实验4 数据查询

# 一、实验目的

1. 掌握SELECT语句的基本语法和查询条件的表示方法；
2. 掌握连接查询的表示方法；
3. 掌握嵌套查询的表示方法；
4. 掌握数据汇总的方法；
5. 掌握ORDER BY子句的作用和使用方法；
6. 掌握HAVING子句的作用和使用方法；

# 二、实验内容

使用SELECT查询语句，在数据库StudentCourse的Student表、Course 表和StuCourse表上进行各种查询，包括单表查询、连接查询、嵌套查询，并进行数据汇总，以及使用GROUP BY子句、ORDER BY子句和HAVING子句对结果进行分组、排序和筛选处理。

1. SELECT语句的基本使用；
2. 连接查询；
3. 嵌套查询；
4. 数据汇总；
5. 使用GROUP BY 子句对结果分组；
6. 使用ORDER BY子句对结果排序；
7. 使用HAVING子句对分组结果进行筛选；

思考与练习：

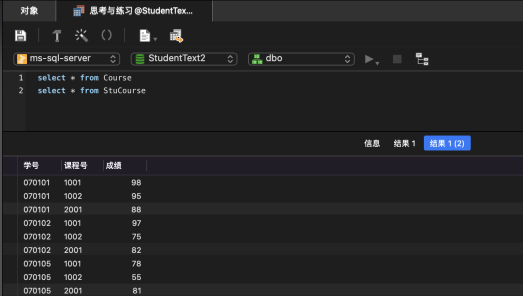
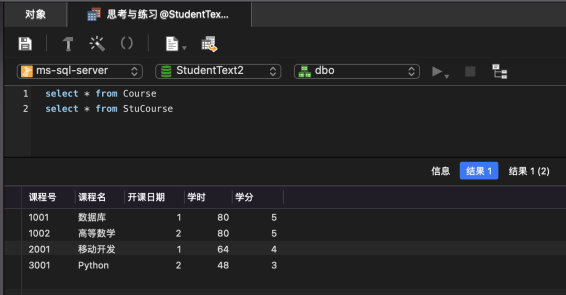
* 1. 用SELECT语句查询Course表和StuCourse表中的所有记录。

SQL语句：

select \* from Course;

select \* from StuCourse;

实验结果：



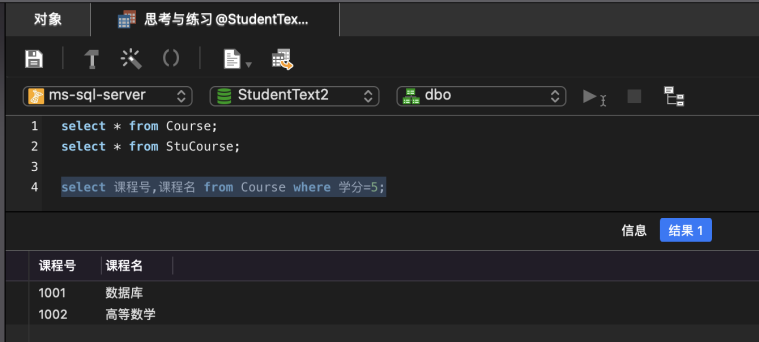
* 1. 用SELECT语句查询Course表和StuCourse表中满足指定条件的一列或若干列。

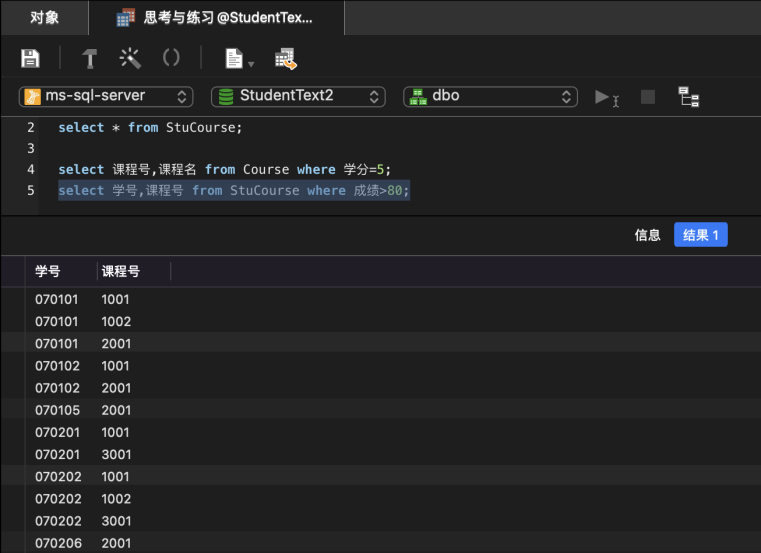
SQL语句：

select 课程号,课程名 from Course where 学分=5;

select 学号,课程号 from StuCourse where 成绩>80;

实验结果：



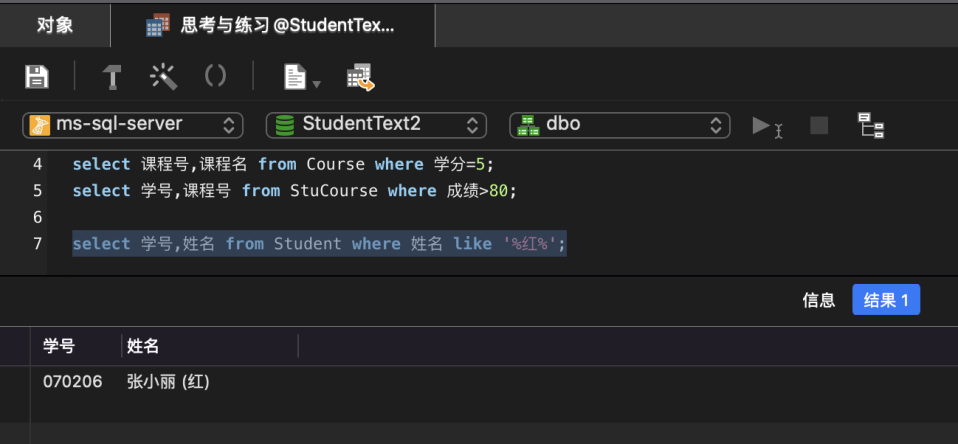


* 1. 查询所有姓名中包含有“红”的学生的学号及姓名。

SQL语句：

select 学号,姓名 from Student where 姓名 like '%红%';

实验结果：



* 1. 用连接查询的方法查找所有选修了“2001” 或“1002”号课程的学生学号和姓名。

SQL语句：

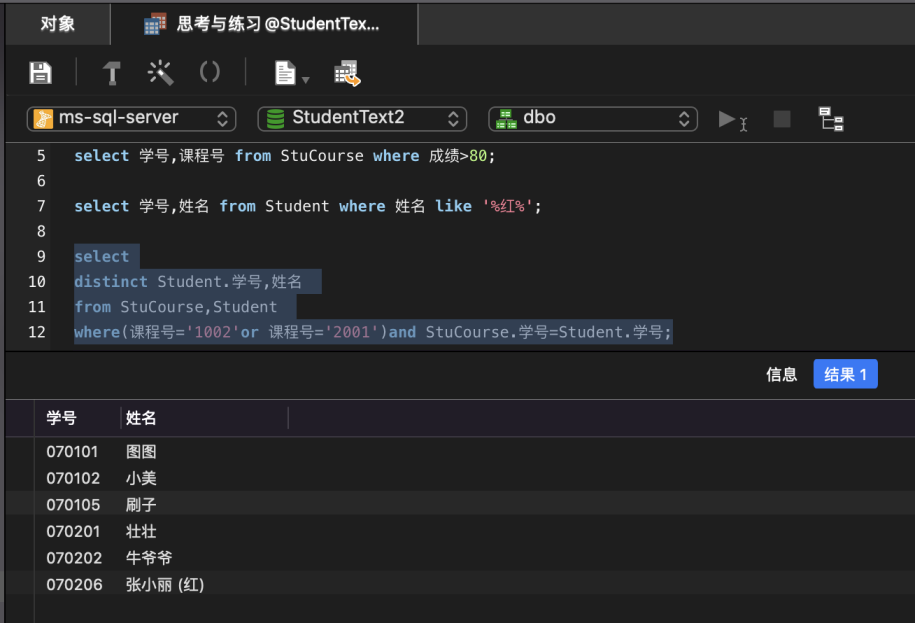
select

distinct Student.学号,姓名

from StuCourse,Student

where(课程号='1002'or 课程号='2001')and StuCourse.学号=Student.学 号;

实验结果：



* 1. 用子查询的方法查找所有选修了“2001”或“1002”号课程的学生学号和姓名。

SQL语句：

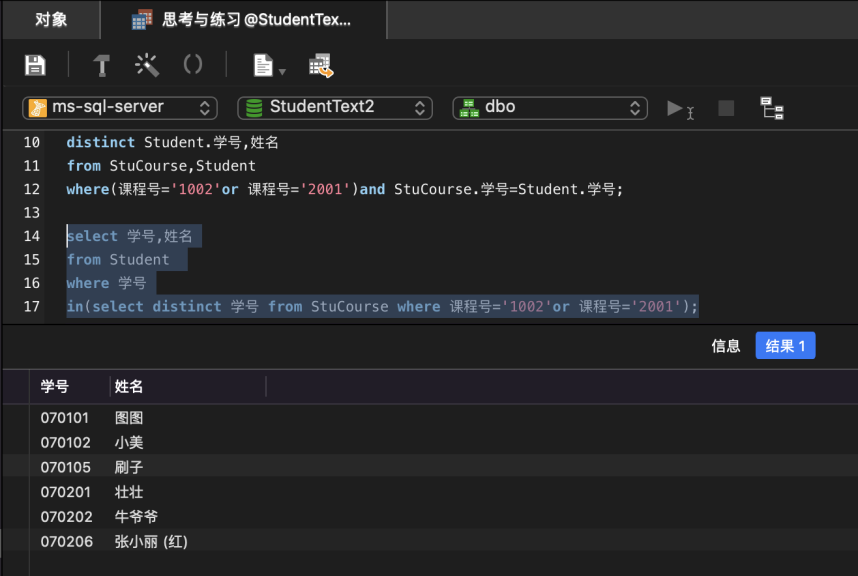
select 学号,姓名

from Student

where 学号

in(select distinct 学号 from StuCourse where 课程号='1002'or 课程 号='2001');

实验结果：



* 1. 用子查询的方法查找“计算机科学与技术”专业“1001”号课程成绩比所有“电子信息工程”专业“1001”号课程成绩都高的学生学号和姓名。

SQL语句：

select

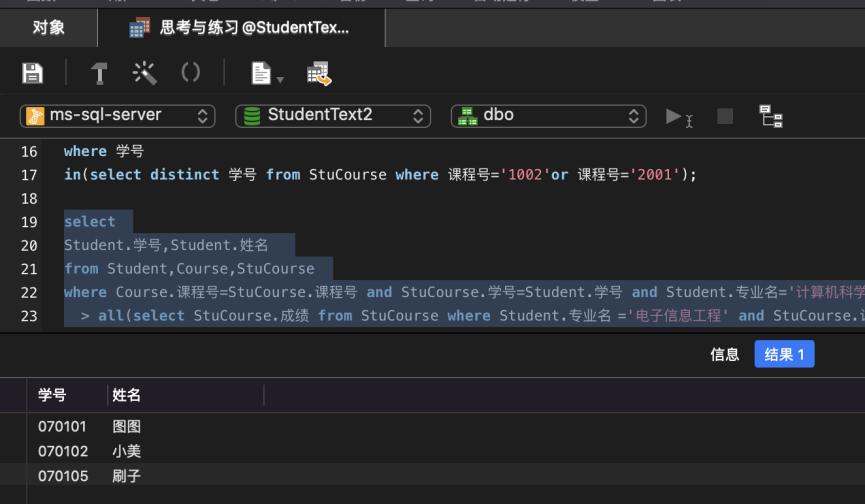
Student.学号,Student.姓名

from Student,Course,StuCourse

where Course.课程号=StuCourse.课程号 and StuCourse.学号=Student. 学号 and Student.专业名='计算机科学与技术' and StuCourse.课程号 ='1001' and StuCourse.成绩

> all(select StuCourse.成绩 from StuCourse where Student.专业 名 ='电子信息工程' and StuCourse.课程号='1001');

实验结果：



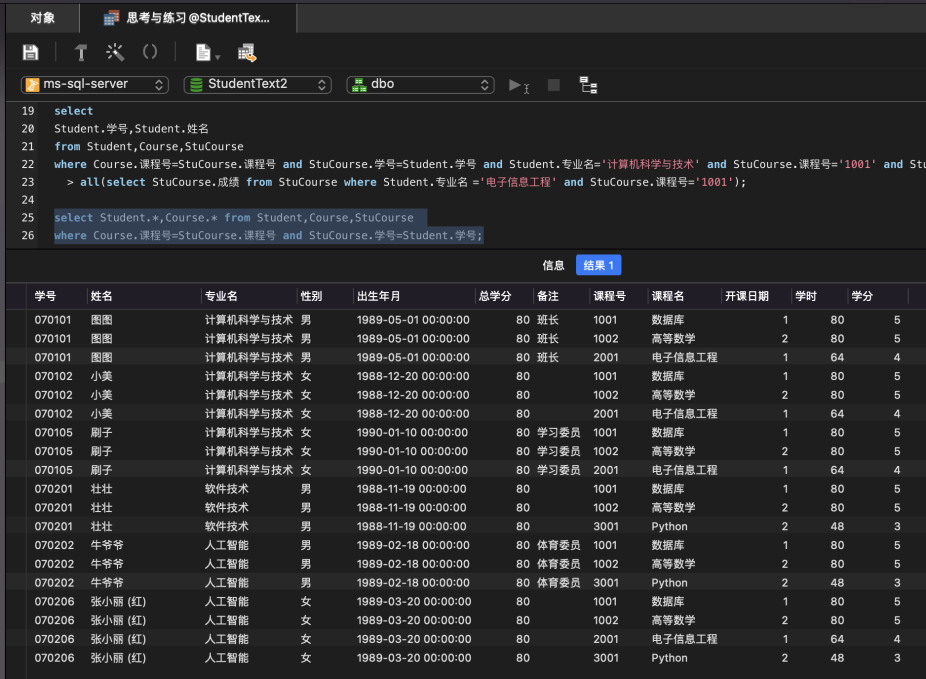
* 1. 查询每个学生的情况及其选修课程的情况。

SQL语句：

select Student.\*,Course.\* from Student,Course,StuCourse

where Course.课程号=StuCourse.课程号 and StuCourse.学号=Student. 学号;

实验结果：



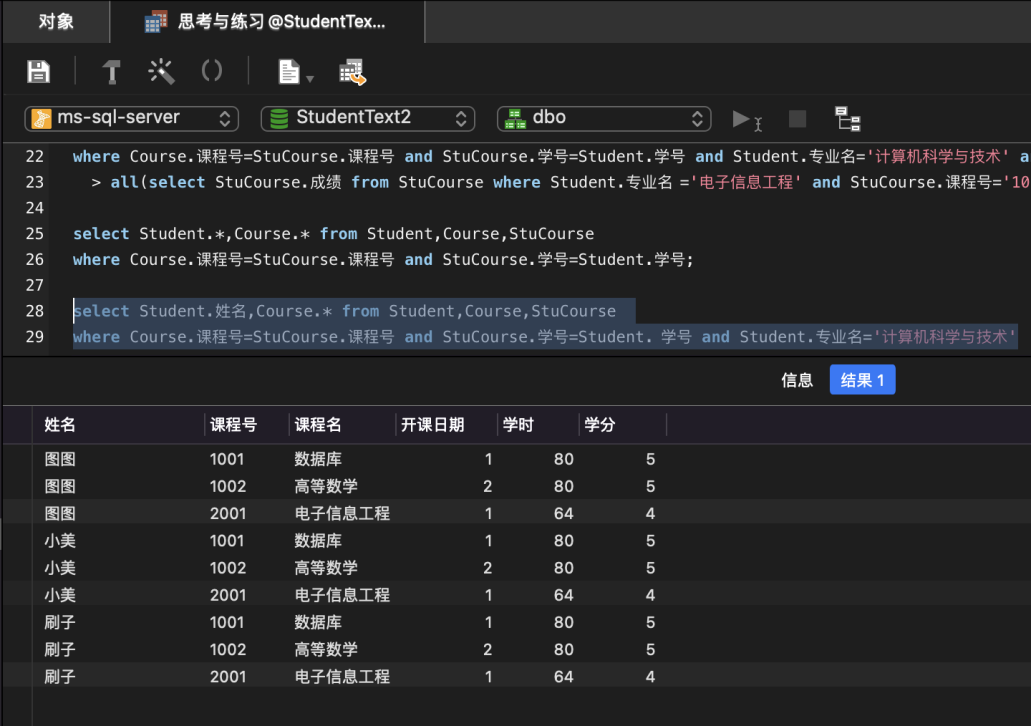
* 1. 查询“计算机科学与技术”专业学生姓名及其选修课程详情。

SQL语句：

select Student.姓名,Course.\* from Student,Course,StuCourse

where Course.课程号=StuCourse.课程号 and StuCourse.学号=Student. 学号 and Student.专业名='计算机科学与技术'

实验结果：



* 1. 查询“计算机科学与技术”专业学生的最高成绩和最低成绩。

SQL语句：

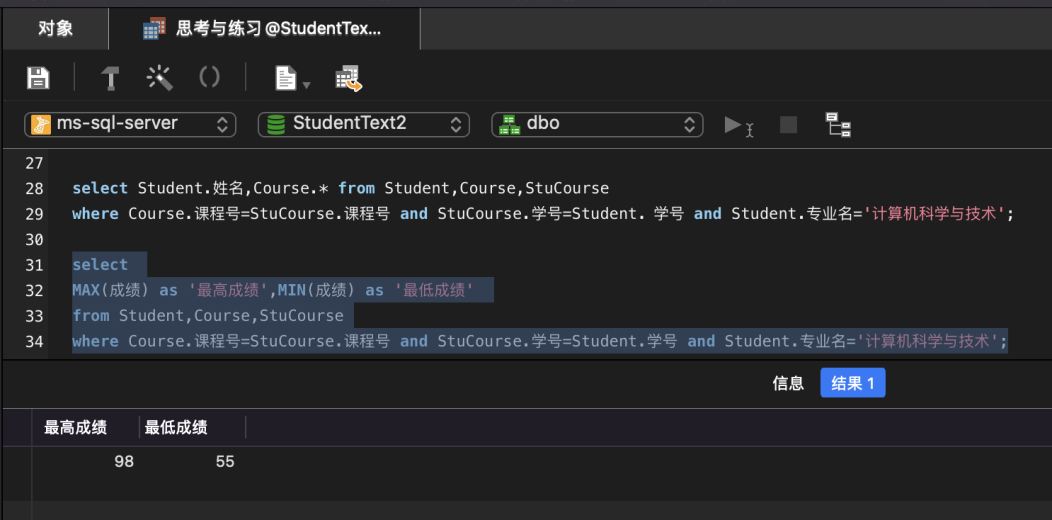
select

MAX(成绩) as '最高成绩',MIN(成绩) as '最低成绩'

from Student,Course,StuCourse

where Course.课程号=StuCourse.课程号 and StuCourse.学号=Student. 学号and Student.专业名='计算机科学与技术';

实验结果：



* 1. 统计“计算机科学与技术”专业各学生的平均成绩。

SQL语句：

select

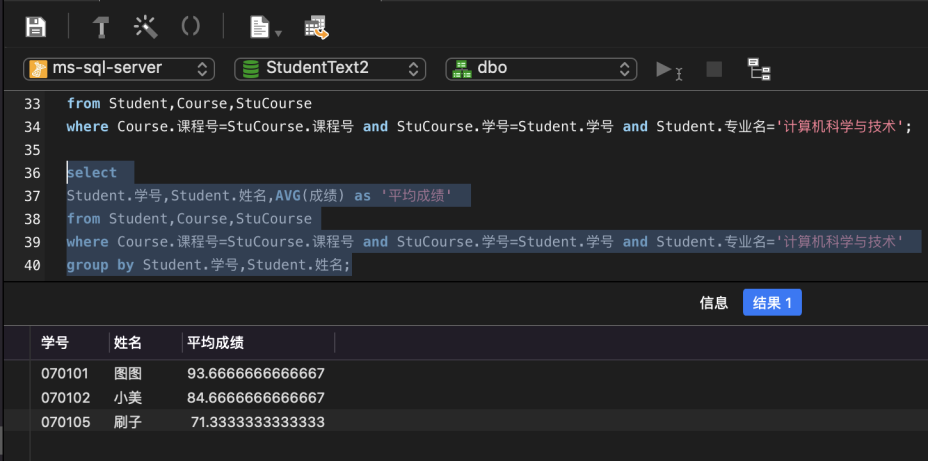
Student.学号,Student.姓名,AVG(成绩) as '平均成绩'

from Student,Course,StuCourse

where Course.课程号=StuCourse.课程号 and StuCourse.学号=Student. 学号 and Student.专业名='计算机科学与技术'

group by Student.学号,Student.姓名;

实验结果：



* 1. 统计各专业平均成绩在80分以上的学生人数。

SQL语句：

select

平均成绩.专业名,

COUNT(\*) as 学生人数

from(

select Student.学号,Student.专业名,AVG(成绩) as '平均成绩' ,count(\*) as '学生人数' from Student,Course,StuCourse

where Course.课程号=StuCourse.课程号 and StuCourse.学号=Student. 学号

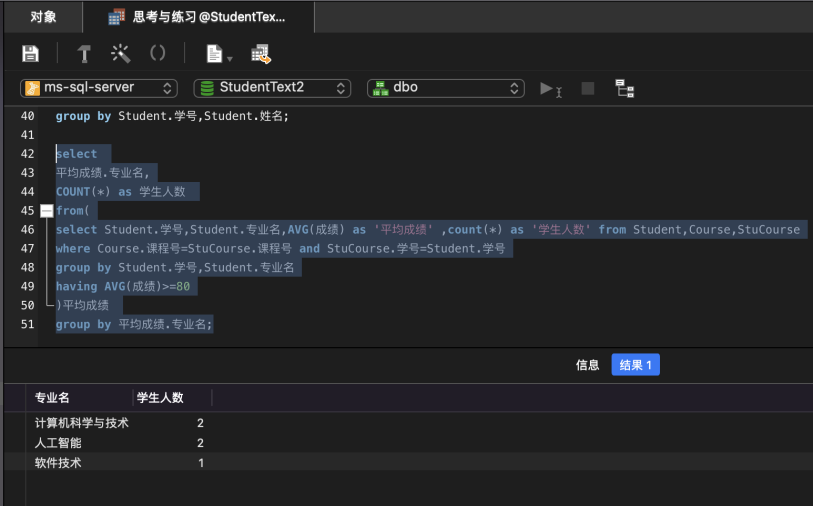
group by Student.学号,Student.专业名

having AVG(成绩)>=80

)平均成绩

group by 平均成绩.专业名;

实验结果：



* 1. 将各学生的学号和姓名按平均成绩降序排列。

SQL语句：

select

Student.学号,Student.姓名,AVG(成绩) as '平均成绩'

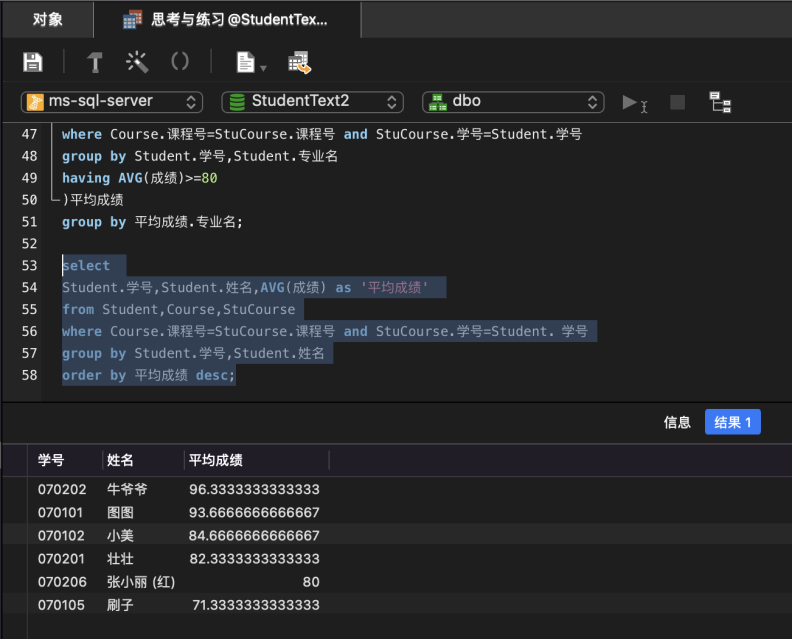
from Student,Course,StuCourse

where Course.课程号=StuCourse.课程号 and StuCourse.学号=Student. 学号

group by Student.学号,Student.姓名

order by 平均成绩 desc;

实验结果：



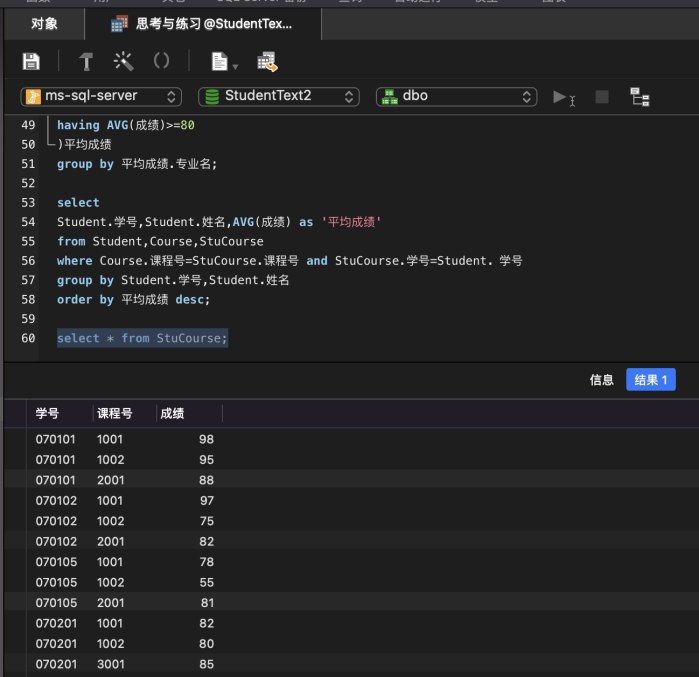
补充查询内容：

1. 查询StuCourse表中的全部数据。

SQL语句：

select \* from StuCourse;

实验结果：



1. 查询计算科学与技术专业学生的姓名和年龄。

SQL语句：

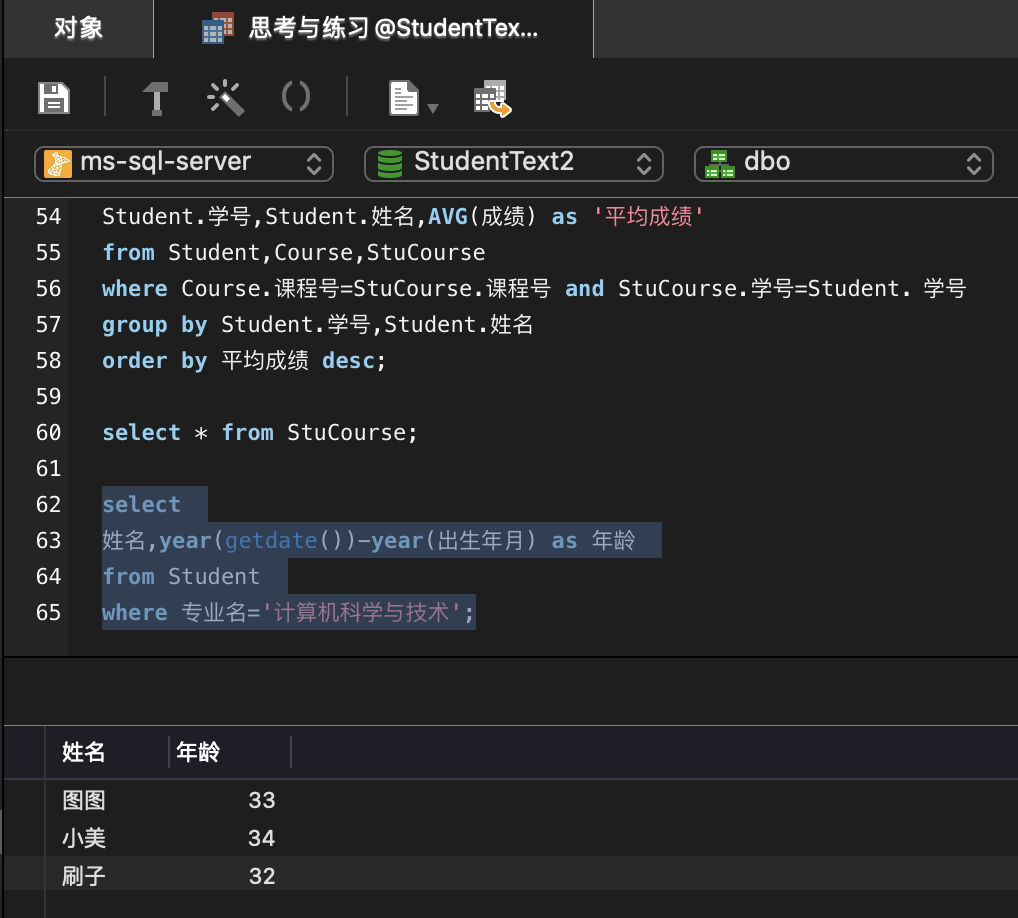
select

姓名,year(getdate())-year(出生年月) as 年龄

from Student

where 专业名='计算机科学与技术';

实验结果：



1. 查询成绩在70～80分的学生的学号、课程号和成绩。

SQL语句：

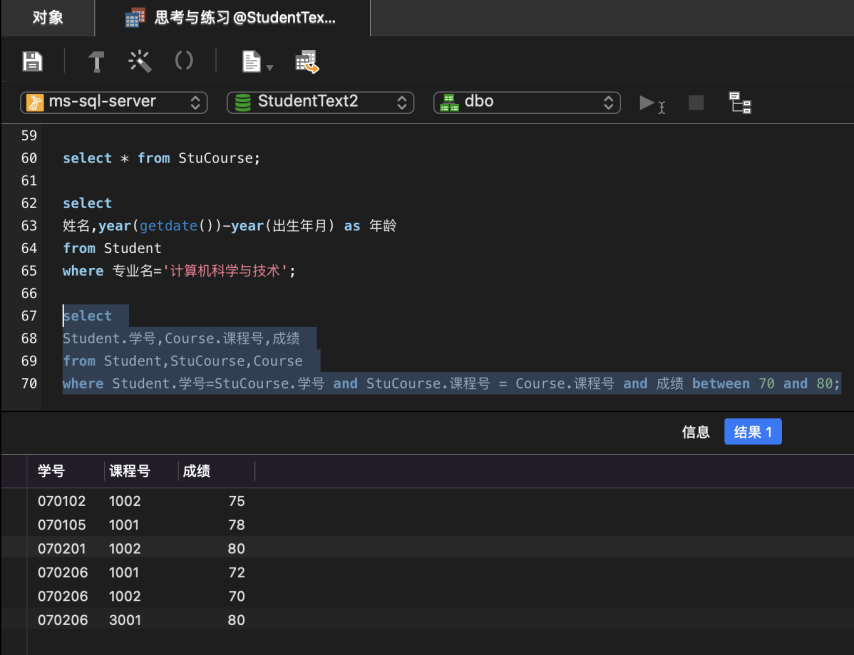
select

Student.学号,Course.课程号,成绩

from Student,StuCourse,Course

where Student.学号=StuCourse.学号 and StuCourse.课程号 = Course.课程号 and 成绩 between 70 and 80;

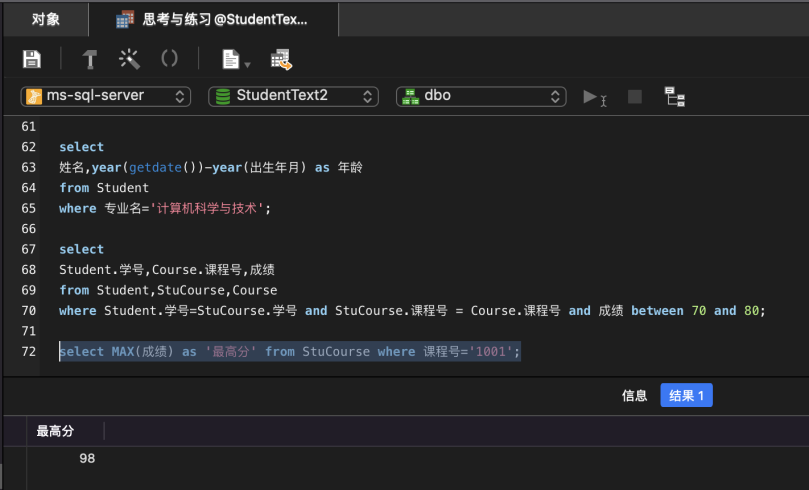
实验结果：



1. 查询1001课程的最高分。

SQL语句：

select MAX(成绩) as '最高分' from StuCourse where 课程号='1001';实验结果：



1. 查询计算科学与技术专业学生的最大年龄和最小年龄。

SQL语句：

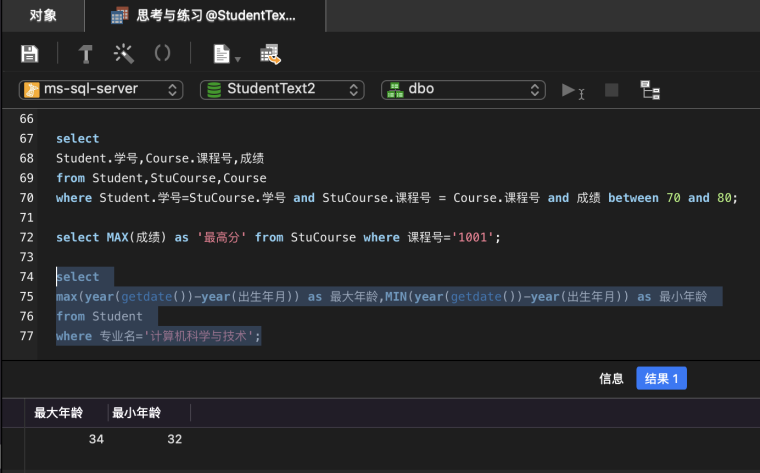
select

max(year(getdate())-year(出生年月)) as 最大年龄,MIN(year(getdate())-year(出生年月)) as 最小年龄

from Student

where 专业名='计算机科学与技术';

实验结果：



1. 统计每个专业的学生人数。

SQL语句：

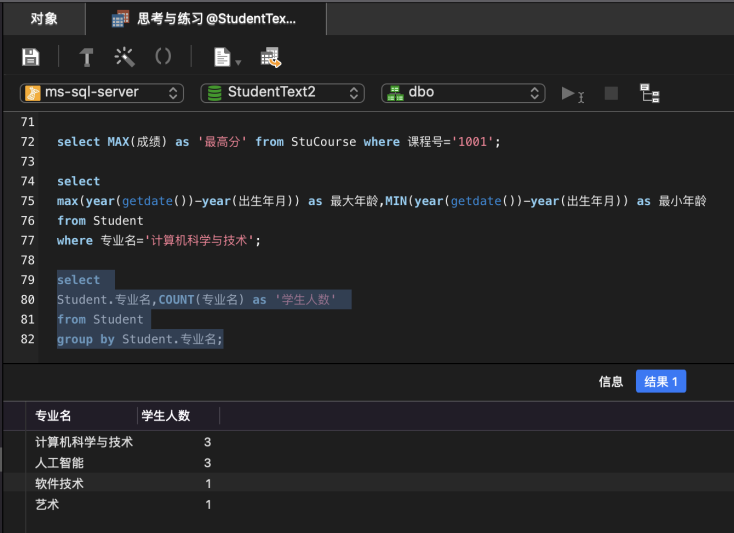
select

Student.专业名,COUNT(专业名) as '学生人数'

from Student

group by Student.专业名;

实验结果：



1. 统计每门课程的选课人数和最高成绩。

SQL语句：

select

distinct Course.课程名,COUNT(课程名)as '选课人数',max(成绩) as '最高成绩'

from Course,StuCourse

where Course.课程号=StuCourse.课程号

group by Course.课程名;

实验结果：



1. 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。

SQL语句：

select Student.学号,count(课程名) as '选课门数',sum(成绩) as '考试总成绩'

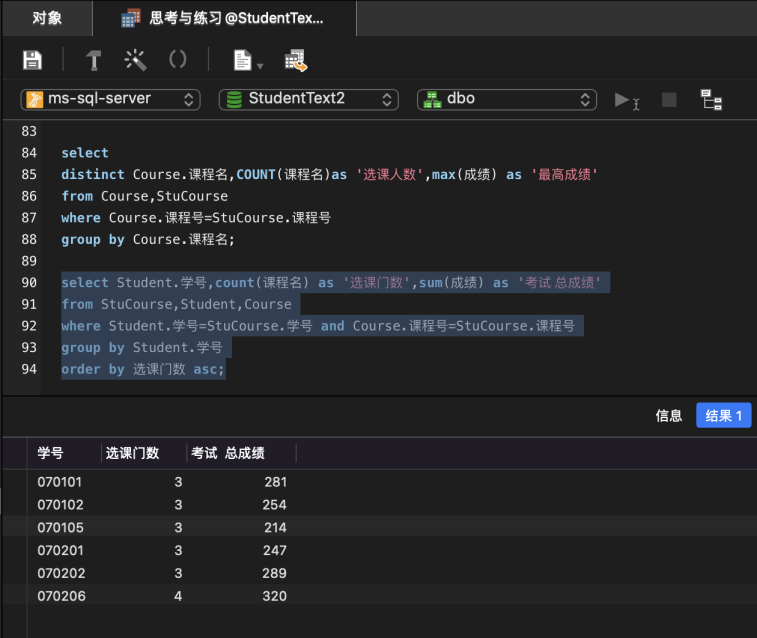
from StuCourse,Student,Course

where Student.学号=StuCourse.学号 and Course.课程号=StuCourse.课程号

group by Student.学号

order by 选课门数 asc;

实验结果：



1. 列出总成绩超过270的学生的学号和总成绩。

SQL语句：

select Student.学号,sum(成绩) as '总成绩'

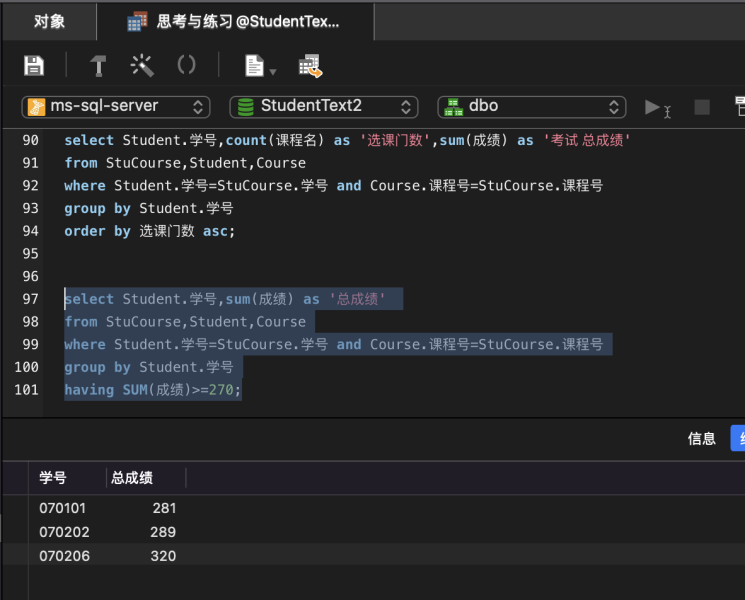
from StuCourse,Student,Course

where Student.学号=StuCourse.学号 and Course.课程号=StuCourse.课程号

group by Student.学号

having SUM(成绩)>=270;

实验结果：



1. 查询选了1002课程的学生姓名和所在专业。

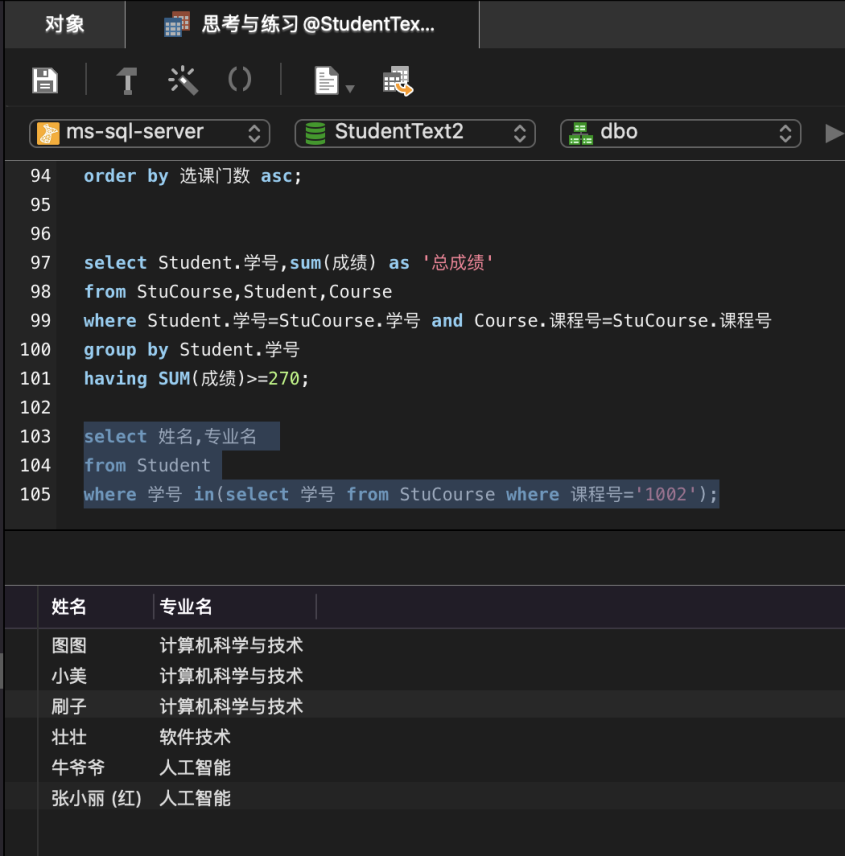
SQL语句：

select 姓名,专业名

from Student

where 学号 in(select 学号 from StuCourse where 课程号='1002');

实验结果：



1. 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。

SQL语句：

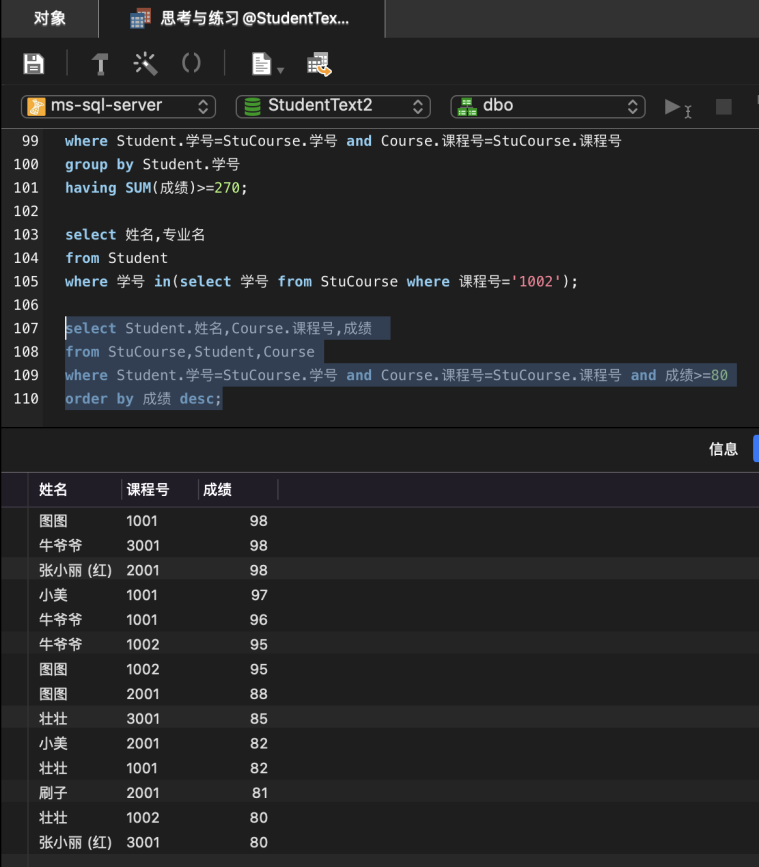
select Student.姓名,Course.课程号,成绩

from StuCourse,Student,Course

where Student.学号=StuCourse.学号 and Course.课程号=StuCourse.课程号 and 成绩>=80

order by 成绩 desc;

实验结果：



1. 查询与“高等数学1”在同一学期开设的课程的课程名和开课学期。

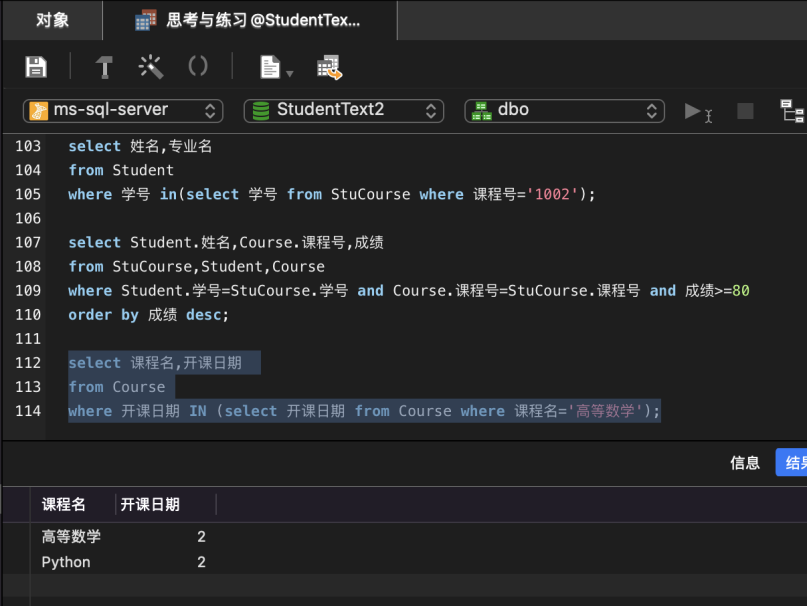
SQL语句：

select 课程名,开课日期

from Course

where 开课日期 IN (select 开课日期 from Course where 课程名='高等数学');

实验结果：



1. 查询与丁一平年龄相同的学生的姓名、所在专业和年龄。

SQL语句：

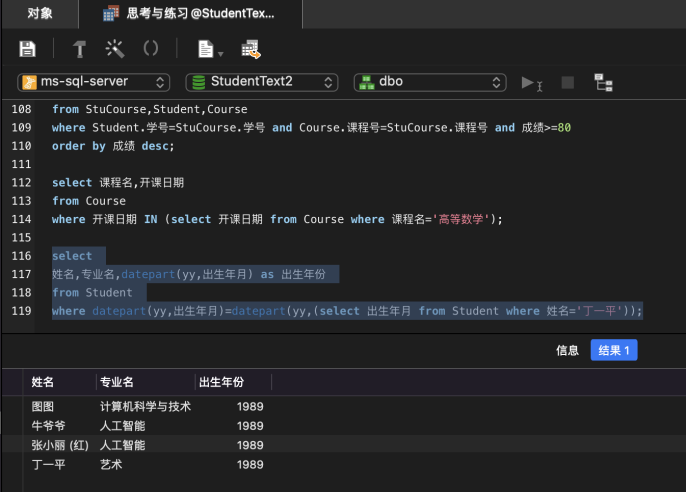
select

姓名,专业名,datepart(yy,出生年月) as 出生年份

from Student

where datepart(yy,出生年月)=datepart(yy,(select 出生年月 from Student where 姓名='丁一平'));

实验结果：



1. 查询哪些课程没有学生选修，列出课程号和课程名。

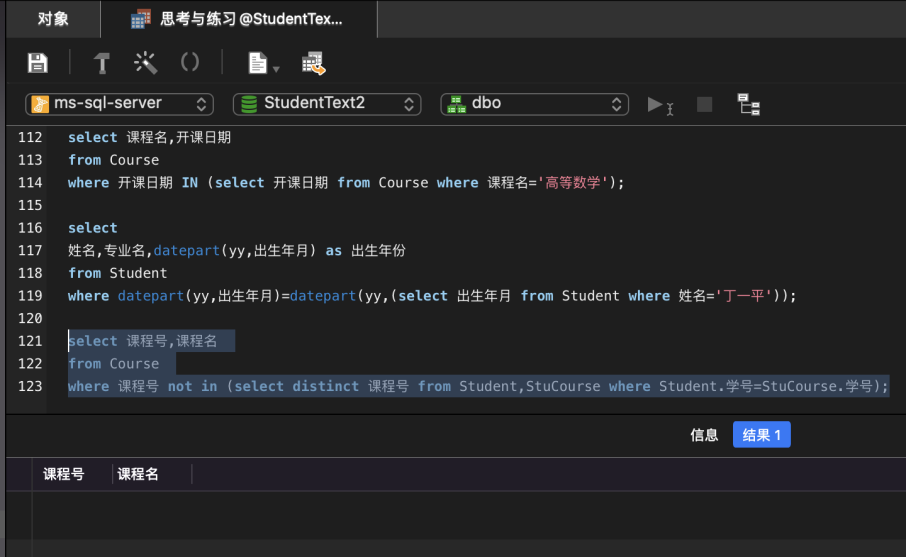
SQL语句：

select 课程号,课程名

from Course

where 课程号 not in (select distinct 课程号 from Student,StuCourse where Student.学号=StuCourse.学号);

实验结果：



1. 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。

SQL语句：

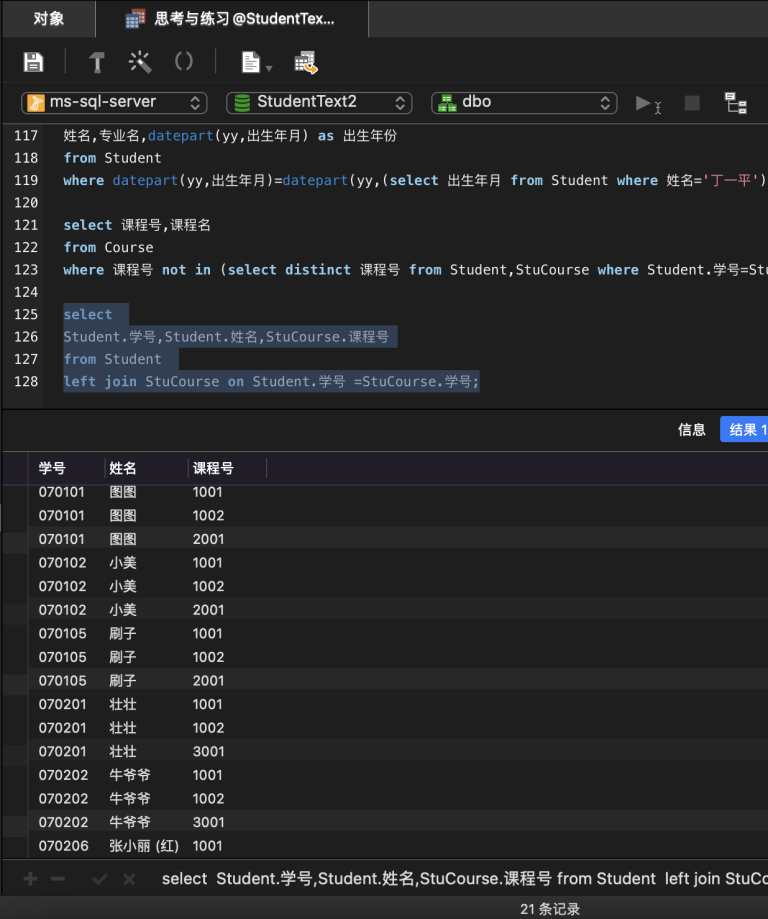
select

Student.学号,Student.姓名,StuCourse.课程号

from Student

left join StuCourse on Student.学号 =StuCourse.学号;

实验结果：



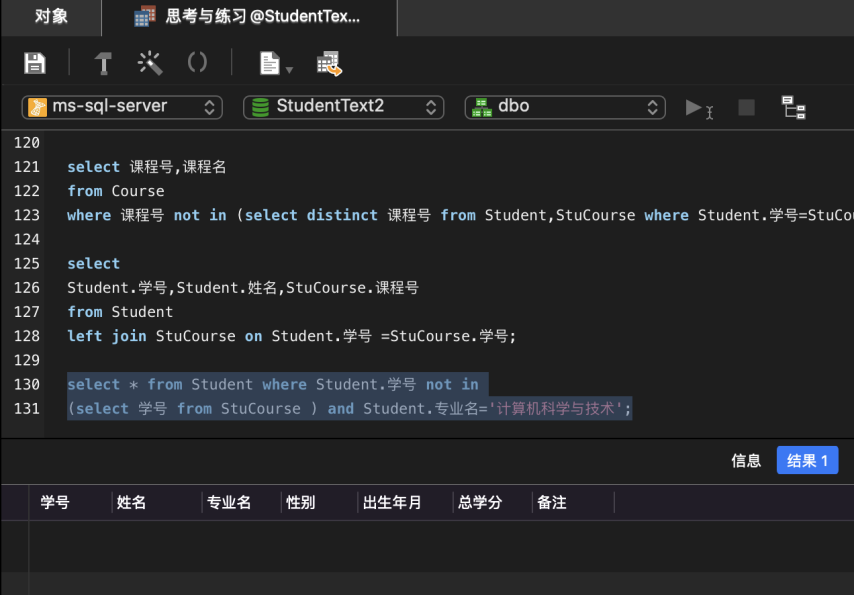
1. 查询计算机科学与技术专业哪些学生没有选课，列出学生姓名。

SQL语句：

select \* from Student where Student.学号 not in

(select 学号 from StuCourse ) and Student.专业名='计算机科学与技';

实验结果：



1. 查询计算机科学与技术专业年龄最大的三个学生的姓名和年龄。top 3

SQL语句：

select

top 3 姓名, 出生年月

from Student

where 专业名='计算机科学与技术'

order by 出生年月 asc;

实验结果：



1. 列出“高等数学”课程考试成绩前三名的学生的学号、姓名、所在系和高等数学的成绩top3。

SQL语句：

select

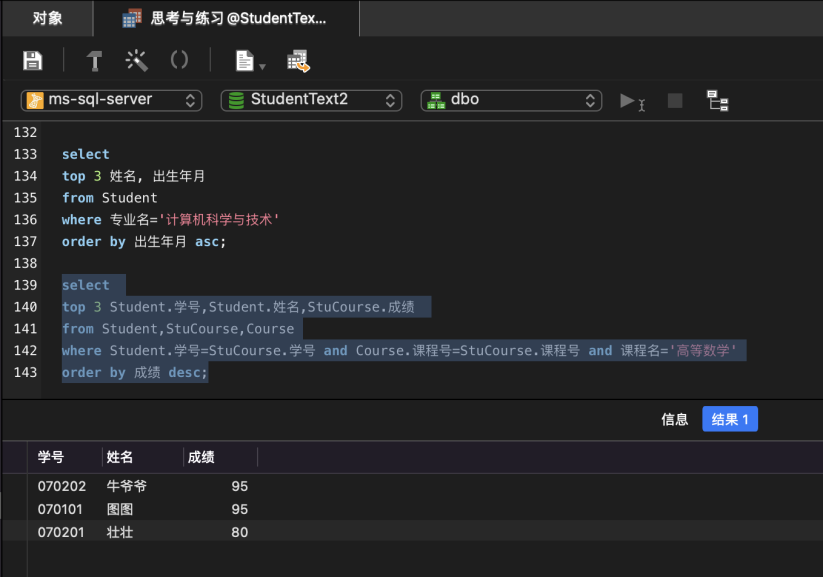
top 3 Student.学号,Student.姓名,StuCourse.成绩

from Student,StuCourse,Course

where Student.学号=StuCourse.学号 and Course.课程号=StuCourse.课程号 and 课程名='高等数学'

order by 成绩 desc;

实验结果：



1. 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。

SQL语句：

select

top 2 Student.学号,count(课程名) as '选课门数'

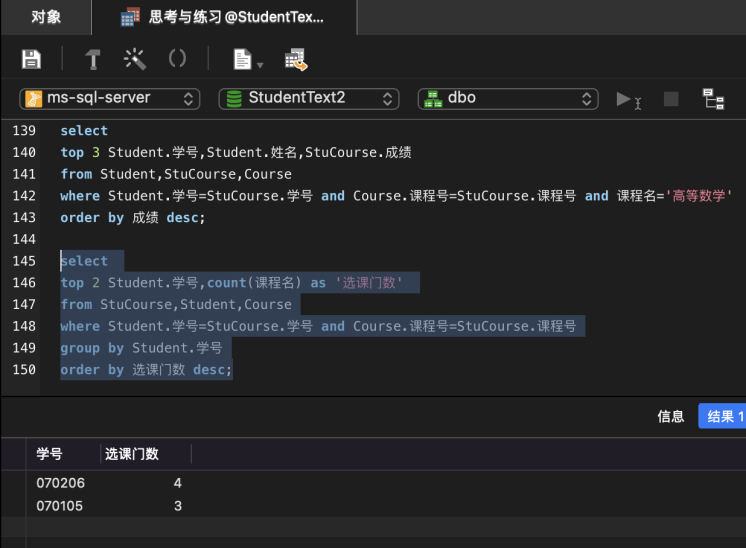
from StuCourse,Student,Course

where Student.学号=StuCourse.学号 and Course.课程号=StuCourse.课程号

group by Student.学号

order by 选课门数 desc;

实验结果：



提示：查询排名前几的方法，select语句中的输出项为top关键字。

用子查询实现如下查询：

1. 查询选了“1001”课程的学生姓名和所在系。

SQL语句：

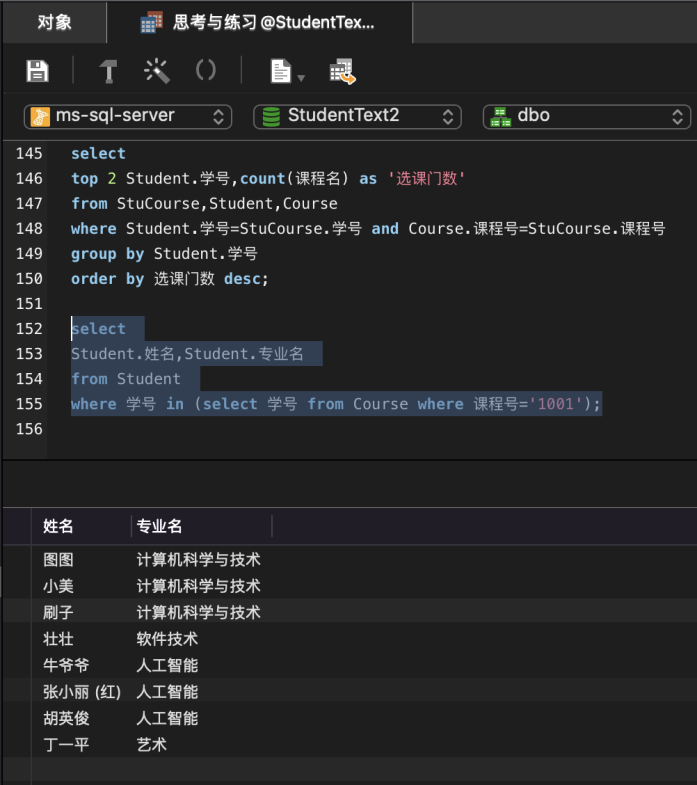
select

Student.姓名,Student.专业名

from Student

where 学号 in (select 学号 from Course where 课程号='1001');

实验结果：



1. 查询电子信息工程专业成绩80分以上的学生的学号和姓名。

SQL语句：

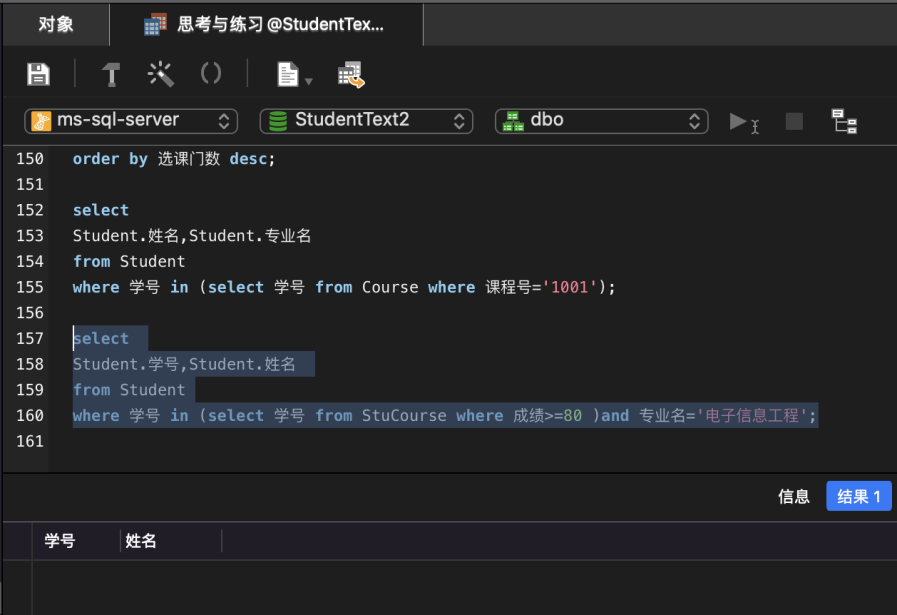
select

Student.学号,Student.姓名

from Student

where 学号 in (select 学号 from StuCourse where 成绩>=80 )and 专业名='电子信息工程';

实验结果：



1. 查询计算机科学与技术专业考试成绩最高的学生的姓名。

SQL语句：

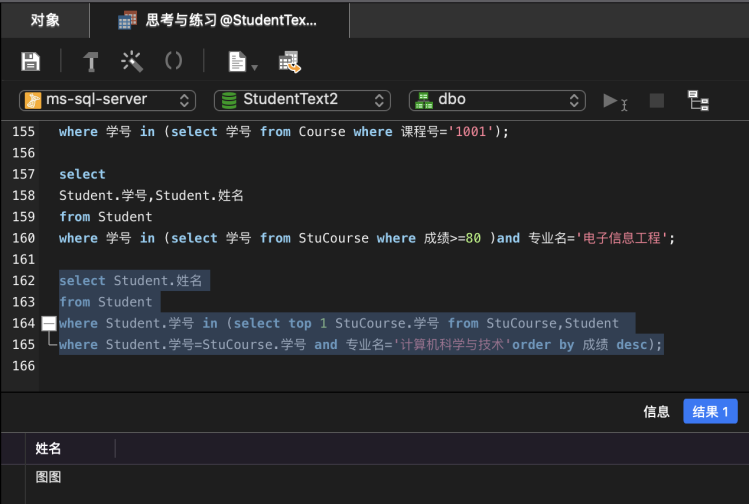
select Student.姓名

from Student

where Student.学号 in (select top 1 StuCourse.学号 from StuCourse,Student

where Student.学号=StuCourse.学号 and 专业名='计算机科学与技术'order by 成绩 desc);

实验结果：



1. 查询年龄最大的男生的姓名、所在专业和年龄。

SQL语句：

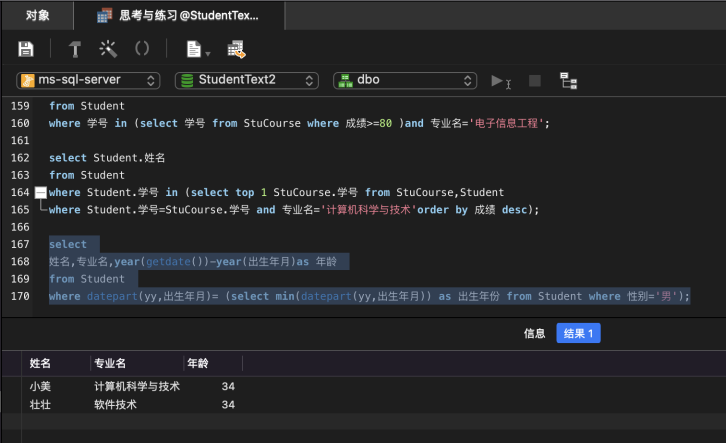
select

姓名,专业名,year(getdate())-year(出生年月)as 年龄

from Student

where datepart(yy,出生年月)= (select min(datepart(yy,出生年月)) as 出生年份 from Student where 性别='男');

实验结果：



1. 查询考试成绩高于1001课程平均成绩的学生的学号和成绩。

SQL语句：

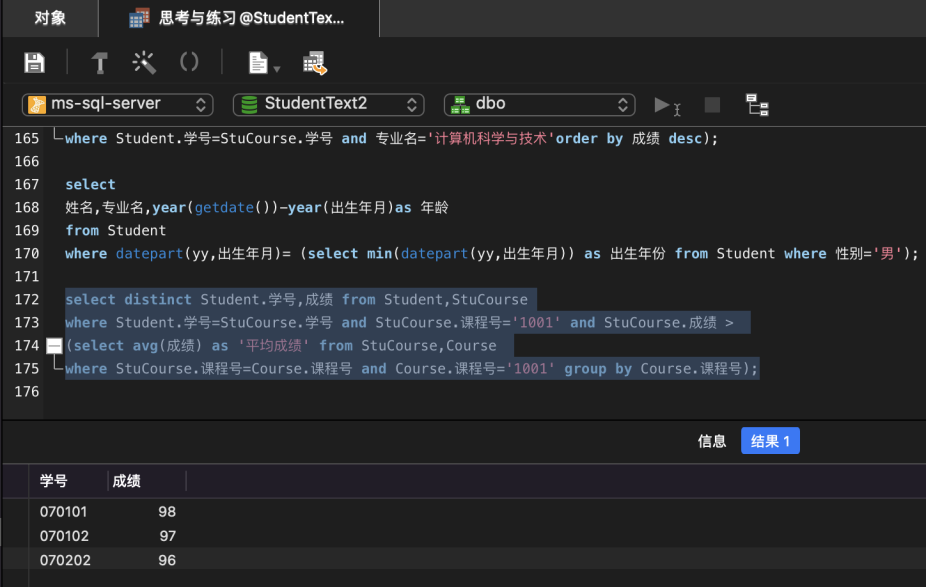
select distinct Student.学号,成绩 from Student,StuCourse

where Student.学号=StuCourse.学号 and StuCourse.课程号='1001' and StuCourse.成绩 >

(select avg(成绩) as '平均成绩' from StuCourse,Course

where StuCourse.课程号=Course.课程号 and Course.课程号='1001' group by Course.课程号);

实验结果：



1. 查询计算机科学与技术专业学生考试成绩高于计算机科学与技术专业学生平均成绩的学生的姓名、考试的课程名和考试成绩。

SQL语句：

select

Student.姓名,StuCourse.课程号,成绩

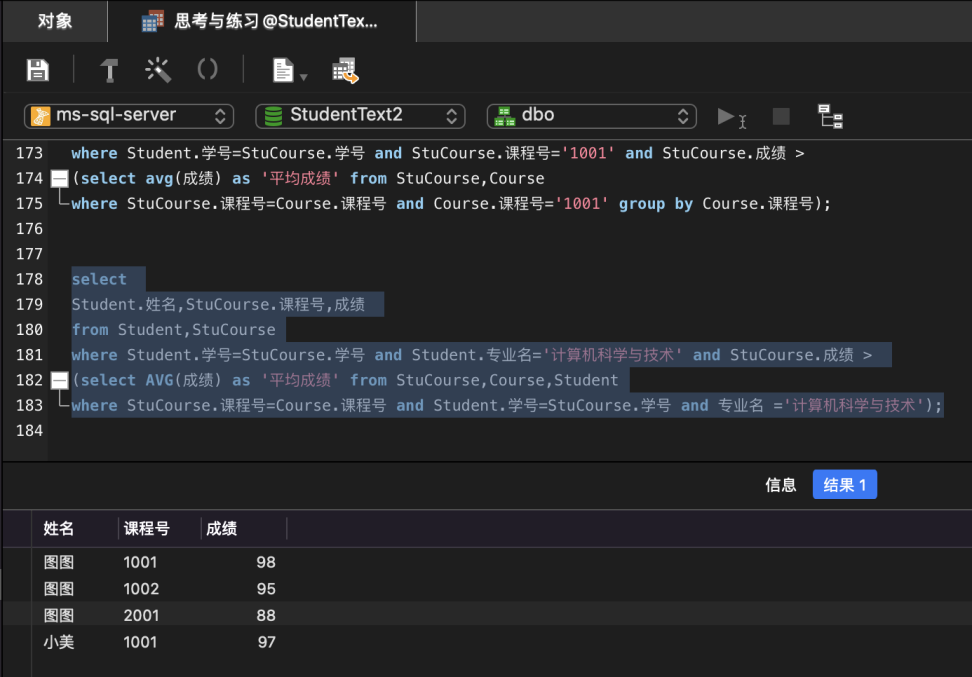
from Student,StuCourse

where Student.学号=StuCourse.学号 and Student.专业名='计算机科学与技术' and StuCourse.成绩 >

(select AVG(成绩) as '平均成绩' from StuCourse,Course,Student

where StuCourse.课程号=Course.课程号 and Student.学号=StuCourse.学号 and 专业名 ='计算机科学与技术');

实验结果：



1. 查询没选“电路基础”的学生姓名和所在专业。

SQL语句：

select

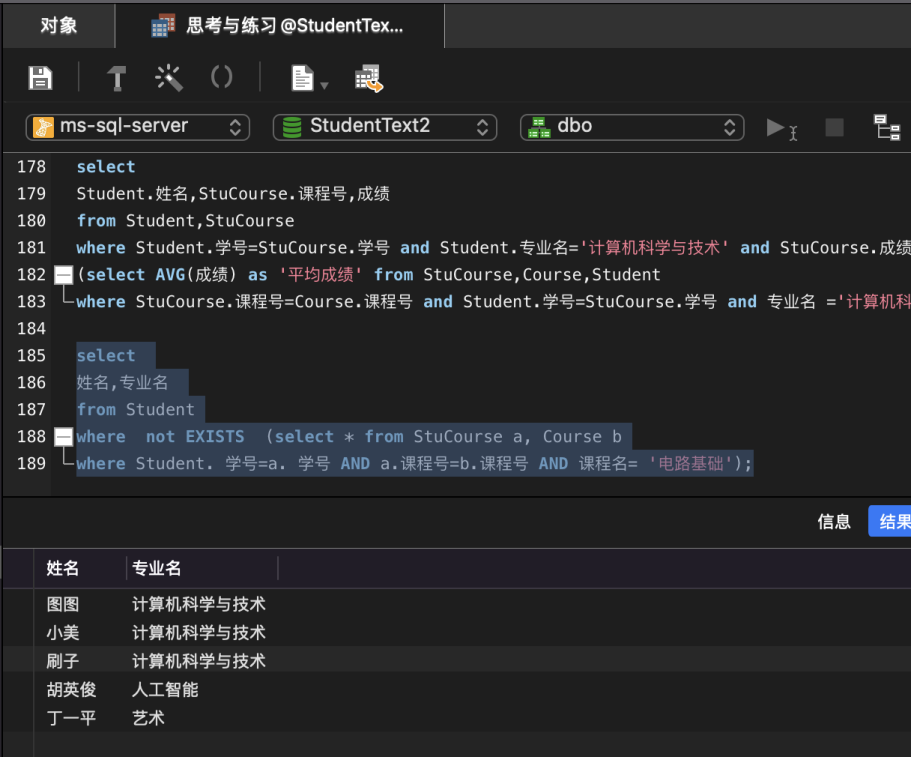
姓名,专业名

from Student

where not EXISTS (select \* from StuCourse a, Course b

where Student. 学号=a. 学号 AND a.课程号=b.课程号 AND 课程名= '电路基础');

实验结果：



1. 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。

SQL语句：

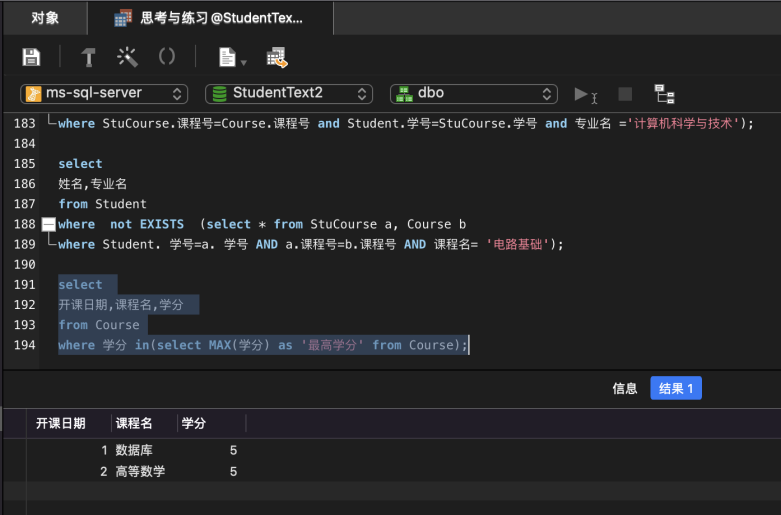
select

开课日期,课程名,学分

from Course

where 学分 in(select MAX(学分) as '最高学分' from Course);

实验结果：



1. 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”。

SQL语句：

select

distinct Course.课程名,Course.课程号,COUNT(课程名)as '选课人数',选课 情况=

case when COUNT(课程名) <100 and COUNT(课程名)>40 then '一般'

when COUNT(课程名)<39 and COUNT(课程名)>1 then '人少' else '无人 选' end

from Course left join StuCourse on Course.课程号=StuCourse.课程 号

group by Course.课程名,Course.课程号;

实验结果：



# 三、实验小结

通过本次实验，掌握了SELECT语句的基本查询和嵌套查询，学会了order by，having等子句的作用和使用方法。学会了使用order by对查询结果进行排序。掌握了一些基础查询和复杂查询。