

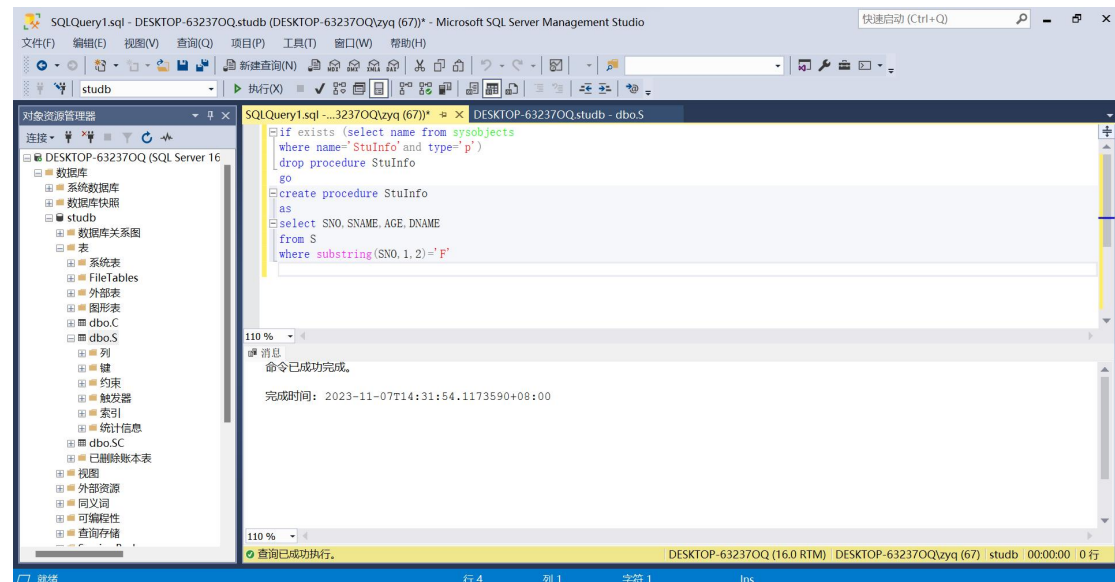
参考下面四个实验，用 SPJ 库，自行设计问题，每个实验设计 3 个不同类型的问题。

(3 个存储过程，3 个触发器，用 SPJ 库中的四张表，自行设计)

实验 1：存储过程

(1) 创建一个名称为 StuInfo 的存储过程，要求完成以下功能：在 S 表中查询女生的学号、姓名、年龄和专业四个字段的内容。

```
if exists (select name from sysobjects
where name='StuInfo' and type='p')
drop procedure StuInfo
go
create procedure StuInfo
as
select SNO, SNAME, AGE, DNAME
from S
where substring(SNO, 1, 2)='F'
```



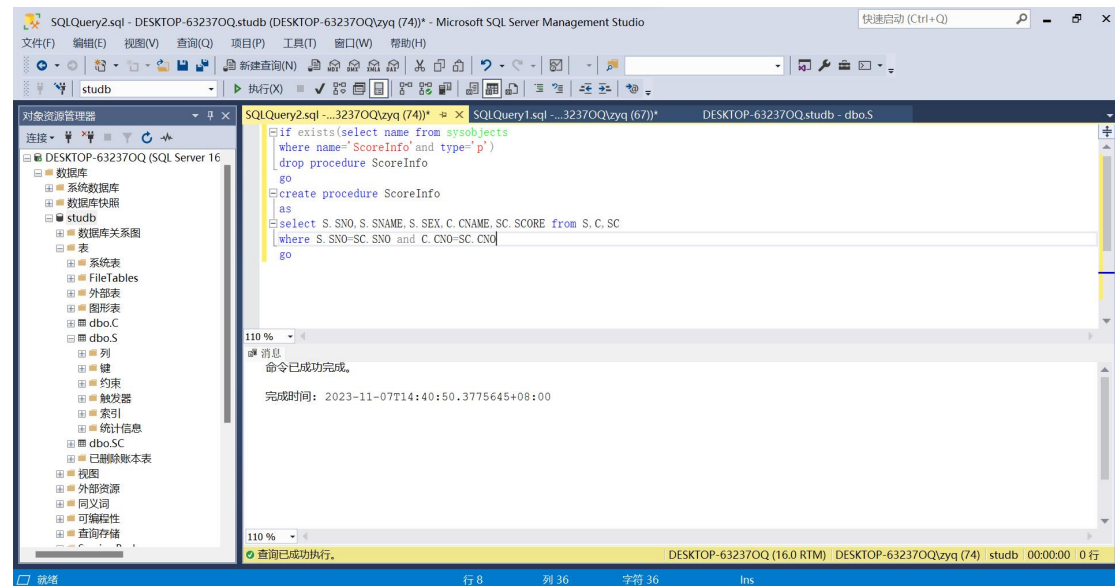
(2) 创建一个存储过程 ScoreInfo，完成的功能是在表 S、表 C 和表 SC 中查询以下字段：学号、姓名、性别、课程名称、分数。

```
if exists(select name from sysobjects
where name='ScoreInfo' and type='p')
drop procedure ScoreInfo
go
create procedure ScoreInfo
```

as

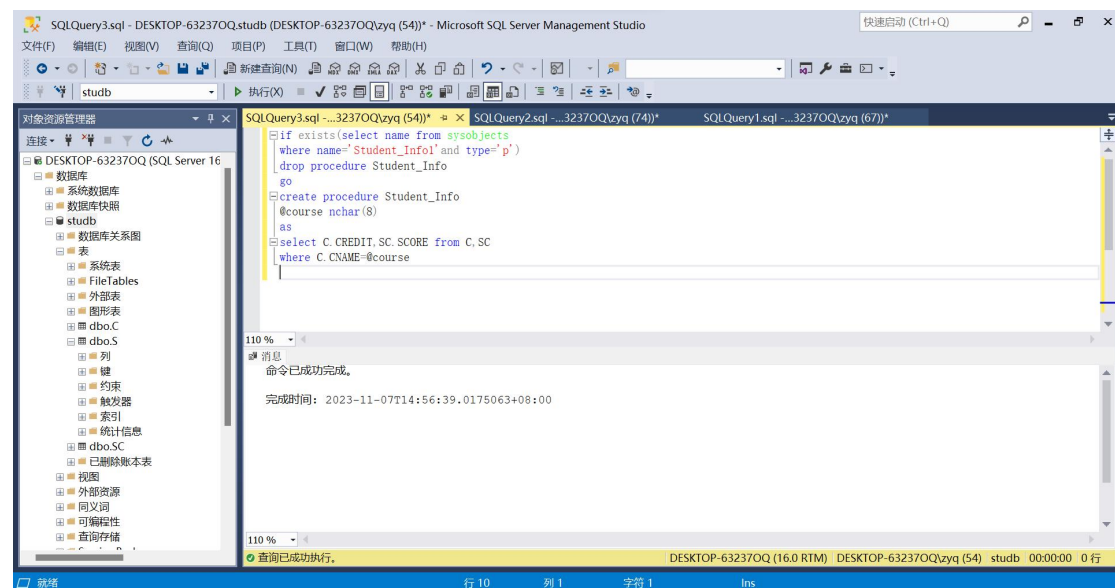
```
select S.SNO,S.SNAME,S.SEX,C.CNAME,SC.SCORE from S,C,SC  
where S.SNO=SC.SNO and C.CNO=SC.CNO
```

go



(3) 创建一个名称为 Student_Info1 的存储过程，要求查询某位同学指定课程的成绩和学分。该存储过程接受与传递参数精确匹配的值。

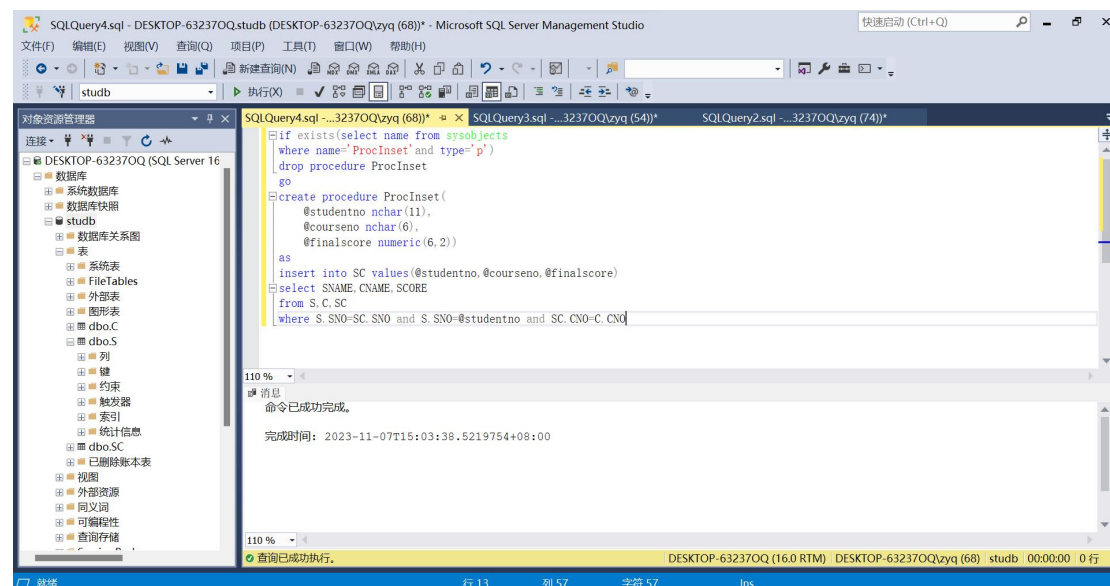
```
if exists(select name from sysobjects  
where name='Student_Info1'and type='p')  
drop procedure Student_Info  
go  
create procedure Student_Info  
@course nchar(8)  
as  
select C.CREDIT,SC.SCORE from C,SC  
where C.CNAME=@course
```



(4) 利用 Transact-SQL 语句创建一个带有参数的存储过程 ProcInsert，向 score 表插入

一条由参数指定的选课记录（学号、课程号、分数），并查询该学生的姓名、选修的所有课程名称、平时成绩和期末成绩。

```
if exists(select name from sysobjects
where name='ProcInset' and type='p')
drop procedure ProcInset
go
create procedure ProcInset(
    @studentno nchar(11),
    @courseno nchar(6),
    @finalscore numeric(6,2))
as
insert into SC values(@studentno,@courseno,@finalscore)
select SNAME,CNAME,SCORE
from S,C,SC
where S.SNO=SC.SNO and S.SNO=@studentno and SC.CNO=C.CNO
```



(5) 统计输入课程的成绩分布情况，即按照各分数段来统计人数：

Rank(division CHAR(20), number INT)

division: '[0,60)', '[60,80)', '[80,100]'

```
CREATE TABLE Rank(
    division char(20),
    sub_sum int
)
INSERT INTO Rank(division)
VALUES (' [0, 60) '), (' [60, 70) '), (' [70, 80) '), (' [80, 90) '), (' [90, 100] ');
```

```
CREATE PROCEDURE printcourse @pcname char(20)
AS
    DECLARE @pcno char(20), @pcount int
    UPDATE C SET CNAME=@pcname WHERE CNAME=@pcname
    IF (@@ROWCOUNT=0)
```

```

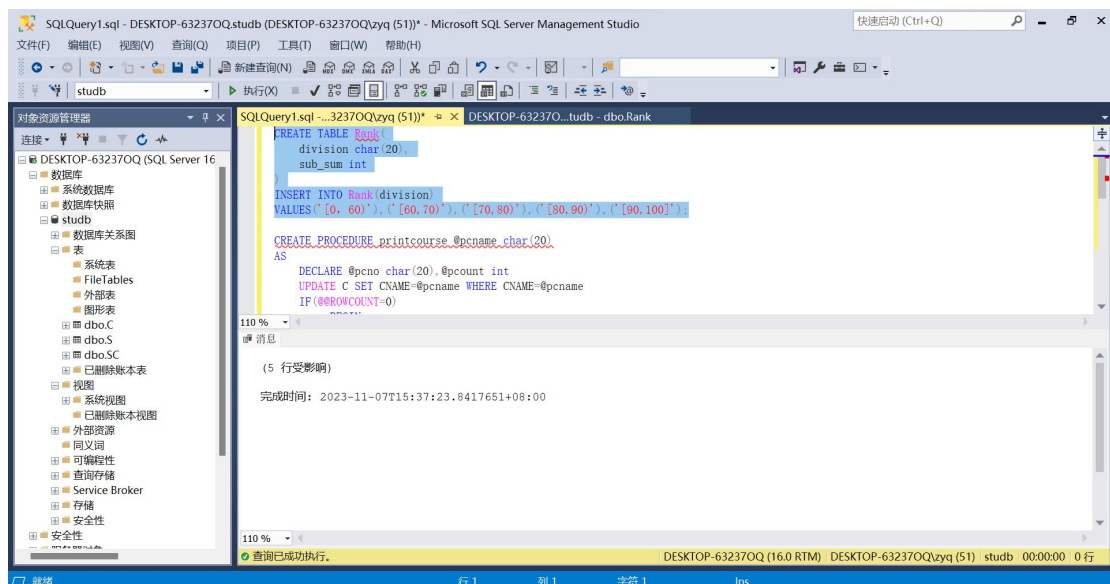
BEGIN
    RAISERROR('您输入的课程号不存在，请重新输入！', 16, 1)
RETURN
END

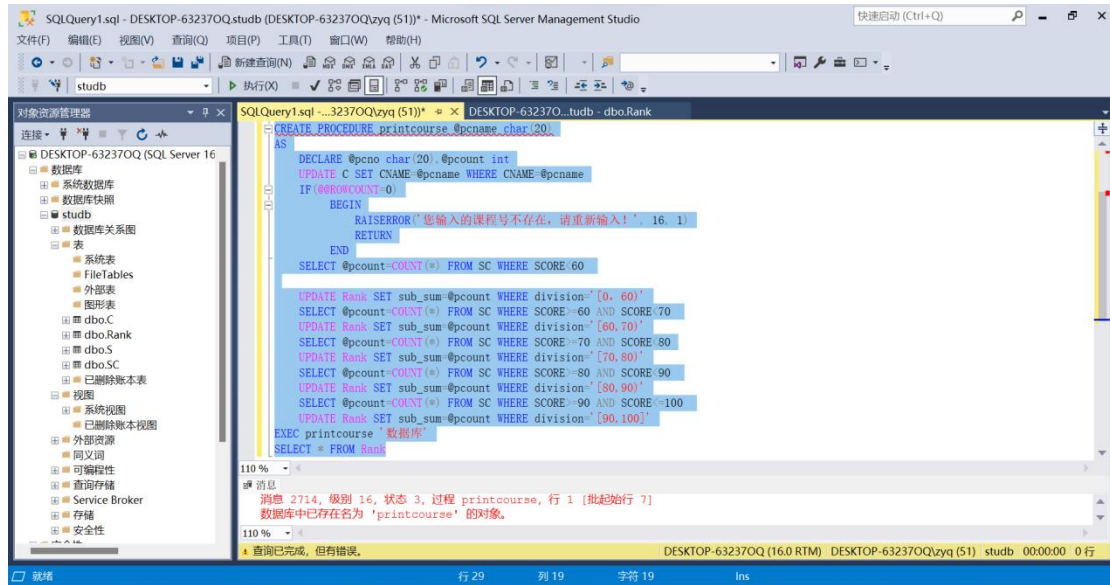
SELECT @pcount=COUNT(*) FROM SC WHERE SCORE<60

UPDATE Rank SET sub_sum=@pcount WHERE division=' [0, 60)'
SELECT @pcount=COUNT(*) FROM SC WHERE SCORE>=60 AND SCORE<70
UPDATE Rank SET sub_sum=@pcount WHERE division=' [60, 70)'
SELECT @pcount=COUNT(*) FROM SC WHERE SCORE>=70 AND SCORE<80
UPDATE Rank SET sub_sum=@pcount WHERE division=' [70, 80)'
SELECT @pcount=COUNT(*) FROM SC WHERE SCORE>=80 AND SCORE<90
UPDATE Rank SET sub_sum=@pcount WHERE division=' [80, 90)'
SELECT @pcount=COUNT(*) FROM SC WHERE SCORE>=90 AND SCORE<=100
UPDATE Rank SET sub_sum=@pcount WHERE division=' [90, 100]'

EXEC printcourse '数据库'
SELECT * FROM Rank

```

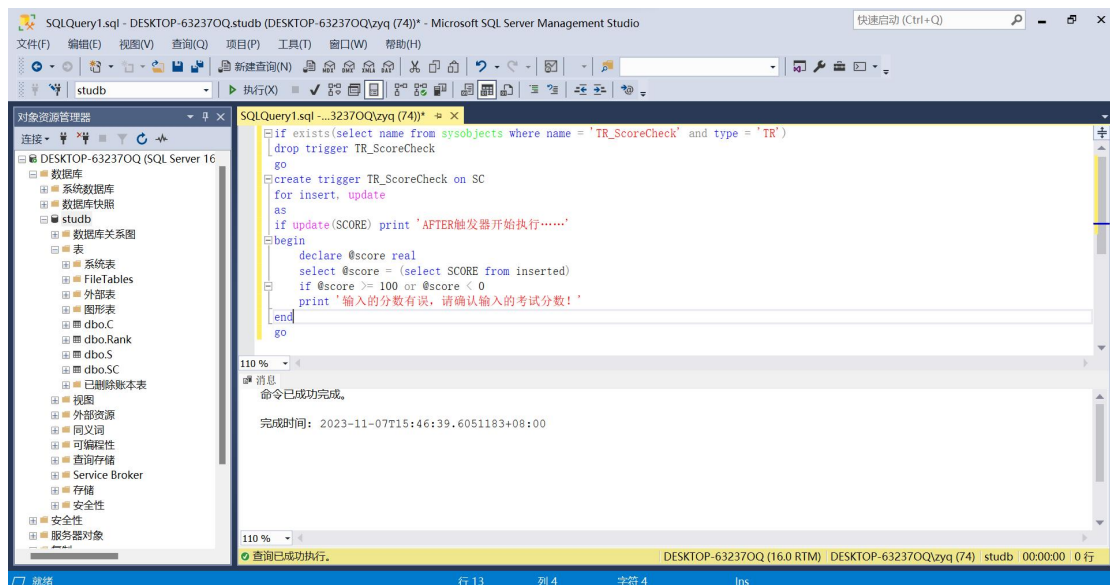




实验 2：触发器

(1) 创建 AFTER 触发器：在 SC 表上创建一个插入、更新类型的触发器 TR_ScoreCheck，当在 score 字段中插入或修改考试分数后，触发该触发器，检查分数是否在 0-100 之间。

```
if exists(select name from sysobjects where name = 'TR_ScoreCheck' and type = 'TR')
drop trigger TR_ScoreCheck
go
create trigger TR_ScoreCheck on SC
for insert, update
as
if update(SCORE) print 'AFTER触发器开始执行……'
begin
    declare @score real
    select @score = (select SCORE from inserted)
    if @score >= 100 or @score < 0
        print '输入的分數有誤，請確認輸入的考試分數！'
end
go
```

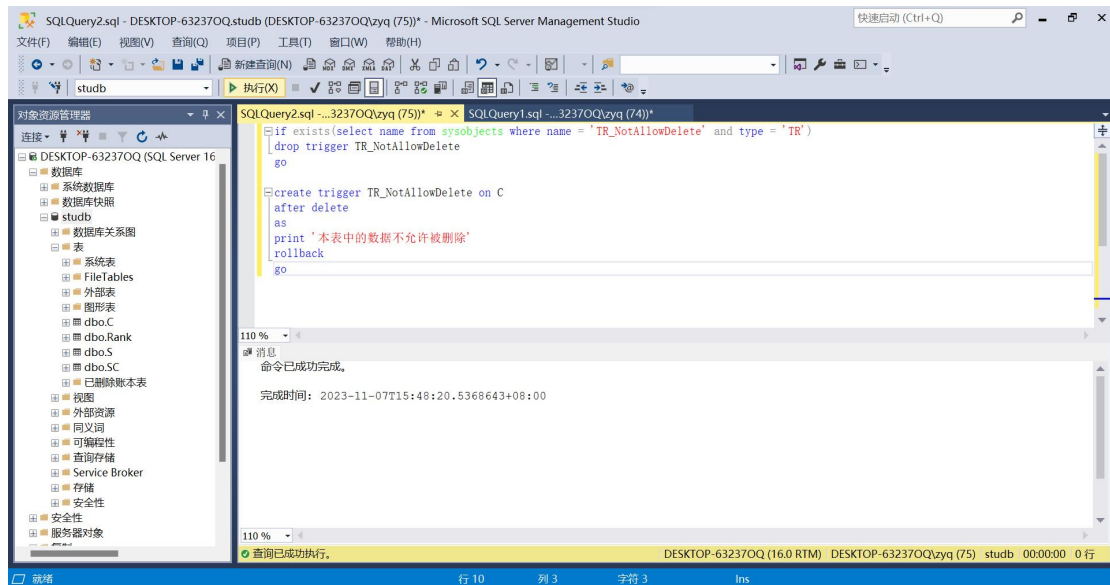


(2) 创建 INSTEAD OF 触发器：在 C 表上创建一个删除类型的触发器 TR_NotAllowDelete，当在 C 表中删除记录时，触发该触发器，显示不允许删除表中数据的提示信息。

```
if exists(select name from sysobjects where name = 'TR_NotAllowDelete' and type = 'TR')
drop trigger TR_NotAllowDelete
go

create trigger TR_NotAllowDelete on C
after delete
as
print '本表中的数据不允许被删除'
rollback
```

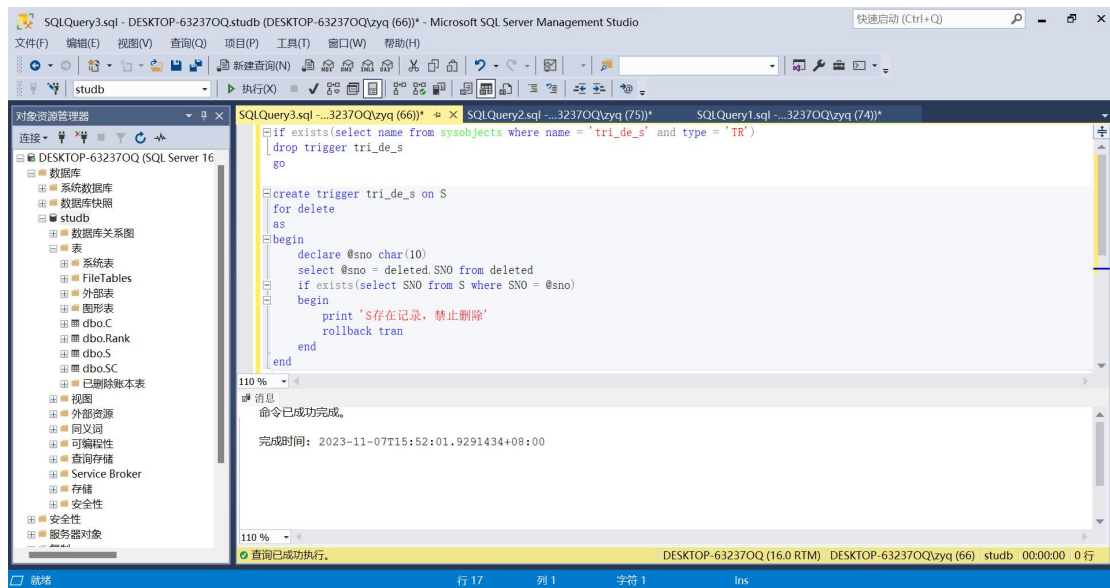
go



(3) 为 S 表创建一个名为“tri_de_s”的后触发器，当删除某条记录时，自动检查表 S 中是否有该学生的选课记录，如果存在则取消删除操作。

```
if exists(select name from sysobjects where name = 'tri_de_s' and type = 'TR')
drop trigger tri_de_s
go
```

```
create trigger tri_de_s on S
for delete
as
begin
    declare @sno char(10)
    select @sno = deleted.SNO from deleted
    if exists(select SNO from S where SNO = @sno)
    begin
        print 'S存在记录，禁止删除'
        rollback tran
    end
end
```



(4) 为 S 表创建触发器，保证学生年龄不能低于 18 岁。

```
if exists(select name from sysobjects where name = 'ST_SalCheck' and type = 'ST')
drop trigger ST_SalCheck
go
```

```
create trigger ST_SalCheck on S
for insert,update
as
begin
declare @s numeric
select @s = (select AGE from inserted)
if(@s < 18)
begin
print '学生年龄不能低于18岁'
rollback
end
end
```