**实验1.1 模式定义与数据完整性操作:**

1. 使用Management Studio图形界面创建studentdb数据库

在左侧的对象资源管理器中“数据库”上点右键，新建数据库，指定数据库名studentdb和各项参数如路径等，确定。

使用SQL语句创建spjdb。

在上方工具条中点“新建查询”，新打开窗口中输入如下SQL语句：

create database spjdb545

然后点“执行”按钮。

2. 在studentdb数据库中**新建查询，使用SQL语句**创建教材第3章中的个表：student、course、sc，同时完成数据完整性的定义（实体完整性、参照完整性和用户定义的域完整性）。注意创建完成后需要刷新当前数据库的表目录。

create table student545

( sno char(9) primary key,

sname char(10) not null,

ssex char(2),

sage smallint constraint ck\_sage545 check(sage>12),

sdept char(15)

);

create table course545

( cno char(4) primary key,

cname char(20),

cpno char(4),

ccredit smallint

);

create table sc545

( sno char(9) not null,

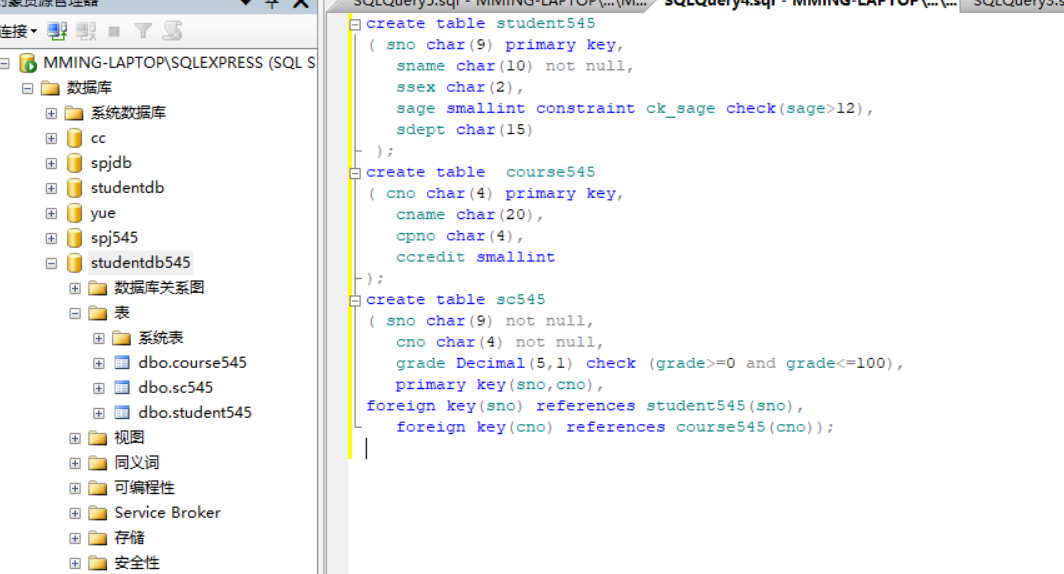
cno char(4) not null,

grade Decimal(5,1) check (grade>=0 and grade<=100),

primary key(sno,cno),

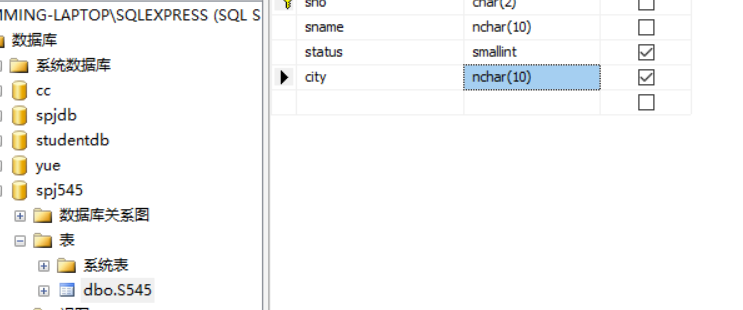
foreign key(sno) references student545 (sno),

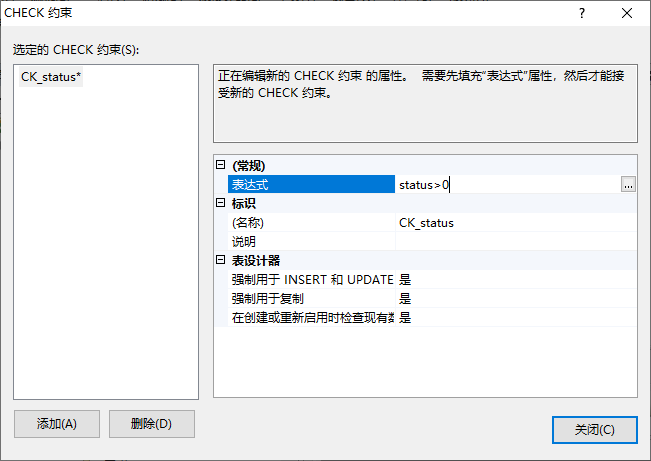
foreign key(cno) references course545 (cno));



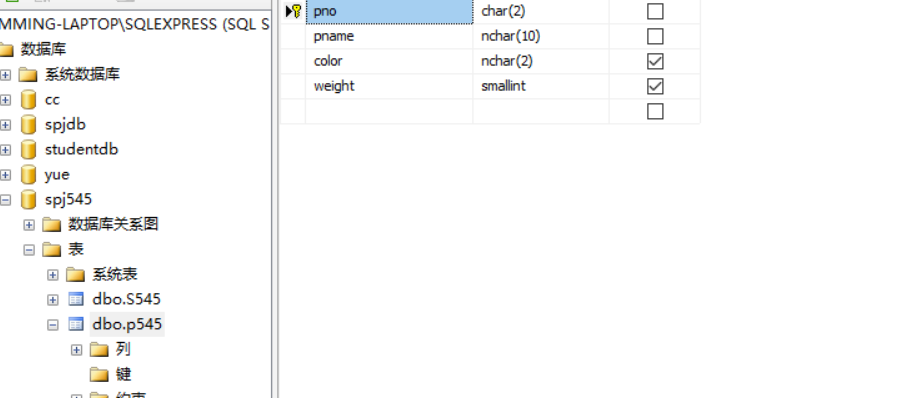
3. 在spjdb数据库中**使用图形界面**创建实验内容中的4个表，同时完成数据完整性的定义（实体完整性、参照完整性和用户定义的域完整性）：

创建表S：右击spjdb数据库中的表容器，选择“新建表”，输入S表的属性等信息，结果如下：

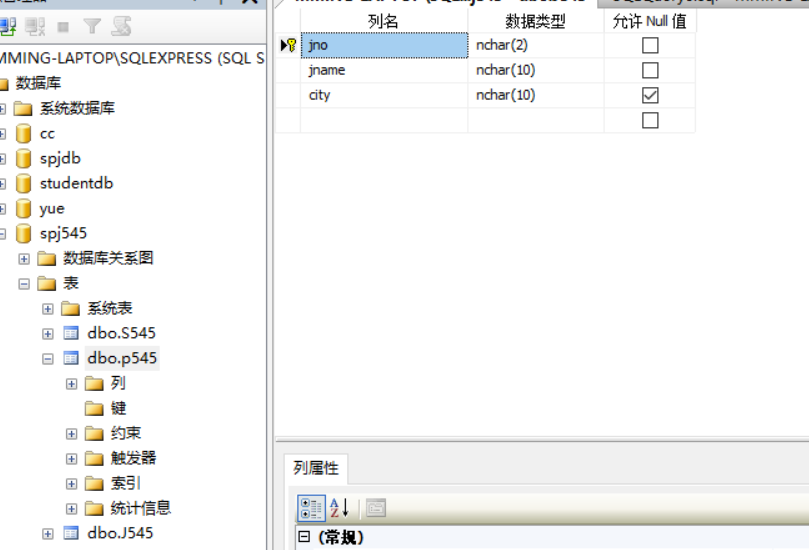




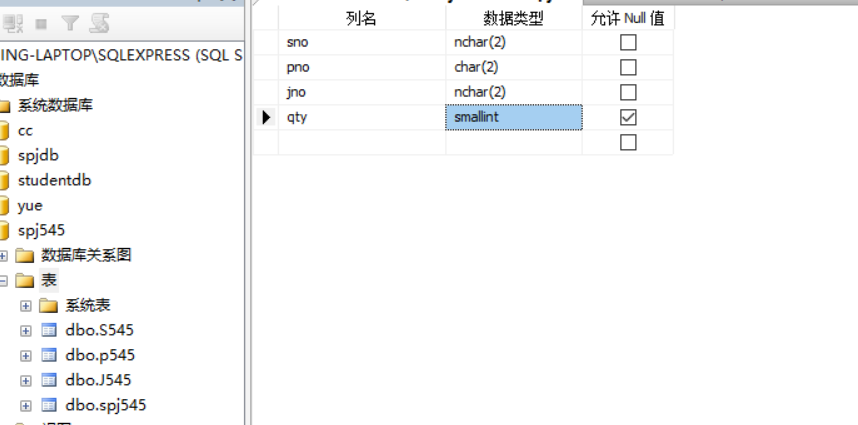
创建表P：右击spjdb数据库中的表容器，选择“新建表”，输入P表的属性、主码及约束等信息，结果如下：



创建表J：右击spjdb数据库中的表容器，选择“新建表”，输入J表的属性、主码等信息，结果如下：

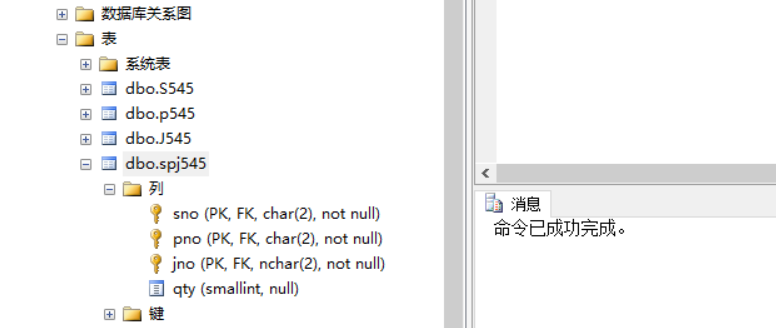


创建表SPJ：右击spjdb数据库中的表容器，选择“新建表”，输入SPJ表的属性信息，并设置主码、外码和约束等信息，操作如下：



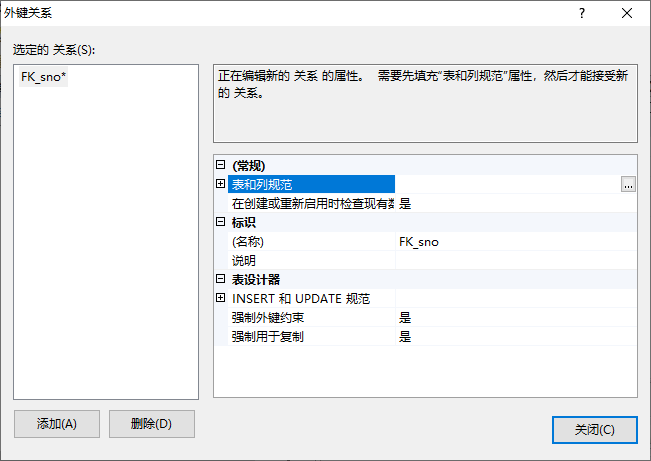
设置主码：（sno,pno,jno）

操作：在右侧输入窗口中，同时选定3个属性sno、pno、jno，单击“设置主键”按钮或“表设计器”菜单中选择“设置主键”。结果如下：

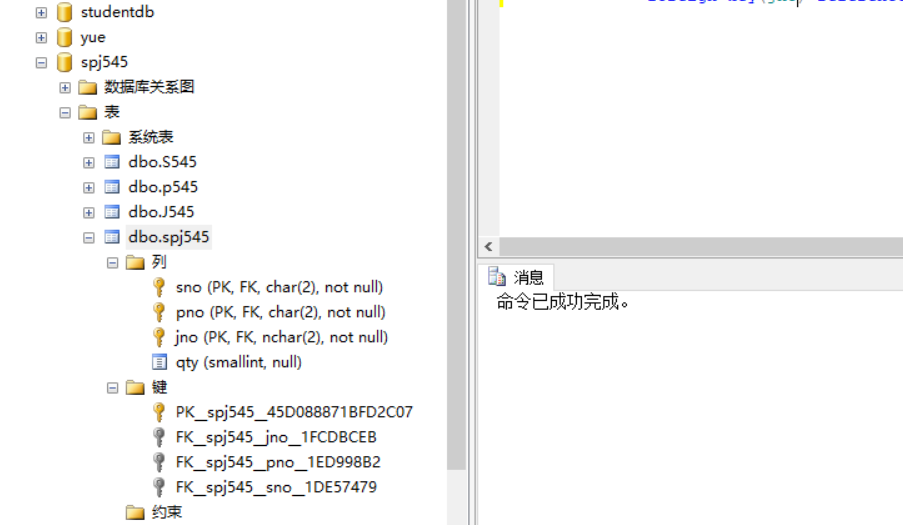


添加外码：sno、pno、jno

操作：在左侧窗口中选定单个属性，如sno。右键菜单中选择“关系”，添加外键FK\_sno，然后，在“表和列规范”选项中设置主键表S及属性sno，外键表SPJ及属性sno。

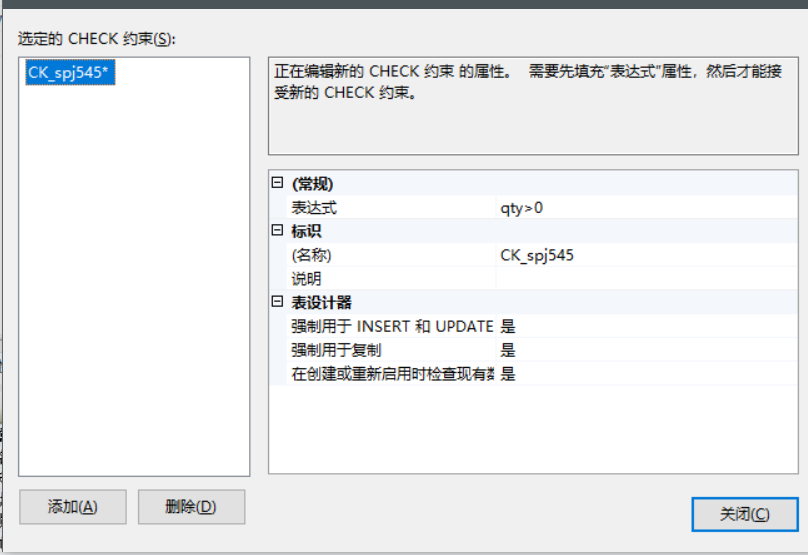


类似添加外码FK\_pno、FK\_jno，必须保存设置（存盘），结果如下：



添加检查约束：CK\_qty大于0

操作：在右侧窗口中右键属性qty，选择“CHECK约束”，添加约束CK\_qty，设置表达式：qty>0，保存设置。最后的结果如下：



4.create schema Production;

create login st with password='suntao123';

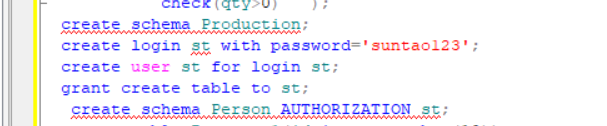
create user st for login st;

grant create table to st;

create schema Person AUTHORIZATION st;

create table Person.t1(id int,name char(10))

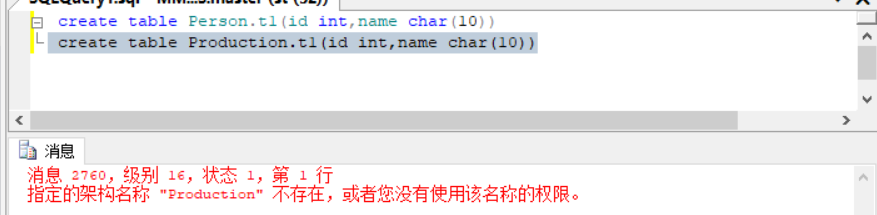
**在DBA的权限下执行下列语句**

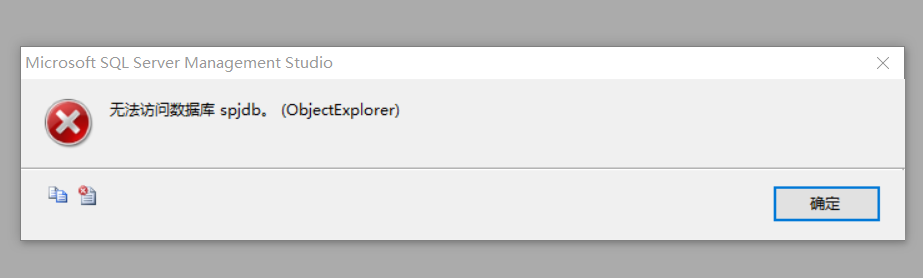


执行完之后切换为st登录



但执行下一条语句时失败

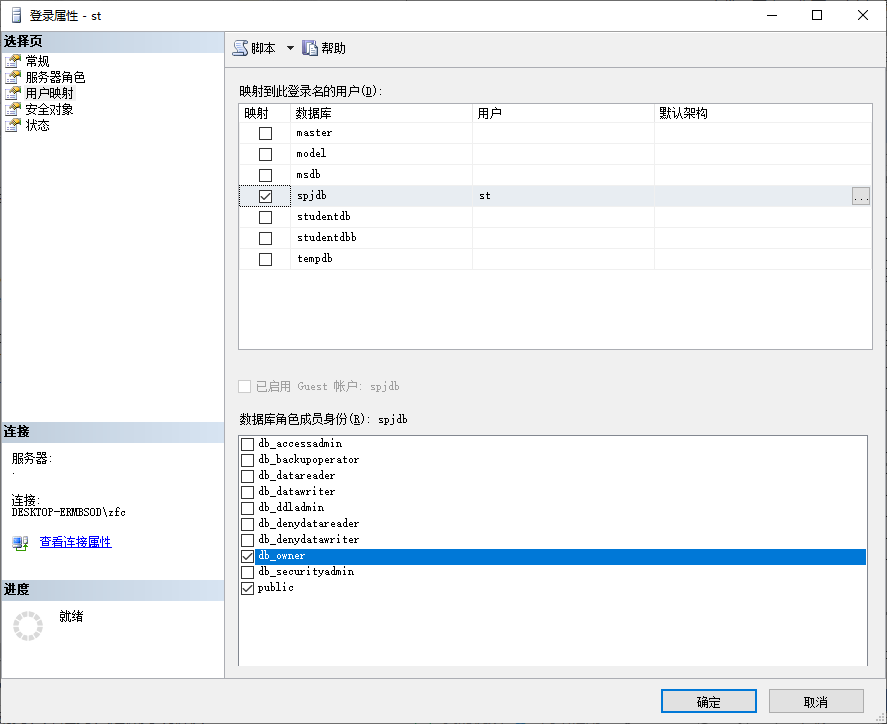




因为没有更改的权限，因为我们在创建用户时只为st创建了person架构。

**权限更改：**

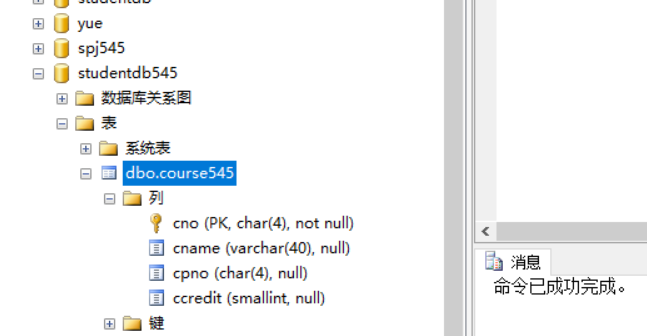
选择windows账号登陆系统，通过右键st登陆账号属性的用户映射功能，设置st为当前数据库(如spjdb)用户，数据库角色成员身份设置为db\_owner。设置结果如下：



**再一次执行create table Person.t1(id int,name char(10)) 成功**

5. 修改表结构

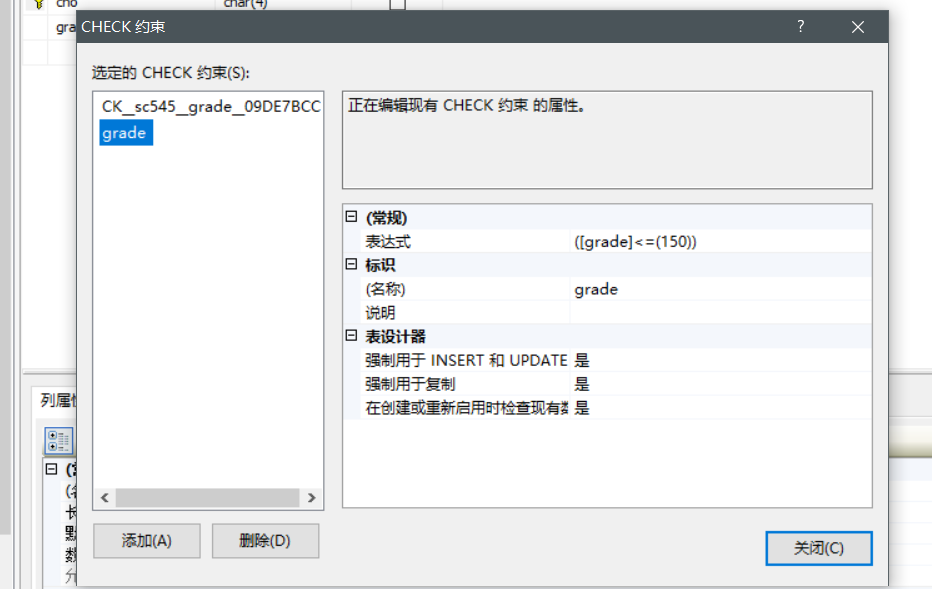
(1) alter table course545 alter column cname varchar(40);



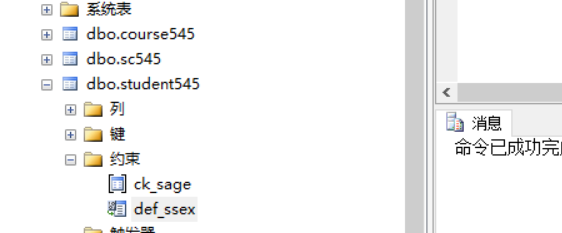
(2) alter table student545 add birthday datetime;



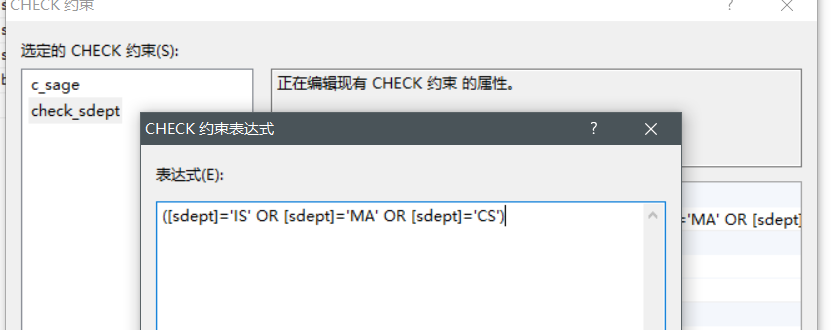
(3) alter table sc545 add constraint CK\_grade check(grade<=150);



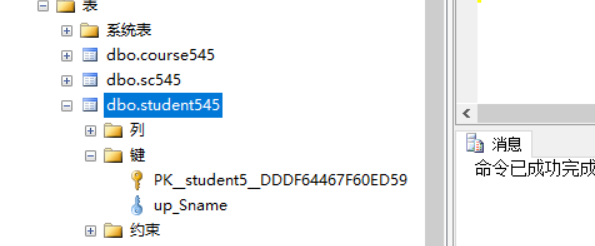
(4) alter table Student545 add constraint def\_ssex default '男' for ssex;



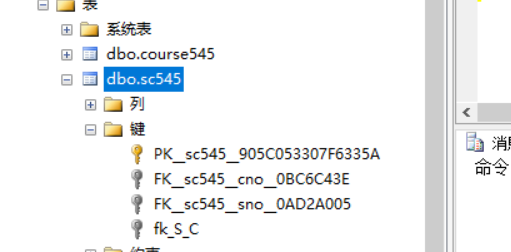
(5) alter table student545 add constraint CK\_sdept check(sdept in ('CS','MA','IS'));



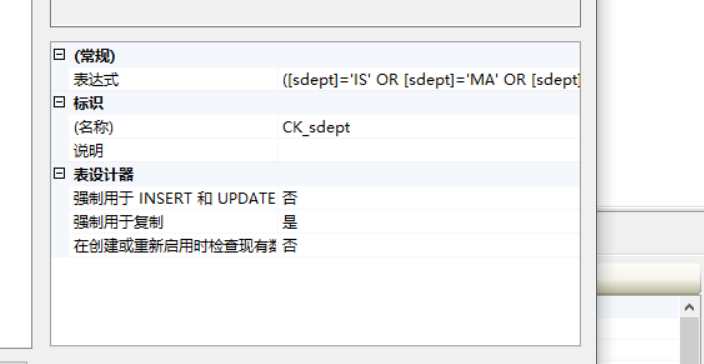
(6) alter table student add constraint UQ\_sname unique(Sname);



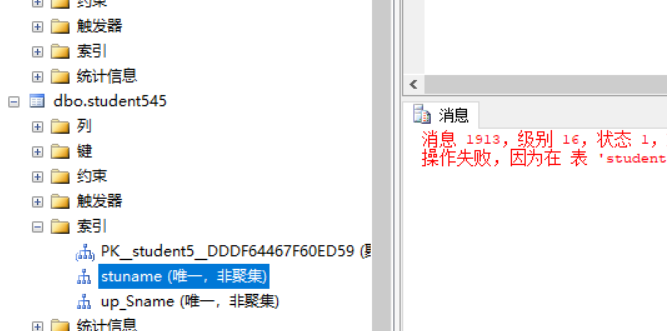
(7) alter table SC545 add constraint fk\_S\_C foreign key(sno) references student545(sno);



(8) alter table student545 nocheck constraint CK\_sdept;

6. 分别建立以下索引(如果不能成功建立,请分析原因)

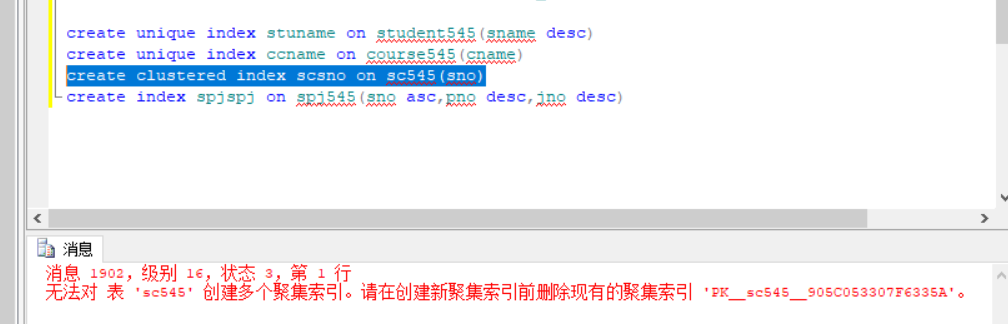
(1) create index index\_sname on student545(sname desc);



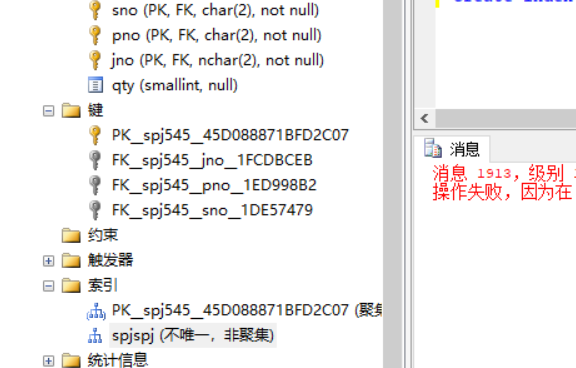
(2) create unique index index\_cname on course545(cname);



(3) create clustered index index\_sno on sc545(sno); --运行出错：无法对表'sc' 创建多个聚集索引。



(4) create index index\_spj on spj545(sno asc,pno asc,jno desc);



7. 索引能提高查询速度，可以使用实验内容中的代码进行过程验证。

在“新建查询”窗口下，复制如下全部代码，直接点“执行”，在消息窗口的最下面，可以看到有索引和无索引的查询耗时。

create table TableIndex

( ID int identity(1,1),

DataValue decimal(18,2)

)

go

/\*---向TestIndex数据库表中插入20000条数据---\*/

declare @r numeric(15,8)

declare @n int

set @n =1

while(1=1)

begin

set @r = rand()

insert into TableIndex (DataValue) values(@r)

set @n = @n + 1

if(@n>20000)

break

end

Go

/\*---查询插入的数据---\*/

select \* from TableIndex

Go

/\*----------------无索引查询-------------------\*/

/\*---计算开始时间---\*/

set nocount on

declare @d datetime

set @d = getDate()

/\*---查询DataValue值在0.3~0.9的数据---\*/

select \* from TableIndex

where DataValue between 0.3 and 0.9

/\*---显示查询所需时间---\*/

declare @time int

set @time = datediff(ms,@d,getDate())

print '无索引查询耗时：'+ convert(varchar(10),@time)

/\*---------------添加索引-------------------\*/

/\*---为DataValue列添加索引---\*/

if exists(select name from sysindexes where name = 'IX\_DataValue')

drop index TableIndex.IX\_DataValue --删除索引

/\*---添加索引---\*/

create clustered index IX\_DataValue

on TableIndex(DataValue)

with fillfactor = 30

Go

/\*----------------有索引查询-------------------\*/

/\*---计算开始时间---\*/

set nocount on

declare @d datetime

set @d = getDate()

/\*---查询DataValue值在0.3~0.9的数据---\*/

select \* from TableIndex with(index=IX\_DataValue)

where DataValue between 0.3 and 0.9

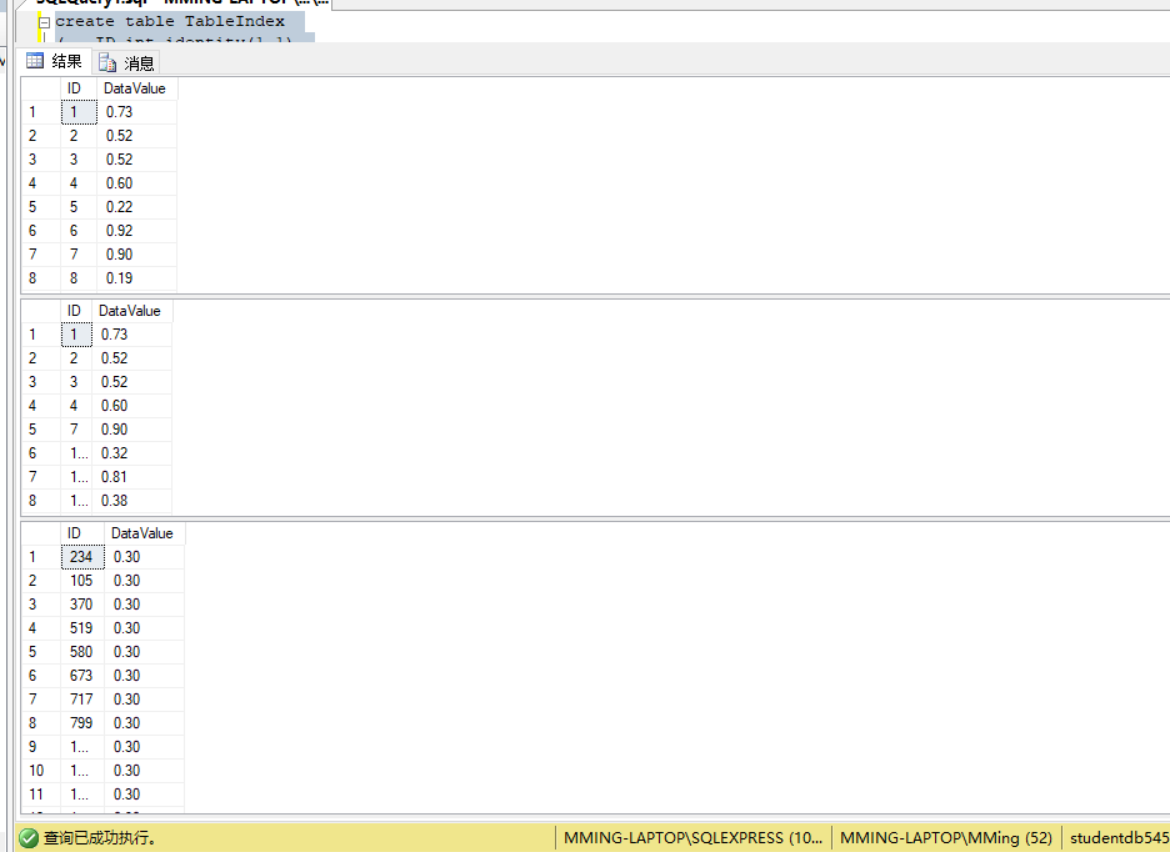
/\*---显示查询所需时间---\*/

declare @time int

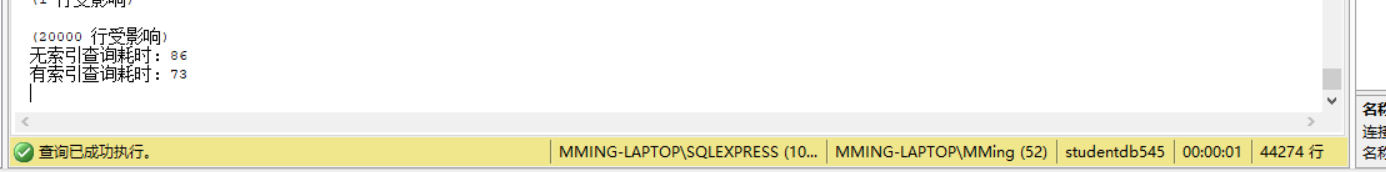
set @time = datediff(ms,@d,getDate())

print '有索引查询耗时：'+ convert(varchar(10),@time)

实验结果如下：



消息卡显示：

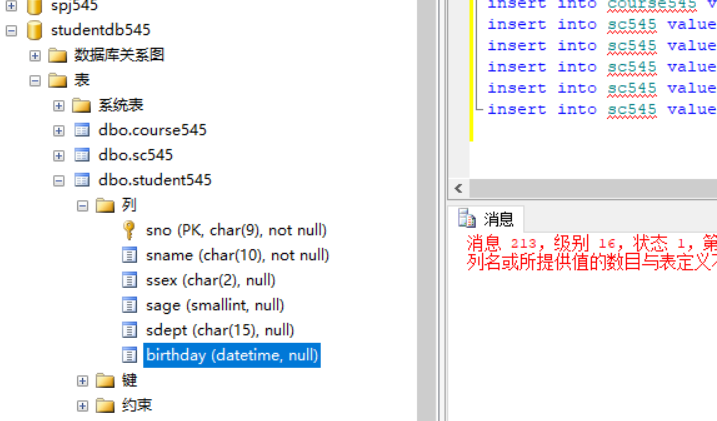


**实验1.2数据操作及查询：**

1. 使用INSERT语句将教材P82表中的数据添加到数据库STUDENTDB中.

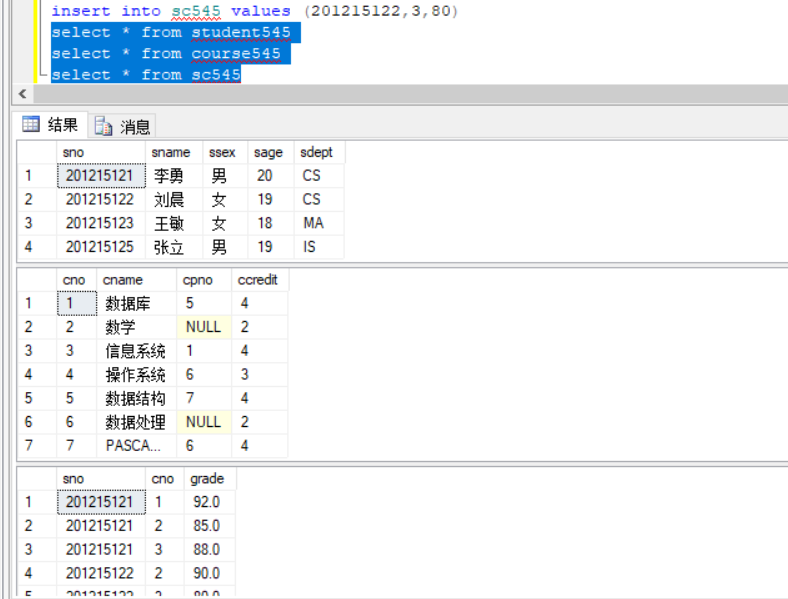
在上方工具条中点“新建查询”，数据库是studentdb，新打开窗口中输入如下SQL语句：

插入之前，先删除birthday属性



执行语句

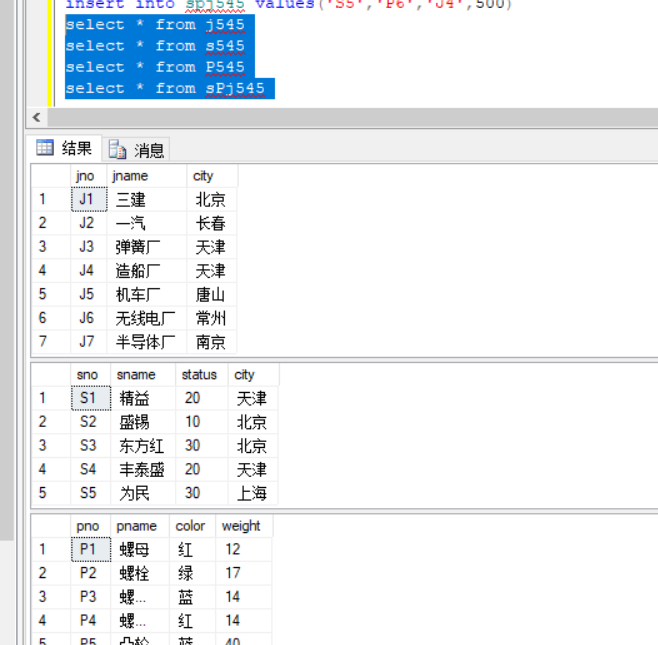
使用查询命令查看结果



1. 将教材P70表中的数据添加到数据库SPJDB中. 体会执行插入操作时检查实体完整性规则、参照完整性规则和用户定义完整性规则的效果.

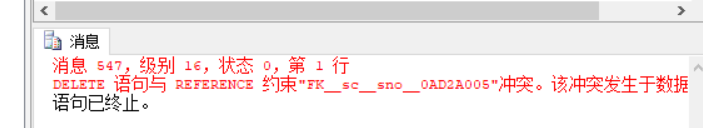
在上方工具条中点“新建查询”，数据库是spjdb，新打开窗口中输入如下SQL语句：

使用查询命令查看插入结果

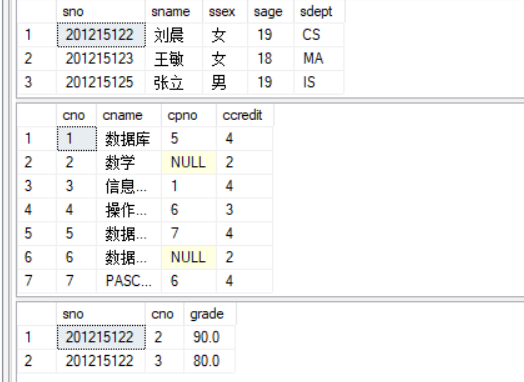


1. 删除student表中学号为201215121的学生，体会执行删除操作时检查参照完整性规则的效果.将参照完整性中的删除规则改为“级联(层叠)”(CASCADE)，重新删除该学生信息。

在数据库为studentdb下执行命令



因为有外码约束，所以不能删除，修改这个constraint更改为级联删除



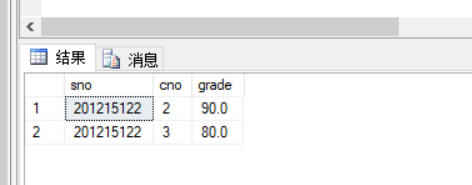
赫然发现在student545 sc545表中都删除了sno为201215121的元组

在数据库为studentdb545下执行命令

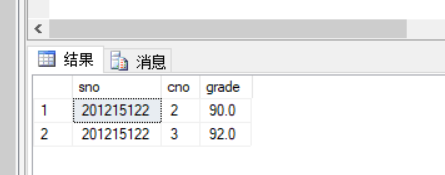
更新student表中201215122学生的新学号为201215128，体会执行更新操作时检查参照完整性规则的效果.将参照完整性中的更新规则改为“级联”，重新更新该学生信息。

1. 设计一组更新操作, 它需要另外一个表中的数据作为更新条件(如将选修了“信息系统”课程的成绩均提高15% )。

查询sc表



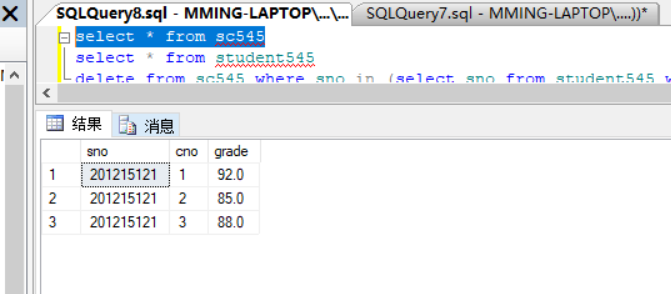
查询sc545表，发现grade已经更新分数



1. 设计一个删除操作，它需要另外一个表中的数据作为删除条件(如将“刘晨”的选课记录删除 )。

将刘晨的选课记录删除

查询sc545表

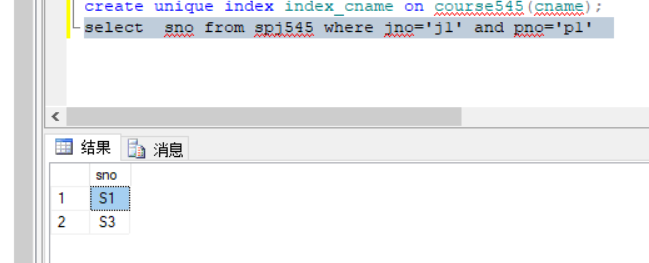


赫然表中没有122（刘晨）的数据

1. 完成P128习题4和习题5中(1)~(7)的查询。

求供应工程J1零件得供应商号码SNO

在spjdb数据库中执行此操作

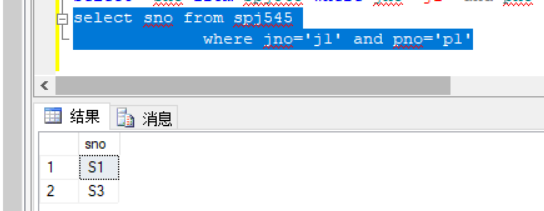
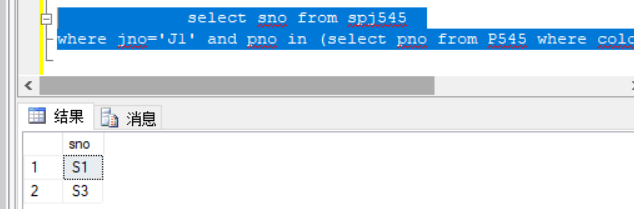


在spjdb数据库中执行此操作

求供应工程J1零件P1得供应商号码SNO

select sno from spj545

where jno=’j1’ and pno=’p1’

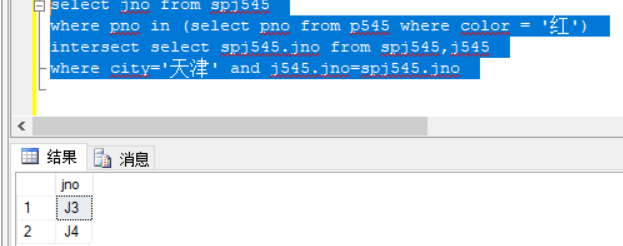
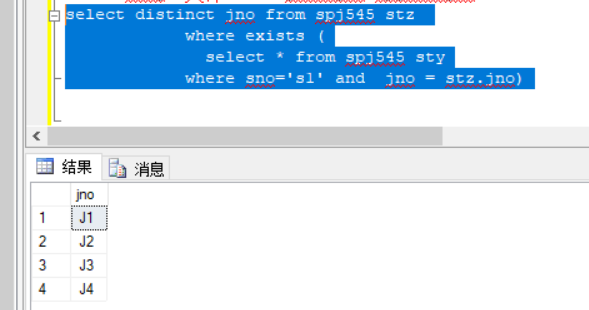
 

求供应工程J1零件为红色得供应商号码SNO

在spjdb数据库中执行此操作

求没有使用天津供应商生产得红色零件得工程号JNO

在spjdb数据库中执行此操作

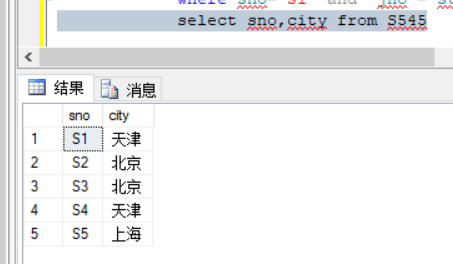
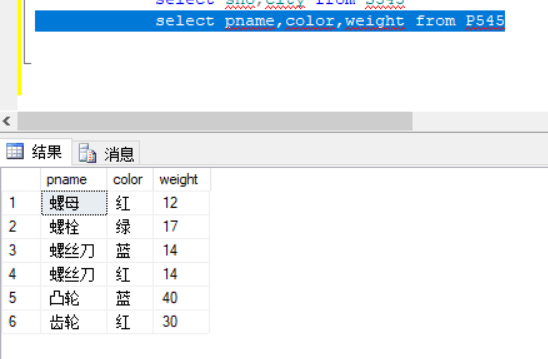
 

求至少用了供应商S1所供应得全部零件得工程号

在spjdb数据库中执行此操作

找出所有供应商得姓名和所在城市

在spjdb数据库中执行此操作

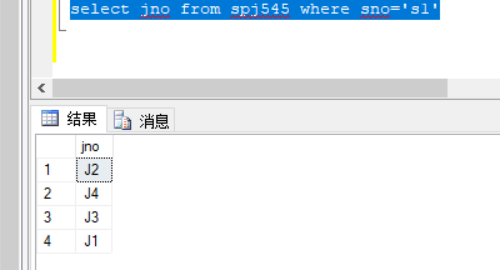
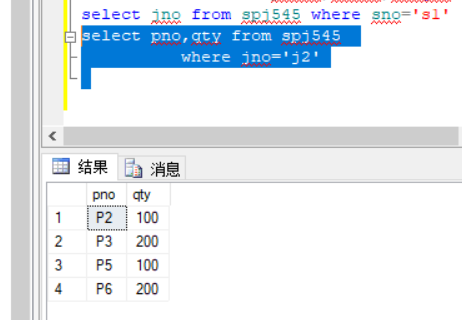
找出所有零件得名称，颜色，重量

select pname,color,weight from P545

找出是使用供应商S1所供应零件得工程号码

在spjdb数据库中执行此操作

select jno from spj545 where sno='s1'

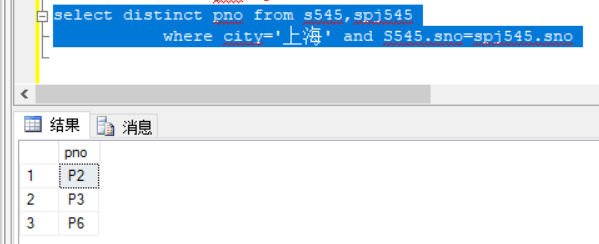
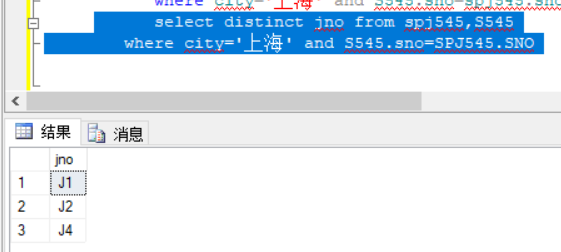
 

找出工程项目J2使用得各种零件得名称及其数量

在spjdb数据库中执行此操作

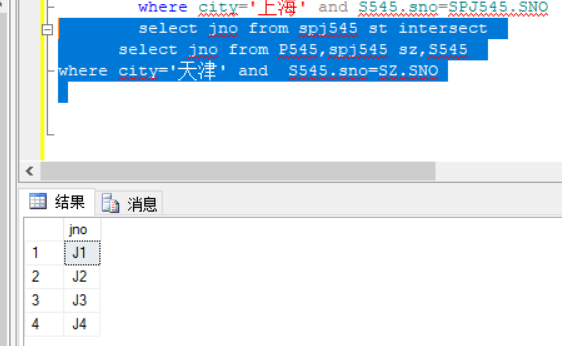
找出上海厂商供应得所有零件号码

在spjdb数据库中执行此操作

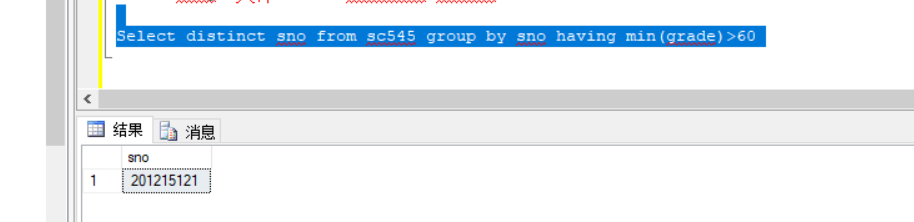
找出使用上海产得零件得工程号码

找出没有使用天津产得零件得工程号码

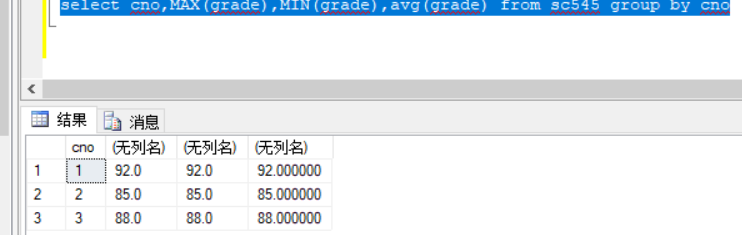


8. 完成以下查询：

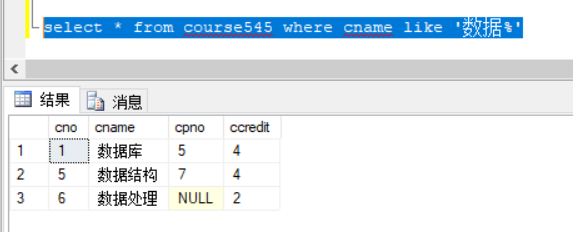
(1)查询所有课程都及格的学生信息。



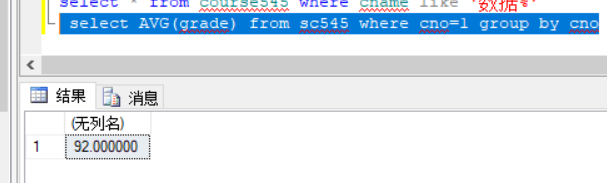
(2)统计每一门课程的最高分、最低分和平均分。



(3)从course表中查询课程名中包含“数据”二字的课程的信息。

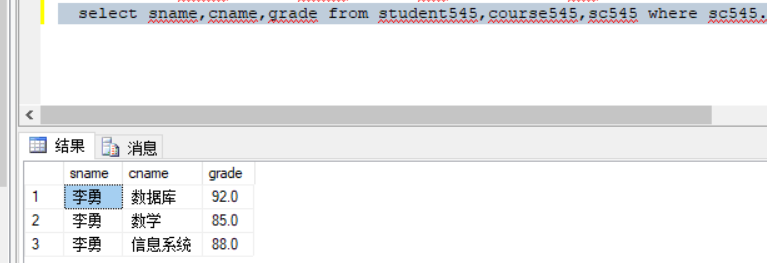


(4)统计所有学生3号课程的平均成绩，要求某个学生3号课程成绩为NULL，如不存在空值，请先插入。



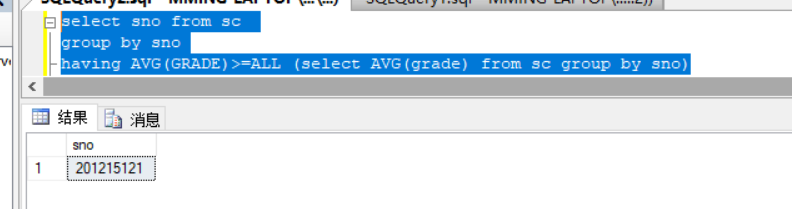
(5)查询每个学生及其选修课情况，显示学生姓名、课程名称和成绩，要求使用外连接输出所有学生和所有课程的信息。

select sname,cname,grade from student545,course545,sc545 where sc545.sno=student545.sno and sc545.cno=course545.cno

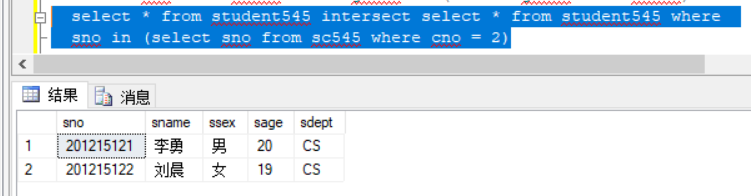


(6)查询所有学生中平均成绩最高的那个学生的学号。

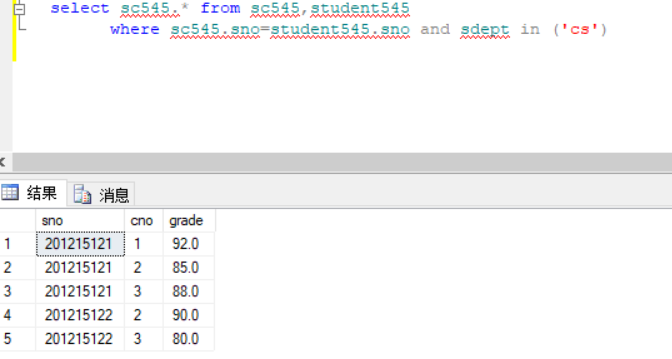
select sno from sc545 where grade>all(select grade from sc545)



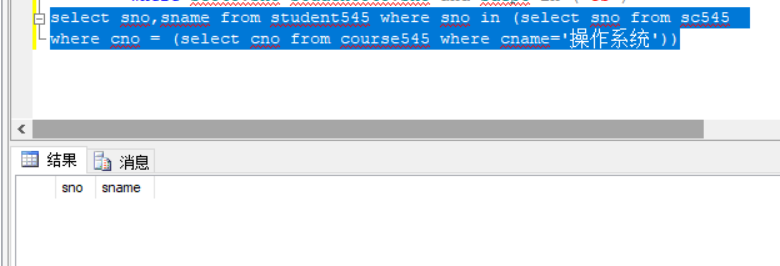
(7)查询只选修了2号课程的学生的信息，可以使用集合查询。



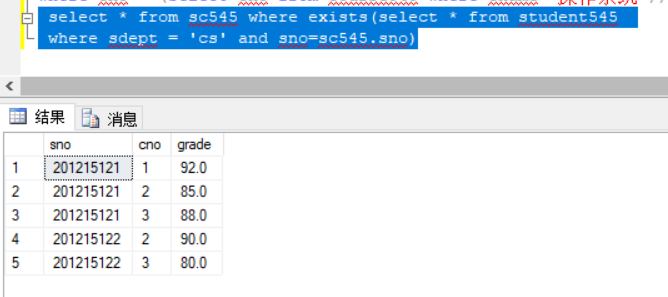
(8)查询所有计算机系学生的学号、选修课程号以及分数。(使用IN谓词)。



(9)查询选修了课程名为“操作系统”的学生学号和姓名。



(10)查询所有计算机系学生的学号、选修课程号以及分数（使用EXISTS谓词）。



**实验1.3 数据库安全性控制**

1.基于实验1中建立的表，完成如下的视图操作：

(1)建立一个计算机系(CS)学生的视图COMPUTE\_S，并要求进行修改和插入操作时仍需保证该视图只有计算机系的学生；

(2) 建立一个信息系(IS)选修了’1’号课程且成绩在80分以上的学生视图IS\_S1；

(3) 建立一个所有计算机系学生的学号、姓名、选修课程名以及成绩的视图COMPUTE\_GRADE;

2. 对视图进行SELECT、INSERT、UPDATE、DELETE操作

（1）通过COMPUTE\_S视图插入一个新学生的信息：201215127，王建，21，男。

要求插入成功后student表中该学生的系值必须为“CS”，如不是，请修改视图后重新插入。

（2）更新COMPUTE\_S视图中所有男生的年龄，都增加一岁，注意观察非计算机系男生的年龄是否被修改。

计算机女生刘晨年龄没有更改，计算机系男生年龄加1

（3）假如你是计算机学院的教务人员，通过COMPUTE\_GRADE视图查询数据库这门课程所有不及格学生的名单，体会视图在简化用户操作过程中的作用。

（4）对COMPUTE\_GRADE进行插入：201215128，张杰，数据库，80。观察结果并分析失败原因。

该视图是由三个关系的个别元组通过联系形成的，这样更改会更改3个基表，因此不允许执行

（5）删除COMPUTE\_S视图中所有学生的信息

3. 创建数据库用户和数据库角色，使之拥有对数据库StudentDB的一定权力。

（1）展开服务器的“安全性”目录树，右击“登录名”,新建登录名test1，SQL Server身份验证，默认数据库为studentdb，用户映射到studentdb数据库，角色成员为public。使用test1用户登录服务器并验证。

（2）使用SQL语句创建用户test2，使之能够访问studentdb数据库并具有建表权限。

（3）使用系统存储过程创建用户test3，使之能够访问studentdb数据库并具有建表权限。

--登录用户的创建

exec sp\_addlogin 'test3', 'test123','studentdb'

--更改当前数据库

use studentdb

--赋予用户访问当前数据库的权限

exec sp\_grantdbaccess 'test3'

--角色的创建

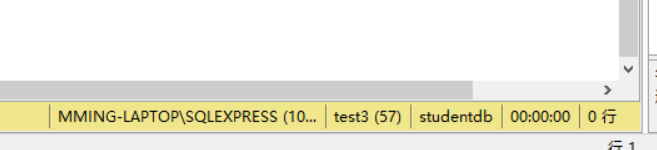
exec sp\_addrole 'teacher'

--添加用户到角色中

exec sp\_addrolemember 'teacher','test3'

--给角色授权

grant create table to teacher



4. 授予用户李勇查看自己成绩的权限，但不能查看其它人的成绩。

提示：先创建用户李勇，然后创建视图S，通过视图S能查看学生“李勇”的成绩，最后把视图S的查询权限赋予用户李勇。

5. 启用SQL Server的审计功能并通过审计日志进行简单的审计跟踪。

右击左侧目录树最上端的服务器——>属性——>安全性，选中“启用c2审核跟踪”。

也可以通过如下存储过程启用c2审核跟踪。

--查看系统参数配置

sp\_configure

--打开系统参数的高级选项

sp\_configure 'show advanced options',1

RECONFIGURE

--再次查看系统参数配置

sp\_configure

--启用c2级别的审计功能

sp\_configure 'c2 audit mode',1

RECONFIGURE

启用c2审核跟踪后，重新启动服务器，在C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Data目录下会产生以当前时间命名的trc跟踪文件。此时可以分别尝试使用错误口令和正确口令登录服务器并进行其它操作，然后停止服务器。



查看审计日志需要使用专门的查看工具，也可以使用操作系统的事件查看器或者SQL Server的扩展存储过程sp\_readerrorlog。

**实验1.4 T-SQL程序设计**

1 定义变量并且通过select语句给变量赋值.

查询学号为“201215121”的学生的姓名和年龄，将其分别赋予变量name和age。

declare @name varchar(10)

declare @age int

select @name=sname,@age=sage from student where sno='201215121'

select @name as 姓名,@age as 年龄

运行结果如下：

姓名 年龄

-------- -----------

李勇 31

练习：查询学号为“201215121”的学生选修“1”号课程的成绩，并赋值给变量grade。

1. if-lese选择结构的使用

查询学生信息，如果学生人数多于10人，则只显示前5名，否则显示所有学生信息。

DECLARE @num int

SELECT @num=count(\*) from student

print '学生人数:'+convert(varchar(5),@num)

IF (@num>10)

BEGIN

print '前五名学生信息'

SELECT TOP 5 \* FROM student ORDER BY sno DESC

END

ELSE

BEGIN

print '所有学生信息'

SELECT \* FROM student

END

练习：查询“1”号课程的平均成绩，如果平均成绩大于等于70，则显示前5名的成绩，否则显示后5名的成绩。

1. while循环结构的使用

创建一个测试表，并使用循环结构快速插入20000条记录。

create table TableIndex (

ID int identity(1,1),

DataValue decimal(18,2))

/\*---向TestIndex数据库表中插入20000条数据---\*/

declare @r numeric(15,8)

declare @n int

set @n = 1

while(1=1)

begin

set @r = rand()

insert into TableIndex (DataValue) values(@r)

set @n = @n + 1

if(@n>20000)

break

end

练习1：检查“1”号课程是否有不及格（60分及格）的学生。如有，每人加2分，直至所有学生选“1”号课程成绩均及格。

练习2：检查“1”号课程是否有不及格（60分及格）的学生。如有，每人加2分，高于95分的学生不再加分，直至所有学生“1”号课程成绩均及格。

1. 编写带有通配符参数的存储过程，查询学生表和成绩表，返回指定姓名的学生姓名、课程名和考试成绩。要求：执行该存储过程时，如果未提供参数，则使用预设的默认值（以“张”打头的姓）

/\*---创建存储过程---\*/

CREATE PROCEDURE sp\_stuInfo

@sname varchar(50)='张%'

AS

PRINT @sname+'同学的考试成绩单如下：'

SELECT sname,cname,grade

FROM student,course,sc

WHERE student.sno=sc.sno and course.cno=sc.cno

and sname LIKE @sname

练习：创建存储过程，查看指定课程及格学生的成绩信息，并返回及格的人数。要求及格分数线作为参数由用户输入，默认为60。

1. 创建触发器，当插入交易记录时，实现自动更新账户余额的功能。建表语句如下：

create table count45(

id char(3) primary key,

balance int)

insert into count45 values('001', 1000)

create table traninfo45(

id char(3) foreign key references count(id),

amount int)

/\*---创建触发器---\*/

create trigger autoupdate\_balance

on traninfo45

for insert

as

update count45 set balance= balance+i.amount

from inserted i

where count45.id=i.id

练习1：删除交易信息时，要求自动备份被删除的数据到表backupTable中

练习2：跟踪用户的交易，交易金额超过20000元，则取消交易，并给出错误提示。

编写事务进行银行转账。

--创建农行帐户表bank

CREATE TABLE bank(

customerName CHAR(10), --顾客姓名

currentMoney int --当前余额

)

/\*---添加约束：根据银行规定，帐户余额不能少于1元，除非销户----\*/

ALTER TABLE bank

ADD CONSTRAINT CK\_currentMoney CHECK(currentMoney>=1)

/\*--插入测试数据：张三开户，开户金额为800 ；李四开户，开户金额1 ---\*/

INSERT INTO bank(customerName,currentMoney) VALUES('张三',1000)

INSERT INTO bank(customerName,currentMoney) VALUES('李四',1)

/\*---创建事务---\*/

/\*--开始事务（指定事务从此处开始，后续的T-SQL语句都是一个整体--\*/

BEGIN TRANSACTION

/\*--定义变量，用于累计事务执行过程中的错误--\*/

DECLARE @errorSum INT

SET @errorSum=0 --初始化为0，即无错误

/\*--转帐：张三的帐户少1000元，李四的帐户多1000元\*/

UPDATE bank SET currentMoney=currentMoney-1000 WHERE customerName='张三'

SET @errorSum=@errorSum+@@error --累计是否有错误

UPDATE bank SET currentMoney=currentMoney+1000 WHERE customerName='李四'

SET @errorSum=@errorSum+@@error --累计是否有错误

/\*--根据是否有错误，确定事务是提交还是撤销---\*/

IF @errorSum<>0 --如果有错误

BEGIN

print '交易失败，回滚事务'

ROLLBACK TRANSACTION

END

ELSE

BEGIN

print '交易成功，提交事务，写入硬盘，永久的保存'

COMMIT TRANSACTION

END

练习：使用第5题中的count表和traninfo表，编写事务实现存取款过程。

要求首先插入交易记录，然后更新账户余额。