

《数据库系统实验》

实验报告

题目	实验 4
姓名	冯大伟
学号	18340040
班级	计科

一. 实验环境:

操作系统: Windows 10

应用: MySQL Workbench 8.0 CE

二. 实验内容与完成情况:

1. 基于 jxgl 数据库, 使用 SQL 语句表达以下查询

(1) 检索年龄大于 23 岁的男学生的学号和姓名

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

结果:

```
mysql> select sno,sname from student where ssex = '男' and sage > 23;
+-----+-----+
| sno  | sname |
+-----+-----+
| 00002 | 狗俊锋 |
| 00004 | 狗俊锋3号 |
| 00006 | 狗俊锋5号 |
+-----+-----+
```

(2) 检索至少选修一门课程的女学生姓名

```
1. select distinct student.sno,sname
2. from student,sc
3. where student.sno = sc.sno and ssex = '女' ;
```

```
mysql> select distinct student.sno,sname
-> from student,sc
-> where student.sno = sc.sno and ssex = '女' ;
+-----+-----+
| sno  | sname |
+-----+-----+
| 00003 | 狗俊锋2号 |
| 00005 | 狗俊锋4号 |
+-----+-----+
```

(3) 检索王林不学的课程的课程号

```
1. select course.cno
```

```

2. from course
3. where course.cno not in (
4.     select sc.cno
5.     from student,sc
6.     where student.sno = sc.sno
7.     and student.sname = '王林'
8. );

```

```

mysql> select course.cno
-> from course
-> where course.cno not in (
->     select sc.cno
->     from student,sc
->     where student.sno = sc.sno
->     and student.sname = '王林'
-> );
+-----+
| cno |
+-----+
| 03  |
| 05  |
| 06  |
| 07  |
| 08  |
+-----+

```

(4) 检索至少选修两门课程的学生学号

```

1. select student.sno
2. from student,sc
3. where student.sno = sc.sno
4. group by sc.sno
5. having count(*) >= 2 ;

```

```

mysql> select student.sno
-> from student,sc
-> where student.sno = sc.sno
-> group by sc.sno
-> having count(*) >= 2 ;
+-----+
| sno |
+-----+
| 00001 |
| 00002 |
| 00003 |
| 00004 |
| 00005 |
| 00006 |
+-----+

```

(5) 检索全部学生都选修的课程的课程号和课程名

```

1. select course.cno,cname

```

```

2. from course,sc
3. where course.cno = sc.cno
4. group by sc.cno
5. having count(*)=
6. (
7.     select count(*)
8.     from student
9. );

```

```

mysql> select course.cno,cname
-> from course,sc
-> where course.cno = sc.cno
-> group by sc.cno
-> having count(*)=
-> (
->     select count(*)
->     from student
-> );

```

cno	cname
02	中单的自我修养

(6) 检索选修了所有 3 学分课程的学生平均成绩

```

1. select avg(grade)
2.     from course join sc using(cno)
3.     where sno in (
4.         select distinct sno
5.         from course join sc using(cno)
6.         where credit=3
7.         group by sno
8.         having count(cno)=(
9.             select count(cno)
10.            from course
11.            where credit=3
12.        )
13.     )
14. group by sno;

```

```
mysql> select avg(grade)
->   from course join sc using(cno)
->   where sno in (
->       select distinct sno
->       from course join sc using(cno)
->       where credit=3
->       group by sno
->       having count(cno)=(
->           select count(cno)
->           from course
->           where credit=3
->       )
->   )
->   group by sno;
```

avg(grade)
70.3333
77.0000

2. 基于 jxgl 数据库，使用 SQL 语句表达以下查询

(1) 统计有学生选修的课程门数

```
1. select count(*)
2. from course
3. where course.cno
4. in
5. (
6.     select sc.cno
7.     from sc
8. );
```

```
mysql> select count(*)
-> from course
-> where course.cno
-> in
-> (
->     select sc.cno
->     from sc
-> );
```

count(*)
8

1 row in set (0.00 sec)

(2) 求选修 4 号课程的学生的平均年龄

```
1. select avg(sage)
2. from student,sc
3. where student.sno = sc.sno and sc.cno = '04' ;
```

```
mysql> select avg(sage)
-> from student,sc
-> where student.sno = sc.sno and sc.cno = '04' ;
```

avg(sage)
23.0000

(3) 求学分为 3 的每门课程的学生平均成绩

1. `select` cname,avg(grade)
2. `from` course,sc
3. `where` credit = 3 and course.cno = sc.cno
4. `group by` cname;

```
mysql> select cname,avg( grade )
-> from course,sc
-> where credit = 3 and course.cno = sc.cno
-> group by cname;
```

cname	avg(grade)
中单的自我修养	80.3333
如何成为外卖骑手	65.0000

(4) 统计每门课程的学生选修人数，要求超过 3 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

1. `select` cno,count(cno)
2. `from` sc
3. `group by` cno
4. `having` count(cno) > 3
5. `order by` count(cno) desc,cno asc;

```
mysql> select cno,count(cno)
-> from sc
-> group by cno
-> having count(cno) > 3
-> order by count(cno) desc,cno asc;
```

cno	count(cno)
02	6
01	4

(5) 检索学号比'王林'同学大而年龄比他小的学生姓名

1. `select` X.sname

2. **from** student **as** X, student **as** Y
3. **where**
4. Y.sname='王林' and X.sno>Y.sno and X.sage<Y.sage;

```
mysql> 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;
+-----+
| sname |
+-----+
| 狗俊锋2号 |
| 狗俊锋4号 |
+-----+
```

(6) 检索姓名以'王'开头的所有学生的姓名和年龄

1. **select** sname, sage
2. **from** student
3. **where** sname like '王%';

```
mysql> select sname, sage
-> from student
-> where sname like '王%';
+-----+-----+
| sname | sage |
+-----+-----+
| 王林 | 25 |
+-----+-----+
```

(7) 在 sc 表中检索成绩为空值的学生的学号和课程号

1. **select** sno, cno
2. **from** sc
3. **where** grade **is** null;

```
mysql> select sno, cno
-> from sc
-> where grade is null;
+-----+-----+
| sno | cno |
+-----+-----+
| 00006 | 04 |
+-----+-----+
```

(8) 求年龄大于女学生平均年龄的男学生的姓名和年龄

1. **select** sname, sage
2. **from** student
3. **where** ssex='男' and sage>
4. (

```

5.     select avg(sage)
6.     from student
7.     group by (ssex)
8.     having ssex='女'
9. );

```

```

mysql> select sname,sage
      -> from student
      -> where ssex='男' and sage>
      -> (
      ->     select avg(sage)
      ->     from student
      ->     group by (ssex)
      ->     having ssex='女',
      -> );

```

sname	sage
王林	25
狗俊锋	31
狗俊锋3号	26
狗俊锋5号	35

(9) 求年龄大于所有女学生年龄的男学生的姓名和年龄

```

1. select sname,sage
2. from student
3. where ssex='男' and sage>all
4. (
5.     select sage
6.     from student
7.     where ssex='女'
8. );

```

```

mysql> select sname,sage
      -> from student
      -> where ssex='男' and sage>all
      -> (
      ->     select sage
      ->     from student
      ->     where ssex='女',
      -> );

```

sname	sage
王林	25
狗俊锋	31
狗俊锋3号	26
狗俊锋5号	35

(10) 检索选修 4 门以上课程的学生总成绩（不统计不及格课程），并要求按总成绩的降序

排列出来

1. `select` sno, `sum`(grade)
2. `from` sc
3. `where` grade>=60
4. `group by` sno
5. `having` `count`(grade)>=4
6. `order by` `sum`(grade) `desc`;

```
mysql> select sno, sum(grade)
-> from sc
-> where grade>=60
-> group by sno
-> having count(grade)>=4
-> order by sum(grade) desc;
```

sno	sum(grade)
00002	318