**注：本作业语法为sql，截图时间在第五题的第二张截图中显示**

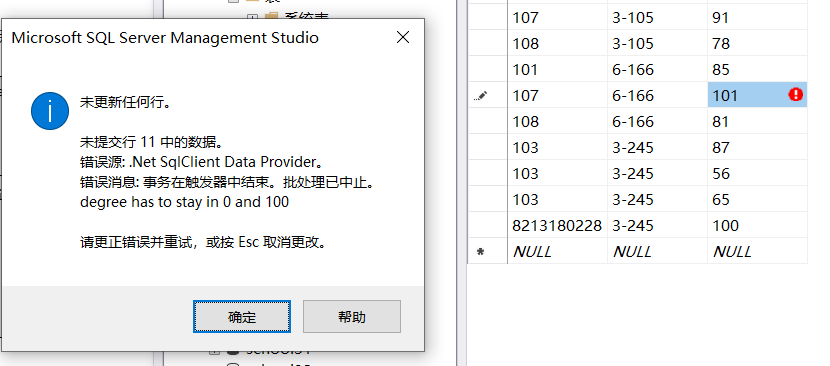
**在建立的学生表、课程表和选课表上完成如下操作。**

1：在选课表中定义一个更新和插入的触发器，保证插入的成绩在0-100之内

代码：

|  |
| --- |
| drop trigger tri\_score\_degree2  use school  go  create trigger tri\_score\_degree2 on score for insert,update as  if exists(select \* from inserted where degree<0 or degree>100)  begin  print'degree has to stay in 0 and 100'  rollback  end  go |

截图：

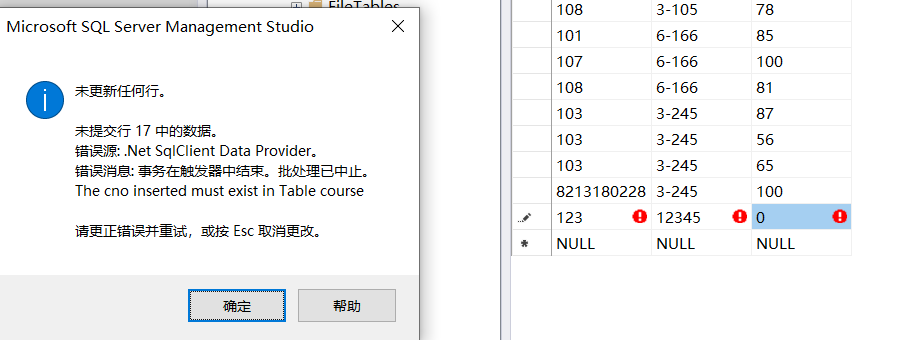


2:在选课表sc表中定义一个保证课程号在课程表的课程范围内的插入触发器

代码：

|  |
| --- |
| use school  go  create trigger tri\_score\_cnoInCourse3 on score for insert as  if not exists(select \* from dbo.course where cno in(select cno from inserted))  begin  print'The cno inserted must exist in Table course'  rollback  end  go |

截图：

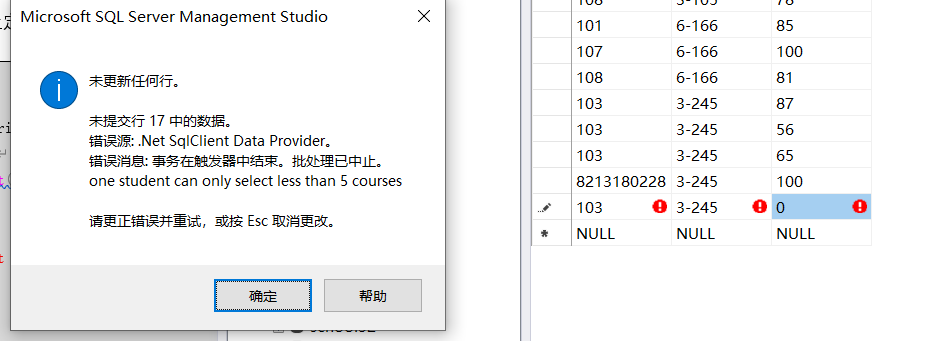


3:在选课表sc表上定义一个限制学生选课门数不能超5门的触发器

代码：

|  |
| --- |
| use school  go  create trigger tri\_score\_less5course on score for insert as  declare @num int  select @num=count(\*) from score where sno in(select sno from inserted)  if(@num>5)  begin  print'one student can only select less than 5 courses'  rollback  end  go |

截图：

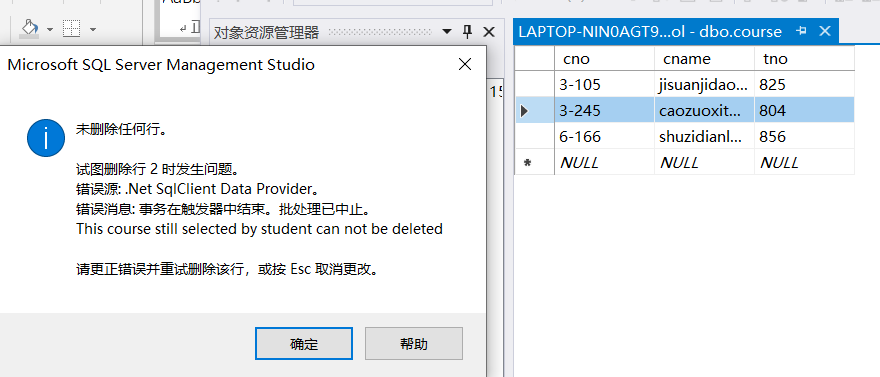


4:在课程表course表上定义一个限制删除有学生选课程的触发器.

代码：

|  |
| --- |
| use school  go  create trigger tri\_course\_NotDeletedCourseSelected on course for delete as  if exists(select \* from deleted where cno in(select cno from score))  begin  print'This course still selected by student can not be deleted'  rollback  end  go |

截图：



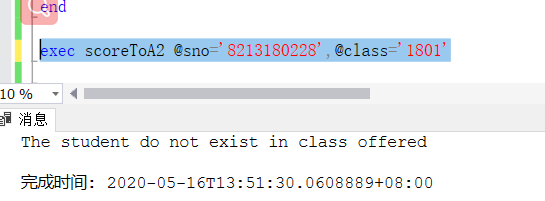
5：建立存储函数，它带有两个参数：输入你的学号和你的班级号(**演示的时候，一定要是你的学号与班级啊，不是你的不行**)。该函数可以将指定的某班学生的成绩改为A+。如果学生不在该班级，请定义例外处理。

代码：

|  |
| --- |
| alter table student alter column sno varchar(10)  alter table score alter column sno varchar(10)  go  create procedure scoreToA2  @sno varchar(10),  @class varchar(10)  as  if(@class in(select class from student where sno=@sno))  begin  update score set degree=100 where sno=@sno  end  else  begin  print'The student do not exist in class offered'  end  exec scoreToA2 @sno='8213180228',@class='1802' |

截图：





6：给teacher表编写一个（before-delete)触发器,当要在teacher表中删除一个老师，将触发该触发器。在触发器中将判断老师是否已经授课？如果已经安排某位教师授课，它将激发一个例外，把无法删除的信息返回用户。

代码：

|  |
| --- |
| use school  go  create trigger tri\_teacher\_beforeDeletedTeacher on teacher for delete as  if exists(select \* from deleted where tno in(select tno from course))  begin  print'This teacher still teach course can not be deleted'  rollback  end  go |

截图：

