# 20171206第十四周上机

完成以下任务, 最终结果保存到文件”学号姓名第十四周作业.sql”中，将结果文件打成压缩包，文件名为“学号姓名第十四周作业.rar”，上传到FTP上UPLOAD文件夹中我们的课程子文件夹内。

首先执行week14\_base.sql和week14\_datainsert.sql，完成库、表的创建，以及数据的导入工作。

1. 设计一个AFTER联动触发器，当在DEPARTMENT表中修改院系编号时，联动修改STUDENT表中相关学生的院系编号。将法政学院的编号从1改成10，观察触发器的执行效果。

答：

use edu

go

create trigger tr\_dno

on department

for update

as

if update(dno)

begin

update student set dno=(select dno from inserted) where dno =(select dno from deleted)

end

return

---- another query

use edu

go

select \* from department

go

select \* from student

go

update department set dno=10 where dno=1

go

select \* from department

go

select \* from student

go

1. 设计一个INSTEAD OF触发器，在向STUDENT\_COURSE表中插入一条记录时，判断选修人数是否达到COURSE表中规定的相关课程的人数上限，如果达到，则不允许插入这条记录，并给出“人数已到上限，无法选修的提示”

答：

use edu

go

create trigger tr\_upperlimit

on student\_course

instead of insert

as

print 'instead of trigger '

declare @cur\_ct int, @limit int

Declare @tsno char(8),@tcno char(10),@tscore tinyint

select @tsno=sno,@tcno=cno,@tscore=score from Inserted

select @cur\_ct= COUNT(sno) from student\_course where cno=@tcno

select @limit = stulimits from course where cno=@tcno

if @cur\_ct >= @limit

begin

print '课程人数已达上限，无法选修'

end

else

begin

insert into student\_course values(@tsno,@tcno,@tscore)

print '选修成功'

end

return

---- another query

use edu

go

insert into student\_course values('20101009','6',NULL)

go

select \* from student\_course

update course set stulimits=20 where cno='6'

insert into student\_course values('20101009','6',NULL)

go

select \* from student\_course

1. 设计一个触发器，在向STUDENT\_COURSE表中插入一条记录时，判断该学号在学生表中是否存在，以及该课程号在课程表中是否存在，如果不满足条件，则不允许插入该记录。

答：

use edu

go

--drop trigger tr\_upperlimit

--go

create trigger tr\_snocnocheck

on student\_course

for insert

as

Declare @tsno char(8),@tcno char(10)

select @tsno=sno,@tcno=cno from Inserted

if NOT exists(select sno from student where sno=@tsno)

begin

print '查无此人，无法选修'

ROLLBACK TRANSACTION

return

end

else

if not exists(select cno from course where cno=@tcno)

begin

print '查无此课，无法选修'

ROLLBACK TRANSACTION

return

end

else

print '选修成功'

return

---- another query

use edu

go

insert into student\_course values('20101033','6',90)

go

insert into student\_course values('20101010','9',90)

go

insert into student\_course values('20101010','6',90)

go

1. 设计一个触发器，在STUDENT\_COURSE表中修改一条记录时，如果修改的是学号或者课程号，则禁止这种修改。

答：

use edu

go

create trigger tr\_upcheck

on student\_course

for update

as

IF UPDATE(sno) OR UPDATE(cno)

BEGIN

PRINT '违背数据的一致性，禁止修改'

ROLLBACK TRANSACTION

END

else

print '成绩成功修改'

return

---- another query

use edu

go

select \* from student\_course

go

update student\_course set sno='20101011' where sno='20101010'

go

1. 设计一个触发器，在向STUDENT\_COURSE表中插入一条记录时，如果分数不是NULL，也不在【0，100】之内，则禁止插入该条记录。

答：

use edu

go

create trigger tr\_scorecheck

on student\_course

for insert

as

Declare @tscore tinyint

select @tscore=score from Inserted

if @tscore is not NULL and (@tscore < 0 OR @tscore > 100)

begin

print '课程成绩非法'

ROLLBACK TRANSACTION

end

else

print '成绩插入成功'

return

---- another query

use edu

go

select \* from student\_course

go

insert into student\_course values('20101010','7',90)

go

select \* from student\_course

go