实验5 连接查询及视图的使用

* **实验环境：**

操作系统：Windows 10 专业版

数据库管理系统：8.0.18 MySQL Community Server - GPL

客户端：MySQL Workbench 8.0 Community+VScode 1.71.2

数据库模式：Test2

学生表：Student(Sno,Sname,Ssex,Sage,Sdept)

课程表：Course(Cno,Cname,Cpno,Ccredit)

选课表：SC(Sno,Cno,Grade)

数据库的值：如下图所示



1. 在学生表和学生选课表中，分别使用内连接、左连接、右连接、全连接查询学生的学号、姓名、性别、成绩。

实验代码：

-- 内连接

select student.Sno,Sname,Ssex,Grade

from Student,Sc

where Student.Sno=Sc.Sno;

-- 左连接

select student.Sno,Sname,Ssex,Grade

from Student left outer join Sc on(Student.Sno=Sc.Sno);

-- 右连接

select student.Sno,Sname,Ssex,Grade

from Student right outer join Sc on(Student.Sno=Sc.Sno);

-- 全连接

-- UNION ALL 和排除联接

-- 这样可以保留同一个表中重复的行，并且保证两个子查询不会产生重复记录。

-- 由于不需要排序和删除重复项，因此对于大型结果集，这可能会大大提高效率。

select Student.Sno,Sname,Ssex,Grade from Student left JOIN Sc on Student.Sno=Sc.Sno

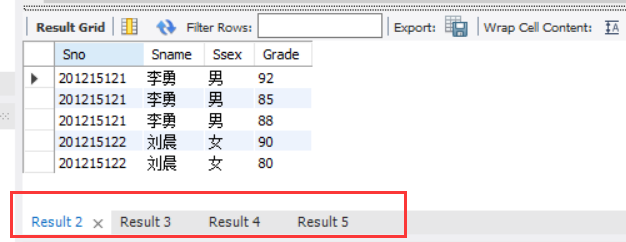
union all

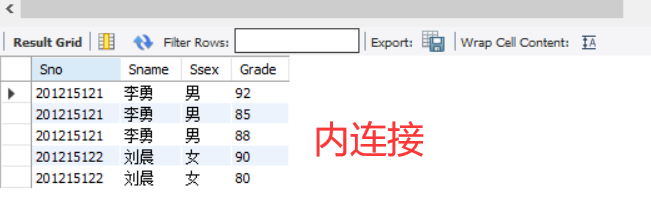
select Student.Sno,Sname,Ssex,Grade from Student right JOIN Sc on Student.Sno=Sc.Sno

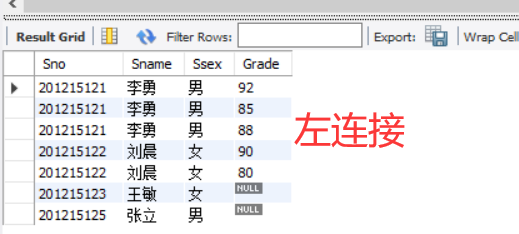
WHERE Sc.Sno is NULL

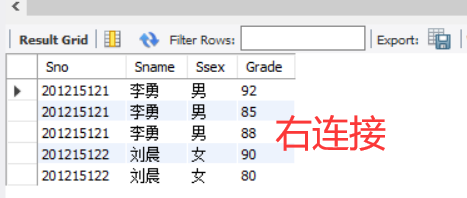
运行结果：

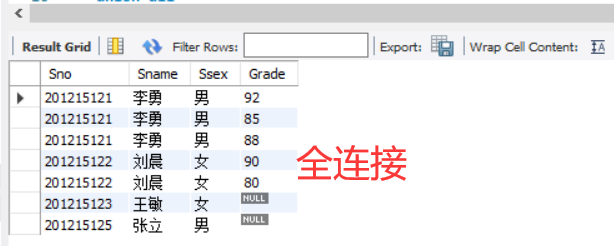
可以看到运行结果里有四个result，依次为内连接、左连接、右连接、全连接











1. 统计各课程平均分、总分、最高分、最低分、参考人数，查询结果包括课程名称（CourseName）、课程总分（SumScore）、课程平均分（AvgScore）、课程最高分（MaxScore）、课程最低分（MinScore）、参考人数（CourseCount）字段（使用INNER JOIN实现）。

实验代码：

-- 2、   统计各课程平均分、总分、最高分、最低分、参考人数，

-- 查询结果包括课程名称（CourseName）、课程总分（SumScore）、课程平均分（AvgScore）、

-- 课程最高分（MaxScore）、课程最低分（MinScore）、参考人数（CourseCount）字段

-- （使用INNER JOIN实现）。

select Course.Cname as Cname,

       sum(Grade) as SumScore,

       floor(avg(Grade)) as AvgScore,

       max(Grade) as MaxScore,

       min(Grade) as MinScore,

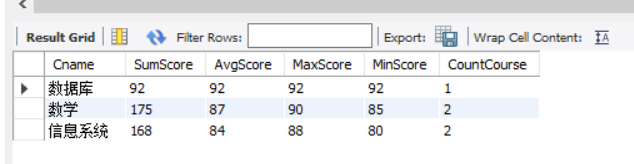
       count(Grade) as CountCourse

from Course

inner join Sc on Course.Cno=Sc.Cno

group by Sc.Cno

运行结果：



1. 将第2题所写的SQL语句创建成视图（V\_GetCourseScore），在视图中查询参考人数大于30小于50的所有课程信息

实验代码：

-- 如果视图已存在，则drop

DROP VIEW IF EXISTS  V\_GetCourseScore;

-- 创建视图

create view V\_GetCourseScore(Cname,SumScore,AvgScore,MaxScore,MinScore,CountCourse)

as

select Course.Cname,sum(Grade),floor(avg(Grade)),max(Grade),min(Grade),count(Grade)

from Course

inner join Sc on Course.Cno=Sc.Cno

group by Sc.Cno;

-- 查询视图

select \* from V\_GetCourseScore

where CountCourse>30 and CountCourse<50

运行结果：



4、创建学生平均成绩视图（V\_StudAvgScore），其中包括学生学号（StudNo）、姓名（StudName）、平均分（AvgScore）、总分（SumScore）、最高分（MaxScore）、最低分（MinScore）、课程门数（CountCourse）字段。

实验代码：

-- 如果视图已存在，则drop

drop view if exists V\_StudAvgScore;

-- 创建视图

create view V\_StudAvgScore(StudNo,StudName,AvgScore,SumScore,MaxScore,MinScore,CountCourse)

as

select Student.Sno,Sname,floor(avg(Grade)),sum(Grade),max(Grade),min(Grade),count(Grade)

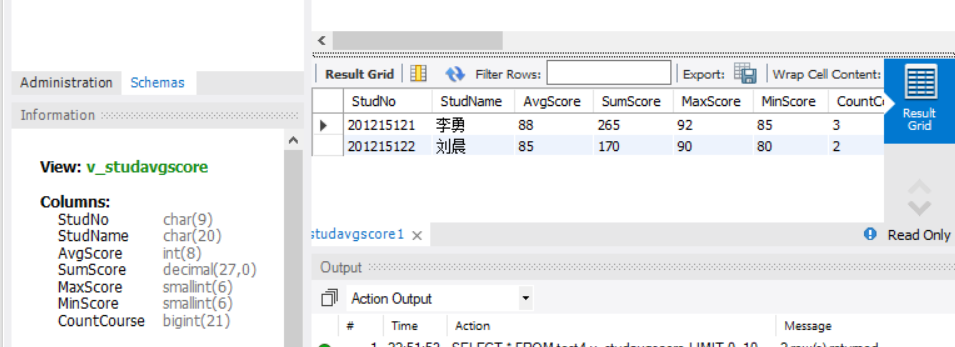
from Student

inner join Sc on Student.Sno=Sc.Sno

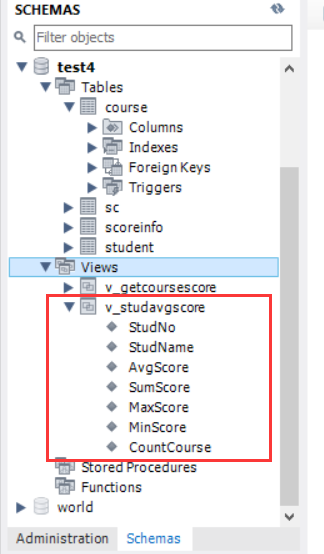
group by Sc.Sno

运行结果：

下图为查询全视图后的结果，左侧为详细属性



下图为创建成功的视图，保存在库中



5、利用第4题的结果，查询平均分在80~85和60~70的数据记录，其中包括学号、姓名、所在系、平均分、课程门数字段。

实验代码：

-- 如果要单独执行这一题的代码不要忘记先建立视图

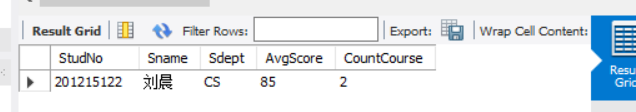
select StudNo,Sname,Sdept,AvgScore,CountCourse

from v\_studavgscore

join Student on StudNo=Sno

where AvgScore between 80 and 85 OR AvgScore between 60 and 70

运行结果：



6、利用第4题的结果，查询平均分大于80的学生基本信息，其中包括学号、姓名、性别、年龄、所在系、平均分字段。

实验代码：

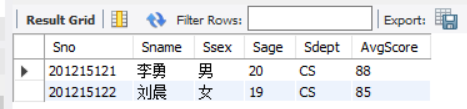
select Sno,Sname,Ssex,Sage,Sdept,AvgScore

from v\_studavgscore

join Student on StudNo=Sno

where AvgScore >80

运行结果：



* 实验总结：

1.关于内连接、左连接、右连接这三种操作我们都可以在MYSQL中直接通过运算符以及join去完美实现，但是全连接在SQL中的full join语法就不能在MYSQL中使用，这时就要考虑将左连接和右连接union all等效替代实现目标，最后添加这样可以保留同一个表中重复的行，并且保证两个子查询不会产生重复记录。且由于不需要排序和删除重复项，因此对于大型结果集，这可能会大大提高效率。

2.视图不同于普通的表，它会对部分属性的一些操作进行相对限制，本实验中尚未体现出视图限制用户的操作，但是充分体现出视图可以作为一个查询后的子窗口供用户去进行浏览以及二次查询，可以代替嵌套查询的一些功能，提高了效率。

3.编程时多考虑一些设备以及数据库的兼容性，例如既然我们在题目中创建了视图，就要考虑有的用户有有的用户没有，可以加上一条件判断，就像3、4题一样该drop就drop。