

## SQL 测验题

现有关系模式如下：

学生（学号，姓名，性别，出生年月）；课程（课程号，课程名，教师姓名）；

选课表（课程号，学号，成绩）

1. 检索年龄大于 20 岁的男生的学号和姓名。

```
select stu_id,stu_name
from student as s
where 2020-year(birthday)>20 and stu_sex='M';
```

2. 检索选修了姓刘的老师所教授的课程的女学生的姓名。

```
select stu_name
from student as s, stu_course as sc, course as c
where s.stu_id=sc.stu_id and sc.course_id=c.course_id and sex='F' and teacher like '刘%';
```

3. 检索李想同学不学的课程的课程号和课程名。

```
select course_id, course_name
from course as c
where not exists
(select * from student as s, stu_course as sc
Where s.stu_id=sc.stu_id and sc.course_id=c.course_id and s.stu_name='李想');
```

4. 检索至少选修了两门课程的学生的学号。

```
select distinct x.stu_id, x.stu_name
from stu_course as x, stu_course as y
where x.stu_id=y.stu_id and x.course_id <> y.course_id;
```

5. 求刘老师所教授课程的每门课的平均成绩。

```
select c.course_id, avg(sc.grade)
from stu_course sc, course c
Where sc.course_id = c.course_id and c.course_teacher like '刘%'
Group by c.course_id;
```

6. 假设不存在重修的情况，请统计每门课的选修人数(选课人数超过两人的课程才统计)。要求显示课程号和人数，查询结果按人数降序排列，若人数相同，按课程号升序排列。

```
Select sc.course_id, count(sc.stu_id)
From stu_course as sc
Group by sc.course_id
Having count(*) > 2
Order by 2 desc, 1;
```

7. 求年龄大于所有女生年龄的男生的姓名和年龄。

```
Select s.stu_name,s.stu_age
      From student as s
      Where s.stu_sex='M' and
            s.stu_age >all      //或者 >(select max(stu_age)...
            (select b.stu_age
              From student as b
      Where b.stu_sex = 'F');
```

8. 假定不存在重修的情况,求选修了所有课程的学生的学号姓名。(可以不用相关子查询做)  
除了用 exists 查询,还有一种做法:

```
select studentnumber from sc group by studentnumber having count(*)=(select count(*) from
course)
```

9. 查询重修次数在 2 次以上的学生学号, 课程号, 重修次数

```
Select 学号,课程号, count(*) from sc group by 学号,课程号 having count(*)>=3
```

10. 查询被单个学生重修次数最多的课程号, 课程名, 教师姓名

```
select 课程 from 选课 group by 学生号,课程 having count(*)>=all(
select count(*) from 选课 group by 学生号,课程 having count(*)>1)
```