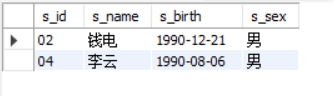
1. 查询"01"课程比"02"课程成绩高的学生的信息及课程分数

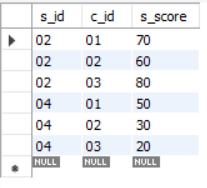
(select \* from student where s\_id in (select s\_id from score s1 where s1.c\_id like'01' and s1.s\_score>(select s\_score from

score s2 where s2.c\_id like '02' and s1.s\_id = s2.s\_id)));



select \* from score where s\_id in (select s\_id from score s1 where s1.c\_id like'01' and s1.s\_score>(select s\_score from

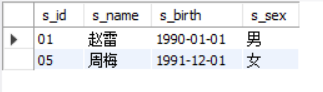
score s2 where s2.c\_id like '02' and s1.s\_id = s2.s\_id));



2、查询"01"课程比"02"课程成绩低的学生的信息及课程分数

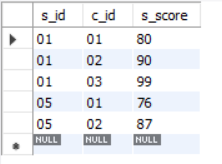
(select \* from student where s\_id in (select s\_id from score s1 where s1.c\_id like'01' and s1.s\_score<(select s\_score from

score s2 where s2.c\_id like '02' and s1.s\_id = s2.s\_id)));



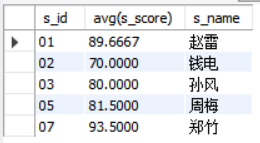
select \* from score where s\_id in (select s\_id from score s1 where s1.c\_id like'01' and s1.s\_score<(select s\_score from

score s2 where s2.c\_id like '02' and s1.s\_id = s2.s\_id));



3、查询平均成绩大于等于60分的同学的学生编号和学生姓名和平均成绩

select avg\_60.\*,student.s\_name from (select s\_id,avg(s\_score) from score group by s\_id having avg(s\_score)>60) avg\_60,student where student.s\_id =avg\_60.s\_id



选择特定表的特定元素时，一定要select student.\* 限定在某一个表不然会select错误

1. 查询平均成绩小于60分的同学的学生编号和学生姓名和平均成绩 (包括有成绩的和无成绩的)

Select （元素1,2,3） from table left outer join table on(条件)

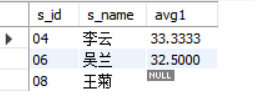
select term.\* from

(select student.s\_id,student.s\_name,avg1

from student left join

(select avg\_60.\*,student.s\_name from (select s\_id ,avg(s\_score) avg1 from score group by s\_id ) avg\_60 ,student where avg\_60.s\_id = student.s\_id) avg2

on avg2.s\_id =student.s\_id) term where avg1 <60 OR term.s\_id like '08'



1. 查询所有同学的学生编号、学生姓名、选课总数、所有课程的总成绩

select ss1.\*,ss2.cnt from

(select student.s\_id,student.s\_name,avg1

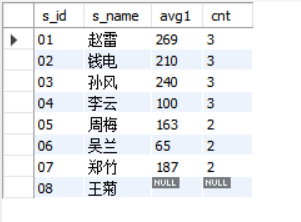
from student left join

(select avg\_60.\*,student.s\_name from (select s\_id ,sum(s\_score) avg1 from score group by s\_id ) avg\_60 ,student where avg\_60.s\_id = student.s\_id) avg2

on avg2.s\_id =student.s\_id) ss1 ,

(select student.s\_id,cnt from student left join (select s\_id, count(c\_id) cnt from score group by s\_id ) s1 on s1.s\_id = student.s\_id) ss2

where ss1.s\_id =ss2.s\_id



6、查询"李"姓老师的数量

select count(t\_name) li from teacher where t\_name like '李%'

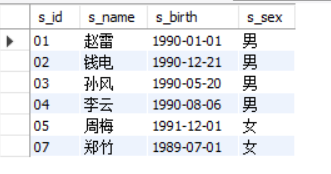


7.查询学过"张三"老师授课的同学的信息

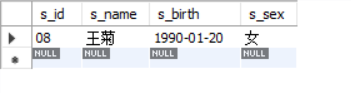
select student.\* from student,

(select score.s\_id from score, (select course.c\_id from course where course.t\_id = '01' ) s1 where score.c\_id = s1.c\_id ) s2

where student.s\_id =s2.s\_id



1. 查询没学过"张三"老师授课的同学的信息



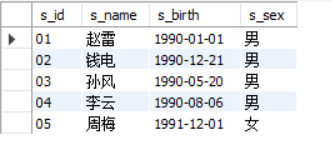
1. 查询学过编号为"01"并且也学过编号为"02"的课程的同学的信息

select student.\* from student,

(select score.s\_id from score, (select course.c\_id from course where course.t\_id = '01' ) s1 where score.c\_id = s1.c\_id ) s2,

(select score.s\_id from score, (select course.c\_id from course where course.t\_id = '02' ) s3 where score.c\_id = s3.c\_id ) s4

where student.s\_id =s2.s\_id and s2.s\_id =s4.s\_id



Not in 后面只能接表

1. 查询学过编号为"01"但是没有学过编号为"02"的课程的同学的信息

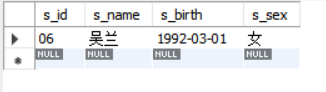
Group by 是聚集操作只能同时对多个数操作

select student.\* from student where student.s\_id in

(select score.s\_id from score where score.c\_id ='01'

)

and student.s\_id not in (select score.s\_id from score where score.c\_id ='02')



11、查询没有学全所有课程的同学的信息

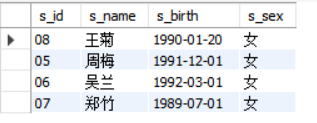
select distinct student.\* from student,

(select s\_id from score group by score.s\_id having count(s\_score) !=3) s1

where student.s\_id =s1.s\_id or student.s\_id in

(select distinct student.s\_id from student,score where student.s\_id not in

(select score.s\_id from score ))



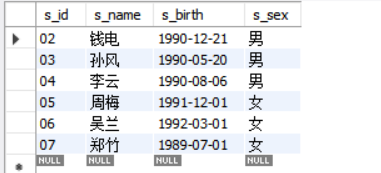
12、查询至少有一门课与学号为"01"的同学所学相同的同学的信息

select student.\* from student where student.s\_id in

(select distinct s1.s\_id from

(select score.s\_id,score.c\_id from score where s\_id !='01') s1

where s1.c\_id = '01' or s1.c\_id ='02' or s1.c\_id ='03' )



13、查询和"01"号的同学学习的课程完全相同的其他同学的信息

select student.\* from student,

(select s1.s\_id from

(select score.s\_id,count(score.s\_score) cnt1 from score group by score.s\_id ) s1

where s1.cnt1=3) s2

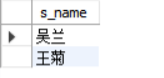
where s2.s\_id = student.s\_id



14、查询没学过"张三"老师讲授的任一门课程的学生姓名

select student.s\_name from student where student.s\_id not in

(select score.s\_id from score where score.c\_id ='02')



15、查询两门及其以上不及格课程的同学的学号，姓名及其平均成绩

select student.s\_id,student.s\_name,s2.avg1

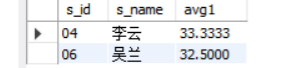
from student,

(select s1.s\_id,avg(s1.s\_score) avg1 from

(select score.\* from score where score.s\_score <60) s1

group by s1.s\_id having count(s1.s\_id) >= 2) s2

where student.s\_id = s2.s\_id



16、检索"01"课程分数小于60，按分数降序排列的学生信息

select student.\*,s3.s\_score from student,

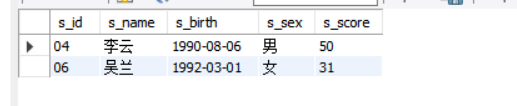
(select s2.\* from

(select s1.s\_id,s1.s\_score from

(select score.\* from score where score.c\_id ='01') s1

where s1.s\_score <60) s2 order by s2.s\_score desc) s3

where s3.s\_id = student.s\_id



17、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 \* from

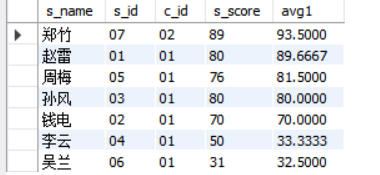
(select student.s\_name,s1.\* from student,

(select score.\*,avg(s\_score) avg1 from score

group by s\_id ) s1

where s1.s\_id = student.s\_id) s2

order by avg1 desc



18.查询各科成绩最高分、最低分和平均分：

以如下形式显示：

课程ID，课程name，最高分，最低分，平均分，及格率，中等率，优良率，优秀率

--及格为>=60，中等为：70-80，优良为：80-90，优秀为：>=90

select s1.c\_id,course.c\_name,s1.\* from course,

(select c\_id,

max(score.s\_score) '最高分',

min(score.s\_score) '最低分',

avg(score.s\_score) '平均分',

sum(case when s\_score>=60 then 1 else 0 end)/count(s\_id) as '及格率',

sum(case when s\_score>=70 and s\_score<=80 then 1 else 0 end)/count(s\_id) as '中等率',

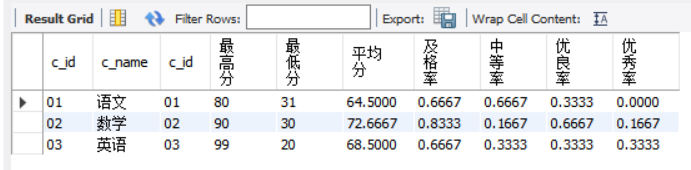
sum(case when s\_score>=80 and s\_score<=90 then 1 else 0 end)/count(s\_id) as '优良率',

sum(case when s\_score>=90 then 1 else 0 end)/count(s\_id) as '优秀率'

from score

group by c\_id) s1

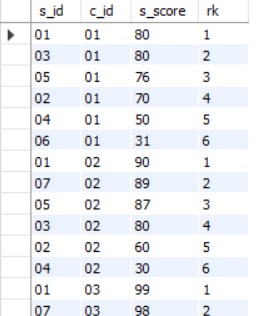
where s1.c\_id=course.c\_id

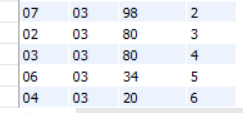


19、按各科成绩进行排序，并显示排名(实现不完全)

select s\_id,c\_id,s\_score,row\_number()

over(partition by c\_id order by s\_score desc) rk from score





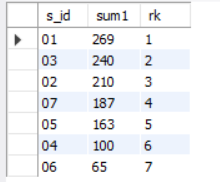
20、查询学生的总成绩并进行排名

select s1.s\_id,s1.sum1,row\_number()

over () rk

from (select s\_id,sum(s\_score) sum1 from score group by s\_id) s1

order by s1.sum1 desc



21、查询不同老师所教不同课程平均分从高到低显示

select teacher.t\_name,s2.avg1,row\_number()

over(order by avg1 desc) rk

from teacher,

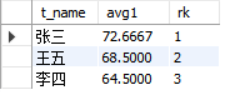
(select course.t\_id,s1.avg1 from course,

(select score.c\_id,avg(s\_score) avg1 from

score group by c\_id) s1

where course.c\_id =s1.c\_id) s2

where s2.t\_id=teacher.t\_id



-- 22、查询所有课程的成绩第2名到第3名的学生信息及该课程成绩

select student.\*,s2.c\_id,s2.s\_score,s2.rk from student,

(select s1.\* from

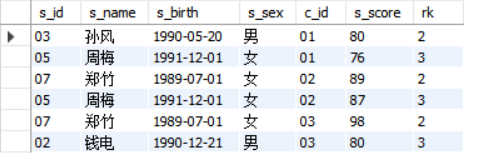
(select score.\* ,row\_number()

over(partition by c\_id order by s\_score desc) rk

from score ) s1

where rk =3 or rk=2) s2

where student.s\_id = s2.s\_id



-- 23、统计各科成绩各分数段人数：

课程编号,课程名称,[100-85],[85-70],[70-60],[0-60]及所占百分比

select course.c\_name,s1.\* from course,

(select c\_id ,

sum(case when s\_score>=0 and s\_score<60 then 1 else 0 end)/count(s\_id) as '[0-60]',

sum(case when s\_score>=60 and s\_score<70 then 1 else 0 end )/count(s\_id) as '[70-60]',

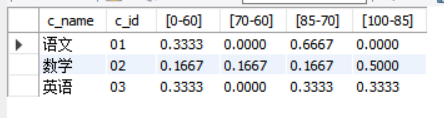
sum(case when s\_score>=70 and s\_score<85 then 1 else 0 end )/count(s\_id) as '[85-70]',

sum(case when s\_score>=85 and s\_score<=100 then 1 else 0 end )/count(s\_id) as'[100-85]'

from score

group by c\_id) s1

where course.c\_id =s1.c\_id



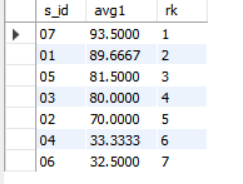
24、查询学生平均成绩及其名次

select s1.\*,row\_number()

over(order by s1.avg1 desc) rk from

(select s\_id,avg(s\_score) avg1 from score

group by s\_id) s1



25、查询各科成绩前三名的记录

1.选出b表比a表成绩大的所有组

2.选出比当前id成绩大的 小于三个的

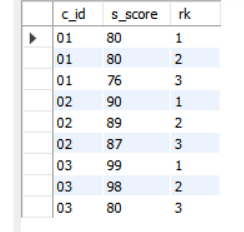
select s1.c\_id,s1.s\_score,s1.rk from

(select \*,row\_number()

over(partition by c\_id order by s\_score desc) rk

from score) s1

where rk = 1 or rk =2 or rk = 3

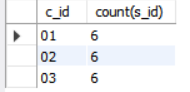


26、查询每门课程被选修的学生数

select score.c\_id ,count(s\_id)

from score

group by c\_id



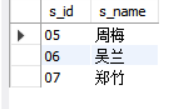
27、查询出只有两门课程的全部学生的学号和姓名

select student.s\_id,student.s\_name from student,

(select \* from score

group by s\_id having count(c\_id) =2) s1

where s1.s\_id =student.s\_id



28、查询男生、女生人数

select

sum(case when s\_sex = '男' then 1 else 0 end) boy,

sum(case when s\_sex = '女' then 1 else 0 end) girl

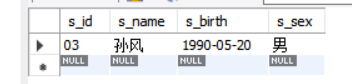
from student



29、查询名字中含有"风"字的学生信息

select student.\* from student

where student.s\_name like '%风%'



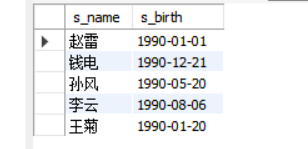
-- 30、查询同名同性学生名单，并统计同名人数

-- 31、查询1990年出生的学生名单

select s\_name,s\_birth

from student

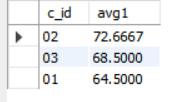
where year(s\_birth) =1990



- 32、查询每门课程的平均成绩，结果按平均成绩降序排列，平均成绩相同时，按课程编号升序排列

select c\_id ,avg(s\_score) avg1 from score

group by c\_id order by avg1 desc,c\_id

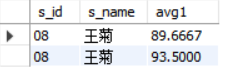


33、查询平均成绩大于等于85的所有学生的学号、姓名和平均成绩

select student.s\_id,student.s\_name,avg(score.s\_score) avg1

from student,score

group by score.s\_id having avg1>=85



34、查询课程名称为"数学"，且分数低于60的学生姓名和分数

select student.s\_name,s2.s\_score from student,

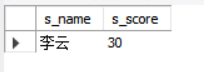
(select s\_id,s\_score from score,

(select c\_id from course

where c\_name like '数学') s1

where s1.c\_id = score.c\_id) s2

where s2.s\_score<60 and student.s\_id = s2.s\_id



35、查询所有学生的课程及分数情况

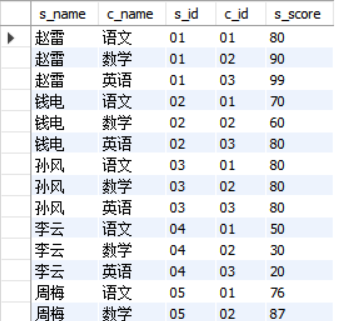
select s1.s\_name,course.c\_name,s1.s\_id,s1.c\_id,s1.s\_score from course,

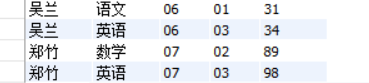
(select student.s\_name,score.\*

from student,score

where student.s\_id = score.s\_id) s1

where s1.c\_id = course.c\_id





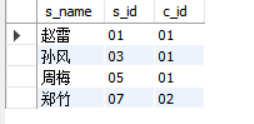
36、查询任何一门课程成绩在70分以上的姓名、课程名称和分数；

select student.s\_name ,s1.\* from student,

(select s\_id,c\_id from score

group by s\_id having min(s\_score)>70) s1

where student.s\_id =s1.s\_id



37、查询不及格的课程

select student.s\_name, s2.\* from student,

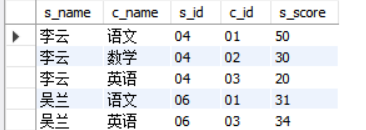
(select course.c\_name ,s1.\* from course,

(select s\_id,score.c\_id,s\_score from score

where score.s\_score <60) s1

where course.c\_id = s1.c\_id) s2

where student.s\_id = s2.s\_id



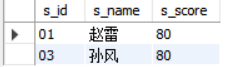
38、查询课程编号为01且课程成绩在80分以上的学生的学号和姓名；

select student.s\_id,student.s\_name,s1.s\_score from student,

(select s\_id,s\_score from score

where c\_id ='01' and s\_score >=80) s1

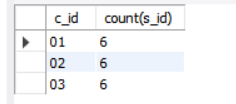
where s1.s\_id =student.s\_id



39、求每门课程的学生人数

select c\_id,count(s\_id) from score

group by c\_id



40、查询选修"张三"老师所授课程的学生中，成绩最高的学生信息及其成绩

select student.\*,s2.c\_id,s2.m1 from student,

(select s\_id,c\_id,max(s\_score) m1 from

(select s\_id,c\_id,s\_score from score where score.c\_id =

(select course.c\_id from teacher,course

where teacher.t\_id =course.t\_id and teacher.t\_name like '张三') ) s1) s2

where s2.s\_id = student.s\_id

