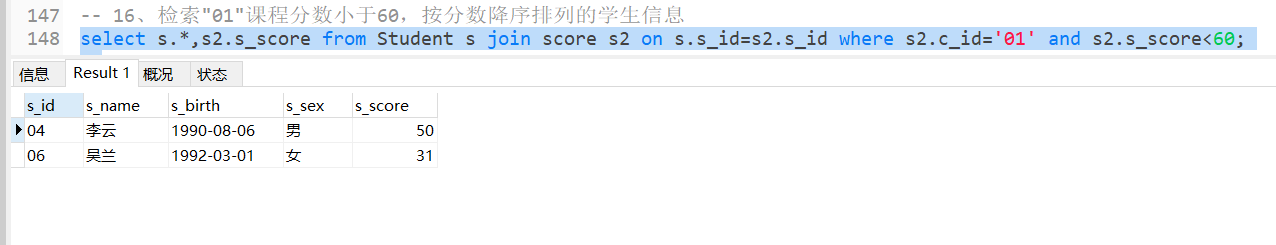
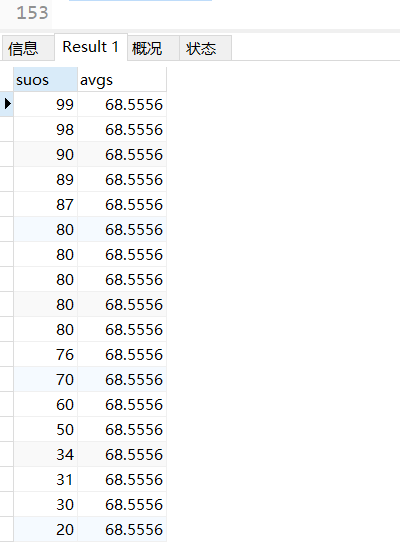
-- 16、检索"01"课程分数小于60，按分数降序排列的学生信息

select s.\*,s2.s\_score from Student s join score s2 on s.s\_id=s2.s\_id where s2.c\_id='01' and s2.s\_score<60;



-- 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

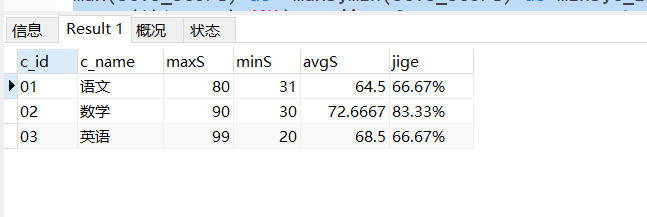
select s2.s\_score as suos,(select avg(score.s\_score) as a from score ORDER BY a desc) as avgs from score s2 order by suos desc;



-- 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

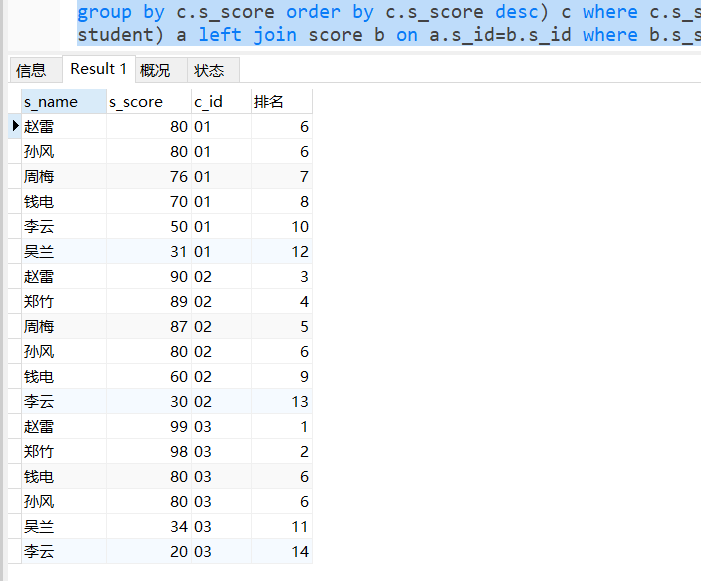
--及格为>=60，中等为：70-80，优良为：80-90，优秀为：>=90

select distinct c.c\_id,c.c\_name,B.maxS,B.minS,B.avgS,b.jige from Course c left join (select avg(sc.s\_score) as avgS,max(sc.s\_score) as maxS,min(sc.s\_score) as minS,c\_id, CONCAT(round(sum(case when s\_score>=60 then 1 else 0 end)/count(\*)\*100,2),'%') as jige from Score sc group by sc.c\_id) B on c.c\_id=B.c\_id;



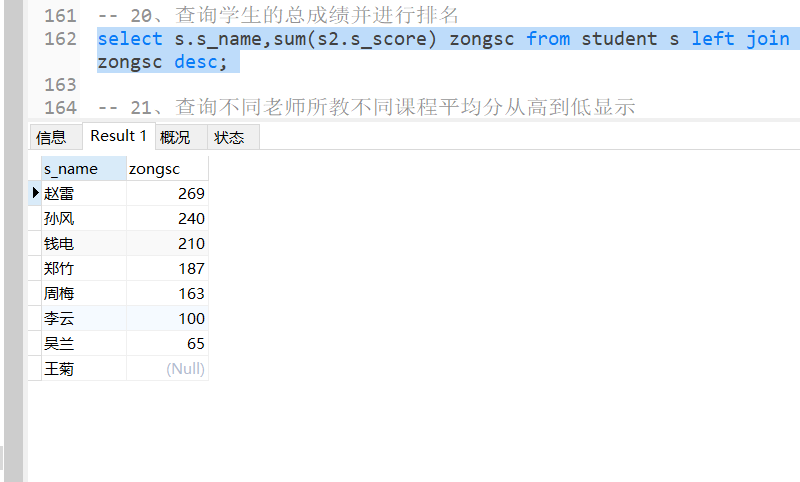
-- 19、按各科成绩进行排序，并显示排名

select a.s\_name,b.s\_score,b.c\_id,(select count(c.s\_score)+1 from (select c.s\_score,count(c.s\_score) from Score c group by c.s\_score order by c.s\_score desc) c where c.s\_score > b.s\_score) as '排名' from (select s\_id,s\_name from student) a left join score b on a.s\_id=b.s\_id where b.s\_score>0 order by b.c\_id,b.s\_score desc;



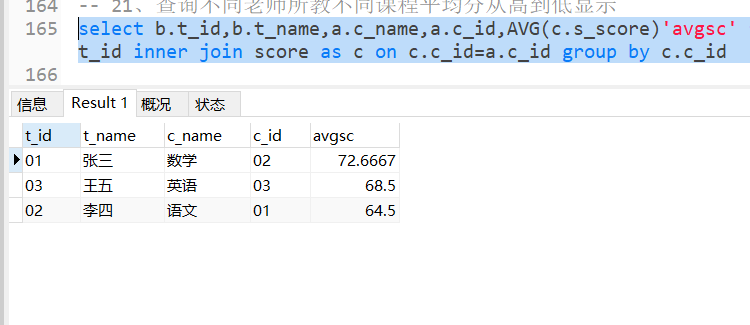
-- 20、查询学生的总成绩并进行排名

select s.s\_name,sum(s2.s\_score) zongsc from student s left join score s2 on s.s\_id=s2.s\_id group by s.s\_id order by zongsc desc;



-- 21、查询不同老师所教不同课程平均分从高到低显示

select b.t\_id,b.t\_name,a.c\_name,a.c\_id,AVG(c.s\_score)'avgsc' from course as a inner join teacher as b on a.t\_id=b.t\_id inner join score as c on c.c\_id=a.c\_id group by c.c\_id order by avgsc desc;



-- 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 a.\* from (

select st.\*,c.c\_id,c.c\_name,sc.s\_score from student st

left join score sc on sc.s\_id=st.s\_id

inner join course c on c.c\_id =sc.c\_id and c.c\_id="01"

order by sc.s\_score desc LIMIT 1,2 ) a

union all

select b.\* from (

select st.\*,c.c\_id,c.c\_name,sc.s\_score from student st

left join score sc on sc.s\_id=st.s\_id

inner join course c on c.c\_id =sc.c\_id and c.c\_id="02"

order by sc.s\_score desc limit 1,2) b

union all

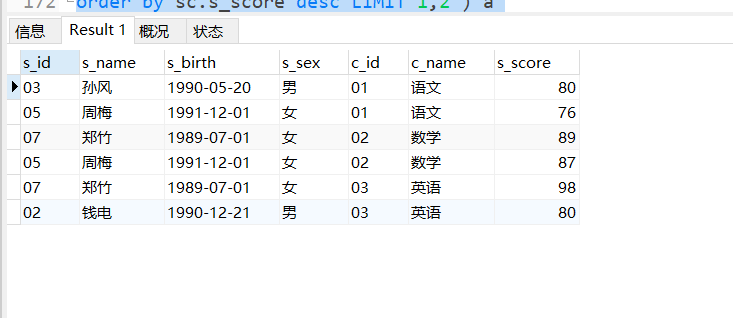
select c.\* from (

select st.\*,c.c\_id,c.c\_name,sc.s\_score from student st

left join score sc on sc.s\_id=st.s\_id

inner join course c on c.c\_id =sc.c\_id and c.c\_id="03"

order by sc.s\_score desc limit 1,2) c



-- 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 sc.c\_id,co.c\_name,SUM(case when sc.s\_score<=100 and sc.s\_score>=85 then 1 else 0 end)'[100-85]',

SUM(case when sc.s\_score<=85 and sc.s\_score>=70 then 1 else 0 end)'[85-70]',

COUNT(case when sc.s\_score<=70 and sc.s\_score>=60 then 521 else null end)'[70-60]',

SUM(case when sc.s\_score<=60 then 1 else 0 end)'[0-60]'

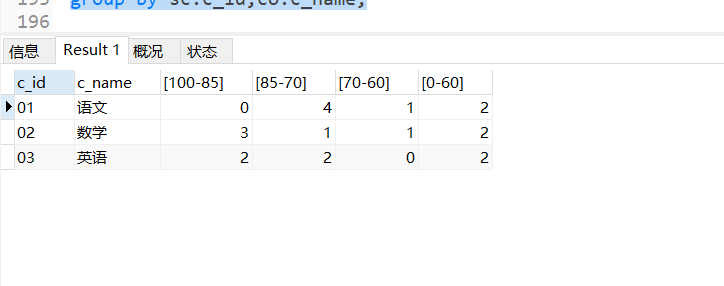
from

score as sc

inner join

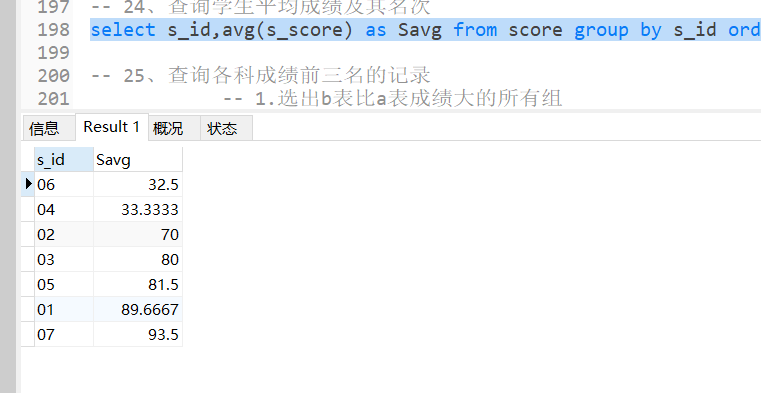
course as co on sc.c\_id=co.c\_id

group by sc.c\_id,co.c\_name;



-- 24、查询学生平均成绩及其名次

select s\_id,avg(s\_score) as Savg from score group by s\_id order by Savg asc;



-- 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 a.\* from (

select st.s\_id,st.s\_name,c.c\_id,c.c\_name,sc.s\_score from student st

left join score sc on sc.s\_id=st.s\_id

inner join course c on c.c\_id=sc.c\_id and c.c\_id='01'

order by sc.s\_score desc limit 0,3) a

union all

select b.\* from (

select st.s\_id,st.s\_name,c.c\_id,c.c\_name,sc.s\_score from student st

left join score sc on sc.s\_id=st.s\_id

inner join course c on c.c\_id=sc.c\_id and c.c\_id='02'

order by sc.s\_score desc limit 0,3) b

union all

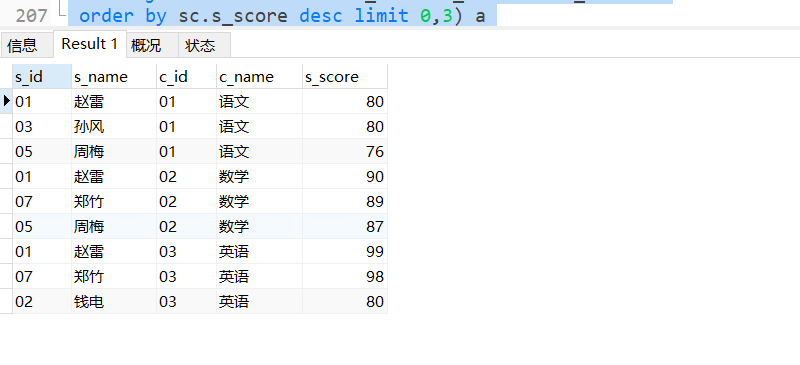
select c.\* from (

select st.s\_id,st.s\_name,c.c\_id,c.c\_name,sc.s\_score from student st

left join score sc on sc.s\_id=st.s\_id

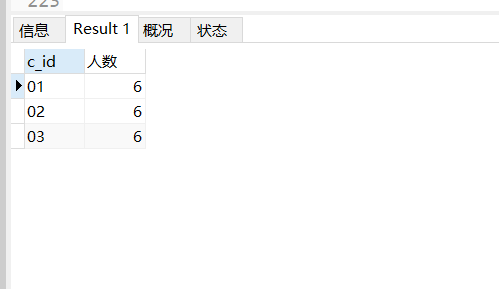
inner join course c on c.c\_id=sc.c\_id and c.c\_id='03'

order by sc.s\_score desc limit 0,3) c



-- 26、查询每门课程被选修的学生数

select c\_id,count(s\_id)'人数' 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 st.s\_id,st.s\_name,count(sc.c\_id)'选修课程数两门'from

student as st

inner join

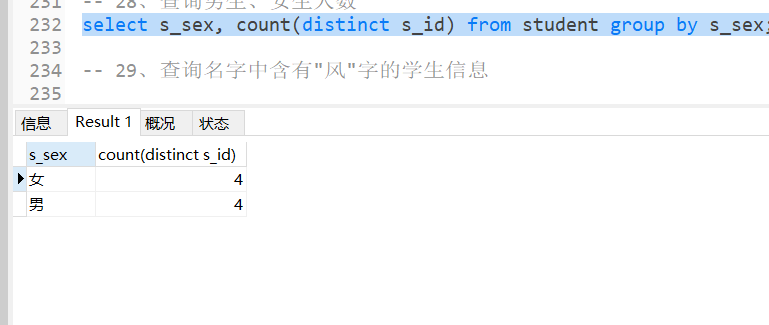
score as sc on st.s\_id=sc.s\_id

group by st.s\_id having count(distinct sc.c\_id)=2;



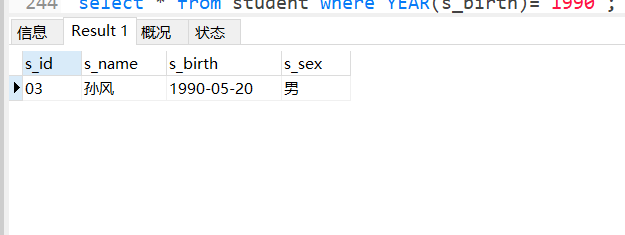
-- 28、查询男生、女生人数

select s\_sex, count(distinct s\_id) from student group by s\_sex;



-- 29、查询名字中含有"风"字的学生信息

select \* from Student where s\_name like "%风%";

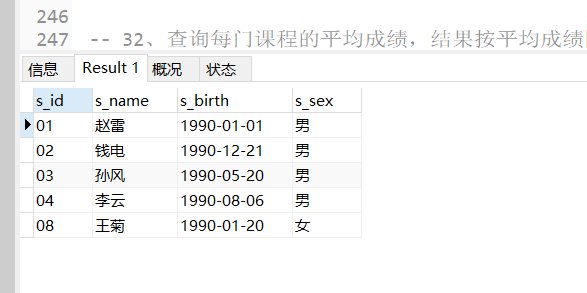


-- 30、查询同名同性学生名单，并统计同名人数

select st.\*,count(1) from student st group by st.s\_name,st.s\_sex having count(1)>1;

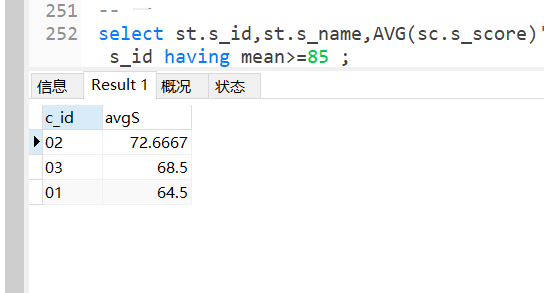
-- 31、查询1990年出生的学生名单

select \* 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) as avgS from score group by c\_id order by avgS desc,c\_id asc;



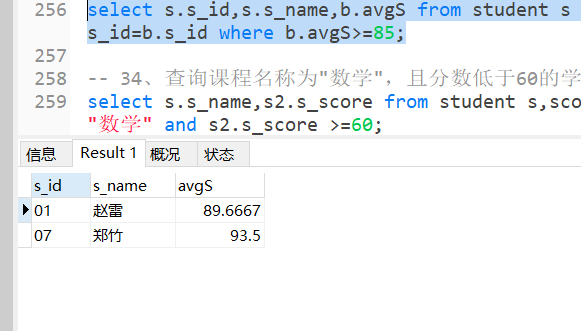
-- 33、查询平均成绩大于等于85的所有学生的学号、姓名和平均成绩

-- 一

select st.s\_id,st.s\_name,AVG(sc.s\_score)'mean' from student as st inner join score as sc on st.s\_id=sc.s\_id group by s\_id having mean>=85 ;

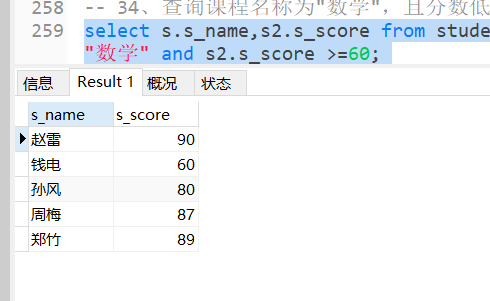
-- 二

select s.s\_id,s.s\_name,b.avgS from student s join (select s\_id,avg(s\_score) avgS from score group by s\_id) b on s.s\_id=b.s\_id where b.avgS>=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 s.s\_name,s2.s\_score from student s,score s2,course c where s.s\_id=s2.s\_id and c.c\_id=s2.c\_id and c.c\_name="数学" and s2.s\_score >=60;



-- 35、查询所有学生的课程及分数情况；

select st.s\_id,st.s\_name

,MAX(case when co.c\_name='语文' then sc.s\_score else null end) as '语文'

,MAX(case when co.c\_name='数学' then sc.s\_score else null end) as '数学'

,MAX(case when co.c\_name='英语' then sc.s\_score else null end) as '英语'

from

student as st

join

score as sc on st.s\_id=sc.s\_id

join

course as co on sc.c\_id=co.c\_id

group by st.s\_id,st.s\_name;

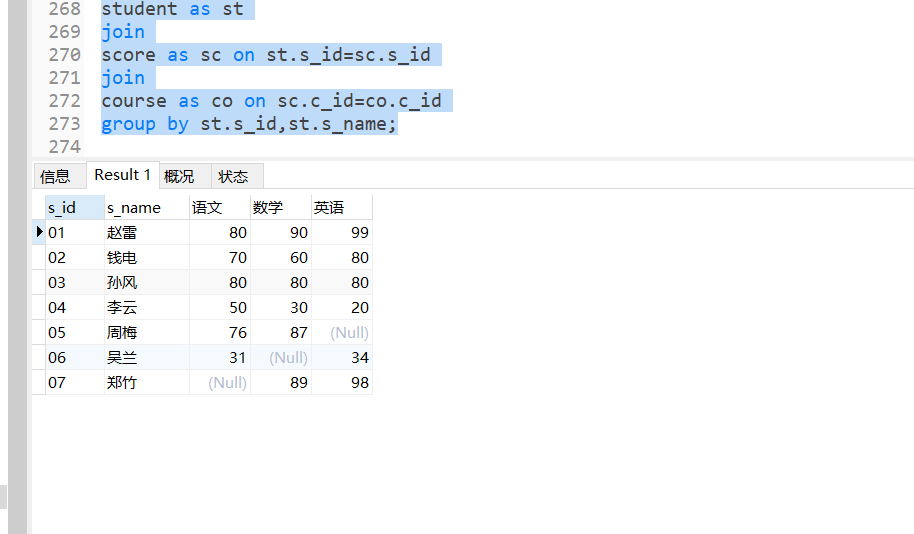
-- 二

select st.s\_id,st.s\_name,c.c\_name,sc.s\_score from student st

left join score sc on sc.s\_id=st.s\_id

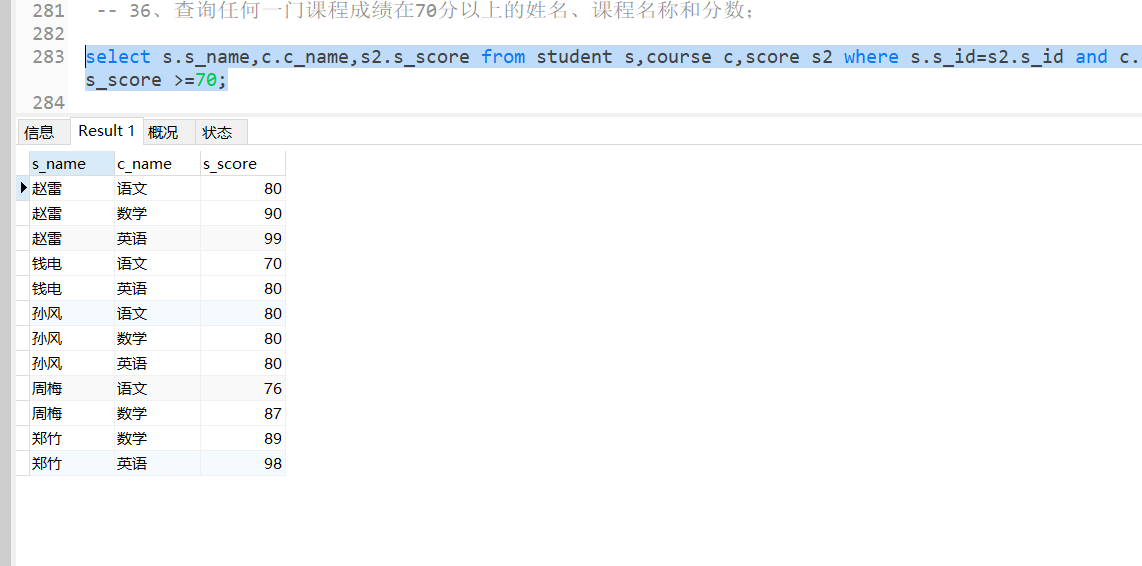
left join course c on c.c\_id =sc.c\_id

order by st.s\_id,c.c\_name;



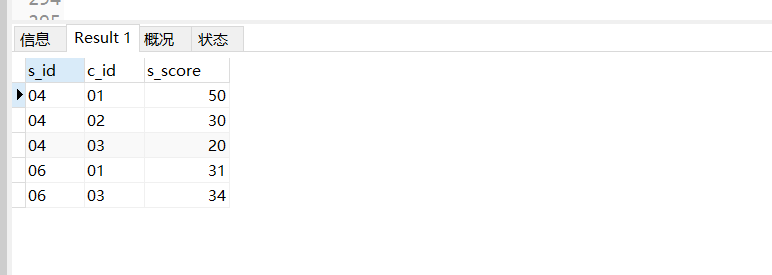
-- 36、查询任何一门课程成绩在70分以上的姓名、课程名称和分数；

select s.s\_name,c.c\_name,s2.s\_score from student s,course c,score s2 where s.s\_id=s2.s\_id and c.c\_id=s2.c\_id and s2.s\_score >=70;



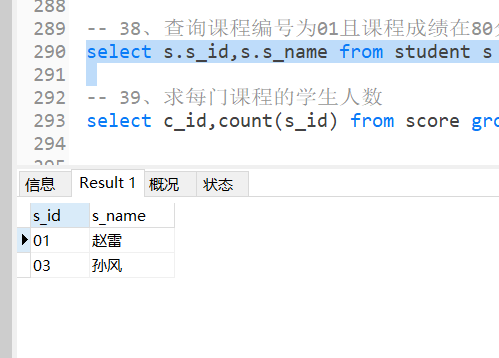
-- 37、查询不及格的课程

select \* from score where s\_score <60;



-- 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 s.s\_id,s.s\_name from student s join score s2 on s.s\_id=s2.s\_id where s2.c\_id=01 and s2.s\_score >=80;



-- 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

-- 查询老师id

-- 查询最高分（可能有相同分数）

-- 查询信息

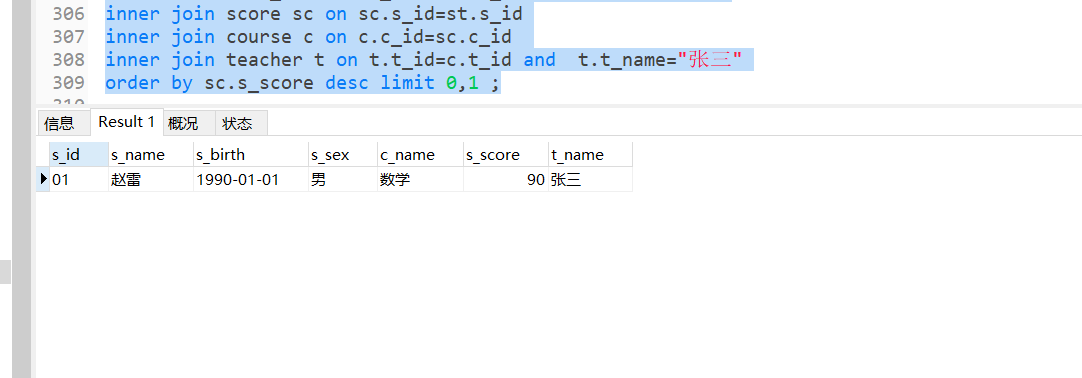
select st.\*,c.c\_name,sc.s\_score,t.t\_name from student st

inner join score sc on sc.s\_id=st.s\_id

inner join course c on c.c\_id=sc.c\_id

inner join teacher t on t.t\_id=c.t\_id and t.t\_name="张三"

order by sc.s\_score desc limit 0,1 ;



-- 41、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 st.s\_id,st.s\_name,sc.c\_id,sc.s\_score from student st

left join score sc on sc.s\_id=st.s\_id

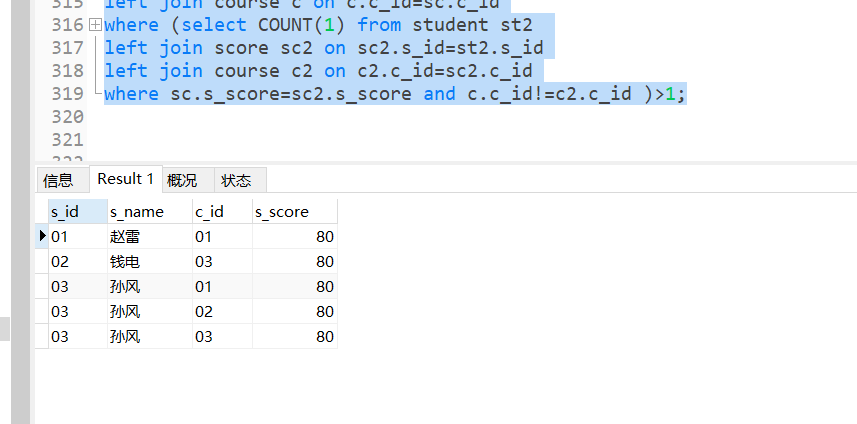
left join course c on c.c\_id=sc.c\_id

where (select COUNT(1) from student st2

left join score sc2 on sc2.s\_id=st2.s\_id

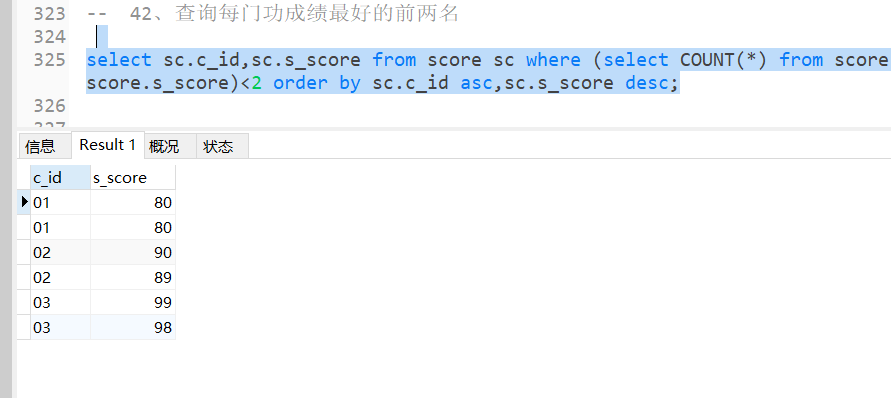
left join course c2 on c2.c\_id=sc2.c\_id

where sc.s\_score=sc2.s\_score and c.c\_id!=c2.c\_id )>1;



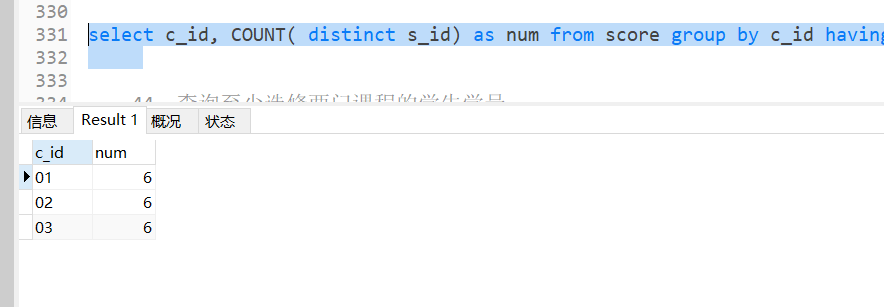
-- 42、查询每门功成绩最好的前两名

select sc.c\_id,sc.s\_score from score sc where (select COUNT(\*) from score where sc.c\_id = score.c\_id and sc.s\_score<score.s\_score)<2 order by sc.c\_id asc,sc.s\_score desc;



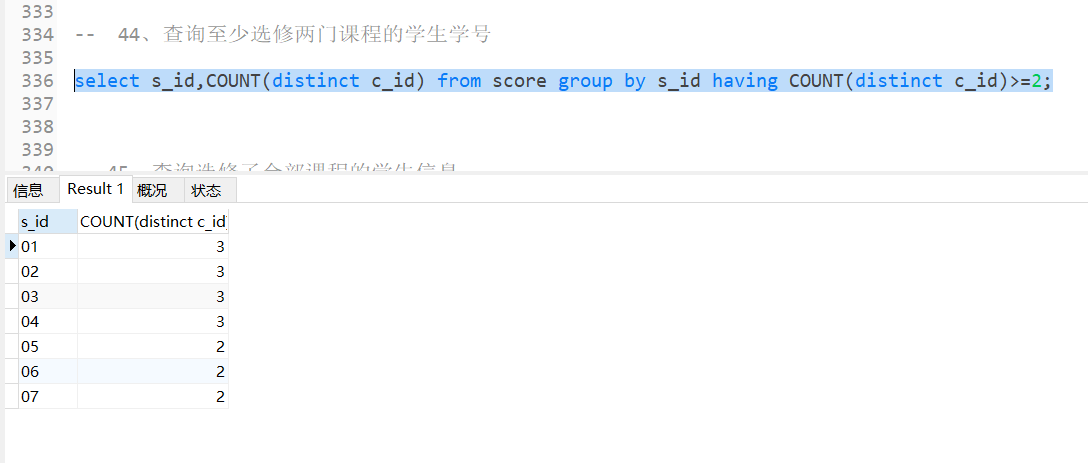
-- 43、统计每门课程的学生选修人数（超过5人的课程才统计）。要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列

select c\_id, COUNT( distinct s\_id) as num from score group by c\_id having num>5 order by num desc,c\_id asc;



-- 44、查询至少选修两门课程的学生学号

select s\_id,COUNT(distinct c\_id) from score group by s\_id having COUNT(distinct c\_id)>=2;

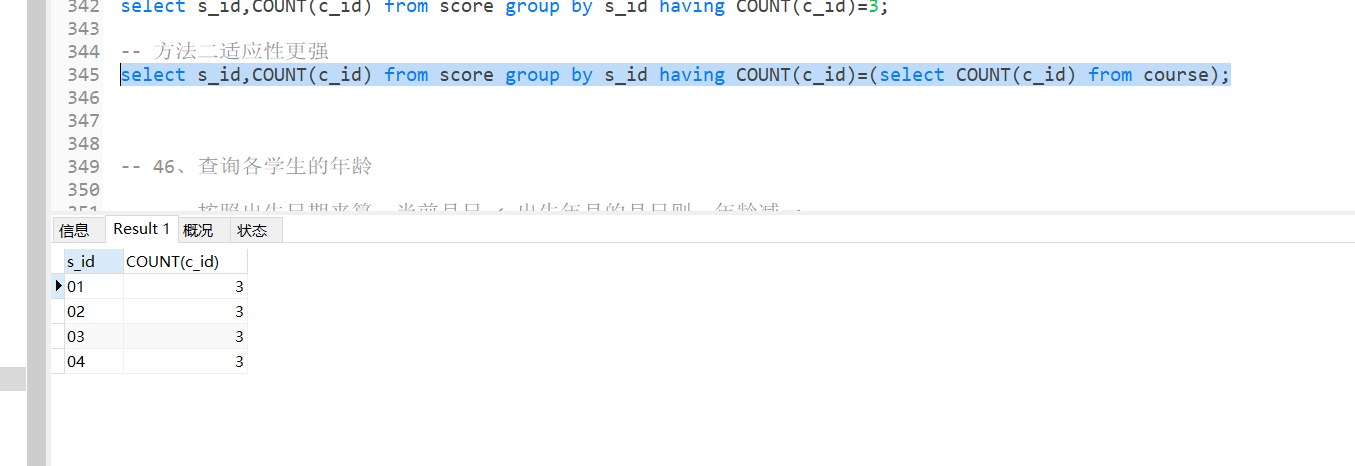


-- 45、查询选修了全部课程的学生信息

select s\_id,COUNT(c\_id) from score group by s\_id having COUNT(c\_id)=3;

-- 方法二适应性更强

select s\_id,COUNT(c\_id) from score group by s\_id having COUNT(c\_id)=(select COUNT(c\_id) from course);



-- 46、查询各学生的年龄

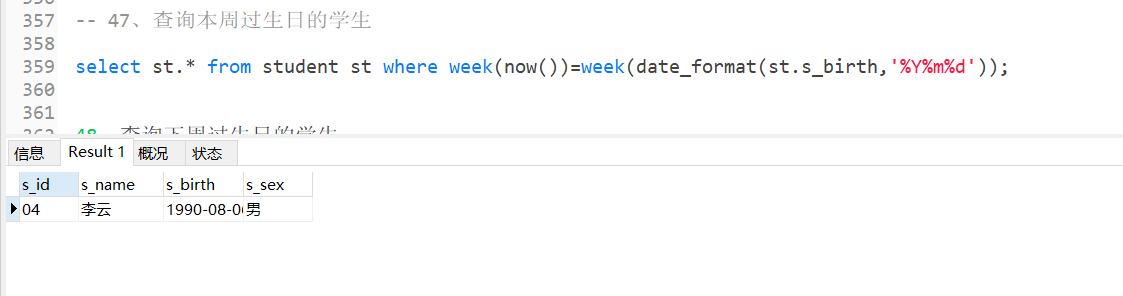
-- 按照出生日期来算，当前月日 < 出生年月的月日则，年龄减一

select \*,(case when convert(int,'1'+substring(convert(varchar(10),s\_birth,112),5,8))<convert(int,'1'+substring(convert(varchar(10),getDate(),112)，5，8))then datediff(yy,s\_birth,getdate()) else datediff(yy,sage,getdate())-1

end) s\_birth from student;

-- 47、查询本周过生日的学生

select st.\* from student st where week(now())=week(date\_format(st.s\_birth,'%Y%m%d'));



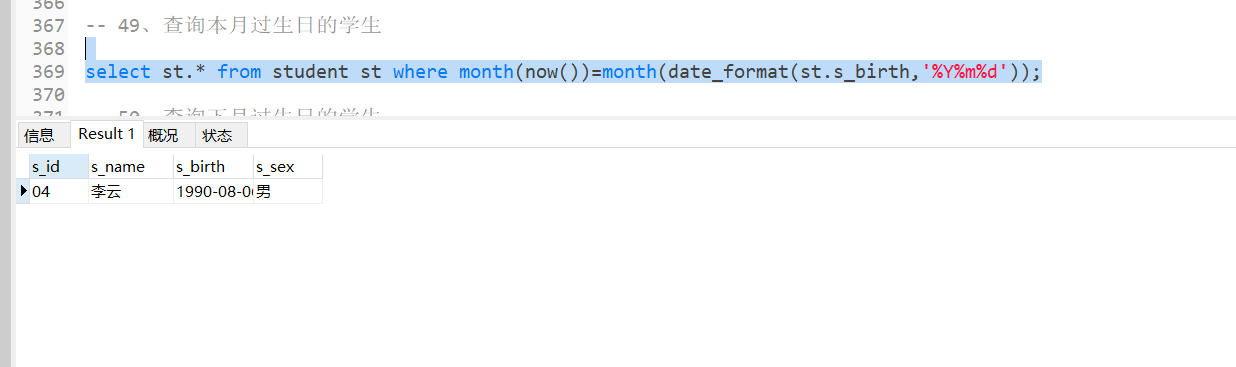
48、查询下周过生日的学生

select st.\* from student st where week(now())+1=week(date\_format(st.s\_birth,'%Y%m%d'));



-- 49、查询本月过生日的学生

select st.\* from student st where month(now())=month(date\_format(st.s\_birth,'%Y%m%d'));



-- 50、查询下月过生日的学生

select st.\* from student st where month(timestampadd(month,1,now()))=month(date\_format(st.s\_birth,'%Y%m%d'));

-- 或

select st.\* from student st where (month(now()) + 1) mod 12 = month(date\_format(st.s\_birth,'%Y%m%d'));

