
30203	唐雷	计算机组成原理	40
30204	吴霏	操作系统	54
30204	吴霏	JAVA 程序设计	50
30204	吴霏	计算机组成原理	40
30206	连洪炽	操作系统	40
30206	连洪炽	JAVA 程序设计	40
30206	连洪炽	计算机组成原理	50
30207	王金柱	操作系统	82
30207	王金柱	JAVA 程序设计	40
30207	王金柱	计算机组成原理	40

3、查询选修了数据库原理的学生的学号和姓名

```
select stu_no, stu_name
from info_sel
where course_name = '数据库原理';
```

```
mysql> select stu_no, stu_name
-> from info_sel
-> where course_name = '数据库原理';
```

stu_no	stu_name
31401	郭子敬
31403	杨磊
31404	冯宇宁
31406	宋月
31407	赵卫京
31408	刘树魁
31409	牛海锋
31411	毛文亮
31412	刘俊亮
31414	王朝远
31415	刘琛
31416	徐亮亮
31417	张超
31418	苏旭浩
31419	崔江平
31420	唐飞
31422	林昕
31423	操牡丹

五、实验小结

本次实验主要是查询部分，在前一次实验建立的数据库的基础上，我主要探索了一些 select 的基本语句，已经一些较为复杂的嵌套查询。对于一些简单的查询，已经可以做到熟练掌握，而对于较复杂的如 10/11 题，在仔细思考过后也能得到正确的结果。总之，本次实验让我对数据库的查询有了更进一步的了解。

此外，对于题 8 中的查询，也让我对 having 和 where 子句有了新的认识。在筛选一些结果信息时，是利用 where 先筛出一些元组后再分组，还是分组之后利用 having 一并筛选，是值得思考的问题。