create table student( 学生表

sno varchar(10) primary key,

sname varchar(20),

sage int(2),

ssex varchar(5)

);

create table teacher( 教师表

tno varchar(10) primary key,

tname varchar(20)

);

create table course( 课程表

cno varchar(10),

cname varchar(20),

tno varchar(20),

constraint pk\_course primary key (cno,tno)

);

create table sc( 分数表

sno varchar(10),

cno varchar(10),

score float(4,2),

constraint pk\_sc primary key (sno,cno)

);

/\*\*\*\*\*\*\*初始化学生表的数据\*\*\*\*\*\*/

insert into student values ('s001','张三',23,'男');

insert into student values ('s002','李四',23,'男');

insert into student values ('s003','吴鹏',25,'男');

insert into student values ('s004','琴沁',20,'女');

insert into student values ('s005','王丽',20,'女');

insert into student values ('s006','李波',21,'男');

insert into student values ('s007','刘玉',21,'男');

insert into student values ('s008','萧蓉',21,'女');

insert into student values ('s009','陈萧晓',23,'女');

insert into student values ('s010','陈美',22,'女');

commit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*初始化教师表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

insert into teacher values ('t001', '刘阳');

insert into teacher values ('t002', '谌燕');

insert into teacher values ('t003', '胡明星');

commit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*初始化课程表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

insert into course values ('c001','J2SE','t002');

insert into course values ('c002','Java Web','t002');

insert into course values ('c003','SSH','t001');

insert into course values ('c004','Oracle','t001');

insert into course values ('c005','SQL SERVER 2005','t003');

insert into course values ('c006','C#','t003');

insert into course values ('c007','JavaScript','t002');

insert into course values ('c008','DIV+CSS','t001');

insert into course values ('c009','PHP','t003');

insert into course values ('c010','EJB3.0','t002');

commit;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*初始化成绩表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

insert into sc values ('s001','c001',78.9);

insert into sc values ('s002','c001',80.9);

insert into sc values ('s003','c001',81.9);

insert into sc values ('s004','c001',60.9);

insert into sc values ('s001','c002',82.9);

insert into sc values ('s002','c002',72.9);

insert into sc values ('s003','c002',81.9);

insert into sc values ('s001','c003','59');

commit;

练习题：

1、查询“c001”课程比“c002”课程成绩高的所有学生的学号；

2、查询平均成绩大于60 分的同学的学号和平均成绩；

3、查询所有同学的学号、姓名、选课数、总成绩；

4、查询姓“刘”的老师的个数；

5、查询没学过“谌燕”老师课的同学的学号、姓名；

6、查询学过“c001”并且也学过编号“c002”课程的同学的学号、姓名；

7、查询学过“谌燕”老师所教的所有课的同学的学号、姓名；

8、查询课程编号“c002”的成绩比课程编号“c001”课程低的所有同学的学号、姓名；

9、查询所有课程成绩小于60 分的同学的学号、姓名；

10、查询没有学全所有课的同学的学号、姓名；

11、查询至少有一门课与学号为“s001”的同学所学相同的同学的学号和姓名；

12、查询至少学过学号为“s001”同学所有一门课的其他同学学号和姓名；

13、把“SC”表中“谌燕”老师教的课的成绩都更改为此课程的平均成绩；

14、查询和“s001”号的同学学习的课程完全相同的其他同学学号和姓名；

15、删除学习“谌燕”老师课的SC 表记录；

16、向SC 表中插入一些记录，这些记录要求符合以下条件：没有上过编号“c002”课程的同学学号、“c002”号课的平均成绩；

17、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

18、按各科平均成绩从低到高和及格率的百分数从高到低顺序

19、查询不同老师所教不同课程平均分从高到低显示

20、统计列印各科成绩,各分数段人数:课程ID,课程名称,[100-85],[85-70],[70-60],[ <60]

21、查询各科成绩前三名的记录:(不考虑成绩并列情况)

22、查询每门课程被选修的学生数

23、查询出只选修了一门课程的全部学生的学号和姓名

24、查询男生、女生人数

25、查询姓“张”的学生名单

26、查询同名同性学生名单，并统计同名人数

27、1981 年出生的学生名单(注：Student 表中Sage 列的类型是number)

28、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

29、查询平均成绩大于85 的所有学生的学号、姓名和平均成绩

30、查询课程名称为“数据库”，且分数低于60 的学生姓名和分数

31、查询所有学生的选课情况；

32、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；

33、查询不及格的课程，并按课程号从大到小排列

34、查询课程编号为c001 且课程成绩在80 分以上的学生的学号和姓名；

35、求选了课程的学生人数

36、查询选修“谌燕”老师所授课程的学生中，成绩最高的学生姓名及其成绩

37、查询各个课程及相应的选修人数

38、查询不同课程成绩相同的学生的学号、课程号、学生成绩

39、查询每门功课成绩最好的前两名

40、统计每门课程的学生选修人数（超过10 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

41、检索至少选修两门课程的学生学号

42、查询全部学生都选修的课程的课程号和课程名

43、查询没学过“谌燕”老师讲授的任一门课程的学生姓名

44、查询两门以上不及格课程的同学的学号及其平均成绩

45、检索“c004”课程分数小于60，按分数降序排列的同学学号

46、删除“s002”同学的“c001”课程的成绩

答案

1.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select a.\* from

(select \* from sc a where a.cno='c001') a,

(select \* from sc b where b.cno='c002') b

where a.sno=b.sno and a.score > b.score;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select \* from sc a

where a.cno='c001'

and exists(select \* from sc b where b.cno='c002' and a.score>b.score

and a.sno = b.sno)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select sno,avg(score) from sc group by sno having avg(score)>60;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select a.\*,s.sname from (select sno,sum(score),count(cno) from sc group by sno) a ,student s where a.sno=s.sno

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select count(\*) from teacher where tname like '刘%';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

5.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select a.sno,a.sname from student a

where a.sno

not in

(select distinct s.sno

from sc s,

(select c.\*

from course c ,

(select tno

from teacher t

where tname='谌燕')t

where c.tno=t.tno) b

where s.cno = b.cno )

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select \* from student st where st.sno not in

(select distinct sno from sc s join course c on s.cno=c.cno

join teacher t on c.tno=t.tno where tname='谌燕')

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

6.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select st.\* from sc a

join sc b on a.sno=b.sno

join student st

on st.sno=a.sno

where a.cno='c001' and b.cno='c002' and st.sno=a.sno;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select st.\* from student st join sc s on st.sno=s.sno

join course c on s.cno=c.cno

join teacher t on c.tno=t.tno

where t.tname='谌燕'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select \* from student st

join sc a on st.sno=a.sno

join sc b on st.sno=b.sno

where a.cno='c002' and b.cno='c001' and a.score < b.score

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select st.\*,s.score from student st

join sc s on st.sno=s.sno

join course c on s.cno=c.cno

where s.score <60

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select stu.sno,stu.sname,count(sc.cno) from student stu

left join sc on stu.sno=sc.sno

group by stu.sno,stu.sname

having count(sc.cno)<(select count(distinct cno)from course)

===================================

select \* from student where sno in

(select sno from

(select stu.sno,c.cno from student stu

cross join course c

minus

select sno,cno from sc)

)

===================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select st.\* from student st,

(select distinct a.sno from

(select \* from sc) a,

(select \* from sc where sc.sno='s001') b

where a.cno=b.cno) h

where st.sno=h.sno and st.sno<>'s001'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

12.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select \* from sc

left join student st

on st.sno=sc.sno

where sc.sno<>'s001'

and sc.cno in

(select cno from sc

where sno='s001')

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

13.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

update sc c set score=(select avg(c.score) from course a,teacher b

where a.tno=b.tno

and b.tname='谌燕'

and a.cno=c.cno

group by c.cno)

where cno in(

select cno from course a,teacher b

where a.tno=b.tno

and b.tname='谌燕')

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

14.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select\* from sc where sno<>'s001'

minus

(

select\* from sc

minus

select \* from sc where sno='s001'

)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

15.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

delete from sc

where sc.cno in

(

select cno from course c

left join teacher t on c.tno=t.tno

where t.tname='谌燕'

)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

16.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

insert into sc (sno,cno,score)

select distinct st.sno,sc.cno,(select avg(score)from sc where cno='c002')

from student st,sc

where not exists

(select \* from sc where cno='c002' and sc.sno=st.sno) and sc.cno='c002';

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

17.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select cno ,max(score),min(score) from sc group by cno;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

18.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select cno,avg(score),sum(case when score>=60 then 1 else 0 end)/count(\*)

as 及格率

from sc group by cno

order by avg(score) , 及格率desc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

19.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select max(t.tno),max(t.tname),max(c.cno),max(c.cname),c.cno,avg(score) from sc , course c,teacher t

where sc.cno=c.cno and c.tno=t.tno

group by c.cno

order by avg(score) desc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

20.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select sc.cno,c.cname,

sum(case when score between 85 and 100 then 1 else 0 end) AS "[100-85]",

sum(case when score between 70 and 85 then 1 else 0 end) AS "[85-70]",

sum(case when score between 60 and 70 then 1 else 0 end) AS "[70-60]",

sum(case when score <60 then 1 else 0 end) AS "[<60]"

from sc, course c

where sc.cno=c.cno

group by sc.cno ,c.cname;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

21.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select \* from

(select sno,cno,score,row\_number()over(partition by cno order by score desc) rn from sc)

where rn<4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

22.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select cno,count(sno)from sc group by cno;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

23.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select sc.sno,st.sname,count(cno) from student st

left join sc

on sc.sno=st.sno

group by st.sname,sc.sno having count(cno)=1;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

24.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select ssex,count(\*)from student group by ssex;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

25.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

26.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select sname,count(\*)from student group by sname having count(\*)>1;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

27.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select sno,sname,sage,ssex from student t where to\_char(sysdate,'yyyy')-sage =1988

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

28.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select cno,avg(score) from sc group by cno order by avg(score)asc,cno desc;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

29.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select st.sno,st.sname,avg(score) from student st

left join sc

on sc.sno=st.sno

group by st.sno,st.sname having avg(score)>85;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

30.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select sname,score from student st,sc,course c

where st.sno=sc.sno and sc.cno=c.cno and c.cname='Oracle' and sc.score<60

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

31.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select st.sno,st.sname,c.cname from student st,sc,course c

where sc.sno=st.sno and sc.cno=c.cno;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

32.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select st.sname,c.cname,sc.score from student st,sc,course c

where sc.sno=st.sno and sc.cno=c.cno and sc.score>70

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

33.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select sc.sno,c.cname,sc.score from sc,course c

where sc.cno=c.cno and sc.score<60 order by sc.cno desc;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

34.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select st.sno,st.sname,sc.score from sc,student st

where sc.sno=st.sno and cno='c001' and score>80;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

35.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select count(distinct sno) from sc;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

36.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select st.sname,score from student st,sc ,course c,teacher t

where

st.sno=sc.sno and sc.cno=c.cno and c.tno=t.tno

and t.tname='谌燕' and sc.score=

(select max(score)from sc where sc.cno=c.cno)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

37.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select cno,count(sno) from sc group by cno;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

38.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select a.\* from sc a ,sc b where a.score=b.score and a.cno<>b.cno

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

39.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select \* from (

select sno,cno,score,row\_number()over(partition by cno order by score desc) my\_rn from sc t

)

where my\_rn<=2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

40.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select cno,count(sno) from sc group by cno

having count(sno)>10

order by count(sno) desc,cno asc;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

41.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select sno from sc group by sno having count(cno)>1;

||

select sno from sc group by sno having count(sno)>1;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

42.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select distinct(c.cno),c.cname from course c ,sc

where sc.cno=c.cno

||

select cno,cname from course c

where c.cno in

(select cno from sc group by cno)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

43.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select st.sname from student st

where st.sno not in

(select distinct sc.sno from sc,course c,teacher t

where sc.cno=c.cno and c.tno=t.tno and t.tname='谌燕')

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

44.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select sno,avg(score)from sc

where sno in

(select sno from sc where sc.score<60

group by sno having count(sno)>1

) group by sno

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

45.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

select sno from sc where cno='c004' and score<90 order by score desc;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

46.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

delete from sc where sno='s002' and cno='c001';

mysql优化版答案

-- where条件优化：把能最大范围缩小查询结果的条件放在最右边，依次类推。

/\*查询“c001”课程比“c002”课程成绩高的所有学生的学号；\*/

select

t.sno

from

(select sno,score from sc where cno = 'c001') t,

(select sno,score from sc where cno = 'c002') t1

where t.sno = t1.sno and t.score > t1.score

/\*查询平均成绩大于60 分的同学的学号和平均成绩；\*/

select sno,score from

(select sno,avg(score) score from sc group by sno) t

where t.score > 60

/\*\*

select sno,avg(score) from sc group by sno having avg(score)>60

select sno,score from sc group by sno

语法错误：查询列不在group by子句中

解决办法：1、去掉查询列中的score

2、在group by后面添加having score > 60

select sno,score from sc group by sno having score > 60

还是语法错误，那为什么上面的就可以呢？？

因为上面avg(score)不是原表里面的列而是新的列，

它查询出来的结果列是 sno avg(score)

而我们查出来的列是sno score

\*/

/\*\*查询所有同学的学号、姓名、选课数、总成绩\*/

select t.sno,t.sname,t1.选课数,t1.总成绩 from

student t,

(select sno , count(cno) 选课数,sum(score) 总成绩 from sc group by sno) t1

where t.sno = t1.sno

/\*查询姓“刘”的老师的个数；\*/

select count(1) from teacher t where t.tname like '%刘%'

/\*查询没学过“谌燕”老师课的同学的学号、姓名；\*/

/\*查询出谌燕所教的课\*/

select cno from course t1,(select tno from teacher where tname = '谌燕') t2

where t1.tno = t2.tno

/\*查询出学过谌燕教的课的学生\*/

select sno from sc t

where t.cno in (select cno from course t1,(select tno from teacher where tname = '谌燕') t2

where t1.tno = t2.tno) group by sno

/\*查询出没有学过谌燕教的课的学生\*/

select sno from student where sno not in (select sno from sc t

where t.cno in (select cno from course t1,(select tno from teacher where tname = '谌燕') t2

where t1.tno = t2.tno) group by sno)

/\*优化，不要使用not in\*/

select cno from course t1,teacher t2 where t1.tno = t2.tno and t2.tname = '谌燕'

select distinct sno from sc t1,

(select cno from course t1,teacher t2 where t1.tno = t2.tno and t2.tname = '谌燕') t2

where t1.cno = t2.cno

select sno from student t1 where t1.sno not in

(select distinct sno from sc t1,

(select cno from course t1,teacher t2 where t1.tno = t2.tno and t2.tname = '谌燕') t2

where t1.cno = t2.cno)

/\*这个语句没有结果

select t1.sno from student t1 where not exists

(select distinct sno from sc t1,

(select cno from course t1,teacher t2 where t1.tno = t2.tno and t2.tname = '谌燕') t2

where t1.cno = t2.cno)

\*/

/\*这个语句结果是全部的学生，没有过滤

select t1.sno from student t1 where exists

(select distinct sno from sc t1,

(select cno from course t1,teacher t2 where t1.tno = t2.tno and t2.tname = '谌燕') t2

where t1.cno = t2.cno)

\*/

select t8.sno from student t8 where exists

/\*\*新增代码\*/

(select sno from

/\*\*新增代码\*/

(select distinct sno from sc t1,

(select cno from course t1,teacher t2 where t1.tno = t2.tno and t2.tname = '谌燕') t2

where t1.cno = t2.cno) t9

/\*\*新增代码\*/

where t8.sno = t9.sno

)

/\*\*新增代码\*/

select t8.sno from student t8 where not exists

(select sno from

(select distinct sno from sc t1,

(select cno from course t1,teacher t2 where t1.tno = t2.tno and t2.tname = '谌燕') t2

where t1.cno = t2.cno) t9

where t8.sno = t9.sno

)

/\*\*查询学过“c001”并且也学过编号“c002”课程的同学的学号、姓名；\*/

explain select s.sno,s.sname from student s, sc c

where c.cno = 'c001' and c.cno = 'c002' and s.sno = c.sno

-- 开启查看sql执行时间

select version()

show variables like "%pro%"

set profiling = 1

show profiles

select \* from student

show profile for query 575

-- 开启查看sql执行时间

select t.sno,t.sname from student t,

(select sno from sc where cno = 'c001') t1,

(select sno from sc where cno = 'c002') t2

where t.sno = t1.sno and t1.sno = t2.sno

/\*\*查询学过“谌燕”老师所教的所有课的同学的学号、姓名；\*/

select cno from course c,teacher t where c.tno = t.tno and t.tname = '谌燕'

select c.cno cno3 from sc c group by c.sno

select s.sname,c.sno,c.cno from student s join sc c on s.sno = c.sno;

select c.tno,s.cno from course c , sc s , teacher t

where c.cno = s.cno and c.tno = t.tno and t.tname = '谌燕';

select st.\* from student st join sc s on st.sno=s.sno

join course c on s.cno=c.cno

join teacher t on c.tno=t.tno

where t.tname='谌燕'

/\*查询课程编号“c002”的成绩比课程编号“c001”课程低的所有同学的学号、姓名\*/

select s.sno,s.sname from student s,

(select t1.sno from

(select sno,score from sc where cno = 'c001') t1,

(select sno,score from sc where cno = 'c002') t2

where t1.sno = t2.sno and t1.score > t2.score

) c

where s.sno = c.sno;

show profiles;

select \* from student st

join sc a on st.sno=a.sno

join sc b on st.sno=b.sno

where a.cno='c002' and b.cno='c001' and a.score < b.score

/\*查询所有课程成绩小于60 分的同学的学号、姓名；\*/

select s.sno,s.sname from student s,

(select c.sno from sc c where c.score < 60) c

where s.sno = c.sno;

/\*查询没有学全所有课的同学的学号、姓名；\*/

select count(1) count from course;

select sno ,count(1) count from sc group by sno;

select sno from

(select count(1) count from course) t1,

(select sno ,count(1) count from sc group by sno) t2

where t2.count < t1.count;

select s.sno , s.sname from student s ,

(select sno from

(select count(1) count from course) t1,

(select sno ,count(1) count from sc group by sno) t2

where t2.count < t1.count) t

where s.sno = t.sno;

select s.sno from student s where not exists (select c.sno from sc c where s.sno = c.sno);

-- 注意下面注释的部分，加上就报语法错误

-- select sno ,sname from

-- (

select s.sno , s.sname from student s ,

(select sno from

(select count(1) count from course) t1,

(select sno ,count(1) count from sc group by sno) t2

where t2.count < t1.count) t

where s.sno = t.sno

-- ) s

union -- 将结果组合到一个结果集中，且会去重。union all 组合到一个结果集中，不去重，比union快

-- (

select s.sno ,s.sname from student s

where not exists (select c.sno from sc c where s.sno = c.sno)

-- ) t

/\*\*查询至少有一门课与学号为“s001”的同学所学相同的同学的学号和姓名；\*/

select cno from sc where sno = 's001';

select c.sno,c.cno from sc c where not exists (select sno from sc where c.sno = sno and sno = 's001');

select distinct s.sno , s.sname

from student s ,

(select c.sno,c.cno from sc c

where not exists (select sno from sc where c.sno = sno and sno = 's001')) t

where exists (select cno from sc where sno = 's001') and s.sno = t.sno;

/\*\*查询至少学过学号为“s001”同学所有一门课的其他同学学号和姓名；没看出来与上一题有什么区别\*/

/\*\* 13 把“SC”表中“谌燕”老师教的课的成绩都更改为此课程的平均成绩；\*/

/\*\* update语法：update table set column = value where 条件 = 某值 \*/

select c.tno,c.cno from course c , teacher t where c.tno = t.tno and t.tname = '谌燕';

select s.cno , avg(score) avg\_score from sc s

where exists

(select c.tno,c.cno from course c , teacher t

where c.tno = t.tno and t.tname = '谌燕' and s.cno = c.cno)

group by s.cno;

/\*

-- mysql 不支持这种语法 ，也不允许对同一个表查询后做update或delete操作

update sc set score value t.avg\_score

from sc c ,(select s.cno , avg(score) avg\_score from sc s

where exists

(select c.tno,c.cno from course c , teacher t

where c.tno = t.tno and t.tname = '谌燕' and s.cno = c.cno)

group by s.cno) t

where c.cno = t.cno;

\*/

update sc c inner join (select s.cno , avg(score) avg\_score from sc s

where exists

(select c.tno,c.cno from course c , teacher t

where c.tno = t.tno and t.tname = '谌燕' and s.cno = c.cno)

group by s.cno) t

on c.cno = t.cno

set c.score = t.avg\_score

/\* 14 查询和“s001”号的同学学习的课程完全相同的其他同学学号和姓名；\*/

-- 查出s001学过的课程数量

select sno,count(cno) ccount from sc where sno = 's001' group by sno;

-- 查出s001学过的课程名称升序排列后组合到一列中并以的“;”号隔开(如果不写separator默认是“,”号隔开)

select s.sno,group\_concat(distinct c.cname order by c.cname asc separator ';') cour

from sc s , course c where s.cno = c.cno and s.sno = 's001';

-- 将上面两句合并一下

select s.sno,

group\_concat(distinct c.cname order by c.cname asc separator ';') cour,

count(c.cno) num

from sc s , course c

where s.cno = c.cno and s.sno = 's001'

group by s.sno;

-- s001以外的其他同学学过的课程数

select c.sno,count(cno) qcount from sc c

where not exists (select sno from sc where sno = 's001' and c.sno = sno )

group by c.sno;

-- s001以外的其他同学学过的课程数及所有课程

select s.sno ,

group\_concat(distinct c.cname order by c.cname asc separator ';') cour ,

count(c.cno) num

from sc s , course c

where not exists (select sno from sc where sno = 's001' and s.sno = sno )

and s.cno = c.cno

group by s.sno;

show profiles;

show profile for query 435;

select t2.sno from

(select s.sno,

group\_concat(distinct c.cname order by c.cname asc separator ';') cour,

count(c.cno) num

from sc s , course c

where s.cno = c.cno and s.sno = 's001'

group by s.sno) t1,

(select s.sno ,

group\_concat(distinct c.cname order by c.cname asc separator ';') cour ,

count(c.cno) num

from sc s , course c

where not exists (select sno from sc where sno = 's001' and s.sno = sno )

and s.cno = c.cno

group by s.sno) t2

where t1.cour = t2.cour and t1.num = t2.num;

-- 查询没有结果，我们来造点数据进去

-- 先让s002跟s001学一样的

insert into sc values ('s002','c003','63');

-- 再让s003跟s001/s002有一个不同

insert into sc values ('s003','c004','74');

-- 再执行之前的语句，只能得到 s002 ，接下来就好办了

select s.sno,s.sname

from student s ,

(select t2.sno from

(select s.sno,

group\_concat(distinct c.cname order by c.cname asc separator ';') cour,

count(c.cno) num

from sc s , course c

where s.cno = c.cno and s.sno = 's001'

group by s.sno) t1,

(select s.sno ,

group\_concat(distinct c.cname order by c.cname asc separator ';') cour ,

count(c.cno) num

from sc s , course c

where not exists (select sno from sc where sno = 's001' and s.sno = sno )

and s.cno = c.cno

group by s.sno) t2

where t1.cour = t2.cour and t1.num = t2.num) p

where s.sno = p.sno;

/\*

不相等优化

a<>0 改为 a>0 or a<0

有一种情况也不要使用or作为连接，因为会导致全表扫描，如：

select id from t where num=10 or num=20

可以这样查询：

select id from t where num=10

union all

select id from t where num=20

a<>’’ 改为 a>’’

看到这个题的时候，一直在想怎么让其他同学学的课程和s001学的不相等，

一直在查询不相等优化资料，可惜并没有关于 !=和<> 的优化资料。

这才慢慢想到用相等来判断。其实一开始就曲解了题意，悲剧！！！

\*/

show profiles;

-- 继续优化，使用join on 比用“,”号连接多表查询快

-- 上面的语句跟下面的这个语句都能实现关联查询,但是使用join更加灵活,效率更高,

-- 比如:加上筛选条件后,使用前者是先将表与表先关联查询,在结果中使用条件筛选,

-- 使用后者是先通过筛选条件筛选后再进行关联;

-- 所以可以将下面用“,”连接的查询都改成使用join on方式，更快

-- join on 默认就是 inner join on

select s.sno,s.sname

from student s

join

(select t2.sno from

(select s.sno,

group\_concat(distinct c.cname order by c.cname asc separator ';') cour,

count(c.cno) num

from sc s , course c

where s.cno = c.cno and s.sno = 's001'

group by s.sno) t1,

(select s.sno ,

group\_concat(distinct c.cname order by c.cname asc separator ';') cour ,

count(c.cno) num

from sc s , course c

where not exists (select sno from sc where sno = 's001' and s.sno = sno )

and s.cno = c.cno

group by s.sno) t2

where t1.cour = t2.cour and t1.num = t2.num) p

on s.sno = p.sno;

-- sql优化多表连接时，表的顺序是：大表在前，小表在后，以小表驱动大表查询

/\*\*15 删除学习“谌燕”老师课的SC 表记录；\*/

/\*\*删除语法：delete from table where \*/

select c.cno from course c join teacher t on c.tno = t.tno;

show profiles;

select c.cno from course c , teacher t where c.tno = t.tno;

delete c from sc c ,

(select c.cno from course c join teacher t on c.tno = t.tno where t.tname = '谌燕') t

where c.cno = t.cno;

delete c from

sc c

join

(select c.cno from course c join teacher t on c.tno = t.tno where t.tname = '谌燕') t

on c.cno = t.cno;

/\*\* 16 向SC 表中插入一些记录，这些记录要求符合以下条件：没有上过编号“c002”课程的同学学号、“c002”号课的平均成绩；\*/

/\*\*insert into tablename( 属性1，属性2) values ('值1'，‘值2’);

https://www.biaodianfu.com/mysql-insert-into-if-exists.html

http://cw.hubwiz.com/card/c/550a8966e564e51f041a15ba/1/4/2/

\*/

-- 学过c002的人

select sno,cno,score from sc c where c.cno = 'c002' group by c.sno;

-- c002的平均成绩

select cno ,avg(score) avg\_score from sc where cno = 'c002' group by cno;

-- 没学过c002的人

select sno from student s

where not exists (select sno,cno,score from sc c where c.cno = 'c002' and s.sno = c.sno group by c.sno);

select t1.sno,t2.cno,t2.avg\_score score

from

(select sno from student s

where not exists

(select sno,cno,score from sc c

where c.cno = 'c002' and s.sno = c.sno

group by c.sno)) t1 ,

(select cno ,avg(score) avg\_score from sc

where cno = 'c002' group by cno) t2;

insert into sc (sno,cno,score)

select sno,cno,score

from (select t1.sno,t2.cno,t2.avg\_score score

from

(select sno from student s

where not exists

(select sno,cno,score from sc c

where c.cno = 'c002' and s.sno = c.sno

group by c.sno)) t1 ,

(select cno ,avg(score) avg\_score from sc

where cno = 'c002' group by cno) t2

) p;

-- 别人写的

insert into sc (sno,cno,score)

select distinct st.sno,sc.cno,(select avg(score) from sc where cno='c002')

from student st,sc

where not exists

(select \* from sc where cno='c002' and sc.sno=st.sno)

and sc.cno='c002';

/\* 17 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\*/

-- 最高分

select score from sc order by score desc limit 1;

-- 按课程 得出最高分 最低分

select cno,max(score),min(score) from sc group by cno;

/\* 18 按各科平均成绩从低到高和及格率的百分数从高到低顺序\*/

-- 每门课程学习的人数

select cno,count(sno) from sc group by cno;

-- 有人学的课程学习的人数，平均值

select cno ,count(sno) num , avg(score) avg\_score from sc c group by cno order by cno;

-- 所有课程

select cno from course ;

-- 没人学过的课程号，人数为0，平均成绩为0，及格率为0.00%，用dual表造一个结果列，且值是 0

select t.cno ,

(select 0 from dual) num ,

(select 0 from dual) avg\_score

from course t

where not exists

(select cno,count(sno) num from sc c where t.cno = cno group by cno);

-- 各科平均成绩

select distinct \* from

(select cno ,count(sno) num , avg(score) avg\_score

from sc c group by cno order by cno

) t1

union all

(select t.cno cno ,(select 0 from dual) num , (select 0 from dual) avg\_score

from course t

where not exists

(select cno,count(sno) num from sc c

where t.cno = cno group by cno)

) t2;

-- 注意一下 union all 连接的两个表要加()号，否则报错：Incorrect usage of UNION and ORDER BY

-- 特别注意，加()好就行了，不用起别名。起别名又会报错

(select cno ,count(sno) num , avg(score) avg\_score

from sc c group by cno order by cno)

union all

(select t.cno cno ,(select 0 from dual) num , (select 0 from dual) avg\_score

from course t

where not exists

(select cno,count(sno) num from sc c

where t.cno = cno group by cno))

-- 按各科平均成绩从低到高，

-- FORMAT（X,D）函数可以控制数据X的小数点为D位(四舍五入运算)

select cno,num,format(avg\_score,2)avg\_score

from (

(select cno ,count(sno) num , avg(score) avg\_score

from sc c group by cno order by cno)

union all

(select t.cno cno ,(select 0 from dual) num , (select 0 from dual) avg\_score

from course t

where not exists

(select cno,count(sno) num from sc c

where t.cno = cno group by cno))) t

order by avg\_score;

-- 求出及格率

-- 每门课程及格的人数

select cno , count(sno) from sc where score > 60 group by cno;

-- 学过的课程的及格率，

select t1.cno ,(t1.num / t2.num ) passing\_rate

from (select cno , count(sno) num from sc where score > 60 group by cno) t1 ,

(select cno ,count(sno) num , avg(score) avg\_score from sc c group by cno order by cno) t2

where t1.cno = t2.cno;

-- 在及格率后面加一个 % 号

-- 下面这个是失败的，case 和 if 都是用来判断的，相当于oracle里面的decode函数

select t1.cno ,

(case when (t1.num / t2.num ) passing\_rate = passing\_rate then passing\_rate = (passing\_rate\*100)+'%')passing\_rate

from (select cno , count(sno) num from sc where score > 60 group by cno) t1 ,

(select cno ,count(sno) num , avg(score) avg\_score from sc c group by cno order by cno) t2

where t1.cno = t2.cno;

-- 下面这个又是失败的，concat()：两个字符串相加，cast(xxx as type):将xxx数据类型转为后面定义的type的类型

select t.cno,concat(cast(t.passing\_rate as varchar) ,'%') passing\_rate

from (select t1.cno ,(t1.num / t2.num ) passing\_rate

from (select cno , count(sno) num from sc where score > 60 group by cno) t1 ,

(select cno ,count(sno) num , avg(score) avg\_score from sc c group by cno order by cno) t2

where t1.cno = t2.cno) t;

-- 上面的结果之所以失败是因为cast类型转换错了，要转成char

select t.cno,concat(cast(t.passing\_rate as char) ,'%') passing\_rate

from (select t1.cno ,format((t1.num / t2.num ),2) passing\_rate -- 保留小数点后两位

from (select cno , count(sno) num from sc where score > 60 group by cno) t1 ,

(select cno ,count(sno) num , avg(score) avg\_score from sc c group by cno order by cno) t2

where t1.cno = t2.cno

) t;

-- 改造一下之前那个没人学过的课程号，人数为0，平均成绩为0，添加及格率为0.00%

select t.cno ,

(select 0 from dual) num ,

(select 0 from dual) avg\_score ,

concat(cast((select 0.00 from dual) as char),'%') passing\_rate

from course t

where not exists

(select cno,count(sno) num from sc c where t.cno = cno group by cno);

-- 结合学过的和没学过的，及格率,按从高到低排序

select cno,passing\_rate

from (

(select t.cno,concat(cast(t.passing\_rate as char) ,'%') passing\_rate

from (select t1.cno ,format((t1.num / t2.num ),2) passing\_rate -- 保留小数点后两位

from (select cno , count(sno) num from sc where score > 60 group by cno) t1 ,

(select cno ,count(sno) num , avg(score) avg\_score from sc c group by cno order by cno) t2

where t1.cno = t2.cno

) t

)

union all

(

select t.cno ,

concat(cast((select 0.00 from dual) as char),'%') passing\_rate

from course t

where not exists

(select cno,count(sno) num from sc c where t.cno = cno group by cno)

)

) t

order by passing\_rate desc;

-- 最后的结果

select t1.cno,t1.avg\_score,t2.passing\_rate

from

(

select cno,num,format(avg\_score,2)avg\_score

from (

(select cno ,count(sno) num , avg(score) avg\_score

from sc c group by cno order by cno)

union all

(select t.cno cno ,(select 0 from dual) num , (select 0 from dual) avg\_score

from course t

where not exists

(select cno,count(sno) num from sc c

where t.cno = cno group by cno))) t

order by avg\_score

) t1 ,

(

select cno,passing\_rate

from (

(select t.cno,concat(cast(t.passing\_rate as char) ,'%') passing\_rate

from (select t1.cno ,format((t1.num / t2.num ),2) passing\_rate -- 保留小数点后两位

from (select cno , count(sno) num from sc where score > 60 group by cno) t1 ,

(select cno ,count(sno) num , avg(score) avg\_score from sc c group by cno order by cno) t2

where t1.cno = t2.cno

) t

)

union all

(

select t.cno ,

concat(cast((select 0.00 from dual) as char),'%') passing\_rate

from course t

where not exists

(select cno,count(sno) num from sc c where t.cno = cno group by cno)

)

) t

order by passing\_rate desc

) t2

where t1.cno = t2.cno;

-- 最后发现如果结果列中包含cno列，平均值排序和及格率排序根本没用了

-- 可是如果没有cno列怎么看得出来哪个平均值或及格率是哪个课程的呢？

-- 我觉得这个题应该分成两道题，求平均值排序和求及格率排序

-- 不管怎样，还是先得到最终结果吧：

-- 结果就是实在搞不出来了。。。。求牛人解答

-- 网上答案，为何人家如此牛逼

select cno,avg(score),sum(case when score>=60 then 1 else 0 end)/count(\*)

as 及格率

from sc group by cno

order by avg(score) , 及格率 desc

/\* 19 查询不同老师所教不同课程平均分从高到低显示\*/

select tno , avg(score) from course c , sc s

where c.cno = s.cno

group by c.tno;

-- 网上答案 注意一个问题：查询列不在group by 或 order by 或 having 子句中出现时，

-- 下面这种方式(max(xxx))就可以避免程序报错

select max(t.tno) tno,max(t.tname),max(c.cno),max(c.cname),c.cno,avg(score) from sc , course c,teacher t

where sc.cno=c.cno and c.tno=t.tno

group by c.cno

order by avg(score) desc;

select max(c.tno) tno , c.cno, avg(score) from course c , sc s

where c.cno = s.cno

group by c.cno

order by avg(score) desc;

/\* 20 统计列印各科成绩,各分数段人数:课程ID,课程名称,[100-85],[85-70],[69-60],[0-59]\*/

select s.cno,

max(c.cname) cname ,

sum(case when score > 85 and score <= 100 then 1 else 0 end) as '[100-85)',

sum(case when score > 69 and score <= 85 then 1 else 0 end) as '[85-70]',

sum(case when score >= 60 and score <= 69 then 1 else 0 end) as '[69-60]',

sum(case when score >= 0 and score < 60 then 1 else 0 end) as '[0-59]'

from course c , sc s

where c.cno = s.cno

group by s.cno;

/\* 21 查询各科成绩前三名的记录:(不考虑成绩并列情况)\*/

select max(sno) sno , cno , abs(score) scor

from sc

group by cno

order by score desc

limit 3;

-- oracle 中有下面的方法来解决这个问题

-- ROW\_NUMBER() OVER()函数用法;(分组，排序），partition by

-- 在使用 row\_number() over()函数时候，

-- over()里头的分组以及排序的执行晚于 where group by order by 的执行。

-- partition by 用于给结果集分组，如果没有指定那么它把整个结果集作为一个分组，

-- 它和聚合函数不同的地方在于它能够返回一个分组中的多条记录，

-- 而聚合函数一般只有一个反映统计值的记录。

-- mysql 怎么干到这个效果呢？

select \* from sc s

left join sc c

on s.cno = c.cno;

select \* from sc s

left join sc c

on s.cno = c.cno and s.sno = c.sno;

select c.cno,c.score ,count(s.cno) from sc s

left join sc c

on s.cno = c.cno and s.sno = c.sno and s.score <= c.score

group by c.cno ,c.sno,c.score

-- 此处分完组后会根据cno和score这两列去重

-- (每行数据中这两列值都和前面行这两列的值相同才会去重)，结果数为12行

order by c.score desc

;

-- 意思就是：我们要查询的这个人，在这一门中分数比他高的少于2个人。

-- 括号里面是查询在这一门中分数比他高的人的数量。

select \* from sc s

where (select count(\*) from sc c where s.`cno` = c.`cno` and c.`score` > s.`score` having count(\*) < 3) < 2

order by s.`cno`,s.`score` desc;

-- 前面的尝试都失败了

-- 大牛博客：http://tdcq.iteye.com/blog/1931981，专门讲这个问题

-- mysql用户自定义变量：http://www.cnblogs.com/genialx/p/5932558.html

-- mysql用户自定义变量实例：https://www.jianshu.com/p/357a02fb2d64

select sno,cno,score

from sc

order by cno,score;

select sno,cno,score

from sc

order by cno,score desc;

select sno,cno,score

from sc

order by cno desc,score desc;

set @num := 0, @type := '';

select sno,cno,score,

@num := if(@type = cno , @num := @num+1 , 1) as num ,

@type := cno

from sc

order by cno,score desc;

set @num := 0, @type := '';

select sno,cno,score

from

(

select sno,cno,score,

@num := if(@type = cno , @num := @num+1 , 1) as num ,

@type := cno

from sc

order by cno,score desc

) as newSc

where newSc.num <= 3;

-- 到此问题解决了，但是还可以进一步优化，就是不用创建临时表的方式

set @num := 0, @type := '';

select sno,cno,score,

@num := if(@type = cno , @num := @num+1 , 1) as num ,

@type := cno

from sc

group by cno,score,sno

having num <= 3;

-- 结果并不是我们预想的那样，我们在cno,score列上建索引

-- 并且把sno字段不参与group by中

alter table sc add key(cno,score);

set @num := 0, @type := '';

select max(sno),cno,score,

@num := if(@type = cno , @num := @num+1 , 1) as num ,

@type := cno

from sc

group by cno,score

having num <= 3;