

(1).创建 SCT 数据库，数据库的两个文件放在 D:\DATA 文件夹下。

(2).数据库数据文件的初始大小为 50M,日志文件的初始大小为 10M, 都 10%增长。

```
create database sct
on
(
name=sct_data,
filename='d:\data\sct_data.mdf',
size=50MB,
filegrowth=10%
)
log on
(
name=sct_log,
filename='d:\data\sct_log.ldf',
size=10MB,
filegrowth=10%
)
go
```

(二)假设要创建的 SCT 有以下几个关系模式

1.学院（学院号，学院名称，院长）

```
coll(collnum,collname,tnum)
use sct
go
create table coll
```

```
(
collnum nchar(2) primary key,
collname nvarchar(20),
tnum nchar(8),
);
```

2.专业(专业号，专业名称，所在学院)

```
d(dnum,dname,collnum)
create table d
(
dnum nchar(4) primary key,
dname nvarchar(20),
collnum nchar(2) foreign key references coll(collnum)
);
```

3.学生（学号，姓名，性别，出生日期，政治面貌，民族，籍贯，入学日期，电话，专业号）

```
s(snum,sname,ssex,sbirth,spolity,snation,sorigin,
enterdate,sphone,dnum)
create table s
(
snum nchar(12) primary key,
```

```

sname nvarchar(20),
ssex nchar(1),
sbirth date,
spolity nvarchar(20),
snation nvarchar(20),
sorigin nvarchar(20),
enterdate date,
sphone nvarchar(13),
dnum nchar(4) foreign key references d(dnum) ,
check((ssex='男')or(ssex='女'))
);

```

4.教师（工号，姓名，性别，出生日期，政治面貌，民资，籍贯，学位，职称，电话，工资，所在学院）

```

t(tnum,tname,tsex,tbirth,tpolity,tnation,torigin,degree,
ttitle,tphone,tsalary,collnum)
create table t
(
tnum nchar(8) primary key,
tname nvarchar(20),
tsex nchar(1),
tbirth date,
tpolity nvarchar(20),
tnation nvarchar(20),
torigin nvarchar(20),
degree nvarchar(20),
ttitle nvarchar(20),
tphone nvarchar(13),
tsalary nvarchar(20),
collnum nchar(2)foreign key references coll(collnum),
check((tsex='男')or(tsex='女'))
);

```

5.课程（课号，课程名，学分）

```

c(cnum,cname,credit)
create table c
(
cnum nchar(4) primary key,
cname nvarchar(20),
credit numeric(2,1)
);

```

6.选修（学号，课号，成绩）

```

sc(snum,cnum,score)
create table sc
(
snum nchar(12) foreign key references s(snum),

```



```
cnum nchar(4) foreign key references c(cnum),
score tinyint,
check((score>=0)and(score<=100)),
primary key (snum,cnum)
);
```

7.任课（工号,课号）

```
tc(tnum,cnum)
create table tc
(
tnum nchar(8) foreign key references t(tnum),
cnum nchar(4) foreign key references c(cnum),
primary key (tnum,cnum)
);
```

1.在已经存在的学生表中增加一个邮件地址“Semail”的新属性列，允许为空；

```
alter table s add semail nvarchar(20);
```

2.修改教师表中 tsalary 的数据类型为 money);

```
alter table t alter column tsalary money;
```

3.把学生表中的电话属性删掉；

4.删除学生性别是男或女的约束条件；

5.增加学生性别是男或女的约束条件。

```
alter table s drop column sphone;
```

```
alter table s drop CK__s__ssex__0AD2A005;
```

```
alter table s add constraint chk_ssex check ((ssex='男')or(ssex='女'));
```

1. 查询系编号为'0501'学生的基本信息（学号、姓名、性别、出生日期）。

2.查询学号为'201305010101'的学生的姓名。

3.查询成绩在[90,100]之间的学生的学号(去掉重复)。

```
select snum,sname,ssex,sbirth
from s
where dnum='0501'
```

```
select sname
from s
where snum='201305010101'
```

```
select distinct snum
from sc
where score>='90' and score<='100'
```

4.查询学院编号为'22'的教师姓名和工资情况，工资增加 1000 后输出。

5.查询有选课记录的所有学生的学号，用 **DISTINCT** 限制结果中学号不重复。

6.查询选修课程'05041017'的学生学号和成绩，结果按成绩的升序排列，如果成绩相同则按学号的降序排列。

```
select tname,tsalary,tsalary+1000
from t
where collnum='22'
select distinct snum
from sc
select snum,score
from sc
where cnum='05041017'
order by score asc,snum desc
```

7.查询全体学生情况，查询结果按所在系的系号升序排列，同一系中的学生按年龄降序排列。

8.查询全体教师情况，查询结果按学院编号升序排列，同一个学院的教师，工资按降序排列。

9.查询选修课程号为'05041017'的平均成绩。

10.查询选修'05041017'号课程的学生最高分数。

```
select *
from s
order by dnum asc,sbirth asc
select *
from t
order by collnum asc,tsalary desc
select AVG(score)
from sc
where cnum='05041017'
select MAX(score)
from sc
where cnum='05041017'
```

11.统计学生人数。

12.统计选修了课程的学生人数。

13.从学生表中查出名字第2个字是'睿' 或 '晓' 或 '子' 或'涵'的学生资料。

14.查询所有籍贯是浙江的学生信息。

```
select COUNT(*)
from s
select COUNT(distinct snum)
from sc
```



```

select *
from s
where sname like '_[睿,晓,子,涵]%'
select *
from t
where torigin like '浙江%'

```

15. 查全体学生的姓名及其出生年份。

16.列出职称是副教授或讲师（高校）或教授的教师信息。

17.列出不在'22'号学院工作的所有副教授的信息。

18.列出 1970 年 1 月 1 日前出生，目前工资低于 5000 元或高于 7000 元的教师信息。

```

select sname,DATEPART(YEAR,sbirth)
from s
select *
from t
where ttitle='副教授' or ttitle='讲师（高校）' or ttitle='教授'
select *
from t
where collnum!='22'
select *
from t
where tbirth<'1970-1-1'and (tsalary<'5000' or tsalary>'7000')

```

19.在职称非空的情况下，连接教师表中的姓名与职称。

20.查询今天是周几（datetime()函数可查询周几，getdate()取系统当前日期）

21. 按照性别统计学生人数。

22.查询平均成绩低于 80 分的学生学号和平均成绩。

```

select tname,ttitle
from t
where ttitle!='NULL'
select DATENAME(weekday,getdate())
select ssex,COUNT(*)
from s
group by ssex
select snum,avg(score)
from sc
group by snum
having AVG(score)<80

```

23.按职称统计所有男教师的最高工资。

24.查询选修了 50 门以上课程是 90 分以上的学生的学号及（90 分以上的）课程数。

25.按平均工资从高到低的顺序分别列出男、女教师各种职称的人数（任职人数多于 100）和平均工资。  
（筛选出工资非空信息）

```
select title,MAX(tsalary)
from t
where tsex='男'
group by title
select snum,COUNT(cnum)
from sc
where score>90
group by snum having COUNT(cnum)>50
select tsex,title,COUNT(tnum),AVG(tsalary)
from t
where tsalary is not null
group by tsex,title having COUNT(tnum)>100
order by AVG(tsalary) desc
```

#### 一. 多表的分组查询

1. 统计各系的学生人数，显示各系的系名称和学生人数。

```
select dname,COUNT(*)
from d,s
where s.dnum=d.dnum
group by dname
```

2.查询有 50 门以上课程是 90 分以上的学生姓名、学号、课程数。

```
select sname,s.snum,COUNT(*)
from s,sc
where s.snum=sc.snum and score>90
group by sname,s.snum
having COUNT(*)>50
```

3.按职称统计信息工程学院平均工资高于 6000 的各种职称的人数和平均工资，查询结果按平均工资从大到小顺序排列。

```
select title,count(*),AVG(tsalary)
from t,coll
where t.collnum=coll.collnum and collname='信息工程学院'
group by title
having avg(tsalary)>6000
order by AVG(tsalary) desc
```

4.统计同名同姓的学生姓名和人数，并按人数从高到低排列。

```
select sname,COUNT(*)
from s
group by sname
```



having COUNT(\*)>1

order by COUNT(\*) desc

5.查询‘计算机科学与技术’专业选修课程的最低成绩大于等于 90 分的学生的姓名。

select sname

from s,sc,d

where d.dnum=s.dnum and s.snum=sc.snum and d.dname='计算机科学与技术'

group by sname

having MIN(score)>90

二.用连接与自身连接实现下列查询操作

1.查询所有学生的学号、姓名、选修的课程号和成绩。

select s.snum,sname,sc.cnum,sc.score

from s,sc

where s.snum=sc.snum

2.查询所有学生的姓名、选修的课程名和相应的学分。

select s.sname,c.cname,c.credit

from s,sc,c

where s.snum=sc.snum and sc.cnum=c.cnum

3.查询选修了数据库原理与技术并且成绩在 90 分以上的学生的姓名。

select sname

from s,sc,c

where s.snum=sc.snum and sc.cnum=c.cnum and c.cname='数据库原理与技术' and sc.score>90

4.查询工资高于杨平的所有教师的姓名、职称和工资。

select t1.tname,t1.title,t1.tsalary

from t t1,t2

where t1.tsalary>t2.tsalary and t2.tname='杨平'

5.查询与“沈羿”在同一个系学习的学生姓名和性别。

select s1.sname,s1.ssex

from s s1,s s2

where s1.dnum=s2.dnum and s2.sname='沈羿'

三、用子查询（嵌套）实验下列查询操作

1.查询选修了数据库原理与技术并且成绩在 90 分以上的学生的姓名。

select sname

from s

where s.snum in

(select snum

from sc

where sc.score>90 and sc.cnum in

(

select cnum

from c

where c.cname='数据库原理与技术'

))

2.查询工资高于杨平的所有教师的姓名、职称和工资。

```
select tname,title,tsalary
from t
where tsalary>
(
select tsalary
from t
where tname='杨平'
)
```

3.查询比‘22’号最低工资高的其他学院教师信息。

```
select t.*
from t
where t.colnum<>'22' and t.tsalary>any
(select t.tsalary
from t
where t.colnum='22')
```

4.查询与“沈羿”在同一个系学习的学生姓名和性别。

```
select sname,ssex
from s
where dnum=
(
select dnum
from s
where sname='沈羿'
)
```

5.查询没有选修‘ B3503021 ’号课程的学生姓名。

```
select sname
from s
where s.snum not in
(
select sc.snum
from sc
where sc.cnum='B3503021'
)
```

#### 一. 利用多种方法实现下列数据查询操作

1.查询选修了‘B3503021 ’号课程的学生的学号和姓名（三种方法实现，连接、不相关与相关子查询）

```
select s.snum,s.sname
from s,sc
where s.snum=sc.snum and sc.cnum='B3503021'
```



```

select snum,sname
from s
where snum in
(
select snum
from sc
where cnum='B3503021'
)
select snum,sname
from s
where exists
(
select *
from sc
where cnum='B3503021' and sc.snum=s.snum
)

```

2.查询没有选修'B3503021'号课程的学生的学号和姓名（两种方法实现）。

```

select snum,sname
from s
where not exists
(
select *
from sc
where cnum='B3503021' and sc.snum=s.snum
)

```

```

select snum,sname
from s
where snum not in
(
select snum
from sc
where cnum='B3503021'
)

```

3.查询工资高于杨平的所有教师的姓名、职称和工资（三种方法实现，相关与不相关子查询、自身连接）。

```

select tname,title,tsalary
from t
where tsalary>
(
select tsalary
from t
where tname='杨平'
)

```

```

select tname,title,tsalary
from t
where exists
(
select *
from t t1
where t1.tname='杨平' and t.tsalary>t1.tsalary
)

```

```

select t1.tname,t1.title,t1.tsalary
from t t1,t t2
where t2.tname='杨平' and t1.tsalary>t2.tsalary

```

4.查询选修了‘数据库原理与技术’的学生姓名（三种方法）

二.利用多种方法实现下列查询操作

1.查询既选修了课程 c01 又选修了课程 c02 的学生的学号。  
（四种方法：集合、自身连接、相关和不相关子查询）。

```

select snum from sc where cnum='c01'
intersect
select snum from sc where cnum='c02'

```

```

select sc1.snum
from sc sc1,sc sc2
where sc1.snum=sc2.snum and sc1.cnum='c01' and sc2.cnum='c02'

```

```

select snum
from sc
where exists
(
select *
from sc sc1
where sc1.snum=sc.snum and sc.cnum='c01' and sc1.cnum='c02'
)
select snum
from sc
where snum in(select snum from sc where cnum='c02') and cnum='c01'

```

2.查询选修了课程 c01 或者选修了课程 c02 的学生的学号。  
（两种方法）。

```

select distinct snum
from sc
where cnum='c01' or cnum='c02'

```



```
select snum from sc where cnum='c01'
union
select snum from sc where cnum='c02'
```

3.查询选修了课程 c01 没有选修课程 c02 的学生的学号。  
(三种方法: 集合、相关和不相关子查询)。

```
select snum from sc where cnum='c01'
except
select snum from sc where cnum='c02'
```

```
select sc.snum
from sc
where sc.cnum='c01' and not exists
(
select *
from sc sc1
where sc1.cnum='c02' and sc1.snum=sc.snum
)
```

```
select snum
from sc
where snum not in(select snum from sc where cnum='c02') and cnum='c01'
```

三、实现下列查询操作

1.查询学校中所有师生的姓名。

```
select sname
from s
union all
select tname
from t
```

2.查询学校中所有与教师同名的学生姓名。

```
select sname
from s
intersect
select tname
from t
```

3.查询选修了全部课程的学生姓名。

```
select sname
from s
where not exists
(select *
from c
where not exists(
select * from sc where sc.snum=s.snum and sc.cnum=c.cnum))
```

4.查询至少选修了学生'201305010101'选修的全部课程的学生姓名。

```
select sname from s where snum in
(
select x.snum from sc x,sc y
where x.cnum=y.cnum and y.snum='201305010101'
group by x.snum having COUNT(*)=(select COUNT(*) from sc where snum='201305010101')
)
```

#### 一. 基本数据更新

1.将新生（学号：200501010101；姓名：伍俊；性别：男；出生日期：1993-08-08；电话：88485561；系编号：0501）的记录插入学生表，并为其选：c01、c02与c03三门课程。

```
insert into s(snum,sname,ssex,sbirth,sphone,dnum)
values('200501010101','伍俊','男','1993-08-08','88485561','0501')
insert into sc(snum,cnum)
values('200501010101','c01')
insert into sc(snum,cnum)
values('200501010101','c02')
insert into sc(snum,cnum)
values('200501010101','c03')
```

2.建立每门课程的课程号，平均成绩与选课人数表。并将数据存入到该表中。

```
create table ccc
(
cnum nchar(12) primary key,
avgscore int,
number int
)
select * from ccc
insert into ccc
select cnum,AVG(score),COUNT(*)
from sc
group by cnum
```

3.事后发现雷吉平考试作弊，将其所有成绩改为0分。

```
update sc
set score=0
where snum=(select snum from s where sname='雷吉平')
```

4.将所有教师的工资增加1000元。

```
update t
set tsalary=tsalary+1000
```



5.将计算机科学与技术专业全体学生的成绩置零。

```
update sc
set score=0
where snum in (select snum from s where dnum=(select dnum from d where dname='计算机科学与技术'))
```

6.将吴达胜任命为‘信息工程学院’院长。

```
update coll
set tnum=(select tnum from t where tname='吴达胜')
where collname='信息工程学院'
```

7.将‘信息工程学院’院长的工资降为该学院的平均工资。

```
update t
set tsalary=(select AVG(tsalary) from t where collnum=(select collnum from coll where collname='信息工程学院'))
where tnum=(select tnum from coll where collname='信息工程学院')
```

8.在学生表中增加 flag 标志列，如果该同学选课，则 flag 置 1,如果未选课，则 flag 置 0。

```
alter table s add flag nchar(1)
select * from s where flag=0
update s
set flag=1
where snum in (select snum from sc)
update s
set flag=0
where snum not in (select snum from sc)
```

9.学号为 S030404 的学生由于退学，删除该学生记录及其选课记录。

```
delete from sc
where snum='S030404'
delete from s
where snum='S030404'
```

10.删除低于平均工资的老师记录以及任课记录。

```
delete from tc
where tnum in (select tnum from t where t.tsalary<(select AVG(t.tsalary) from t))
delete from t
where tnum in (select tnum from t where t.tsalary<(select AVG(t.tsalary) from t))
```

11.删除计算机科学与技术专业全体学生的选课记录。

```
delete sc where snum in (select snum from s where dnum=(select dnum from d where dname='计算机科学与技术'))
```

12.删除没有选课的学生信息。

```
delete s where snum not in (select snum from sc)
```

二. 技术提升（选做）

1.将信息工程学院工资大于 6000 元的教师信息备份到 a 表中

```
select * into aaa from t where tsalary>(select AVG(tsalary) from t) and collnum=(select collnum from coll where collname='信息工程学院')
```

2.将学生的姓名、学号和选修课程的总学分备份到 b 表中。



```

select s.snum,sname,SUM(credit)
from s,sc,c
where s.snum=sc.snum and sc.cnum=c.cnum
group by s.snum,sname

```

## 一. 视图

1.创建一个视图 VIEW\_S，显示成绩等于 100 分的学生的学号、姓名、课程名和成绩。

```

create view view_s
as select s.snum,sname,cname,score
from s,sc,c
where s.snum=sc.snum and sc.cnum=c.cnum and score='100'
select *
from view_s

```

2.建立‘计算机科学与技术’专业选修了‘数据库原理与技术’这门课程的学生视图。(学号，姓名，成绩)

```

create view view_ss
as select s.snum,sname,score
from s,d,c,sc
where s.snum=sc.snum and sc.cnum=c.cnum and c.cname='数据库原理与技术' and
d.dnum=s.dnum and dname='计算机科学与技术'
select *
from view_ss

```

3.建立‘计算机科学与技术’专业选修了‘数据库原理与技术’这门课程且成绩在 90 分以上的学生视图。(学号，姓名，成绩)

要求创建基于视图的视图。

```

create view view_sss
as select s.snum,sname,score
from s,d,c,sc
where s.snum=sc.snum and sc.cnum=c.cnum and c.cname='数据库原理与技术' and
d.dnum=s.dnum and dname='计算机科学与技术' and score>=90

```

```

select *
from view_sss

```

4.建立‘信息管理与信息系统’专业的学生的学号和平均成绩的视图。

```

create view view_avg
as select s.snum , AVG(score) as avgs
from s,sc,d
where s.snum=sc.snum and d.dnum=s.dnum and d.dname='信息管理与信息系统'
group by s.snum
select *
from view_avg

```



5.将学院名称及该学院教师的平均工资定义为一个视图。

```
create view view_coll
as select collname,AVG(tsalary) as avgtsalary
from coll,t
where coll.collnum=t.collnum
group by collname
select *
from view_coll
```

6.建立一视图显示教师的工号，姓名和年龄。（可考虑用 datediff()函数）

```
create view view_t
as select tnum,tname,DATEDIFF(YEAR,2016,t.tbirth)as nianlin
from t
select *
from view_t
```

## 二.权限

7.在 SSMS 中设置验证模式为 SQL Server 和 Windows 混合认证。

8.在 SSMS 中建立一个以你名字拼音命名的登录名,然后再将该登录名映射到 master 及用户数据库。

以下题目要求用数据控制语言(grant、revoke)实现:

9.将创建数据库，创建表的权限授予“你”。

```
grant create database,create table
to wuyuanshuang
```

10.将对学生表查询和修改学生电话号码的权限授予“你”，并且“你”可以进行权限传播。

```
grant select,update (sphone)
on s
to wuyuanshuang
```

with grant option

11.“你”具有对 SC 表的所有权利（读、插入、修改、删除）。

```
grant select,insert,update,delete
on sc
to wuyuanshuang
```

12.将“你”对 sc 表的修改权限收回来。

```
revoke update
on sc
from wuyuanshuang
```

13.将“你”对修改学生电话号码的权限收回来。

```
revoke update (sphone)
on s
from wuyuanshuang cascade
```



### 1.变量与函数的应用

(1)定义两个变量，将 sc 表中的最高分和最低分分别赋值给这两个变量，并显示变量值。

```
declare @max int,@min int
select  @max=(select MAX(score) from sc)
select  @min=(select MIN(score) from sc)
print @max
print @min
```

(2)获取当前系统日期和当前月份。

```
print getdate()
print datepart(month,getdate())
```

(3)查询数据库所在的计算机名。（可以用函数，也可以用全局变量）

```
select @@SERVERNAME,HOST_NAME()
```

### 2.while 循环

(1)求整数 0 到 100 的和，并且屏幕输出。（while 循环）

```
declare @i int,@sum int
set @i=1
set @sum=0
while @i<=100
begin
    set @sum=@sum+@i
    set @i=@i+1
end
print @sum
go
```

(2)在数据库中创建一个表 nums（number int primay key）,并为这张表从 1 到 1000，填充 1000 行数据。

```
create table nums
(num nvarchar(10) primary key)
declare @i int
set @i=1
while @i<=1000
begin
    insert into nums values(@i)
    set @i=@i+1
end
```

### 3.if...else

从 T（教师表）中求出‘信息工程学院’教师的平均工资，如果平均工资大于 6000，则输出‘工资高于平均水平’，否则输出‘工资低于平均水平’。

```
if (select AVG(tsalary) from t,coll where t.collnum=coll.collnum and coll.collname='信息工程学院')>6000
begin
    print '工资高于平均水平'
```



```

end
else
begin
    print '工资低于平均水平'
end

```

#### 4.case

(1).从教师表 T 中，选取 tnum,tname,tsex,title，如果 Title 为‘教授’,则显示‘正高级’,如果为‘副教授’，则显示‘副高级’，  
如果为‘讲师’，则显示‘中级’，如果为助教，则显示‘初级’。

```

select tnum,tname,tsex,
case
when title='教授' then '正高级'
when title='副教授' then '副高级'
when title='讲师' then '中级'
when title='助教' then '初级'
end as "title"
from t

```

(2).查询教师的姓名,性别和年龄(表中为 Tbirth 出生日期),年龄小于 30 岁,显示年龄偏小,30 到 55 岁之间(包括边界值),

显示年龄居中,大于 55 岁,显示年龄偏大。(提示:使用 datediff()和 getdate()函数)

```

select tname,tsex,
case
    when datediff(YY,tbirth,GETDATE())<30 then '年龄偏小'
    when datediff(YY,tbirth,GETDATE())>=30 and datediff(YY,tbirth,GETDATE())<=55 then '年龄居中'
    when datediff(YY,tbirth,GETDATE())>55 then '年龄偏大'
end as "old"
from t

```

--(3)从 sc 表中查询所有同学选课成绩情况，将百分制转换为五分制：[null]:缺考；[0,60):不及格；[60,70):及格；

--[70,80): 中等；[80,90):良好；[90,100): 优秀。

```

select snum,cnum,
case
    when score IS null then '缺考'
    when score<60 then '不及格'
    when score>=60 and score<70 then '及格'
    when score>=70 and score<80 then '中等'
    when score>=80 and score<90 then '良好'
    when score>=90 then '优秀'
end as "等级"
from sc

```

5..使用 if exists()语句判断于森同学是否选课，如果已选课，输出‘该同学已选课’，否则，输出该同学未选课。

```

if exists(select * from sc where snum=(select snum from s where sname='于森'))
begin

```



```

        print '该同学已选课'
    end
else
begin
    print '该同学未选课'
end

```

6..延迟 10 秒，查询学生的基本信息。

```

waitfor delay '00:00:10'
select *
from s

```

## --一、定义简单的存储过程

--1 创建一个加密存储过程，从学生选课数据库中查询学号,姓名,课程名和成绩。执行该存储过程，并用 sp\_helptext 查看该存储过程。

```

if exists(select * from sysobjects
where name='p_s' and type='p')
drop proc p_s
go
create proc p_s
with encryption
as
select s.snum,sname,cname,score
from s,c,sc
where s.snum=sc.snum and c.cnum=sc.cnum
go
exec p_s
go
sp_helptext p_s
go

```

## --二、定义带输入参数的存储过程

--2.创建一存储过程，完成向课程表（C）中插入一条记录，记录值通过输入参数传递到存储过程中，并执行该存储过程

```

if exists(select * from sysobjects
where name='p_c' and type='p')
drop proc p_c
go
create proc p_c @cnum nchar(12),@cname nvarchar(50),@credit numeric(5,1)
as
begin
insert into c(cnum,cname,credit) values(@cnum,@cname,@credit)
end

```



```

go
exec p_c 'c111122',3.0
go
sp_helptext p_c
go
--3.创建一存储过程，显示某一同学的平均成绩，学号通过输入参数传递到存储过程中，并执行该存储过程。

```

```

if exists(select * from sysobjects
where name='sc_av' and type='p')
drop proc sc_av
go
create proc sc_av @snum nchar(12)
as
begin
select AVG(score) from sc where snum=@snum
end
go
exec sc_av '10103060209'
go

```

--4.创建一存储过程，在删除学院(coll)表时,同时删除该学院的教师及其任课记录，该学院的系，  
--所属系的学生以及选修记录，输入参数为学院编号。并执行该存储过程。

```

select * from coll
if exists(select * from sysobjects
where name='de_coll' and type='p')
drop proc de_coll
go
create proc de_coll @collnum nchar(2)
as
begin
delete from tc where tnum in (select tnum from t where collnum=@collnum)
delete from sc where snum in (select snum from s,d where collnum=@collnum and s.dnum=d.dnum)
delete from s where snum in (select snum from s,d where collnum=@collnum and s.dnum=d.dnum)
delete from t where collnum=@collnum
delete from d where collnum=@collnum
delete from coll where collnum=@collnum
end
exec de_coll '01'

```

--三、(信管专业选做)创建带返回参数（output）的存储过程

--5.创建一个存储过程，该存储过程接受外部传入的 snum,在 sc 表中查找该 snum 的选课记录，然后输出该学生所选课程的平均成绩（输出参数）。

```

create proc av_sc @snum nchar(12) ,@chenji int output
as
select @chenji=AVG(score) from sc where snum=@snum
go

```

--6. 执行上一题的存储过程,输出测试结果(如果平均成绩在 90-100 之间为'优秀',在 80-89 之间为'良好',



--在 70-79 之间为'中等', 在 69-70 之间为'及格', 在 0-59 之间为'不及格')

```
declare @chenji int
```

```
exec av_sc 's030101',@chenji out
```

```
if(@chenji>=80 and @chenji<90) print '良好'
```

```
if(@chenji>=70 and @chenji<80) print '中等'
```

```
if(@chenji>=60 and @chenji<70) print '及格'
```

```
if(@chenji<60) print '不及格'
```

--1.(1)在选课表 SC 上创建触发器, 当向该表中插入记录时, 若学生选课门数超过 5 门时, 则向客户端发出警告信息,

-- 同时将插入记录自动删除。

-- (2)验证该触发器。

```
if exists(select * from sysobjects
```

```
where name='ins_sc' and type='tr')
```

```
drop trigger ins_sc
```

```
go
```

```
create trigger ins_sc
```

```
on sc
```

```
after insert
```

```
as
```

```
if((select count(sc.snum) from Inserted,sc where sc.snum=Inserted.snum)>5)
```

```
begin
```

```
print '学生选课门数超过 5 门,不再插入'
```

```
rollback transaction
```

```
end
```

```
insert into sc
```

```
values ('201405070530','01021043','99')
```

--2.(1)在 sc 表上创建触发器, 当删除学生的选课记录时, 首先检查要删除的行数, 如果是删除多行, 则返回错误信息。

--(2)验证该触发器。

```
if exists(select * from sysobjects
```

```
where name='del_sc' and type='tr')
```

```
drop trigger del_sc
```

```
go
```

```
create trigger del_sc
```

```
on sc
```

```
after delete
```



```

as
if((select count(*) from deleted)>1)
begin
print '删除多行,错误'
rollback transaction
end

```

```

delete from sc
where snum='201405070530'

```

--3.--(1)创建一触发器,在对学生的姓名进行更新时，显示更新前后的学生姓名。

--(2)更新自己的姓名验证该触发器是否生效。

```

if exists(select * from sysobjects
where name='up_s' and type='tr')
drop trigger up_s
go

```

```

create trigger up_s
on s
after update
as
if update(sname)
begin
declare  @a nvarchar(50),@b nvarchar(50)
set @a=(select sname from deleted)
set @b=(select sname from inserted)
print @a
print @b
end

```

```

update s
set sname='吴双双'

```

```

where snum='201405070530'

```

--4.--(1)建立一个触发器,在向 “ 教师 ” 表插入记录时引发该触发器,检查所插入元组中教师的工资是否大于 3000 元且小于 10000 元,若不是则提示用户“工资必须大于 3000 元 ,小于 10000 元”,并且回滚事务。

--(2)插入一个老师信息，设定工资为 2800 元，测试该触发器。

```

if exists(select * from sysobjects
where name='tr_inst' and type='tr')
drop trigger tr_inst
go

```

```

create trigger tr_inst
on t
after insert
as

```

```
if not exists(select * from inserted where tsalary>3000 and tsalary<10000)
begin
rollback tran
raiserror('教师的工资必须大于 3000 元小于 10000 元! ',16,1)
end
```

```
insert into t
values('1111111111','张三','男','2013-2-2','null','null','null','null','null','null','2800','26')
```

--5.--(1)创建一触发器实现级联删除。在删除学院(coll)表时,同时删除该学院的教师及其任课记录, 该学院的系,

--所属系的学生以及选修记录。

--(2)删除某一学院, 验证该触发器。

```
if exists(select * from sysobjects
where name='del_coll' and type='tr')
drop trigger del_coll
go
```

```
create trigger del_coll
on coll
instead of delete
as
begin
delete from sc where snum in(select snum from s where dnum in(select dnum from d,deleted where
d.collnum=(select collnum from deleted)))
delete from tc where tnum in(select tnum from t where t.collnum=(select collnum from deleted))
delete from s where dnum in(select dnum from t where collnum=(select collnum from deleted))
delete from d where d.collnum=(select collnum from deleted)
delete from t where t.collnum=(select collnum from deleted)
end
```

```
delete from coll
where collnum='22'
```