**《数据库系统原理》实验报告**

### **实验题目：**实验9 表数据的高级查询

**姓名：**

**实验日期：**

**实验内容及完成情况：**（可续页）

### 实验9 表数据的高级查询

**实验学时：2学时**

**实验类型：设计**

**实验要求：必做**

**一、实验目的**

本实验的目的是使学生熟练掌握MySQL查询分析器的使用方法，加深SQL语言查询语句的理解。熟练掌握数据查询中的分组、统计、计算和集合的操作方法。

**二、实验内容**

在数据库hub中完成以下操作。

1. 查询所有学生姓名及相应的选修课程名、成绩。

USE hub;

SELECT

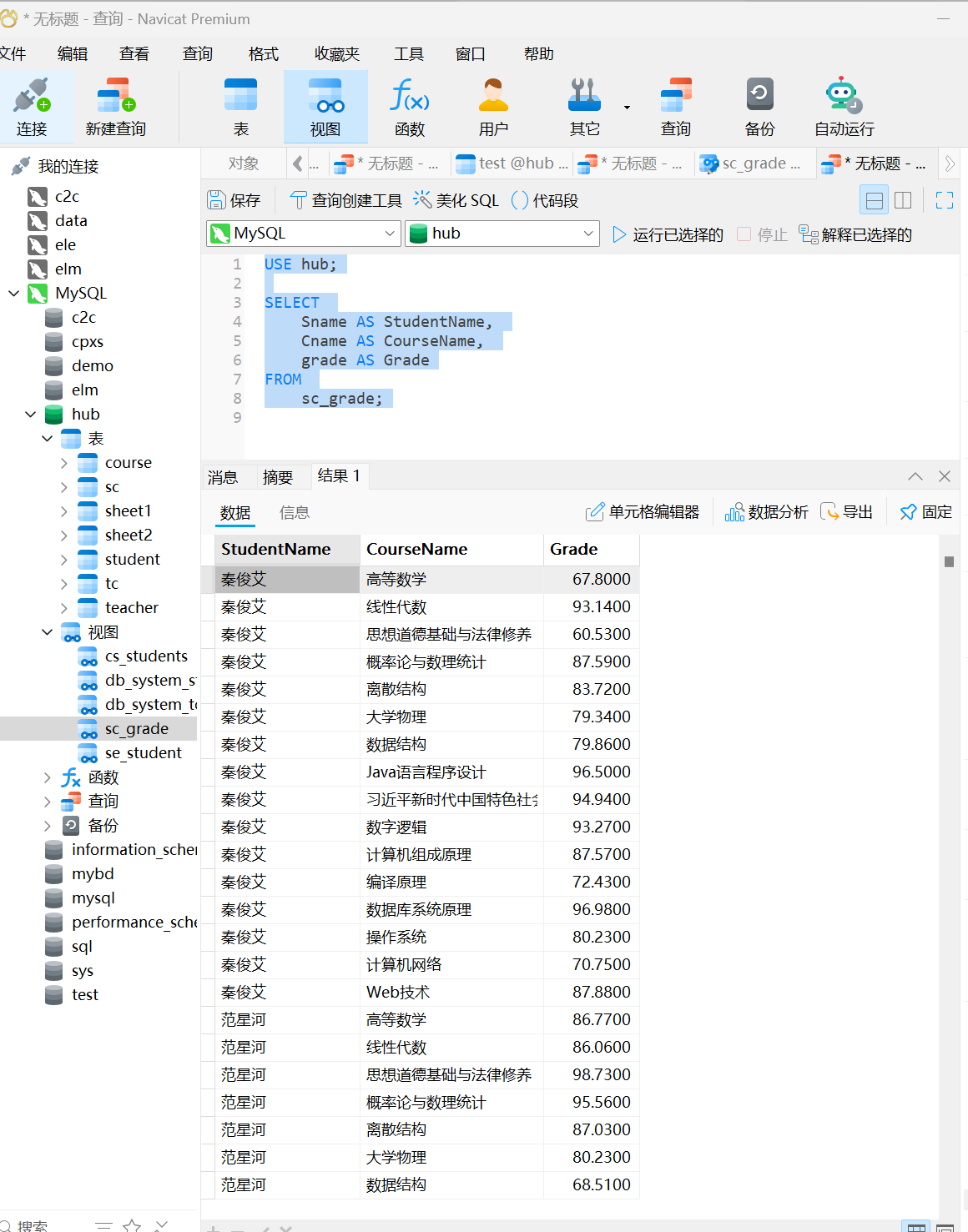
Sname AS StudentName,

Cname AS CourseName,

grade AS Grade

FROM

sc\_grade;



1. 查询系部为软件学院且数据库系统原理成绩在90分以上的学生名字和专业名称以及所属班级。

USE hub;

SELECT

se\_student.Sname AS StudentName,

se\_student.major AS MajorName,

se\_student.class AS ClassName

FROM

se\_student

JOIN

sc\_grade

ON

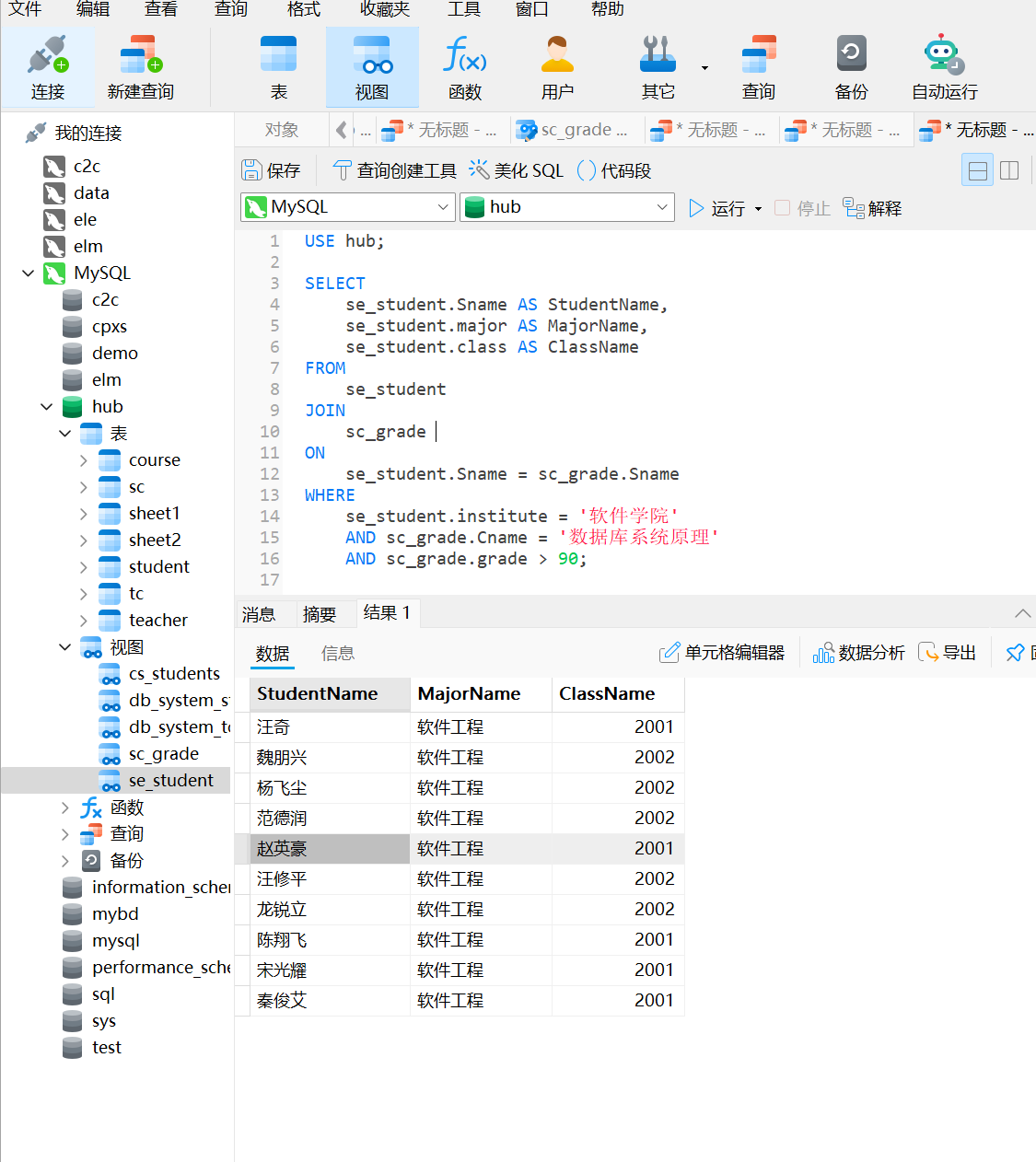
se\_student.Sname = sc\_grade.Sname

WHERE

se\_student.institute = '软件学院'

AND sc\_grade.Cname = '数据库系统原理'

AND sc\_grade.grade > 90;



1. 统计student表中院系为软件学院的学生数。

USE hub;

SELECT

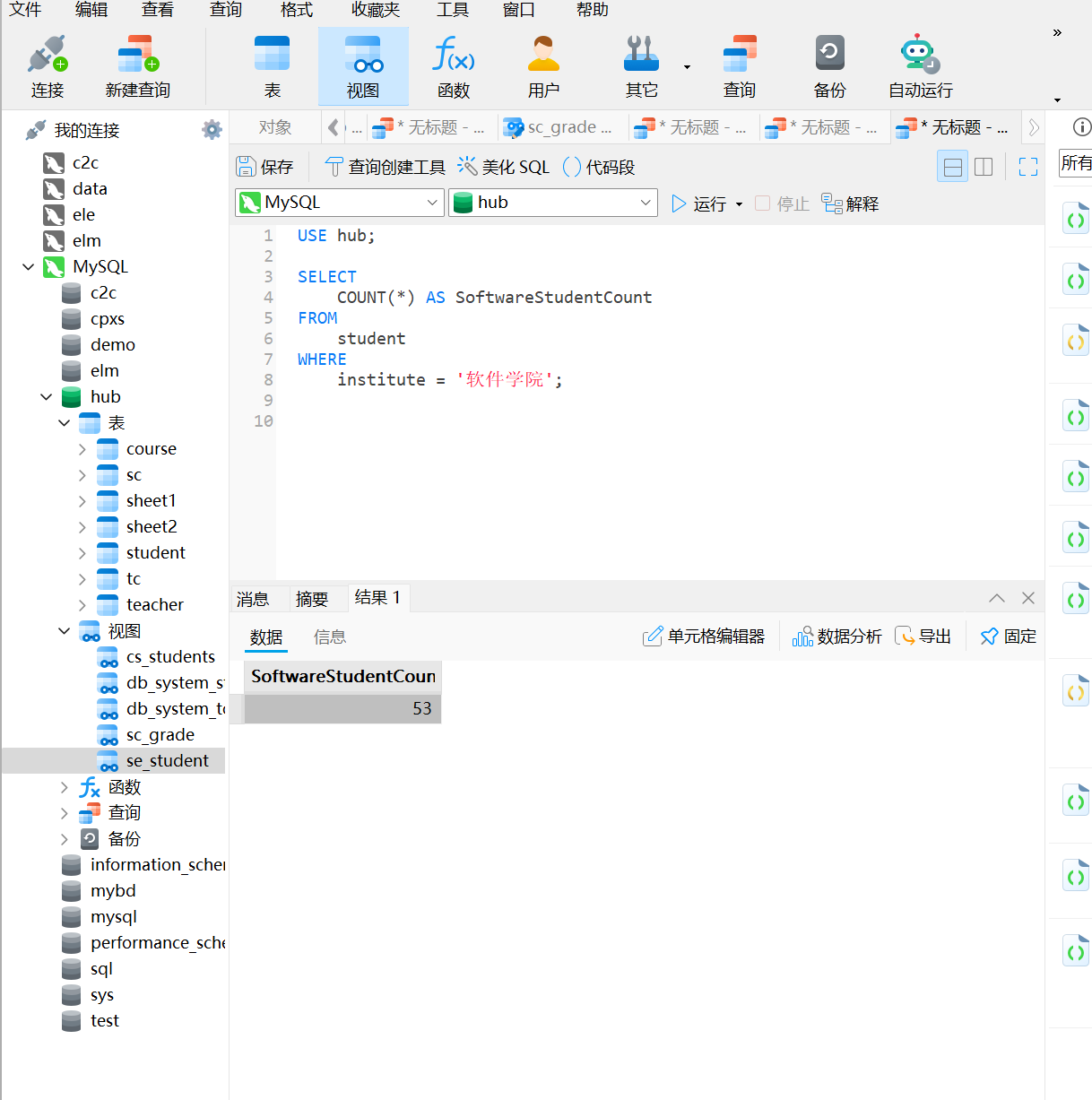
COUNT(\*) AS SoftwareStudentCount

FROM

student

WHERE

institute = '软件学院';



1. 统计sc表中数据库系统原理课程的平均成绩。

USE hub;

SELECT

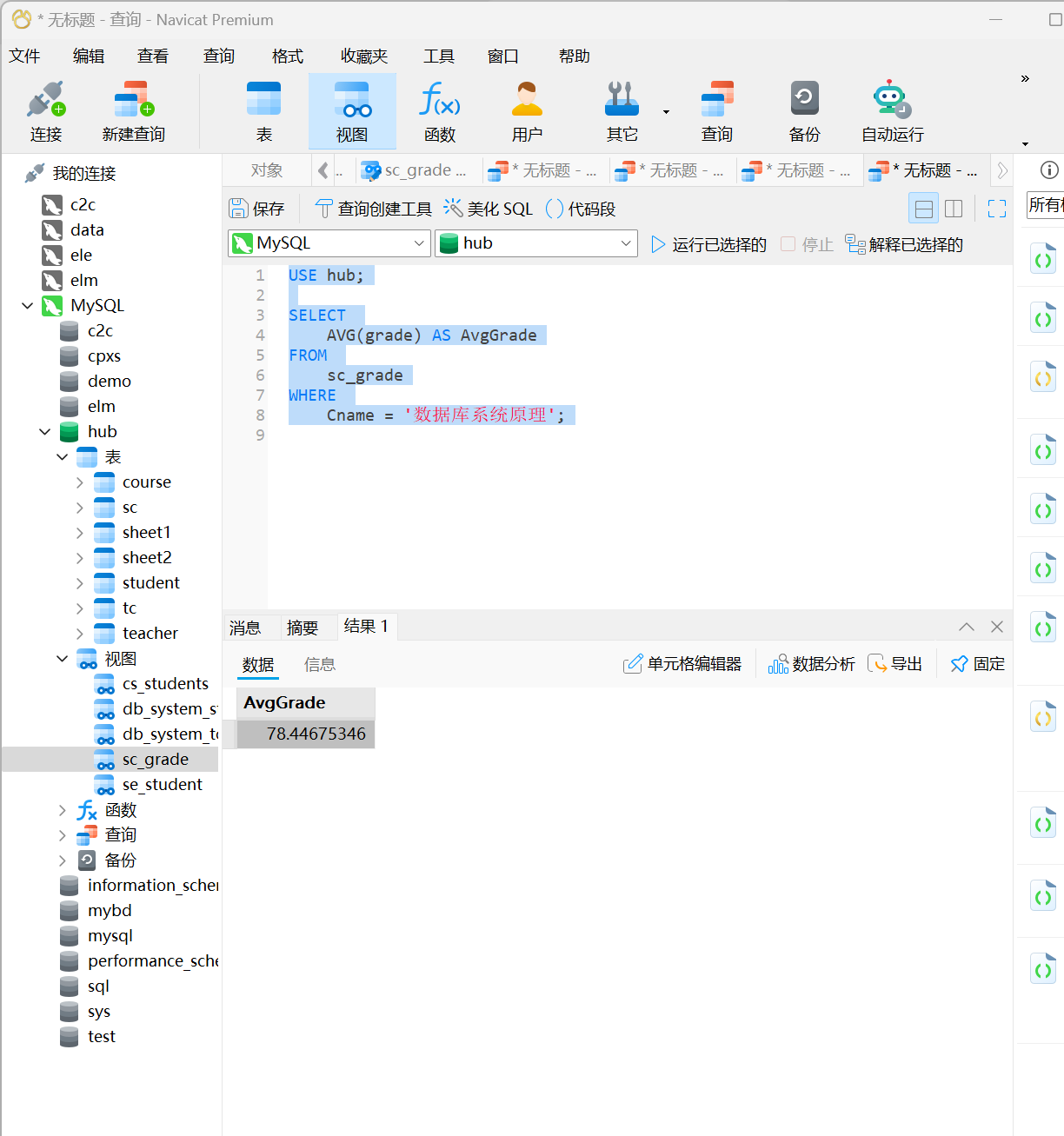
AVG(grade) AS AvgGrade

FROM

sc\_grade

WHERE

Cname = '数据库系统原理';



1. 统计course中开设课程总数。

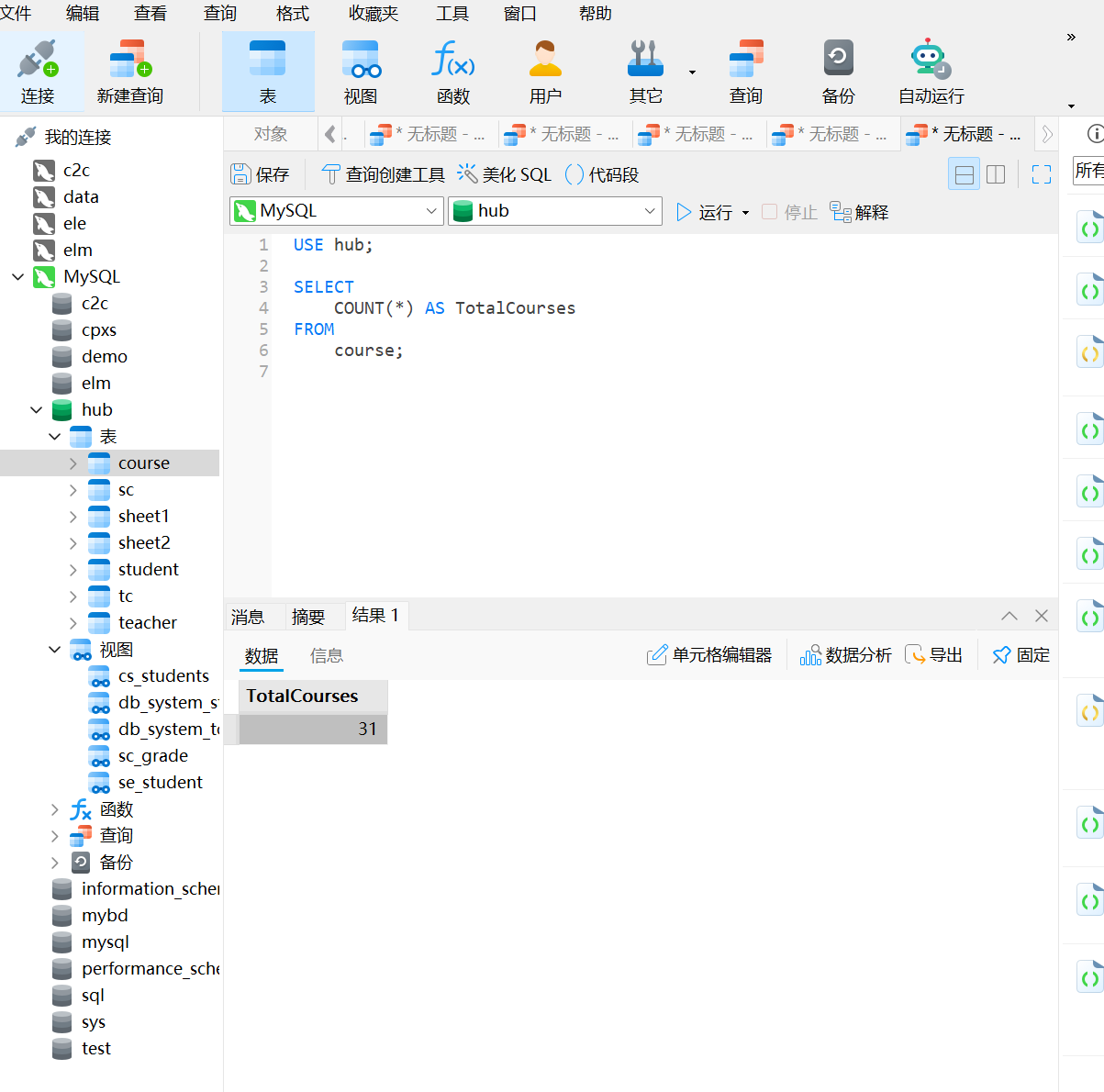
USE hub;

SELECT

COUNT(\*) AS TotalCourses

FROM

course;



1. 分组统计student表中各院系的学生人数。

USE hub;

SELECT

institute AS Department,

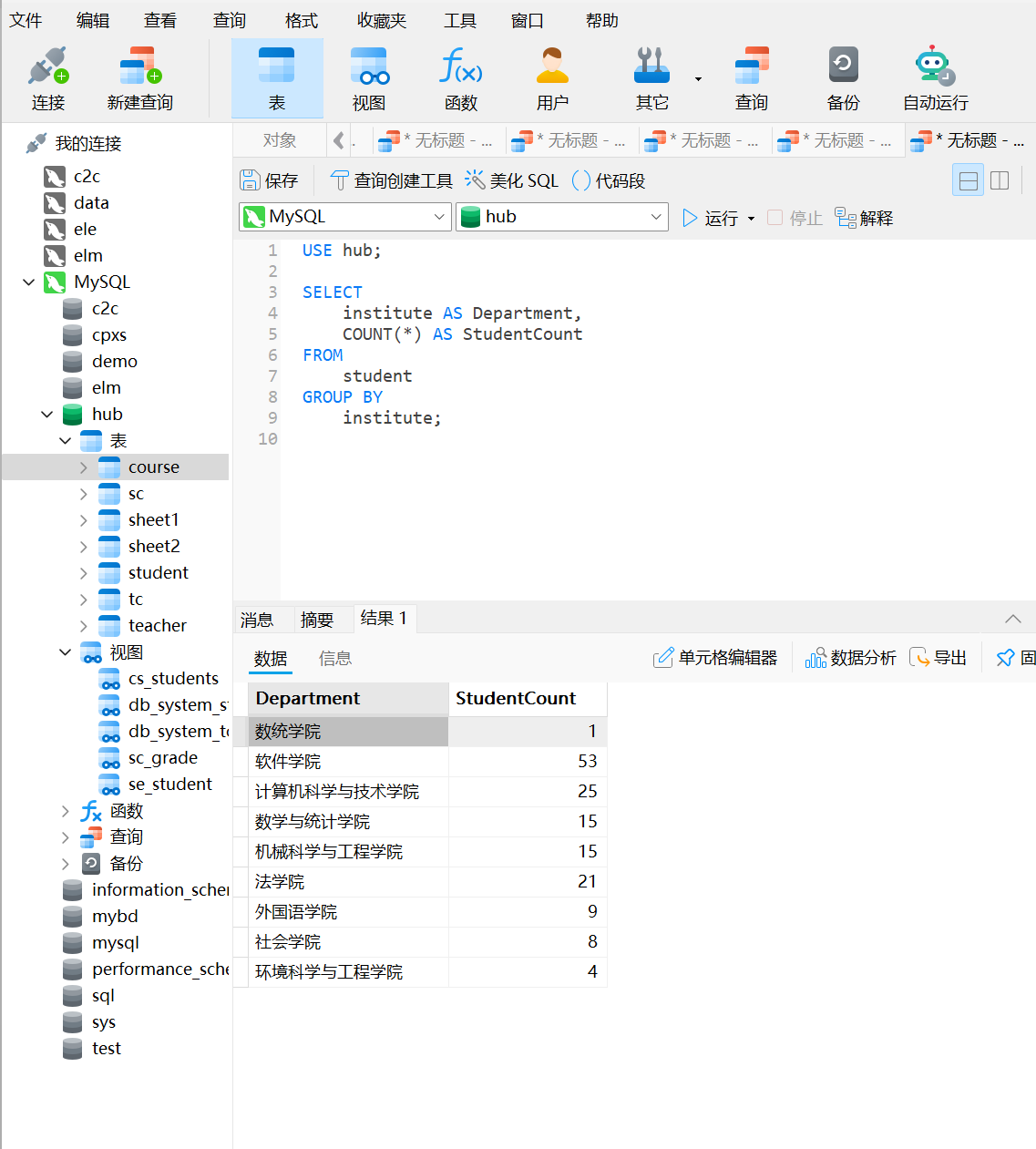
COUNT(\*) AS StudentCount

FROM

student

GROUP BY

institute;



1. 分组统计各学院平均的思想道德基础与法律修养课程的成绩。

USE hub;

SELECT

student.institute AS Institute,

AVG(sc\_grade.grade) AS AvgGrade

FROM

student

JOIN

sc\_grade

ON

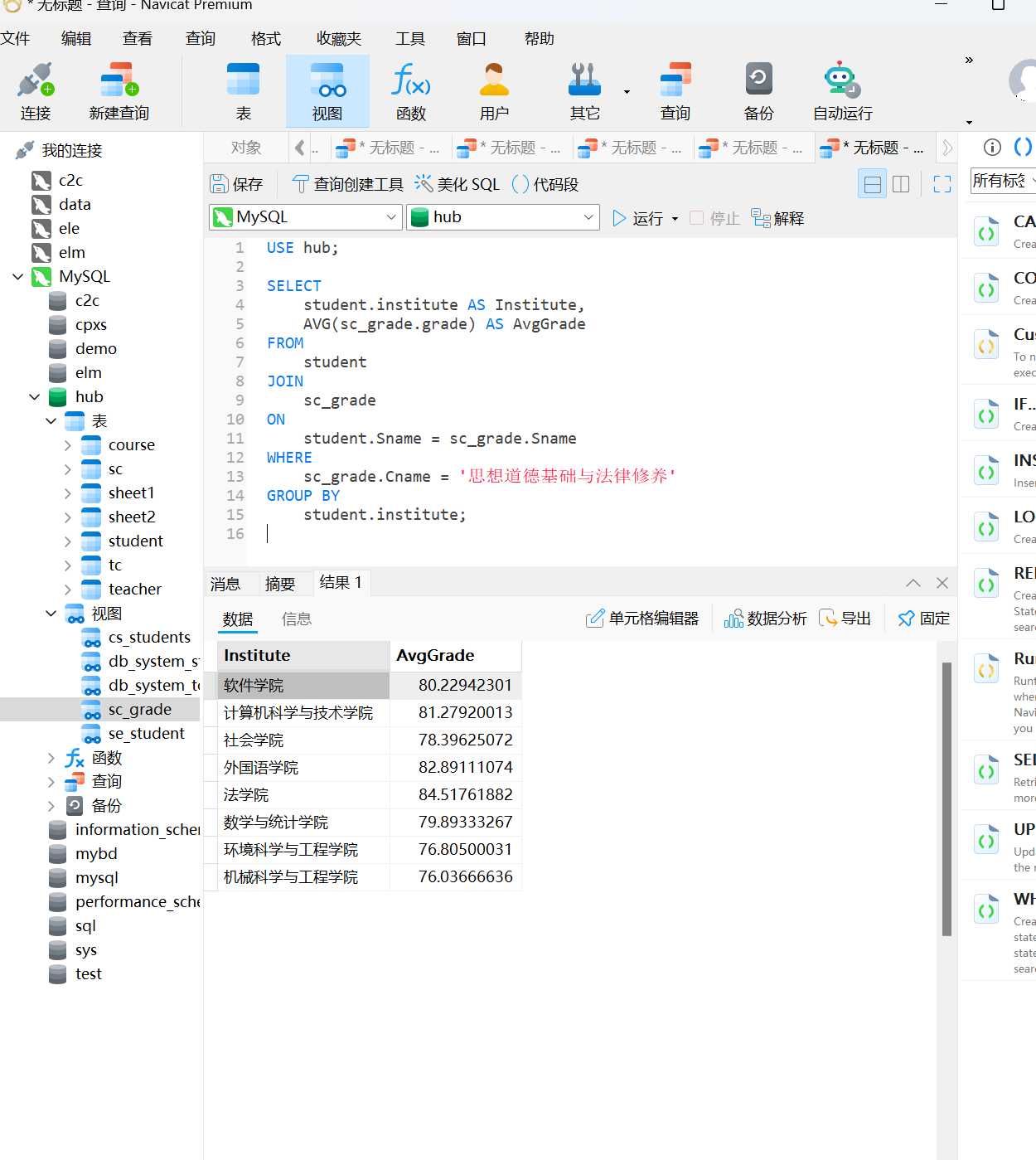
student.Sname = sc\_grade.Sname

WHERE

sc\_grade.Cname = '思想道德基础与法律修养'

GROUP BY

student.institute;



1. 查询有超过两条教授课程记录的教师的名字和所属院系。

USE hub;

SELECT

Tname AS TeacherName,

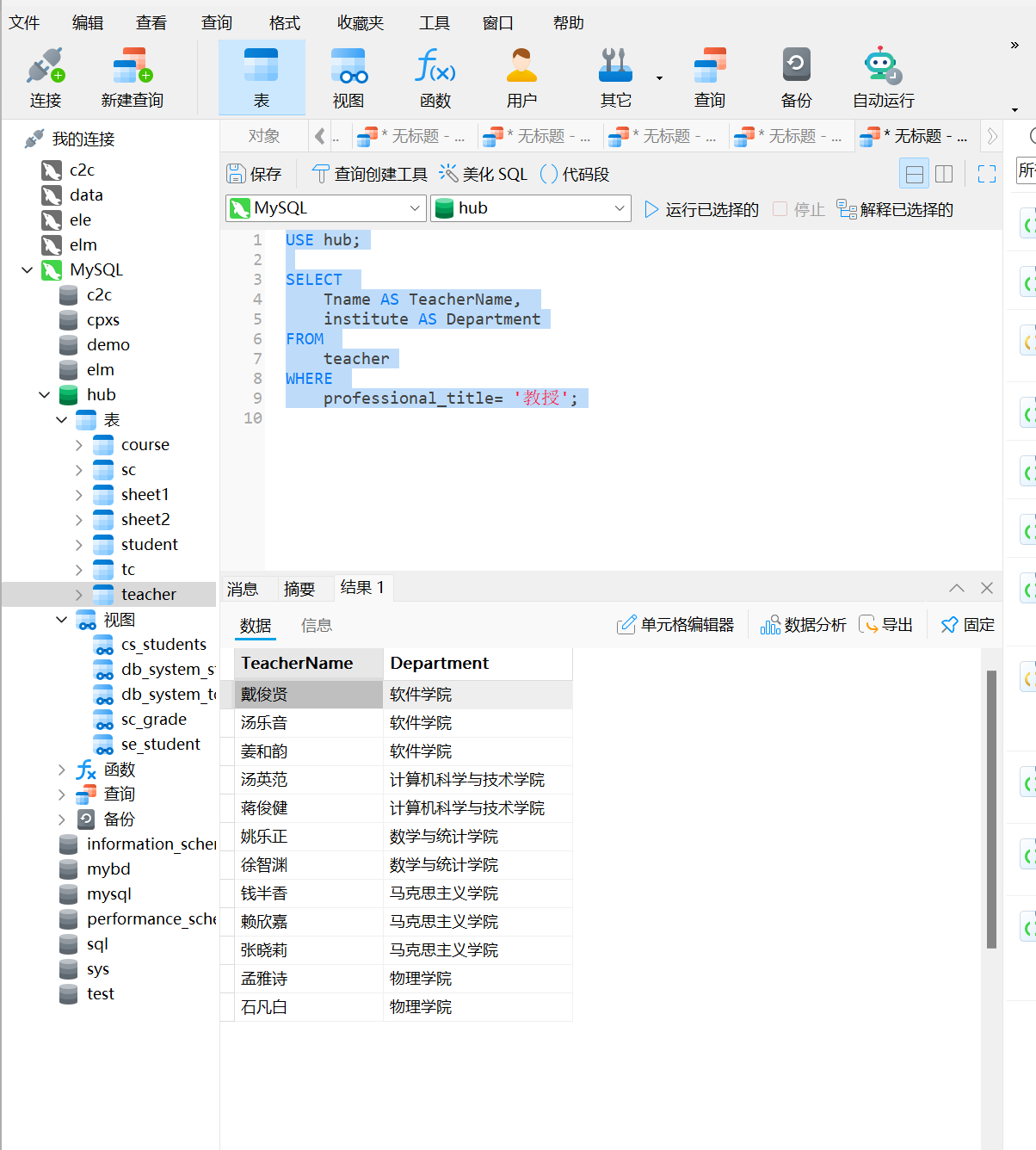
institute AS Department

FROM

teacher

WHERE

professional\_title= '教授';



1. 查询不属于软件学院的专业名称和所属院系。

USE hub;

SELECT

major AS MajorName,

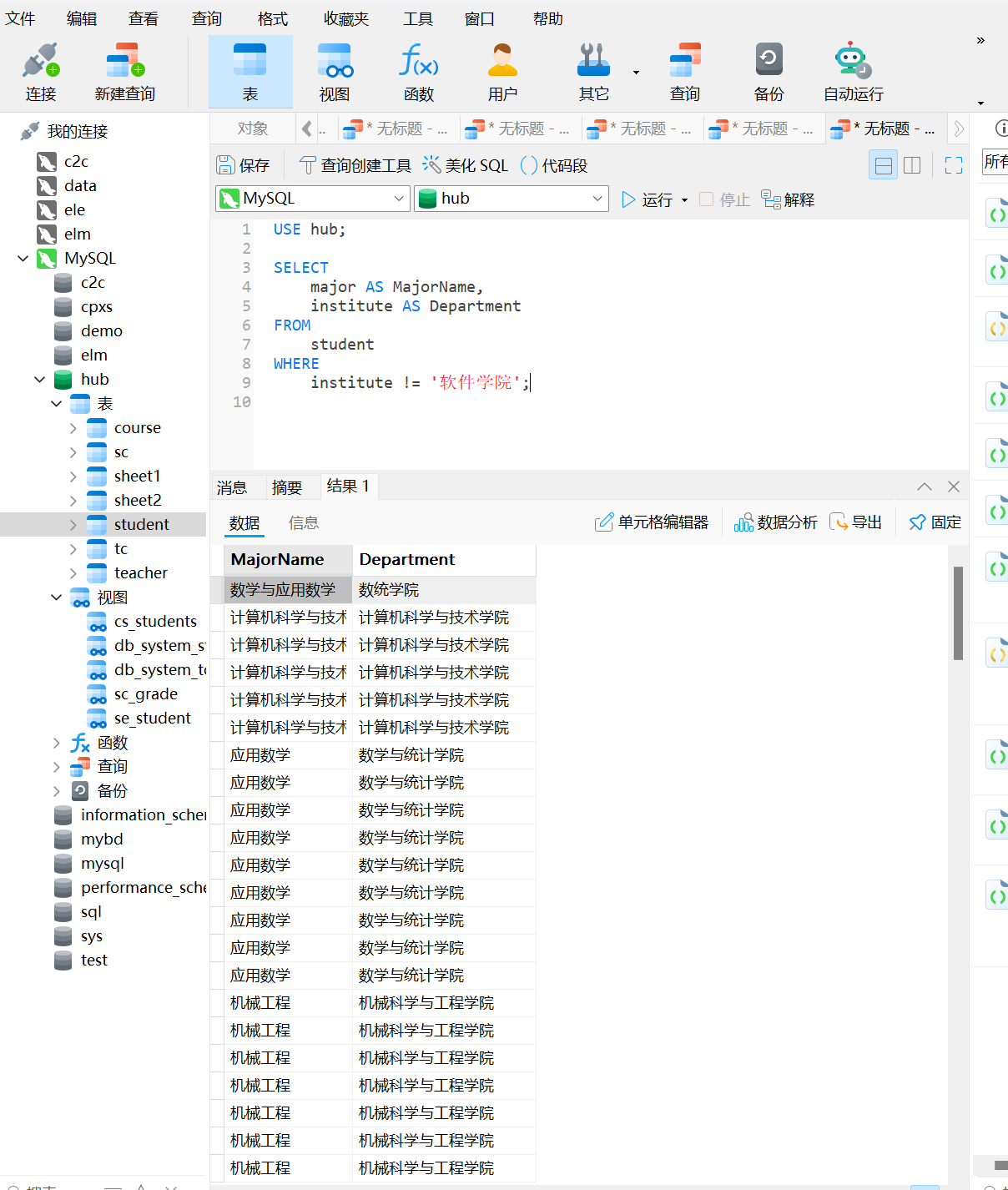
institute AS Department

FROM

student

WHERE

institute != '软件学院';



1. 查询选修课程成绩在80分以下的课程名称、学生名字。

USE hub;

SELECT

Cname AS CourseName,

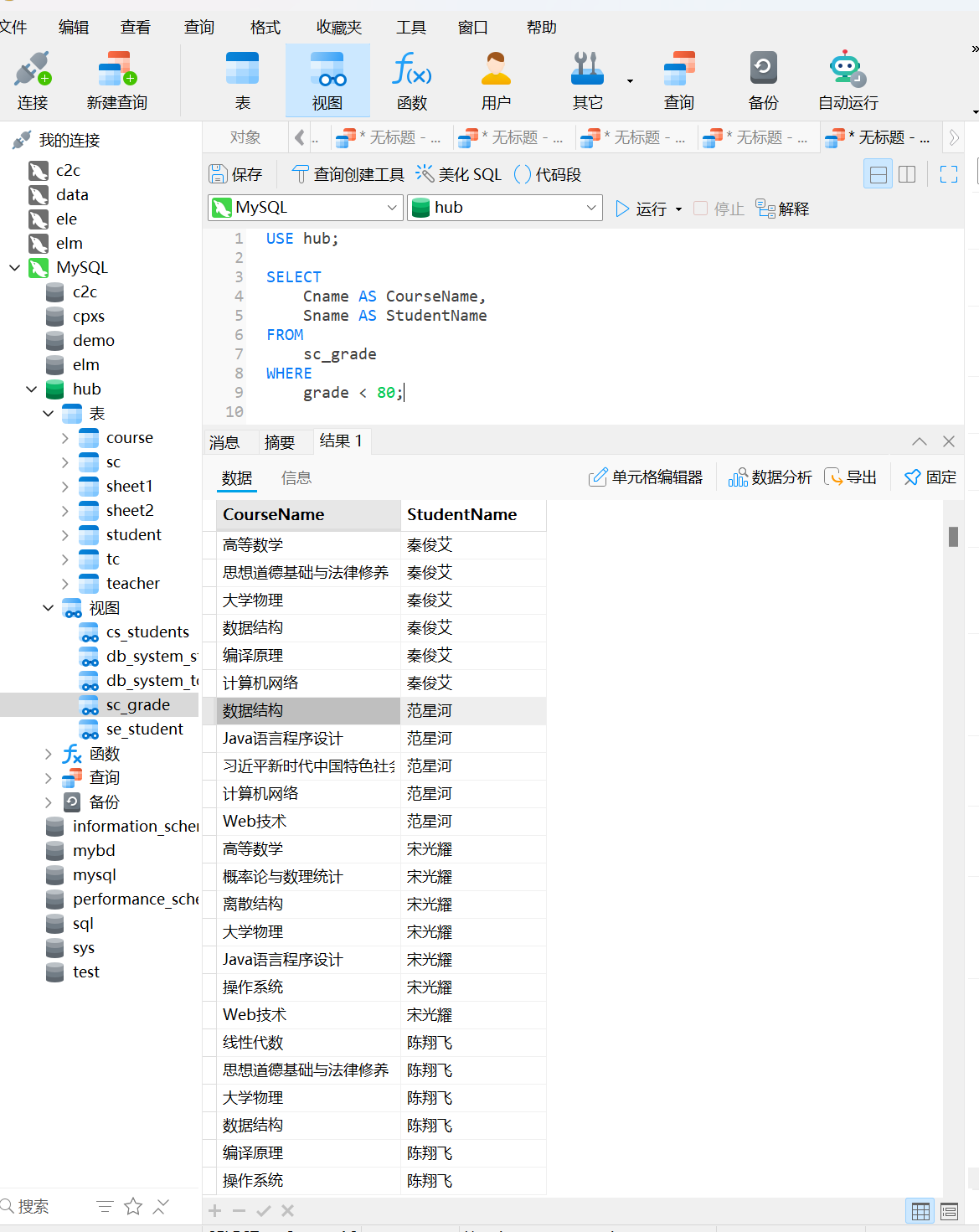
Sname AS StudentName

FROM

sc\_grade

WHERE

grade < 80;



1. 查询选修数据库系统原理成绩排名前20的学生名字和成绩。

USE hub;

SELECT

Sname AS StudentName,

grade AS Grade

FROM

sc\_grade

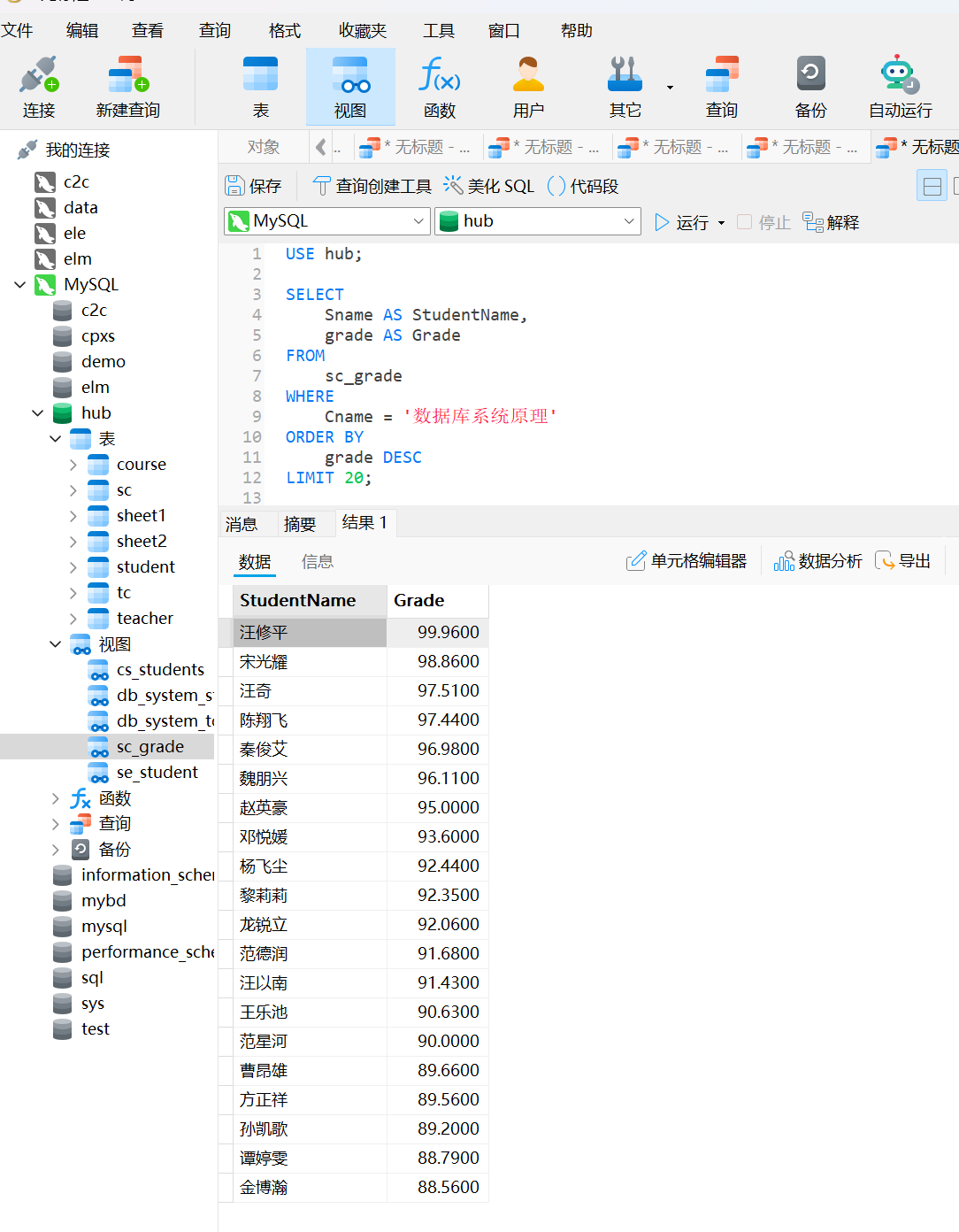
WHERE

Cname = '数据库系统原理'

ORDER BY

grade DESC

LIMIT 20;



1. 查询高等数学课程平均成绩低于软件学院的院系名称及其平均成绩，以平均数的降序排列。

USE hub;

WITH AvgScores AS (

-- 计算每个院系的《高等数学》课程平均成绩

SELECT

student.institute AS Institute,

AVG(sc\_grade.grade) AS AvgGrade

FROM

student

JOIN

sc\_grade

ON

student.Sname = sc\_grade.Sname

WHERE

sc\_grade.Cname = '高等数学'

GROUP BY

student.institute

),

SoftwareAvg AS (

-- 计算软件学院的《高等数学》平均成绩

SELECT

AVG(sc\_grade.grade) AS SoftwareAvgGrade

FROM

student

JOIN

sc\_grade

ON

student.Sname = sc\_grade.Sname

WHERE

student.institute = '软件学院'

AND sc\_grade.Cname = '高等数学'

)

-- 查询低于软件学院平均成绩的院系及其平均成绩

SELECT

AvgScores.Institute AS Institute,

AvgScores.AvgGrade AS AvgGrade

FROM

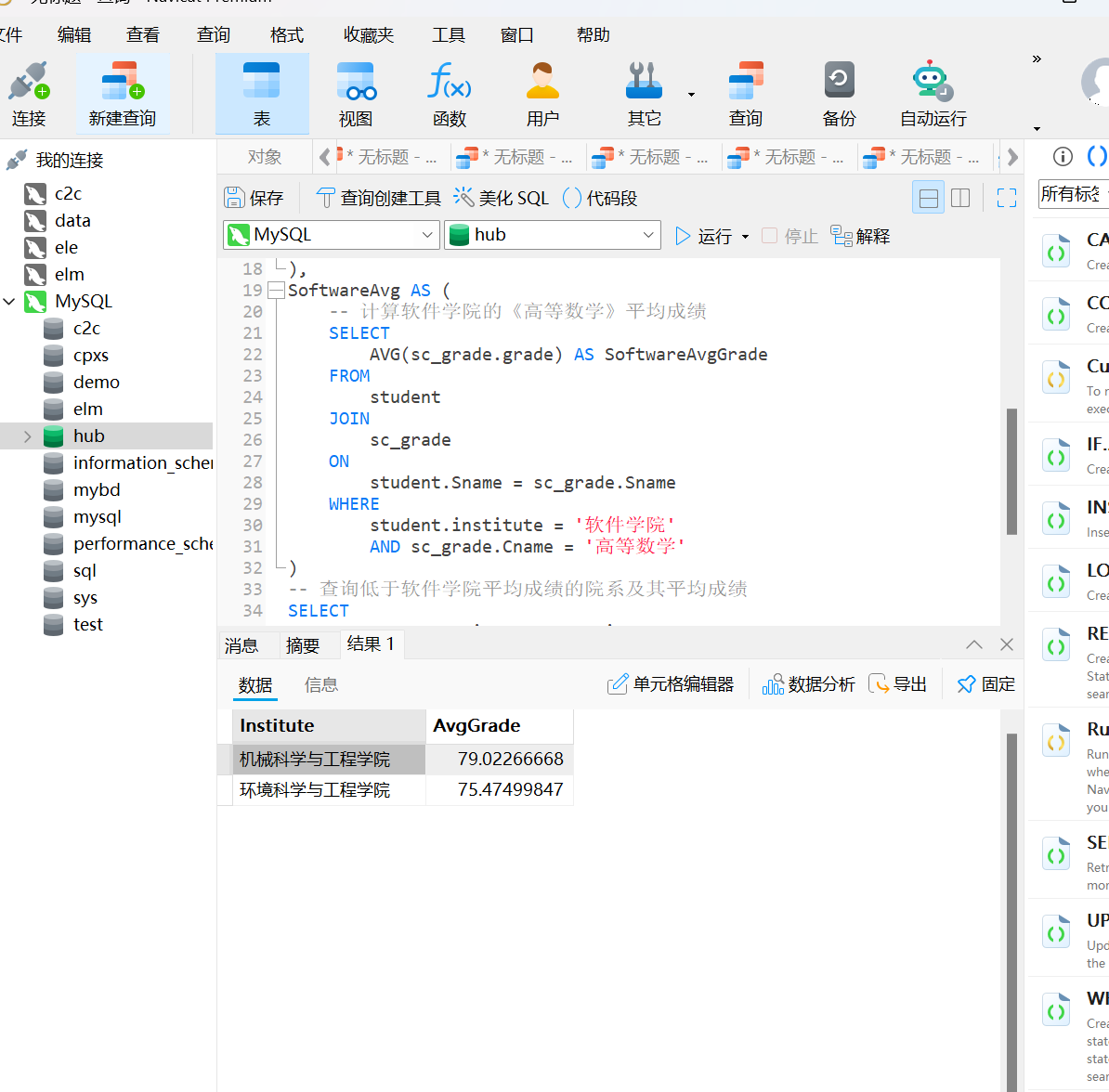
AvgScores, SoftwareAvg

WHERE

AvgScores.AvgGrade < SoftwareAvg.SoftwareAvgGrade

ORDER BY

AvgScores.AvgGrade DESC;



1. 查询sc表中多于3条成绩记录且名字以“李”开头的学生的名字和平均成绩，以平均成绩的升序排列。

USE hub;

SELECT

Sname AS StudentName,

AVG(grade) AS AvgGrade

FROM

sc\_grade

WHERE

Sname LIKE '李%'

GROUP BY

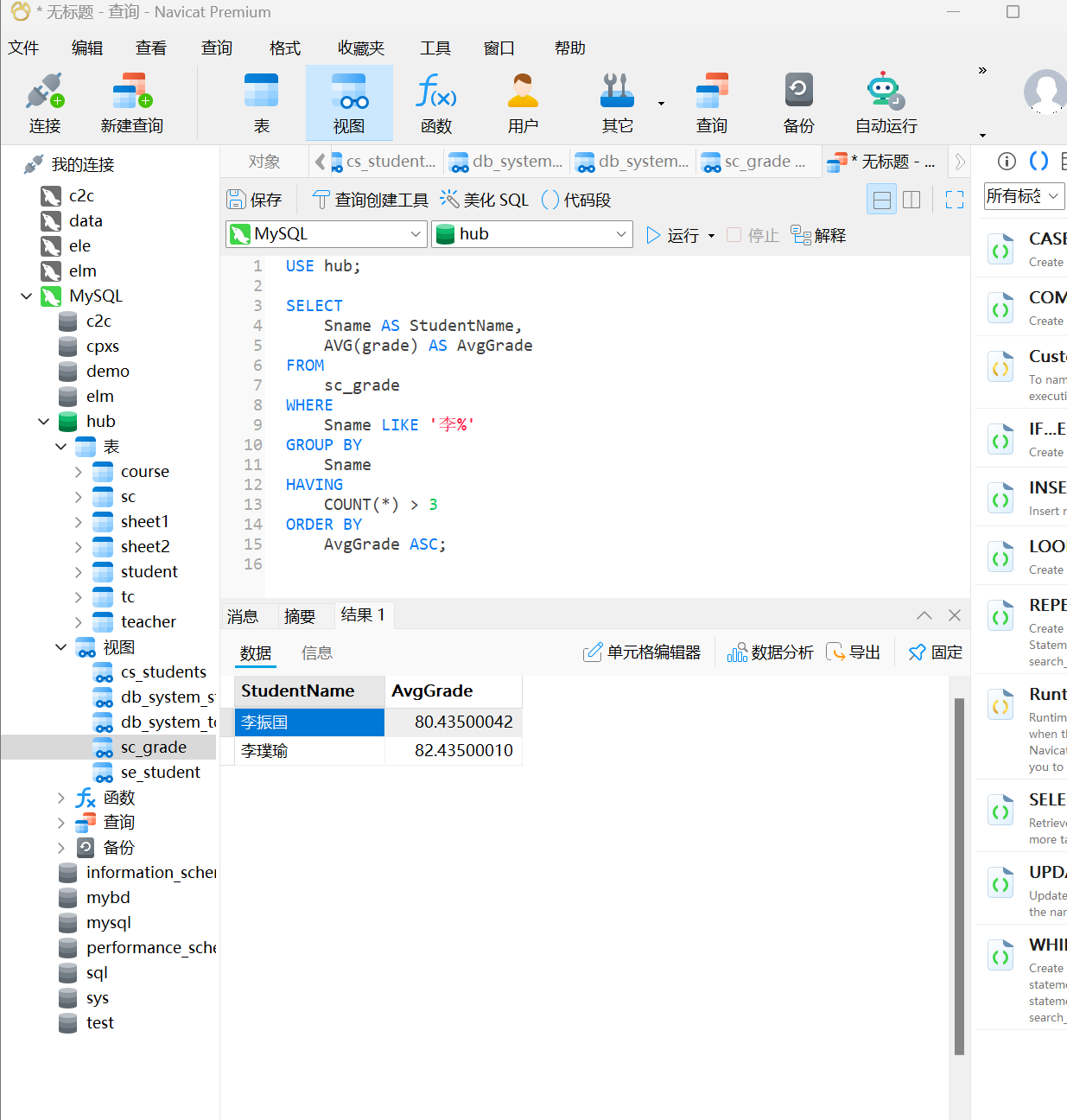
Sname

HAVING

COUNT(\*) > 3

ORDER BY

AvgGrade ASC;



**三、课后练习题**

1、建立名为SPJ的数据库。它包括S、P、J、SPJ 4个关系模式：

S(SNO,SNAME,STATUS,CITY);

P(PNO,PNAME,COLOR,WEIGHT);

J(JNO,JNAME,CITY);

SPJ(SNO,PNO,JNO,QTY)

供应商表S由供应商代码（SNO）、供应商姓名（SNAME）、供应商状态（STATUS）、供应商所在城市（CITY）组成；

零件表P由零件代码（PNO）、零件名(PNAME)、颜色(COLOR)、重量(WEIGHT)组成；

工程项目表J由工程项目代码(JNO)、工程项目名(JNAME)、工程项目所在城市(CITY)组成；

供应情况表SPJ由供应商代码(SNO)、零件代码(PNO)、工程项目代码(JNO)、供应数量(QTY)组成，标识某供应商 供应某种零件 给某工程项目的数量为QTY。

创建数据库：

CREATE DATABASE SPJ;

USE SPJ;

创建供应商表 S：

CREATE TABLE S (

SNO CHAR(5) PRIMARY KEY, -- 供应商代码

SNAME VARCHAR(50) NOT NULL, -- 供应商姓名

STATUS INT CHECK (STATUS >= 0), -- 供应商状态

CITY VARCHAR(50) -- 供应商所在城市

);

创建零件表 P：

CREATE TABLE P (

PNO CHAR(5) PRIMARY KEY, -- 零件代码

PNAME VARCHAR(50) NOT NULL, -- 零件名称

COLOR VARCHAR(20), -- 颜色

WEIGHT DECIMAL(10, 2) CHECK (WEIGHT > 0) -- 重量（大于 0）

);

创建工程项目表 J：

CREATE TABLE J (

JNO CHAR(5) PRIMARY KEY, -- 工程项目代码

JNAME VARCHAR(50) NOT NULL, -- 工程项目名称

CITY VARCHAR(50) -- 工程项目所在城市

);

创建供应情况表 SPJ：

CREATE TABLE SPJ (

SNO CHAR(5), -- 供应商代码

PNO CHAR(5), -- 零件代码

JNO CHAR(5), -- 工程项目代码

QTY INT CHECK (QTY >= 0), -- 供应数量（大于等于 0）

PRIMARY KEY (SNO, PNO, JNO), -- 主键为三者的组合

FOREIGN KEY (SNO) REFERENCES S(SNO), -- 外键约束

FOREIGN KEY (PNO) REFERENCES P(PNO),

FOREIGN KEY (JNO) REFERENCES J(JNO)

);

2、在数据库SPJ中，合理地为库中每一个数据表定义主键和外键。

3、适当为各数据表及其字段添加约束（唯一性、默认值等）

今有若干数据如下,利用navicat进行插入

S表

|  |  |  |  |
| --- | --- | --- | --- |
| SNO | SNAME | STATUS | CITY |
| S1 | 精益 | 20 | 天津 |
| S2 | 盛锡 | 10 | 北京 |
| S3 | 东方红 | 30 | 北京 |
| S4 | 丰泰盛 | 20 | 天津 |
| S5 | 为民 | 30 | 上海 |

P表

|  |  |  |  |
| --- | --- | --- | --- |
| PNO | PNAME | COLOR | WEIGHT |
| P1 | 螺母 | 红 | 12 |
| P2 | 螺栓 | 绿 | 17 |
| P3 | 螺丝刀 | 蓝 | 14 |
| P4 | 螺丝刀 | 红 | 14 |
| P5 | 凸轮 | 蓝 | 40 |
| P6 | 齿轮 | 红 | 30 |

J表

|  |  |  |
| --- | --- | --- |
| JNO | JNAME | CITY |
| J1 | 三建 | 北京 |
| J2 | 一汽 | 长春 |
| J3 | 弹簧厂 | 天津 |
| J4 | 造船厂 | 天津 |
| J5 | 机车厂 | 唐山 |
| J6 | 无线电厂 | 常州 |
| J7 | 半导体厂 | 南京 |

SPJ表

|  |  |  |  |
| --- | --- | --- | --- |
| SNO | PNO | JNO | QTY |
| S1 | P1 | J1 | 200 |
| S1 | P1 | J3 | 100 |
| S1 | P1 | J4 | 700 |
| S1 | P2 | J2 | 100 |
| S2 | P3 | J1 | 400 |
| S2 | P3 | J2 | 200 |
| S2 | P3 | J4 | 500 |
| S2 | P3 | J5 | 400 |
| S2 | P5 | J1 | 400 |
| S2 | P5 | J2 | 100 |
| S3 | P1 | J1 | 200 |
| S3 | P3 | J1 | 200 |
| S4 | P5 | J1 | 100 |
| S4 | P6 | J3 | 300 |
| S4 | P6 | J4 | 200 |
| S5 | P2 | J4 | 100 |
| S5 | P3 | J1 | 200 |
| S5 | P6 | J2 | 200 |
| S5 | P6 | J4 | 500 |

或利用SQL语句插入，执行如下INSERT语句

INSERT INTO S (SNO, SNAME, STATUS, CITY) VALUES

('S1', '精益', 20, '天津'),

('S2', '盛锡', 10, '北京'),

('S3', '东方红', 30, '北京'),

('S4', '丰泰盛', 20, '天津'),

('S5', '为民', 30, '上海');

INSERT INTO P (PNO, PNAME, COLOR, WEIGHT) VALUES

('P1', '螺母', '红', 12),

('P2', '螺栓', '绿', 17),

('P3', '螺丝刀', '蓝', 14),

('P4', '螺丝刀', '红', 14),

('P5', '凸轮', '蓝', 40),

('P6', '齿轮', '红', 30);

INSERT INTO J (JNO, JNAME, CITY) VALUES

('J1', '三建', '北京'),

('J2', '一汽', '长春'),

('J3', '弹簧厂', '天津'),

('J4', '造船厂', '天津'),

('J5', '机车厂', '唐山'),

('J6', '无线电厂', '常州'),

('J7', '半导体厂', '南京');

INSERT INTO SPJ (SNO, PNO, JNO, QTY) VALUES

('S1', 'P1', 'J1', 200),

('S1', 'P1', 'J3', 100),

('S1', 'P1', 'J4', 700),

('S1', 'P2', 'J2', 100),

('S2', 'P3', 'J1', 400),

('S2', 'P3', 'J2', 200),

('S2', 'P3', 'J4', 500),

('S2', 'P3', 'J5', 400),

('S2', 'P5', 'J1', 400),

('S2', 'P5', 'J2', 100),

('S3', 'P1', 'J1', 200),

('S3', 'P3', 'J1', 200),

('S4', 'P5', 'J1', 100),

('S4', 'P6', 'J3', 300),

('S4', 'P6', 'J4', 200),

('S5', 'P2', 'J4', 100),

('S5', 'P3', 'J1', 200),

('S5', 'P6', 'J2', 200),

('S5', 'P6', 'J4', 500);

第4～10题在数据库SPJ中完成。

1. 查询重量最轻的零件的零件代码。

USE SPJ;

SELECT

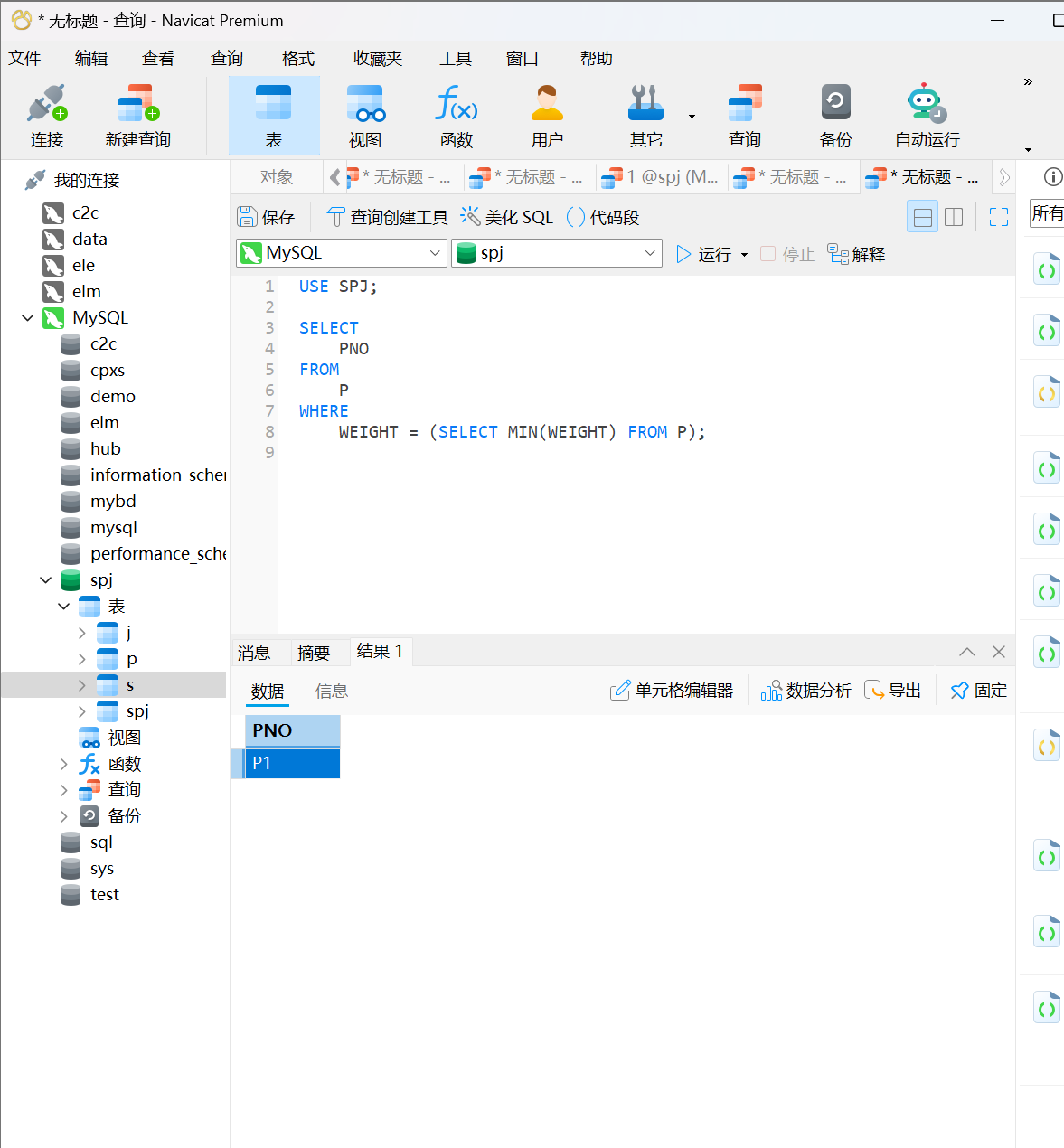
PNO

FROM

P

WHERE

WEIGHT = (SELECT MIN(WEIGHT) FROM P);



1. 查询由供应商S1提供零件的工程项目名

USE SPJ;

SELECT

J.JNAME AS ProjectName

FROM

SPJ

JOIN

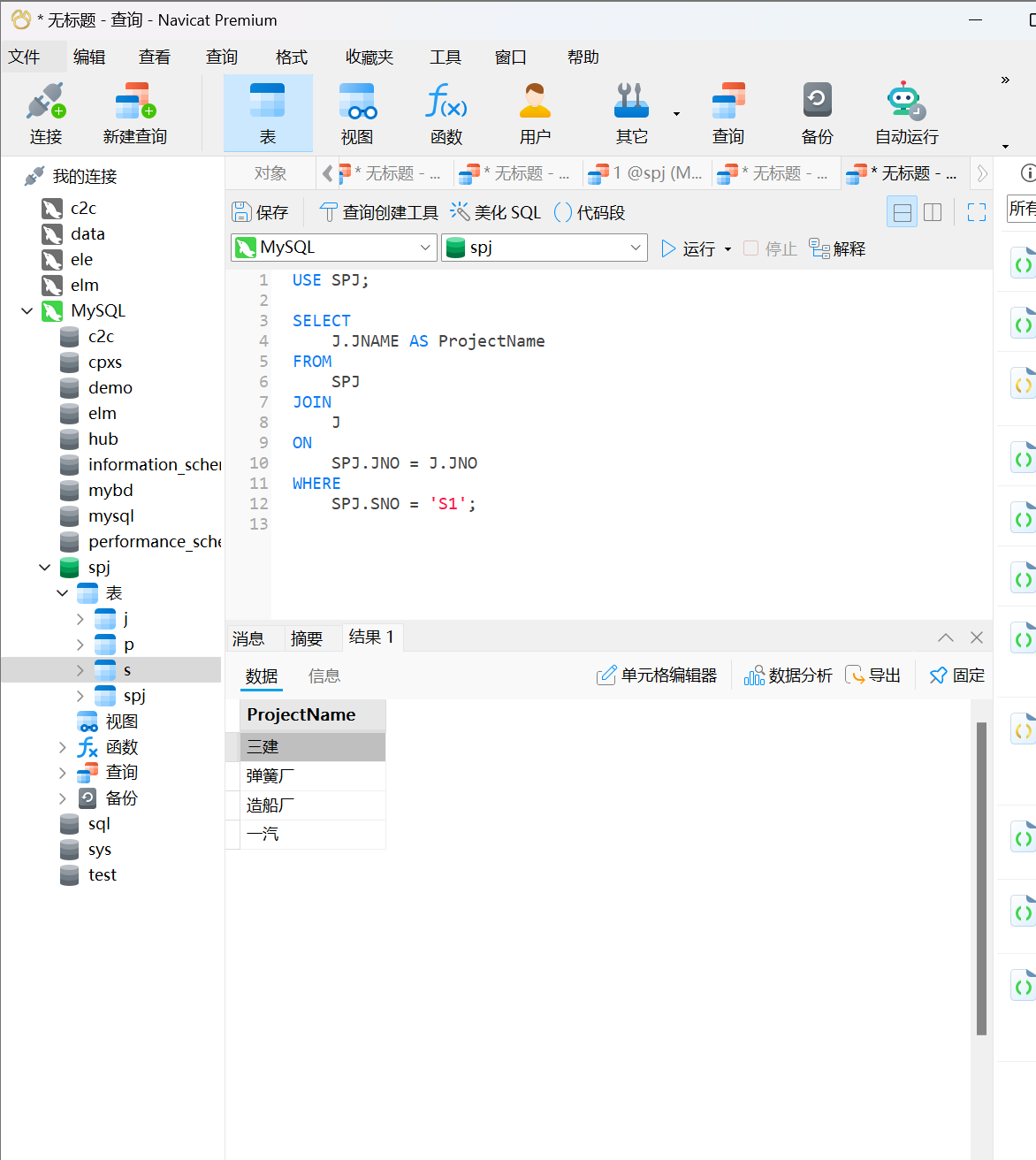
J

ON

SPJ.JNO = J.JNO

WHERE

SPJ.SNO = 'S1';



6、查询同时为工程J1和J2提供零件的供应商代码。

USE SPJ;

SELECT

SNO

FROM

SPJ

WHERE

JNO = 'J1'

INTERSECT

SELECT

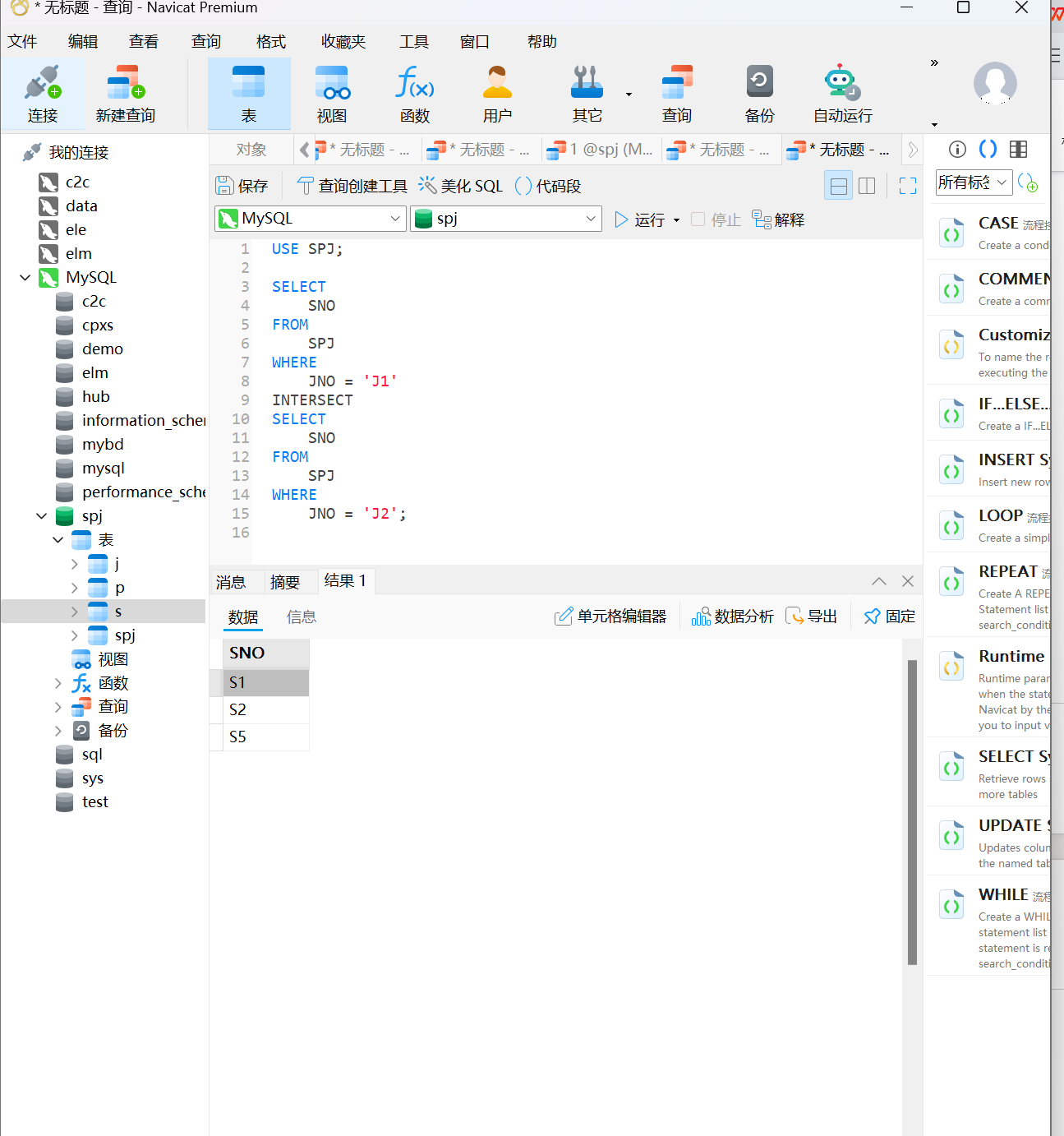
SNO

FROM

SPJ

WHERE

JNO = 'J2';



7、查询为位于天津的工程提供零件的供应商代码。

USE SPJ;

SELECT DISTINCT

SPJ.SNO AS SupplierCode

FROM

SPJ

JOIN

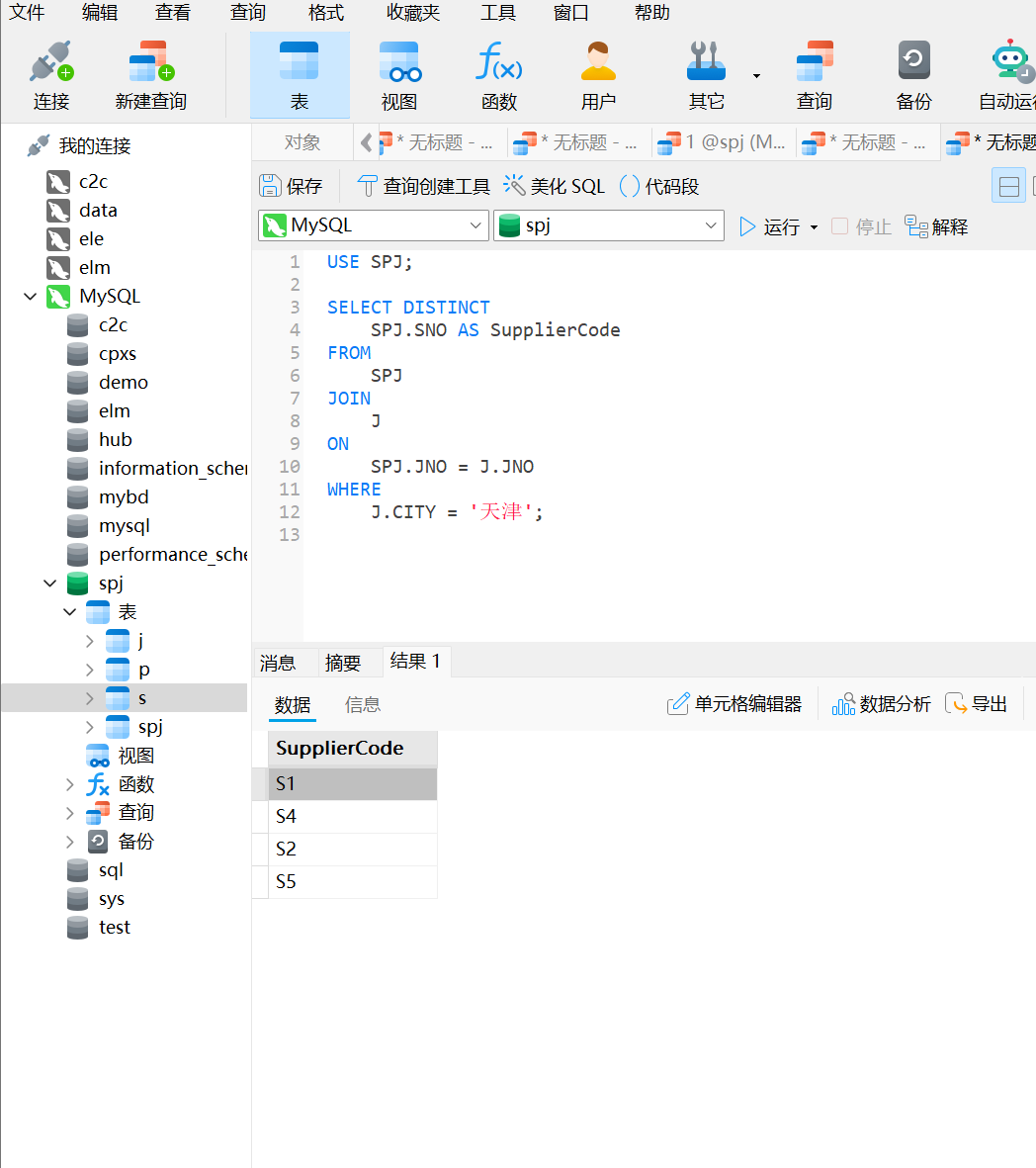
J

ON

SPJ.JNO = J.JNO

WHERE

J.CITY = '天津';



1. 查询同时为位于天津或北京的工程提供红色零件的供应商代码。

USE SPJ;

SELECT DISTINCT

SPJ.SNO AS SupplierCode

FROM

SPJ

JOIN

J

ON

SPJ.JNO = J.JNO

JOIN

P

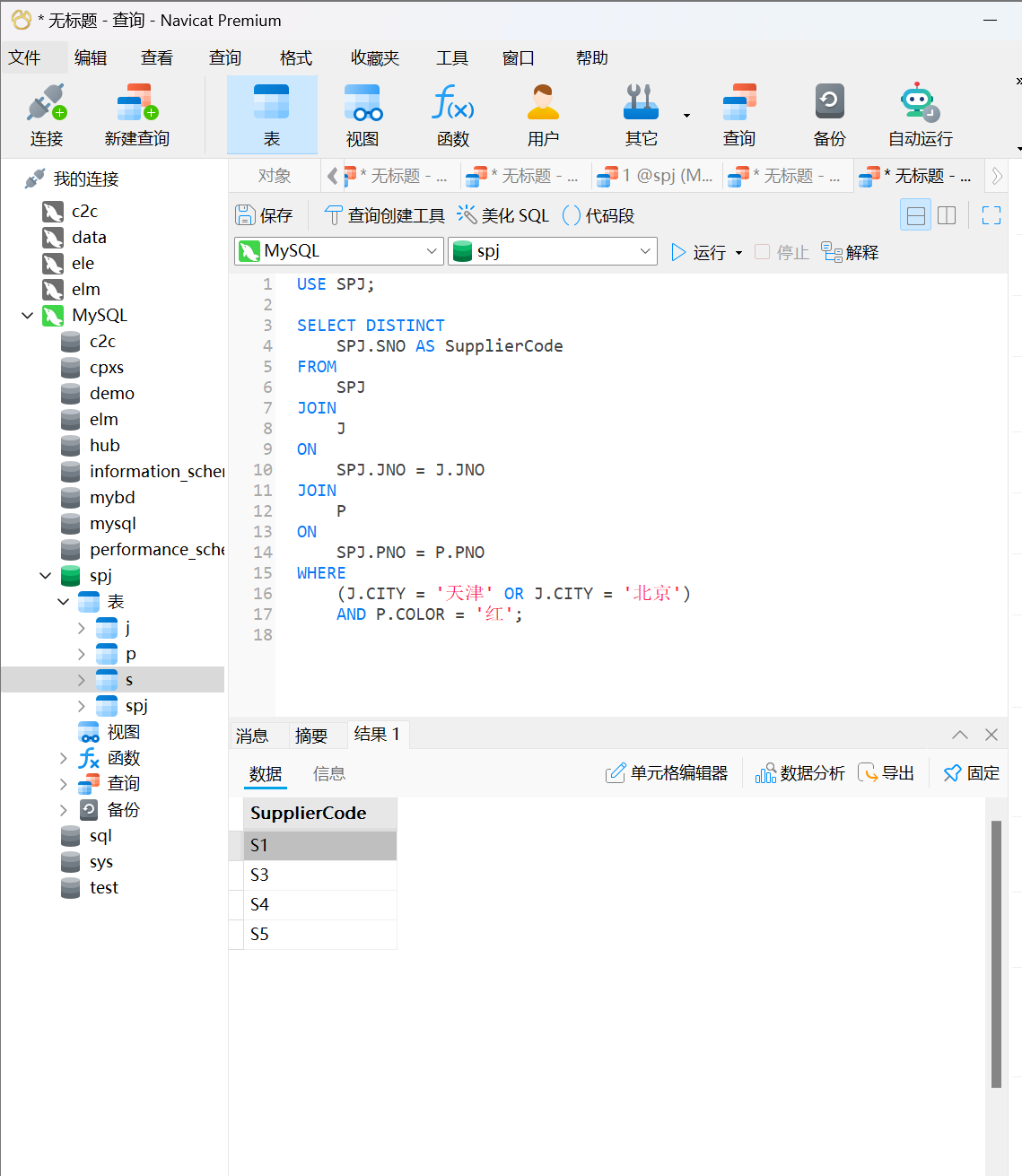
ON

SPJ.PNO = P.PNO

WHERE

(J.CITY = '天津' OR J.CITY = '北京')

AND P.COLOR = '红';



9、查询供应商和工程所在城市相同的供应商能提供的零件代码。

USE SPJ;

SELECT DISTINCT

SPJ.PNO AS PartCode

FROM

SPJ

JOIN

S

ON

SPJ.SNO = S.SNO

JOIN

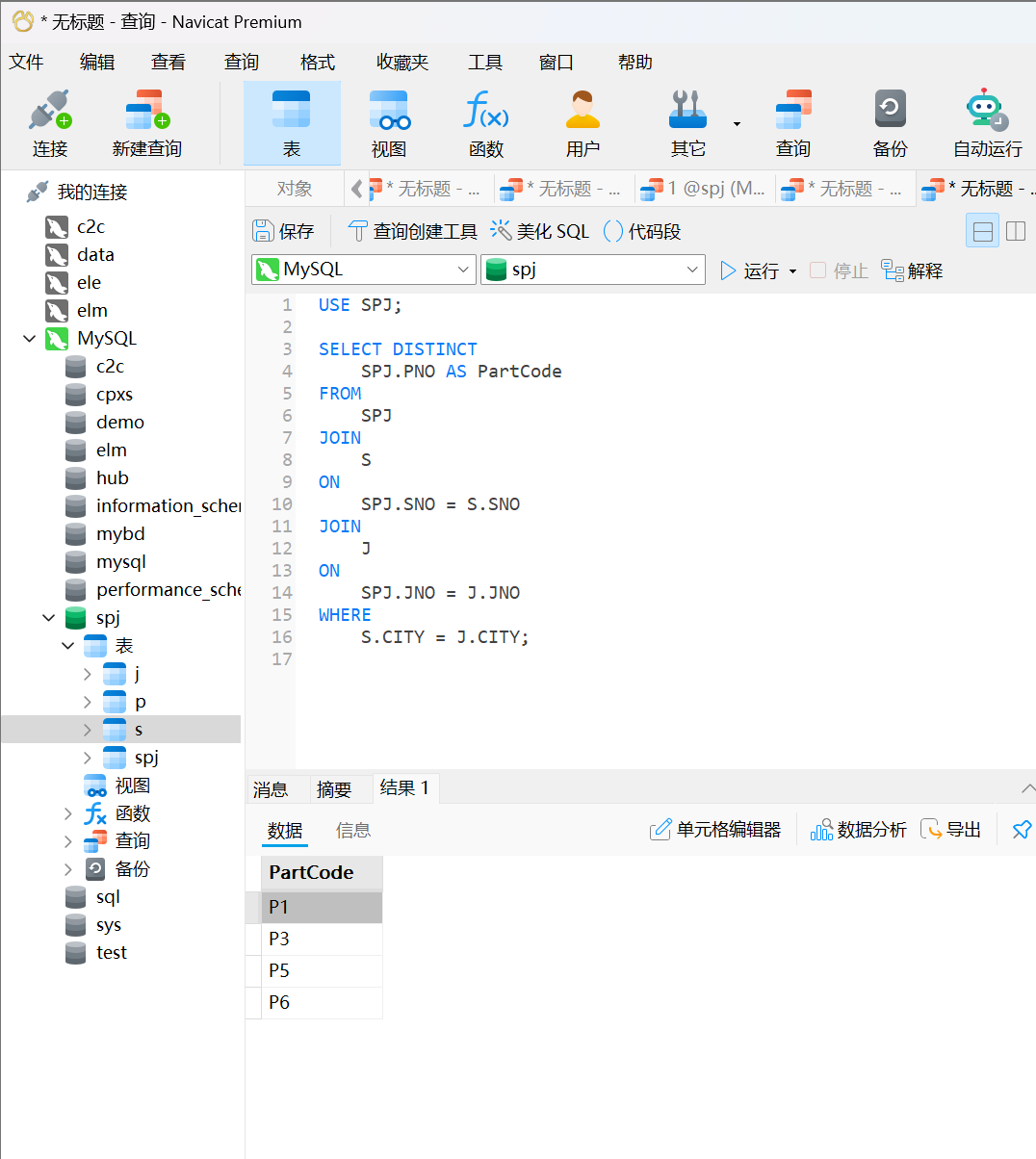
J

ON

SPJ.JNO = J.JNO

WHERE

S.CITY = J.CITY;



10.查询上海供应商不提供任何零件的工程代码。

USE SPJ;

SELECT DISTINCT

J.JNO AS ProjectCode

FROM

J

WHERE

J.JNO NOT IN (

SELECT DISTINCT

SPJ.JNO

FROM

SPJ

JOIN

S

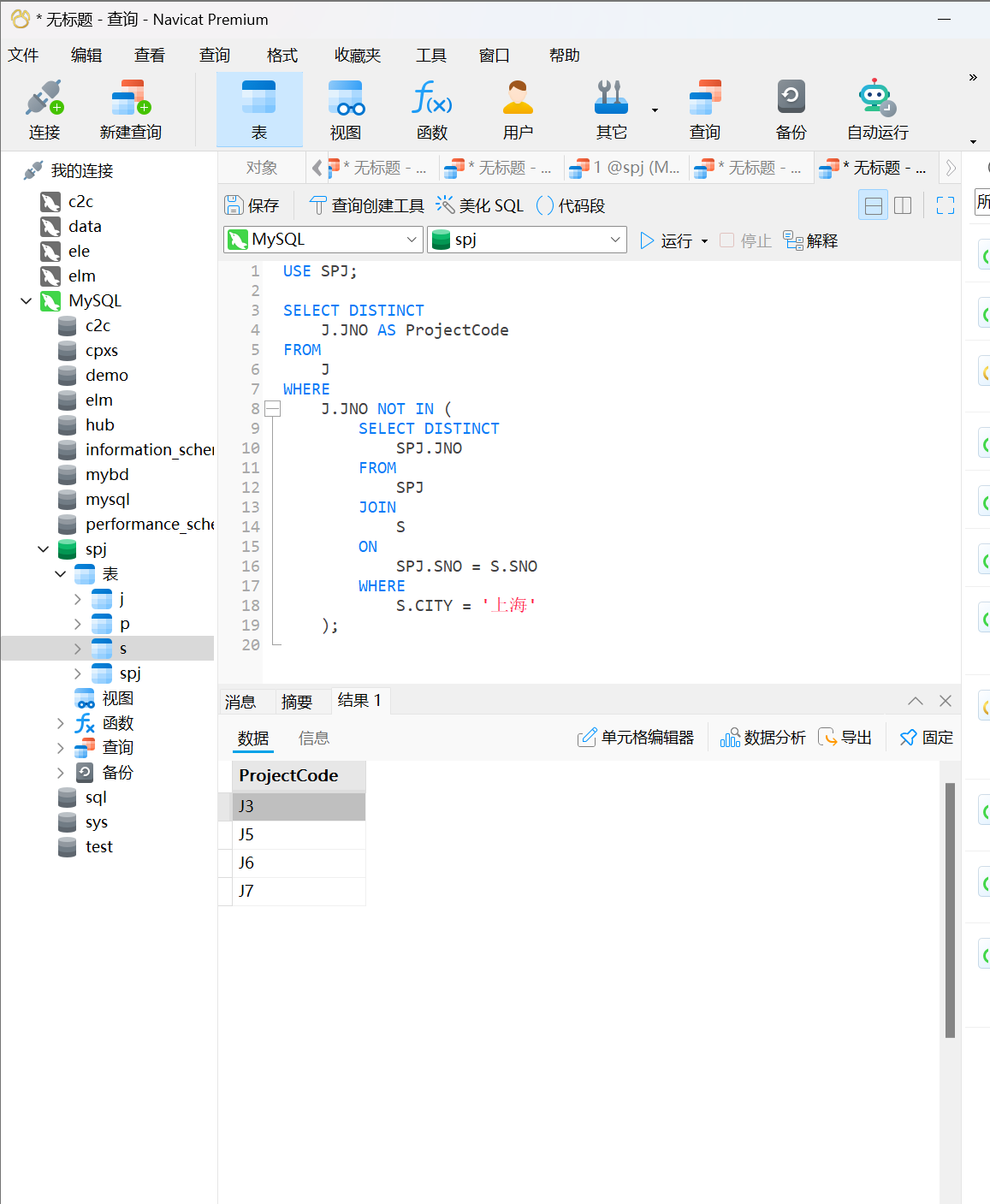
ON

SPJ.SNO = S.SNO

WHERE

S.CITY = '上海'

);



**实验总结：**

· **MySQL查询分析器的熟练使用**  
通过本次实验，熟悉了 MySQL 查询分析器的基本使用，包括编写、运行和调试 SQL 查询语句。掌握了在实际场景中高效分析和解决数据问题的方法。

· **SQL查询语句的深入理解**  
在实验中，通过对简单和复杂查询的实践操作，进一步加深了对 SQL 查询语句的理解，尤其是 WHERE、GROUP BY、HAVING、ORDER BY 等关键字的功能和应用。

· **数据分组和统计操作的掌握**  
实验练习了分组统计、聚合函数（如 SUM、AVG、COUNT）、以及结合条件的分组计算，强化了对数据分组统计的理解与实践能力。

· **集合操作的实践**  
通过集合操作（如 UNION、INTERSECT 和 NOT IN 等）的应用，掌握了在多表查询和复杂逻辑条件下的解决方案，提升了处理多维数据的能力。

· **实验收获**  
本次实验不仅提高了对 SQL 查询语句的灵活运用能力，还增强了数据分析和逻辑处理的能力，为后续的数据库课程和实践打下了坚实的基础。通过动手操作，更深刻体会到数据库查询在实际应用中的重要性和高效性。

**教师评语及成绩**：