```
CREATE TABLE class(cid int not NULL
auto_increment primary key,
caption char(32)
)engine=innodb default charset=utf8;

insert into class(caption) values('三年级一班'),
('三年级二班'),('一年级一班');

create table student(sid int not null
auto_increment primary key,
sname char(8),
gender char(4),
class_id int not null,
constraint fk_class_table foreign key(class_id)
references class(cid)
)engine=innodb default charset=utf8;


insert into student(sname,gender,class_id)
values('钢蛋','女',1),('杨艳','女',2),('张
扬','男',1),('王大锤','男',3);


create table teacher(tid int not null
auto_increment primary key,
tname char(10)
```

```
)engine=innodb default charset=utf8;

insert into teacher(tname) values('博多'),('仓颉'),
('黎明');


create table course(cid int not null auto_increment primary key,
cname char(32),
teacher_id int,
constraint fk_teacher_course foreign key(teacher_id) references teacher(tid)
)engine=innodb default charset=utf8;

insert into course(cname,teacher_id) values('生物',1),('体育',2),('物理',3);


create table score(sid int not null auto_increment primary key,
student_id int not null,
course_id int not null,
number int not null,
unique uq_st_co (student_id,course_id),
constraint fk_score_student foreign key(student_id) references student(sid),
constraint fk_score_course foreign key(course_id) references course(cid)
```

```
)engine=innodb default charset=utf8;

insert into score(student_id,course_id,number)
values(1,1,60),(1,2,59),(2,2,100);
```

创建数据：

```
CREATE TABLE class(cid int not NULL
auto_increment primary key,
caption char(32)
)engine=innodb default charset=utf8;

insert into class(caption) values('三年级二班'),
('三年级三班'),('一年级二班'),('二年级一班');

create table student(sid int not null
auto_increment primary key,
sname char(8),
gender char(4),
class_id int not null,
constraint fk_class_table foreign key(class_id)
references class(cid)
)engine=innodb default charset=utf8;
```

```
13
14
15 insert into student(gender,class_id,sname)
   values('男', '1', '理解'), ('女', '1', '钢蛋'),
   ('男', '1', '张三'), ('男', '1', '张一'), ('女',
   '1', '张二'), ('男', '1', '张四'), ('女', '2',
   '铁锤'), ('男', '2', '李三'), ('男', '2', '李
   一'), ('女', '2', '李二'), ('男', '2', '李四'),
   ('女', '3', '如花'), ('男', '3', '刘三'), ('男',
   '3', '刘一'), ('女', '3', '刘二'), ('男', '3',
   '刘四');
16
17 create table teacher(tid int not null
   auto_increment primary key,
18 tname char(10)
19 )engine=innodb default charset=utf8;
20
21 insert into teacher values('1', '张磊老师'),
   ('2', '李平老师'), ('3', '刘海燕老师'), ('4', '朱
   云海老师'), ('5', '李杰老师');
22
23
24 create table course(cid int not null
   auto_increment primary key,
25 cname char(32),
26 teacher_id int,
27 constraint fk_teacher_course foreign
   key(teacher_id) references teacher(tid)
28 )engine=innodb default charset=utf8;
```

```
29
30 insert into course(cname,teacher_id) values('生
   物', '1'), ('物理', '2'), ('体育', '3'), ('美术',
   '2');
31
32
33 create table score(sid int not null
   auto_increment primary key,
34 student_id int not null,
35 course_id int not null,
36 number int not null,
37 unique uq_st_co (student_id,course_id),
38 constraint fk_score_student foreign
   key(student_id) references student(sid),
39 constraint fk_score_course foreign
   key(course_id) references course(cid)
40 )engine=innodb default charset=utf8;
41
42 insert into score values('1', '1', '1', '10'),
   ('2', '1', '2', '9'), ('5', '1', '4', '66'),
   ('6', '2', '1', '8'), ('8', '2', '3', '68'),
   ('9', '2', '4', '99'), ('10', '3', '1', '77'),
   ('11', '3', '2', '66'), ('12', '3', '3', '87'),
   ('13', '3', '4', '99'), ('14', '4', '1', '79'),
   ('15', '4', '2', '11'), ('16', '4', '3', '67'),
   ('17', '4', '4', '100'), ('18', '5', '1', '79'),
   ('19', '5', '2', '11'), ('20', '5', '3', '67'),
   ('21', '5', '4', '100'), ('22', '6', '1', '9'),
   ('23', '6', '2', '100'), ('24', '6', '3', '67'),
```

```
('25', '6', '4', '100'), ('26', '7', '1', '9'),
('27', '7', '2', '100'), ('28', '7', '3', '67'),
('29', '7', '4', '88'), ('30', '8', '1', '9'),
('31', '8', '2', '100'), ('32', '8', '3', '67'),
('33', '8', '4', '88'), ('34', '9', '1', '91'),
('35', '9', '2', '88'), ('36', '9', '3', '67'),
('37', '9', '4', '22'), ('38', '10', '1', '90'),
('39', '10', '2', '77'), ('40', '10', '3',
'43'), ('41', '10', '4', '87'), ('42', '11',
'1', '90'), ('43', '11', '2', '77'), ('44',
'11', '3', '43'), ('45', '11', '4', '87'),
('46', '12', '1', '90'), ('47', '12', '2',
'77'), ('48', '12', '3', '43'), ('49', '12',
'4', '87'), ('52', '13', '3', '87');
43
44
45
46 select score.sid,score.student_id as '学
   号',score.number as '分数',student.sname as '姓
   名',student.gender as '性别',class.caption as
   '班级',course.cname as '课程',teacher.tname as
   '老师' from score
47 left join student on
   score.student_id=student.sid
48 left join class on student.class_id=class.cid
49 left join course on score.course_id=course.cid
50 left join teacher on
   course.teacher_id=teacher.tid
51
```

班级表：class	
cid	caption
1	三年二班
2	一年三班
3	三年一班

学生表：student			
sid	sname	gender	class_id
1	钢蛋	女	1
2	铁锤	女	1
3	山炮	男	2

老师表：teacher	
tid	tname
1	波多
2	苍空
3	饭岛

课程表：course		
cid	cname	tearch_id
1	生物	1
2	体育	1
3	物理	2

成绩表：score			
sid	student_id	corse_id	number
1	1	1	60
2	1	2	59
3	2	2	100

操作表：

每个老师任课的个数是多少

```
1 select teacher_id,count(cid) as '课程数量' from
  course group by teacher_id;
```

学生里面男生和女生的个数

```
1 select gender,count(sid) from student GROUP BY
```

```
  gender;
```

1、自行创建测试数据

2、查询“生物”课程比“物理”课程成绩高的所有学生的学号；

```
1 select * from
2 (select
  score.sid,score.student_id,course.cname,score.nu
  mber from score left join course on
  score.course_id=course.cid where
  course.cname='生物') as A
3 inner join
4 (select
  score.sid,score.student_id,course.cname,score.nu
  mber from score left join course on
  score.course_id=course.cid where
  course.cname='物理' ) as B
5 on A.student_id=B.student_id
6 where A.number>B.number;
```

3、查询平均成绩大于60分的同学的学号和平均成绩；

```
1 select student_id,avg(number) from score GROUP
  BY student_id having avg(number)>60;
2 把学生姓名显示出来
```

```
select B.student_id,student.sname,B.avg from
(select student_id,avg(number) as avg from score
GROUP BY student_id having avg(number)>60) as B
left join student on B.student_id=student.sid;

```

4、查询所有同学的学号、姓名、选课数、总成绩；

```
select
student.sid,student.sname,student.gender,B.`选课
数`,B.`总成绩` from student
left join (select student_id,count(sid) as '选课
数',sum(number) as '总成绩' from score GROUP BY
student_id) as B on student.sid=B.student_id;
老师：select
score.student_id,student.sname,count(student_id)
,sum(number) from score left join student on
score.student_id=student.sid group by
score.student_id;
```

5、查询姓“李”的老师的个数；

```
select count(tid) from teacher WHERE tname like
'李%' ;
```

6、查询没学过“李平”老师课的同学的学号、姓名；

```
select * from student where
sid not in (
select student_id from score where course_id in
(
select
course.cid
from
course
left join teacher on
course.teacher_id=teacher.tid
where
teacher.tname='李平老师'
) group by student_id);
```

7、查询学过“001”并且也学过编号“002”课程的同学的学号、姓名；

```
select student.sid,student.sname from score
left join student on student_id=student.sid
where course_id=1 or course_id=2 group by
student_id having count(course_id)>1;
```

8、查询学过“李平”老师所教的所有课的同学的学号、姓名；

```sql
select student.sid,student.sname from
(select * from score where course_id in(
select cid from course left join teacher on
course.teacher_id=teacher.tid
where teacher.tname="李平老师")
group by student_id having count(course_id)=
(select count(cid) from course left join teacher
on course.teacher_id=teacher.tid where
teacher.tname="李平老师")) as A left join student
on A.student_id=student.sid;
```

9、查询课程编号“002”的成绩比课程编号“001”课程低的所有同学的学号、姓名；

```sql
select student.sid,student.sname from (select *
from score where course_id=2) as A
inner join (select * from score where
course_id=1) as B on A.student_id=B.student_id
inner join student on A.student_id=student.sid
where A.number<B.number;
```

10、查询有课程成绩小于60分的同学的学号、姓名；

```sql
select distinct student.sid,student.sname from
```

```
  score
2 left join student on student_id=student.sid
  where number<60;
```

distinct 能去重，但效率不高

11、查询没有学全所有课的同学的学号、姓名；

```
1 select student.sid,student.sname from score
2 left join student on student_id=student.sid
  group by student_id having count(score.sid)
  <(select count(cid) from course);
```

count() 主键或1效率高

12、查询至少有一门课与学号为“001”的同学所学相同的同学的学号和姓名；

```
1 select student.sid,student.sname from score
2 left join student on student_id=student.sid
  where student_id !=1 and course_id in (select
  course_id from score where student_id=1) group
  by student_id;
```

13、查询至少学过学号为“001”同学所选课程中任意一门课的**其他同**

学学号和姓名；

```
select student_id,sname, count(course_id)
from score left join student on score.student_id = student.sid
where student_id != 1 and course_id in (select course_id from score where student_id = 1) group by student_id having count(course_id) = (select count(course_id) from score where student_id = 1)

```

14、查询和“002”号的同学学习的课程完全相同的其他同学学号和姓名；

a 获取和2号同学选课数相同的同学

b 筛选至少有一门课与2号同学相同课程的同学

c 筛选与2号同学课程数相同的同学

```
select student_id from score where student_id in (
select student_id from score where student_id !=2 group by student_id
having count(1) =(select count(1) from score where student_id =2)
) and course_id in (select course_id from score where student_id=2) group by student_id having
```

```
count(1)=(select count(1) from score where
student_id=2);
```

15、删除学习“叶平”老师课的score表记录；

```
delete from score where course_id in(select cid
from course left join teacher on
course.teacher_id=teacher.tid where
teacher.name='叶平')
```

16、向SC表中插入一些记录，这些记录要求符合以下条件：①没有上过编号“002”课程的同学学号；②插入“002”号课程的平均成绩；

```
 insert into score(student_id,course_id,number)
select student_id,2,(select avg(number) from
score where course_id=2) from score where
course_id !=2 group by stuent_id;
```

17、按平均成绩从低到高显示所有学生的“语文”、“数学”、“英语”三门的课程成绩，按如下形式显示： 学生ID,语文,数学,英语,有效课程数,有效平均分；

```
select
```

```
student_id,
(select number from score as s2 where
s2.student_id=s1.student_id and course_id=1) as
语文,
(select number from score as s2 where
s2.student_id=s1.student_id and course_id=2) as
数学,
(select number from score as s2 where
s2.student_id=s1.student_id and course_id=3) as
英语,
count(1),
avg(number)
from score as s1 group by student_id desc;
```

其他写法:

```
select sc.student_id,
        (select number from score left join
course on score.course_id = course.cid where
course.cname = "生物" and
score.student_id=sc.student_id) as sy,
        (select number from score left join
course on score.course_id = course.cid where
course.cname = "物理" and
score.student_id=sc.student_id) as wl,
        (select number from score left join
course on score.course_id = course.cid where
```

```
  course.cname = "体育" and
  score.student_id=sc.student_id) as ty,
5         count(sc.course_id),
6         avg(sc.number)
7     from score as sc
8     group by student_id desc;
```

18、查询各科成绩最高和最低的分：以如下形式显示：课程ID，最高分，最低分；

```
1 select
  course_id,max(number),min(number),min(number)+1,
  case when min(number)<10 then 0 else min(number)
  end as c
2 from score group by course_id;
3
```

19、按各科平均成绩从低到高和及格率的百分数从高到低排序；

```
1 select course_id,avg(number),sum(case when
  number<60 then 0 else 1 end),sum(1),sum(case
  when number<60 then 0 else 1 end)/sum(1) as jgl
  from score group by course_id order by
  avg(number),jgl desc;
2
```

20、课程平均分从高到低显示（显示任课老师）；

```
select
score.course_id,course.cname,avg(if(isnull(score
.number),0,score.number)),teacher.tname from
score
left join course on score.course_id=course.cid
left join teacher on
teacher.tid=course.teacher_id
group by score.course_id;

```

21、查询各科成绩前三名的记录:(不考虑成绩并列情况)

```
select * from score where course_id=1 order by
number desc;
# 每科学生数量大于3时可以使用此方法，考虑了成绩并列
的情况
select * from (select student_id,course_id,
number,
1,
(select number from score as s2 where
s2.course_id=s1.course_id group by s2.number
order by s2.number desc limit 3,1) as cc
```

```
from score as s1) as B where B.number>B.cc
order by course_id,number desc;

# 每科学生数量小于3时使用此方法，没考虑成绩并列情况
select
    score.sid,
    score.student_id,
    score.course_id,
    score.number,
    t1.first_score,
    t1.second_score,
    t1.third_score
from score inner join(
                    select
                        s1.sid,
                        (select number from
score as s2 where s1.course_id=s2.course_id
order by number  desc limit 0,1) as
first_score,
                        (select number from
score as s3 where s1.course_id=s3.course_id
order by number  desc limit 1,1) as
second_score,
                        (select number from
score as s4 where s1.course_id=s4.course_id
order by number  desc limit 2,1) as third_score

                    from score as s1
                    ) as t1 on score.sid =
```

```
  t1.sid
where score.number in(
    t1.first_score,
    t1.second_score,
    t1.third_score);

```

22、查询每门课程被选修的学生数;

```
select course_id,count(1) from score group by
course_id;
```

23、查询出只选修了一门课程的全部学生的学号和姓名;

```
select student_id,count(1) from score group by
student_id having count(1)=1;
```

24、查询男生、女生的人数;

```
select gender,count(1) from student group by
gender;
```

25、查询姓“张”的学生名单；

```
select * from student where sname like '张%';
```

26、查询同名同姓学生名单，并统计同名人数；

```
select sname,count(1) from student group by
sname;
```

27、查询每门课程的平均成绩，结果按平均成绩升序排列，平均成绩相同时，按课程号降序排列；

```
select
course_id,avg(if(isnull(number),0,number)) from
score group by course_id order by avg(number)
asc,course_id desc;
```

28、查询平均成绩大于85的所有学生的学号、姓名和平均成绩；

```
select
student_id,student.sname,avg(if(isnull(number),0
```

```
,number)) as avg from score
left join student on student_id=student.sid
group by student_id having avg>85;
```

29、查询课程名称为“数学”，且分数低于60的学生姓名和分数；

```
select
score.student_id,student.sname,score.number from
score
left join course on score.course_id=course.cid
left join student on
score.student_id=student.sid
where score.number<60 and course.cname='生物';
```

30、查询课程编号为003且课程成绩在80分以上的学生的学号和姓名；

```
select student_id,sname from score
left join student on student_id=student.sid
where course_id=3 and number >80 ;
31、求选了课程的学生人数
select count(distinct student_id) from score;
select count(c) from (
        select count(student_id) as c from score
group by student_id) as A;
```

32、查询选修“张磊”老师所授课程的学生中，成绩最高的学生姓名及其成绩；

```
select student.sname,max(number) from score
left join course on course_id=cid
left join teacher on teacher_id=tid
left join student on student_id=student.sid
where tname='张磊老师';
 select sname,num from score
 left join student on score.student_id =
student.sid
 where score.course_id in (select course.cid
from course left join teacher on
course.teacher_id = teacher.tid where tname='张
磊老师') order by num desc limit 1;
```

33、查询各个课程及相应的选修人数；

```
select course_id,count(1) from score group by
course_id;
```

34、查询不同课程但成绩相同的学生的学号、课程号、学生成绩；

笛卡尔机：每个数据都相互对应

```
select s1.student_id,s1.course_id,s1.number from
score as s1,score as s2 where s1.sid !=s2.sid
and s1.course_id !=s2.course_id and
s1.number=s2.number;
? ? ?
select DISTINCT
s1.course_id,s2.course_id,s1.num,s2.num from
score as s1, score as s2 where s1.num = s2.num
and s1.course_id != s2.course_id;
```

35、查询每门课程成绩最好的前两名；

```
select B.student_id,B.course_id,B.number from
(select student_id,course_id,
number,
(select number from score as s2 where
s2.course_id=s1.course_id group by s2.number
order by s2.number desc limit 2,1) as cc
from score as s1) as B where B.number>B.cc
order by course_id,number desc;

select
score.sid,score.course_id,score.number,T.first_
num,T.second_num from score left join
    (
    select
```

```
        sid,
        (select number from score as s2 where
s2.course_id = s1.course_id order by number
desc limit 0,1) as first_num,
        (select number from score as s2 where
s2.course_id = s1.course_id order by number
desc limit 1,1) as second_num
    from
        score as s1
    ) as T
    on score.sid =T.sid
    where score.number <= T.first_num and
score.number >= T.second_num
```

36、检索至少选修两门课程的学生学号；

```
select student_id,count(1) from score group by
student_id having count(1)>1;
37、查询全部学生都选修的课程的课程号和课程名；
select course_id,cname from score
left join course on course_id=cid
group by course_id having count(1)=(select
count(sid) from student);

```

38、查询没学过“叶平”老师讲授的任一门课程的学生姓名；

```
select * from student where sid not in (
select student_id from score where course_id in
(select cid from course left join teacher on
course.teacher_id=teacher.tid where
teacher.tname='李平老师'));
```

39、查询两门以上不及格课程的同学的学号及其平均成绩；

```
select A.student_id,avg(number) from score inner
join (
select student_id from score  where number<60
group by student_id having count(1)>1) as A on
score.student_id=A.student_id ;
```

40、检索“004”课程分数小于60，按分数降序排列的同学学号；

```
select * from score where course_id=4 and
number<60 order by number desc;
```

41、删除“002”同学的“001”课程的成绩；

```
1 delete from score where course_id = 1 and
  student_id = 2;
```