



-- 查询课程编号为“01”的课程比“02”的课程成绩高的所有学生的学号（重点）

```
select a.s_id, a.s_score, b.s_score from
(
select * from score where c_id = '01'
) as a
inner join
(
select * from score where c_id = '02'
) as b
on a.s_id = b.s_id
where a.s_score > b.s_score;
```

-- 查询平均成绩大于 60 分的学生的学号和平均成绩（简单，第二道重点）

```
select s_id, avg(s_score)
from score
group by s_id
having avg(s_score) > 60;
```

-- 查询所有学生的学号、姓名、选课数、总成绩（不重要）

```
select a.s_id, a.s_name,
count(b.c_id) 'c_count',
sum(case when b.s_score is null then 0 else b.s_score end) 'sum'
from
(
student as a
left join
score as b
on a.s_id = b.s_id
)
group by a.s_id, a.s_name;
```

-- 查询姓猴的老师的个数

```
select count(t_id) from teacher where t_name like '张%';
```

```
select count(distinct t_name) 'dis_name' from teacher where t_name like '张%';
```

-- 查询 **没学过张三老师** 的课的学生学号和姓名

```
select s_id, s_name from student
where s_id not in(
select s_id from score
where c_id =
    (
        select c_id from course
        where t_id =
            (select t_id from teacher where t_name = '张三')
    )
);
```

```
select s_id, s_name from student
```

```
where s_id not in
```

```
(
    select s.s_id from
    score as s inner join
    course as c on s.c_id = c.c_id
    inner join
    teacher as t on c.t_id = t.t_id
    where t.t_name = '张三'
);
```

-- 查询 学过张三老师 所教的课的学生学号和姓名

```
select s_id, s_name from student
```

```
where s_id in
```

```
(
    select s_id from score
    where c_id in
    (
        select c_id from course
        where t_id =
        (
            select t_id from teacher where t_name = '张三'
        )
    )
);
```

-- 查询学过编号为“01”的课程并且也学过编号为“02”的课程的学生的学号、姓名（重点）

```
select s_id, s_name from student where s_id in
(
    select a.s_id
    from
    (
        (
            select s_id from score where c_id = '01'
        ) as a
        inner join
        (
            select s_id from score where c_id = '02'
        ) as b
        on a.s_id = b.s_id
    )
);
```

-- 查询课程编号为“02”的总成绩（不重点）

```
select sum(s_score), avg(s_score) from score group by c_id having c_id = '02';
```

-- 查询所有课程成绩小于 60 分的学生的学号、姓名

```
select s_id, s_name from student where s_id in(
    select a.s_id from
    (
        select s_id, count(c_id) 'cnt' from score where s_score < 60 group by s_id
    ) as a
    inner join
    (
        select s_id, count(c_id) 'cnt' from score group by s_id
    ) as b
    on a.s_id = b.s_id
    where a.cnt = b.cnt
);
```

-- 查询没有学全所有课的学生的学号、姓名(重点)

```
select a.s_id, a.s_name from
(
    select st.s_id, st.s_name, count(c_id) 'cnt' from
    student as st left join score as s on st.s_id = s.s_id group by st.s_id
) as a
inner join
(select count(c_id) 'cnt' from course) as b
on a.cnt < b.cnt;
```

-- 查询至少有一门课和学号 01 的学生重合的学生学号和姓名

```
select s_id, s_name from student where s_id in
(
    select distinct s_id from score where c_id in
    (
        select c_id from score where s_id = '01'
    ) and s_id != '01'
);
```

```
select a.s_id, a.s_name from student as a
inner join
(
    select distinct s_id from score where c_id in
    (
        select c_id from score where s_id = '01'
    ) and s_id != '01'
) as b on a.s_id = b.s_id;
```

-- 查询和 '01' 号同学所学课程完全相同的其他同学的学号

```
select s_id from score group by s_id having count(distinct c_id) = (
    select count(distinct c_id) from score where s_id = '01'
) and s_id != '01' and s_id not in
(
    select distinct s_id from score where c_id not in
    (
        select c_id from score where s_id = '01'
    )
);
```

-- 查询两门及以上不及格课程的同学的学号姓名及其平均成绩

```
select a.s_id, a.s_name, avg(b.s_score) from
    student as a
inner join
    score as b on a.s_id = b.s_id
where a.s_id in
(
    select s_id from score where s_score < 60 group by s_id having count(c_id) >= 2
)
group by a.s_id, a.s_name;
```

-- 检索 '01' 课程分数小于 60，按分数降序排序的学生信息

```
select a.*, s.s_score from
student as a
inner join
score as s
on a.s_id = s.s_id
where s.s_score < 60 and c_id = '01'
order by s.s_score desc;
```

-- 按平均成绩从高到低显示所有学生的所有课程的成绩以及平均成绩

```
select s_id,
max(case when c_id = '01' then s_score else null end) '语文',
max(case when c_id = '02' then s_score else null end) '数学',
max(case when c_id = '03' then s_score else null end) '英语',
avg(case when s_score is null then 0 else s_score end) 'avg_score'
from score
group by s_id
order by avg_score desc;
```

-- 查询各科成绩最高分、最低分、和平均分

```
select c.c_id, c.c_name, b.max, b.min, b.avg_s, b.pass, b.ok, b.good, b.perfect
from course as c
inner join
(
    select c_id,
    sum(case when s_score >= 60 then 1 else 0 end) / count(s_id) 'pass',
    sum(case when s_score >= 70 and s_score < 80 then 1 else 0 end) / count(s_id) 'ok',
    sum(case when s_score >= 80 and s_score < 90 then 1 else 0 end) / count(s_id) 'good',
    sum(case when s_score >= 90 then 1 else 0 end) / count(s_id) 'perfect',
    max(s_score) 'max',
    min(s_score) 'min',
    avg(s_score) 'avg_s'
    from score
    group by c_id
) as b
on c.c_id = b.c_id;
```

-- 按各科成绩进行排序，并显示排名

-- row_number 没有重复值的连续排序

-- dense_rank 连续排序（有重复值）

-- rank 跳跃排序

```
select s_id, c_id, s_score, rank() over(partition by c_id order by s_score desc) '排名' from score;
```

-- 查询学生的总成绩并进行排名

```
select s_id,  
max(case when c_id = '01' then s_score else null end) '语文',  
max(case when c_id = '02' then s_score else null end) '数学',  
max(case when c_id = '03' then s_score else null end) '英语',  
sum(s_score) as '总成绩'  
from score group by s_id order by 总成绩 desc;
```

-- 查询不同老师所教不同课程平均分从高到低显示(以课程为主体求平均分)

```
select c.t_id, c.c_name, t.t_name, avg(s.s_score) 'avg_s' from  
course as c  
left join  
score as s  
on c.c_id = s.c_id  
left join  
teacher as t  
on c.t_id = t.t_id  
group by c.c_id, c.c_name  
order by avg_s desc;
```

-- 查询所有课程的成绩第 2 名和第 3 名的学生信息及该课程成绩

```
select * from  
(  
select s.*, sc.c_id 'c_id', row_number() over(partition by sc.c_id order by sc.s_score) order_s  
from  
student as s  
left join  
score as sc  
on s.s_id = sc.s_id  
) as m  
where m.order_s in (2, 3)  
order by m.c_id;
```

-- 使用分段统计各科成绩、各分数段人数，课程 ID 和课程名称

```
select c.c_id, c.c_name,  
sum(case when s.s_score > 85 and s.s_score <= 100 then 1 else 0 end) '[100-85]',  
sum(case when s.s_score > 70 and s.s_score <= 85 then 1 else 0 end) '[85-70]',  
sum(case when s.s_score >= 60 and s.s_score <= 70 then 1 else 0 end) '[70-60]',  
sum(case when s.s_score < 60 then 1 else 0 end) '<60',  
count(s.s_id) 'total'  
from score as s inner join course as c on s.c_id = c.c_id  
group by c.c_id, c.c_name;
```

```
-- 查询学生平均成绩及其名次
-- 省略 partition by 是把所有的数据看成一个组
select s_id, avg(s_score) 'avg_s', row_number() over(order by avg(s_score) desc) '名次'
from score group by s_id;
```

```
-- 查询每门课程被选修的学生数
select c.c_id, c.c_name, count(distinct sc.s_id) 'count' from
score as sc
inner join
course as c
on sc.c_id=c.c_id group by c.c_id, c.c_name;
```

```
-- 查询只有两门课程的全部学生的学号和姓名
select st.s_id, st.s_name
from student as st
inner join
score as sc
on st.s_id = sc.s_id
group by st.s_id, st.s_name having count(distinct c_id) = 2;
```

```
-- 查询男生女生的数量
select count(s_id), s_sex from student group by s_sex;
```

```
select sum(case when s_sex = '男' then 1 else 0 end) '男生人数',
sum(case when s_sex = '女' then 1 else 0 end) '女生人数'
from student;
```

```
-- 查询名字中含有“风”的学生信息
select * from student where s_name like "%风%";
```

```
-- 查询 1990 年出生的学生名单
select * from student where year(s_birth)=1990;
select * from student where month(s_birth)=1;
```

```
-- 查询平均成绩大于 85 的所有学生的学号、姓名、平均成绩
select st.s_id, st.s_name, avg(sc.s_score) 'avg_s'
from student as st inner join score as sc on st.s_id = sc.s_id
group by sc.s_id having avg_s >= 85;
```

```
-- 查询没门课的平均成绩，先按平均成绩升序，再按课程降序
select c_id, AVG(s_score) 'avg_s' from score group by c_id order by avg_s, c_id desc;
```

```

-- 数学这门课成绩低于 60 的学生信息和成绩
select sc.s_id, st.s_name, c.c_name, sc.s_score
from score as sc inner join course as c on sc.c_id = c.c_id
inner join student as st on st.s_id = sc.s_id
where c.c_name = '数学' and sc.s_score < 60;

-- 查询所有学生的课程及分数情况
select st.s_id, st.s_name,
sum(case when sc.c_id = '01' then sc.s_score else null end) '语文',
sum(case when sc.c_id = '02' then sc.s_score else null end) '数学',
sum(case when sc.c_id = '03' then sc.s_score else null end) '英语'
from
student as st left join score as sc on st.s_id = sc.s_id left join course as c on c.c_id = sc.c_id
group by st.s_id, st.s_name order by 数学 desc;

-- 查询课程成绩在 70 分以上的课程名称、分数和学生姓名
select st.s_name, c.c_name, sc.s_score from
(student as st inner join score as sc on st.s_id = sc.s_id inner join course as c on c.c_id = sc.c_id)
where sc.s_score > 70;

-- 查询不及格的课程并按课程号从大到小排列
select st.s_id, st.s_name, c.c_id, c.c_name, sc.s_score from
(student as st inner join score as sc on st.s_id = sc.s_id inner join course as c on c.c_id = sc.c_id)
where s_score < 60 order by c_id desc;

-- 查询课程 ID=03 中超过 80 分的同学的信息和分数
select st.s_id, st.s_name, c.c_id, c.c_name, sc.s_score from
(student as st inner join score as sc on st.s_id = sc.s_id inner join course as c on c.c_id = sc.c_id)
where sc.c_id = '03' and sc.s_score > 80 order by sc.s_score desc;

-- 查询每门课的选课人数
select c.c_id, c.c_name, count(sc.s_id) from
score as sc inner join course as c on sc.c_id = c.c_id group by c.c_id, c.c_name;

-- 查询选修张三老师所授课程的学生中成绩最高的学生姓名及其成绩
select m.sname, m.sscore, m.tname, m.top from
(select st.s_name 'sname', sc.s_score 'sscore', t.t_name 'tname', row_number() over(order by
sc.s_score desc) 'top'from
student as st
inner join score as sc on st.s_id = sc.s_id
inner join course as c on sc.c_id = c.c_id
inner join teacher as t on c.t_id = t.t_id
where t.t_name = '张三'
) as m where m.top = 1;

```


-- 或者

```
select st.s_id, st.s_name 'sname', sc.s_score 'sscore', c.c_id, t.t_name 'tname' from
student as st
inner join score as sc on st.s_id = sc.s_id
inner join course as c on sc.c_id = c.c_id
inner join teacher as t on c.t_id = t.t_id
where t.t_name = '张三' order by sc.s_score desc limit 0, 1;
```

-- 查询不同课程成绩相同的学生的学生编号、课程编号、学生成绩

```
select * from score where s_id in(
    select s_id from(
        select s_id, s_score from score group by s_id, s_score
    ) as m group by m.s_id having count(s_id) = 1
);
```

-- 查询选课人数超过 5 人的课程 id，并且按照选课人数降序排列，课程 id 升序排列

```
select c_id, count(distinct s_id) 'num' from score group by c_id having num > 5 order by num
desc, c_id;
```

-- 查询至少选了两门课的学生

```
select s_id, count(distinct c_id) 'c_count' from score group by s_id having c_count >= 2;
```

-- 查询选了所有课程的学生

```
select * from student where s_id in
(
    select s_id from score group by s_id having count(distinct c_id) = (
        select count(distinct c_id) from course
    )
);
```

-- 查询学生的年龄

```
select s_id, s_birth, year(now()) - year(s_birth) from student;
select s_id, s_birth, floor(datediff(now(), s_birth) / 365) 'age' from student;
```

-- 查询没有学过张三老师讲授的任意一门课的学生姓名

```
select st.s_id, st.s_name,
sum(case when t.t_name = '张三' or s.c_id = null then 0 else 1 end) as sumExp,
sum(case when s.c_id = null then 0 else 1 end) as total
from student as st left join score as s on st.s_id = s.s_id
left join course as c on s.c_id = c.c_id left join teacher as t on c.t_id = t.t_id
group by s_id having sumExp = total;
```

-- 或者

```
select s_id, s_name from student where s_id not in(  
    select s.s_id  
    from student as st left join score as s on st.s_id = s.s_id  
    left join course as c on s.c_id = c.c_id left join teacher as t on c.t_id = t.t_id  
    where t_name = '张三'  
);
```

-- 查询下周过生日同学（1998 年的 5 月 24 月 2022 年的 5 月 24 不一定是五月的第同一周）

```
select s_id, s_name, week(s_birth, 1) from student;  
select s_id, s_name from student where week(concat('2022-', substring(s_birth, 6, 5)), 1) =  
week(now(), 1) + 1;  
select week('2022-07-06', 1) - week(now(), 1);
```

-- 查询本月过生日的人

```
select * from student where month(now()) = month(s_birth);
```

-- 查询下月过生日的人(12 月的下个月是一月)

```
select * from student where (month(now()) + 1) % 12 = month(s_birth);
```