数据库系统上机（二）：数据查询

一、实验目的

掌握SQL程序设计基本规范，熟练运用SQL语言实现数据基本查询，包括单表查询、分组统计查询和连接查询等。

二、实验内容和要求及试验结果。

1.  查询SC表中的全部数据。

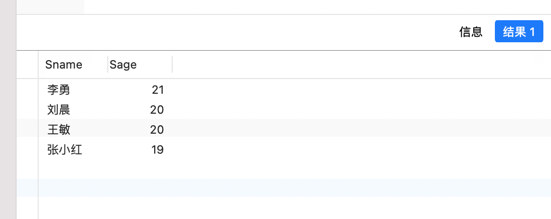
SELECT\*FROM CSC

2.  查询计算机系学生的姓名和年龄。

SELECT Sname,Sage

FROM Student

WHERE Sdept=N'计算机系'

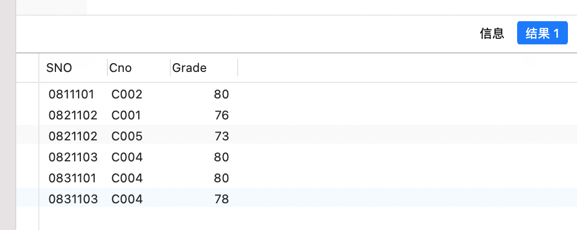


3.  查询成绩在70～80分的学生的学号、课程号和成绩。

SELECT SNO,Cno,Grade

FROM CSC

WHERE Grade BETWEEN 70 AND 80

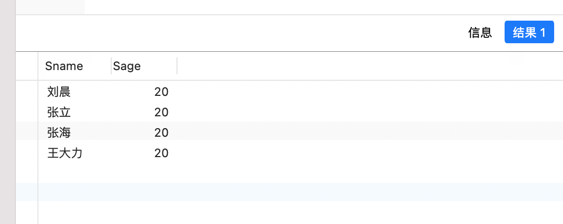


4.  查询计算机系年龄在18～20岁的男生姓名和年龄。

SELECT Sname,Sage

FROM Student

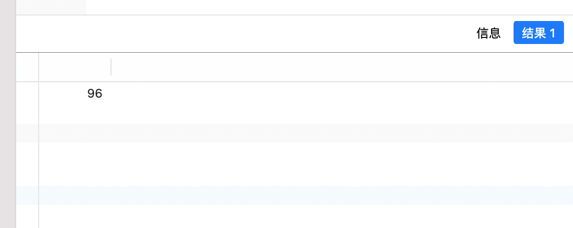
WHERE Sage BETWEEN 18 AND 20 AND Ssex=N'男'



5.  查询C001课程的最高分。

SELECT MAX(Grade)

FROM CSC

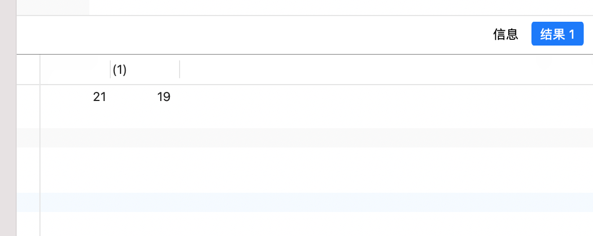


6.  查询计算机系学生的最大年龄和最小年龄。

SELECT MAX (Sage),MIN (Sage)

FROM Student

WHERE Sdept=N'计算机系'

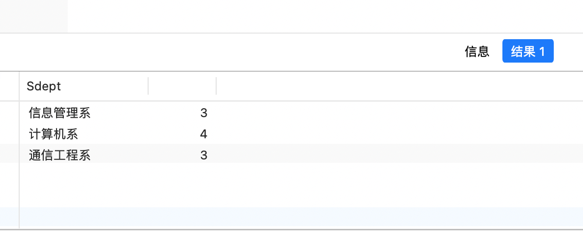


7.  统计每个系的学生人数。

SELECT Sdept,COUNT (\*)

FROM Student

GROUP BY Sdept

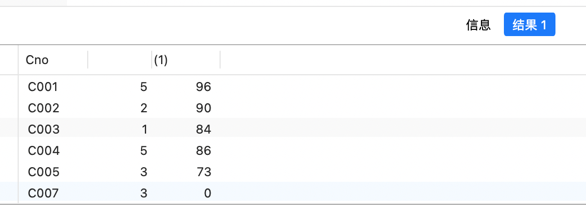


8.  统计每门课程的选课人数和最高成绩。

SELECT Cno,COUNT (\*),MAX (Grade)

FROM CSC

GROUP BY Cno



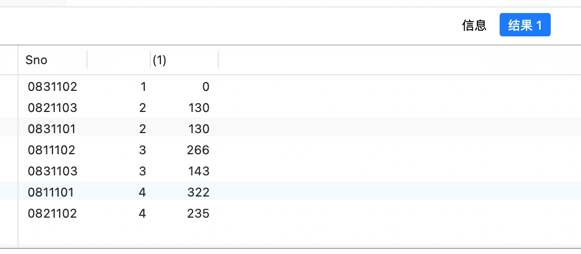
9.  统计每个学生的选课门数和考试总成绩，并按选课门数升序显示结果。

SELECT Sno,COUNT(Sno),SUM(Grade)

FROM CSC

GROUP BY Sno

ORDER BY COUNT(Sno)



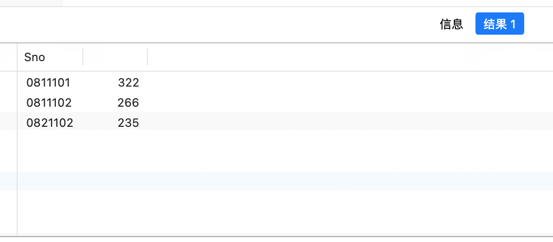
10.  列出总成绩超过200的学生的学号和总成绩。

SELECT Sno,SUM(Grade)

FROM CSC

GROUP BY Sno

HAVING SUM(Grade)>=200

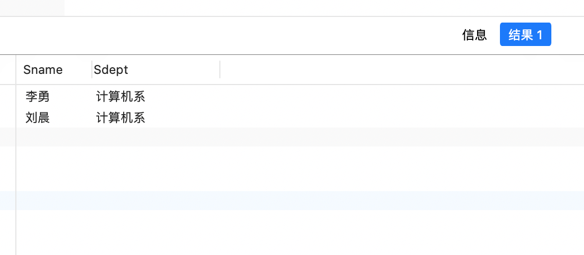


11.  查询选了C002课程的学生姓名和所在系。

SELECT Sname,Sdept

FROM CSC,Student

WHERE Student.Sno=CSC.Sno AND CSC.Cno='C002'



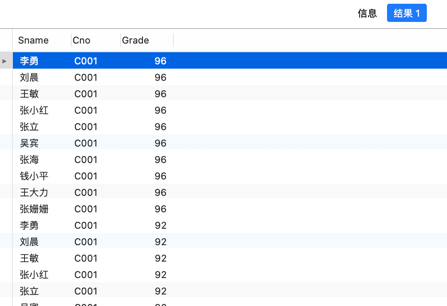
12.  查询考试成绩80分以上的学生姓名、课程号和成绩，并按成绩降序排列结果。

SELECT Sname,Cno,Grade

FROM CSC,Student

WHERE Grade>80

ORDER BY Grade DESC

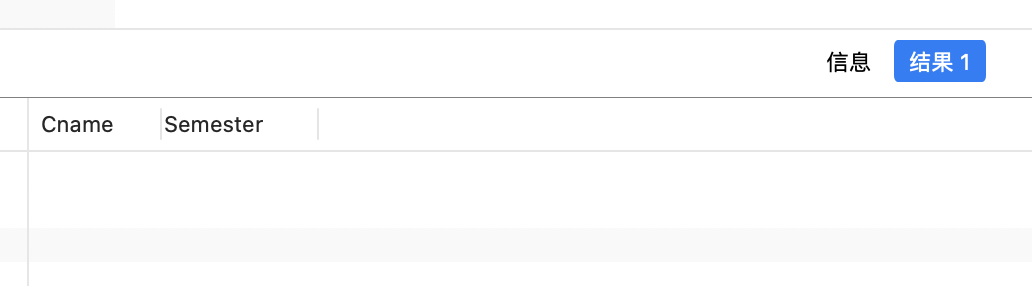


13.  查询与VB在同一学期开设的课程的课程名和开课学期。

SELECT C2.Cname,C1.Semester

FROM Course C1 JOIN Course C2 ON C1.Semester = C2.Semester

WHERE C1.Cname='VB' AND C2.Cname!='VB'



14.  查询与李勇年龄相同的学生的姓名、所在系和年龄。

SELECT C2.Sname,C2.Sdept,C2.Sage

FROM Student C1 JOIN Student C2

ON C1.Sage=C2.Sage

WHERE C1.Sname=N'李勇' AND C2.Sname!=N'李勇'

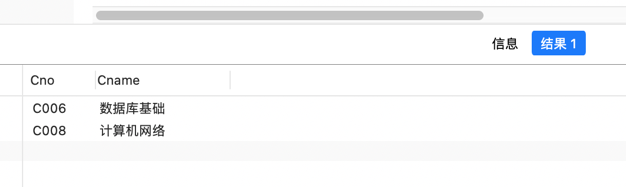


15.  查询哪些课程没有学生选修，列出课程号和课程名。

SELECT S.Cno,Cname

FROM Course S left JOIN CSC ON S.Cno=CSC.Cno

WHERE CSC.Cno IS NULL



//16.  查询每个学生的选课情况，包括未选课的学生，列出学生的学号、姓名、选的课程号。

select Student.Sno,Sname, CSC.Cno

from CSC, Student

where CSC.Sno=Student.Sno;

select Sname,Sno

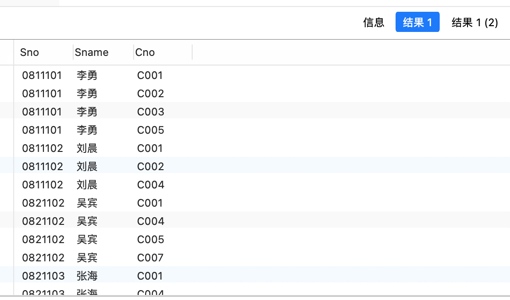
from Student

where Not exists(

select \*

from CSC

where Sno = Student.sno)





17.  查询计算机系哪些学生没有选课，列出学生姓名。

select Sname

from Student

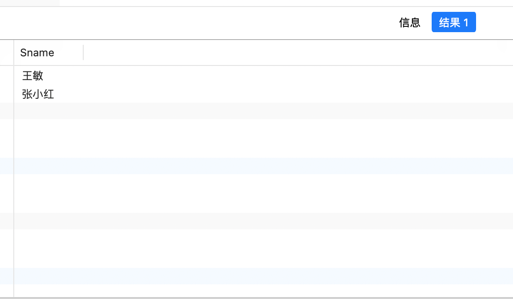
where Not exists(

select \*

from CSC

where Sno = Student.sno

) and Sdept = N'计算机系';

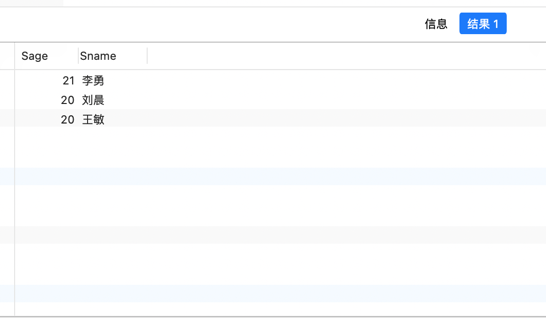


18.  查询计算机系年龄最大的三个学生的姓名和年龄。top 3

select top 3 Sage,Sname

from Student

where Sdept = N'计算机系'



19.  列出“VB”课程考试成绩前三名的学生的学号、姓名、所在系和VB成绩。top3

select top 3 Grade, Student.Sno, Sname, Sdept

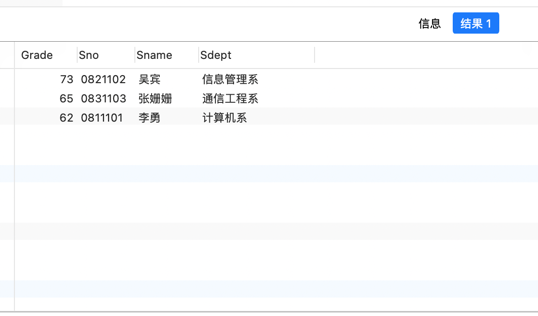
from Student, CSC, Course

where Student.Sno = CSC.Sno

and CSC.Cno = Course.Cno

and Cname = 'VB'

order by Grade DESC;

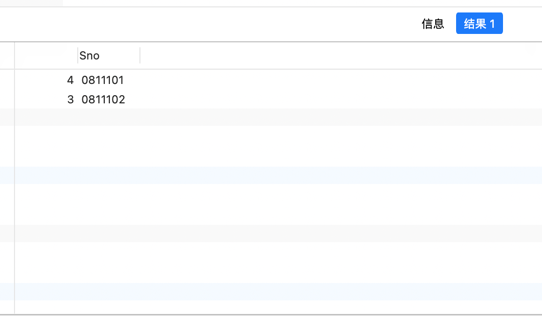


20.  查询选课门数最多的前2位学生，列出学号和选课门数。 top 2

SELECT top 2 count(Cno), Sno

FROM CSC

GROUP BY Sno



提示：查询排名前几的方法，select语句中的输出项为top关键字。

21.  用子查询实现如下查询：

（1）查询选了“C001”课程的学生姓名和所在系。

SELECT Sname,Sdept

FROM Student

WHERE Sno IN(

SELECT Sno

FROM CSC

WHERE Cno = 'C001')



（2）查询通信工程系成绩80分以上的学生的学号和姓名。

select Sno, Sname

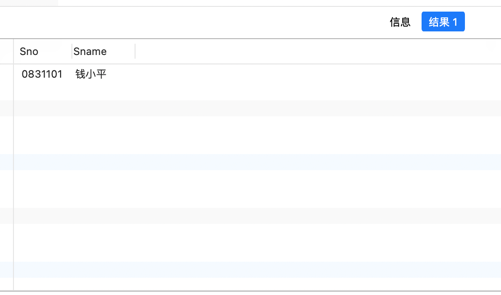
from Student

where Sno IN(

select DISTINCT Student.Sno

from CSC, Student

where CSC.Sno = Student.Sno AND Sdept=N'通信工程系' AND Grade >= 80)



（3）查询计算机系考试成绩最高的学生的姓名。

select Sname

from Student

where Sdept = N'计算机系' AND Sno =(select Sno

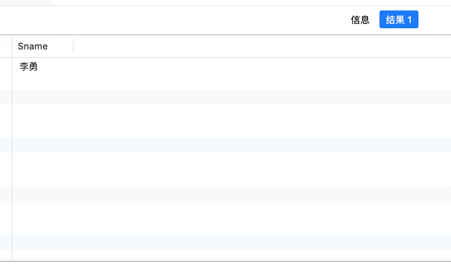
from CSC

where Grade = (

select MAX(Grade)

from CSC,Student

where CSC.Sno = Student.Sno AND Sdept = N'计算机系'))



（4）查询年龄最大的男生的姓名、所在系和年龄。

select Sname, Sdept, Sage

from Student

where Ssex = N'男' AND Sage =(

select Max(Sage)

from Student

where Ssex = N'男') ;

select Sname, Sdept, Sage

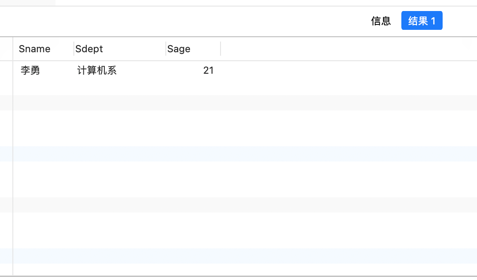
from Student

where Ssex = N'男' AND Sage =(

select Max(Sage)

from Student

where Ssex = N'男') ;



22.  查询C001课程的考试成绩高于该课程平均成绩的学生的学号和成绩。

select Student.Sno, Grade

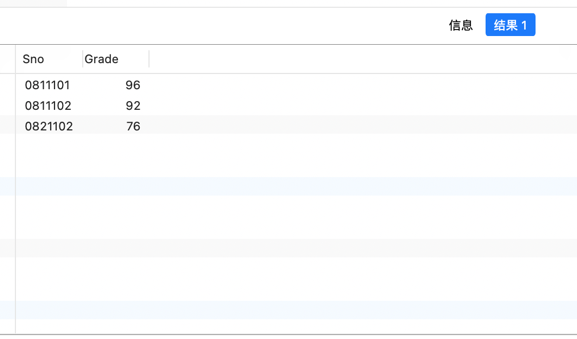
from Student, CSC

where Student.Sno = CSC.Sno AND Cno = 'C001' AND Grade >(

select AVG(Grade)

from CSC

where Cno = 'C001' );



23.  查询计算机系学生考试成绩高于计算机系学生平均成绩的学生的姓名、考试的课程名和考试成绩。

select Sname, Cname, Grade

from Student, CSC, Course

where Student.Sno = CSC.Sno AND

CSC.Cno = Course.Cno AND

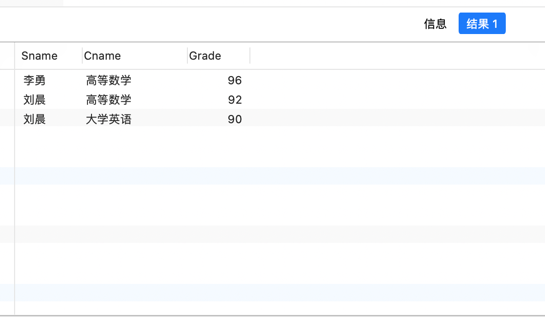
Sdept = N'计算机系' AND

Grade > (

select AVG(Grade)

from CSC,Student

where CSC.Sno = Student.Sno AND Sdept = N'计算机系')



24.  查询VB课程考试成绩高于VB平均成绩的学生姓名和VB成绩。

select Sname, Grade

from Student, CSC, Course

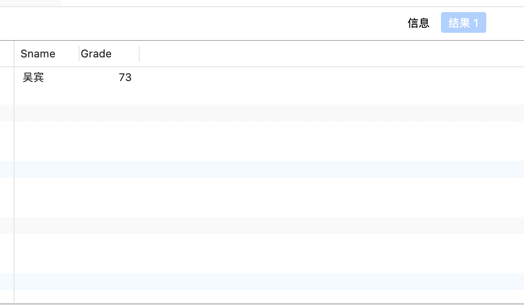
where Student.Sno = CSC.Sno AND Course.Cno = CSC.Cno AND

Cname = 'VB' AND Grade >(

select AVG(Grade)

from Course, CSCa

where CSC.Cno = Course.Cno AND Cname = 'VB');



25.  查询没选VB的学生姓名和所在系。

select DISTINCT Sname, Sdept

from Student,CSC,Course

where Student.Sno = CSC.Sno AND CSC.Cno=Course.Cno AND Cname <> 'VB'



26.  查询每个学期学分最高的课程信息，列出开课学期、课程名和学分。

select Semester , Cname, Credit

from Course x1

where Credit = (

select MAX(Credit)

from Course x2

where x1.Semester = X2.Semester)



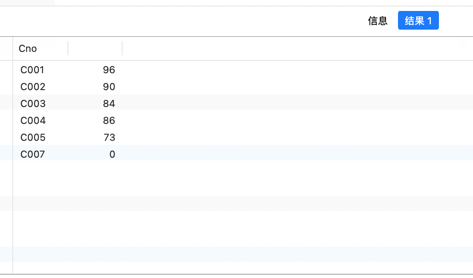
27.  查询每门课程考试成绩最高的学生信息，列出课程号、学生姓名和最高成绩，结果按课程号升序排序，不包括没考试的课程。

select Cno, max(Grade)

from Student, CSC

where Student.Sno = CSC.Sno

Group by Cno



28.  查询计算机系学生姓名、年龄和年龄情况，其中年龄情况为：如果年龄小于18，则显示“偏小”；如果年龄在18-22，则显示“合适”；如果年龄大于22，则显示“偏大”。

select Sname , Sage,

case when Sage <18 then N'偏小'

when Sage between 18 and 22 then N'合适'

when Sage >22 then N'偏大'

end 年龄情况

from Student

where Sdept = N'计算机系'



29.  统计每门课程的选课人数，包括有人选的课程和没有人选的课程，列出课程号，选课人数及选课情况，其中选课情况为：如果此门课程的选课人数超过100人，则显示“人多”；如果此门课程的选课人数在40～100，则显示“一般”；如果此门课程的选课人数在1～39，则显示“人少”；如果此门课程没有人选，则显示“无人选”。

Cno 人数 选课情况

人多

一般

select Course.Cno 课程号, COUNT(Student.Sno) 选课人数,

case when COUNT(Student.Sno) > 100 then N'人多'

when COUNT(Student.Sno) between 40 and 100 then N'一般'

when COUNT(Student.Sno) < 40 then N'人少'

else N'无人选'

end 选课情况

from Course left outer join CSC on (CSC.Cno = Course.Cno), Student

where Student.Sno = CSC.Sno

group by Course.Cno



三、实验小结

问题1、第二题查询所在系统学生的时候输出空值，但是同样的条件选择年龄等其他就可以。

解决方案：把SELECT Sname,Sage FROM Student WHERE Sdept='计算机系'

改成SELECT Sname,Sage FROM Student WHERE Sdept=N'计算机系'

原因：因为我使用的数据库是英文版，使用的是UNICODE编码，所以查询中文的时候在中文面前前面加个N解决。

问题2、第十题使用如下语句查询SELECT Sno,SUM(Grade) FROM CSC WHERE SUM(Grade)>=200

GROUP BY Sno报错An aggregate may not appear in the WHERE clause unless it is in a subquery contained in a HAVING clause or a select list, and the column being aggregated is an outer reference.

解决方案：用having语句代替where语句。

原因：where语句作用与基本表或者试图，而having语句用于group分好的组。