# 实验九 触发器的创建与使用

**一、实验目的**

本实验的目的是使学生进一步掌握SQL Server触发器的创建及使用方法，加深SQL触发器的理解。通过对数据的更新操作体会其触发器的作用。

**二、实验准备**

结合课堂教学内容，了解触发器的相关知识，掌握触发器的定义，理解触发器的使用方法及其特点。

**三、实验要求**

1. 掌握T-SQL创建触发器语句Create trigger的使用。

2. 完成所规定的触发器定义要求。

3. 通过数据查询检查触发器的使用效果。

4. 注意操作结果的截图与保存，供撰写实验报告使用。

**四、实验内容**

1. 在班级表class中增加班级人数（c\_total）字段。

2. 为学生表(student)创建INSERT触发器t\_inst\_stu：新增一名学生时，若其班级编号非空，则将班级表（class）中相应班级的人数（c\_total）自动加1。

3. 为学生表(student)创建DELETE触发器t\_dele\_stu：删除一名学生时，若其班级编号非空，则将班级表（class）中相应班级的人数（c\_total）自动减1。

4. 为学生表(student)创建UPDATE触发器t\_update\_stu: 当某学生所在班号发生变化时（即调到另一班级后），将其原先所在班级的人数（c\_total）减1，将新调入的班级班级的人数（c\_total）加1。

5. 体会触发器的作用。

**五、实验步骤**

在查询分析器中完成实验内容。

1. 使用alter table语句在班级表class中增加班级人数（c\_total）字段，默认值（DEFAULT）是0；

2. 为学生表(student)分别创建：

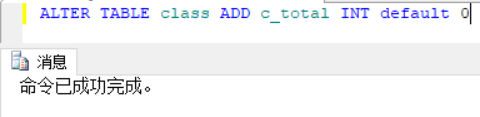
* INSERT触发器t\_inst\_stu
* DELETE触发器t\_dele\_stu
* UPDATE触发器t\_update\_stu

3. 体会触发器的作用：

1) 查看班级表（class）

2）对学生表(student)分别插入（INSERT）、删除（DELETE）和修改（UPDATE）元组

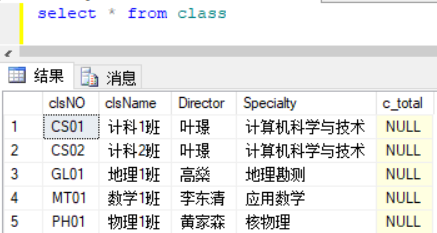
3）再次查看班级表（class），检查其数据的变化情况，体会触发器的作用。



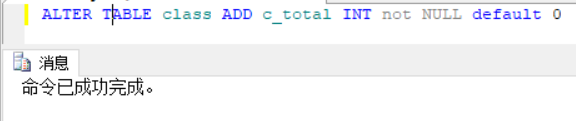
1. 直接创建总是得到这样的结果

全为NULL

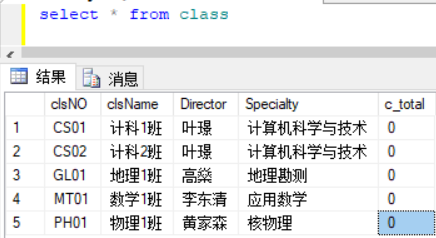
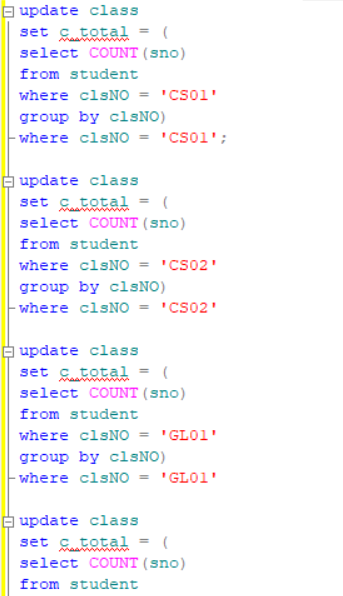
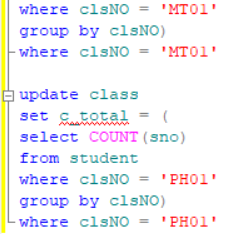
我们将创建代码改为



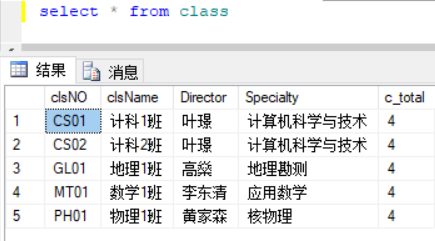
于是结果正确了



接下来，我们根据实际情况对其进初始化

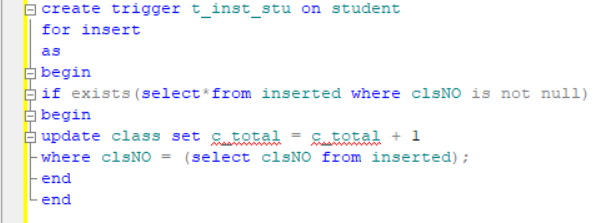
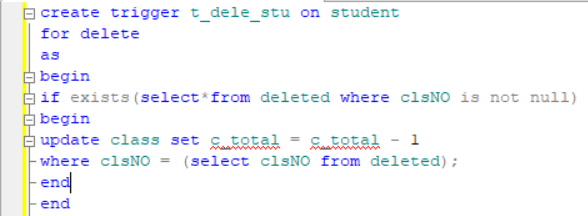


此时，再进行查询，发现已经初始化完毕

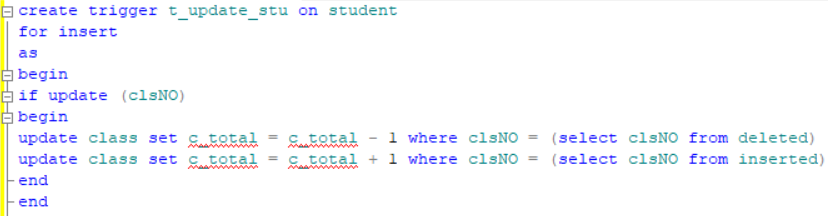


2.

3.



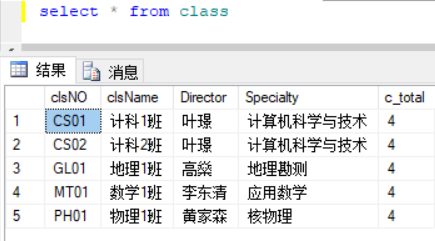
4．



接下来，我们检查触发器作用是否正确

Class表进行初始化后的情况为

接下来，我们对student表进行插入



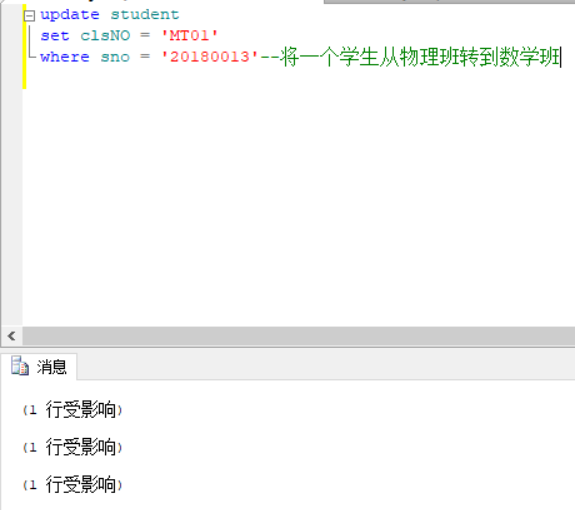
对student表进行删除



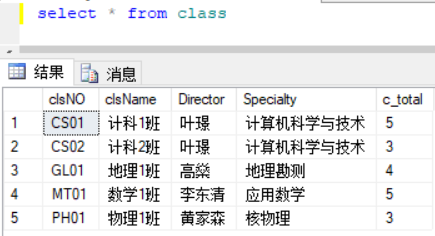
最后，对一个物理班的学生进行换班，换到数学班



全部完成后，我们对class表进行检查



增加了学生的CS01班人数从4减到3



删除了学生的CS02班人数从4加到5

转走了学生的PH01班人数减1

转来了学生的MT01班人数加1

结果正确

# 实验十 存储过程的创建与使用

**一、实验目的**

本实验的目的是使学生进一步掌握SQL Server存储过程的创建及使用方法，加深对SQL存储过程的理解。通过对存储过程的调用体会其作用。

**二、实验准备**

结合课堂教学内容，了解存储过程的相关知识，掌握存储过程的定义，理解存储过程的调用及其参数的使用方法。

**三、实验要求**

1. 掌握T-SQL创建触发器语句Create procedure的使用。

2. 完成所规定的存储过程定义要求。

3. 调用（EXEC）所创建的过程，检查过程的执行结果。

4. 注意操作结果的截图与保存，供撰写实验报告使用。

**四、实验内容**

1、创建一个不带参数的存储过程p\_stu\_info1，实现对满足要求的学生基本信息的查询。

要求：所有年龄<21岁的男同学

1. 创建一个带有参数的存储过程p\_stu\_info2，实现对满足要求的学生基本信息的查询。

要求：输入参数为学号，与指定学号的学生同龄的所有同学。

1. 创建一个存储过程p\_stu\_info3,根据输入的学号，查询某学生的基本信息。

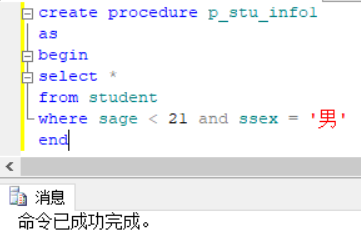
要求：输入参数为学号。

1. 创建一个存储过程p\_stu\_grade,根据输入的学号，返回其选课及其成绩。

要求：输入参数为学号。

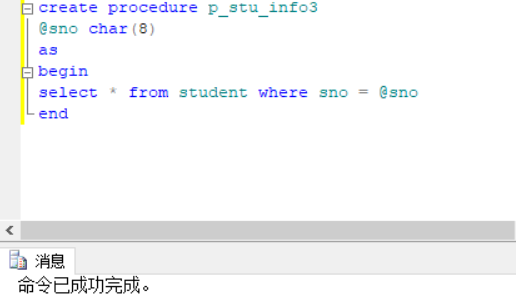
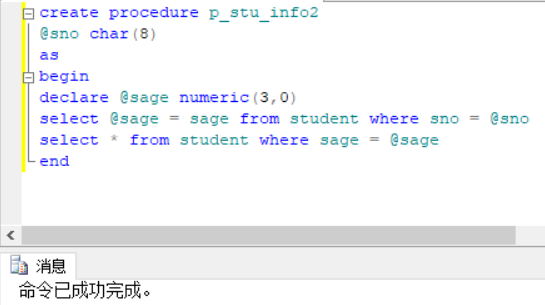
5.使用SQL语句分别执行p\_stu\_info1、p\_stu\_info2、p\_stu\_info3和p\_stu\_grade，并查看显示结果。

1.



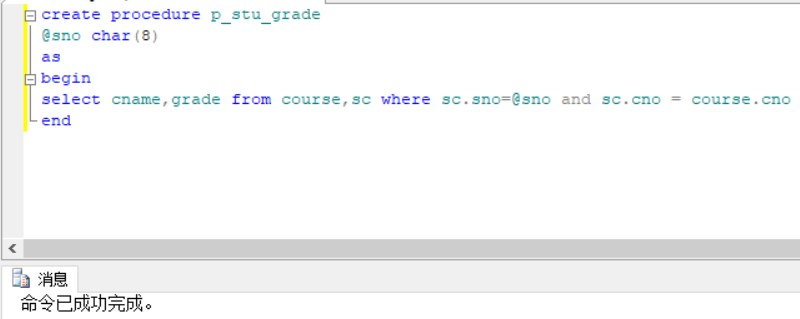
2.

3.



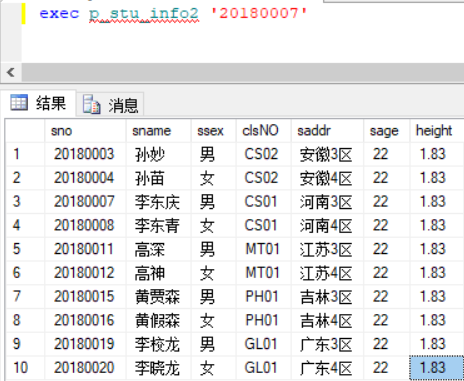
4.

最后，挨个进行验证



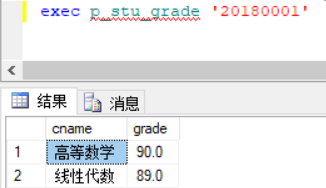
**1.运行**p\_stu\_info1，得到了所有年龄小于21岁的男同学

2运行p\_stu\_info2，得到了和学号为‘20180007’的同学年龄相同的同学的信息。



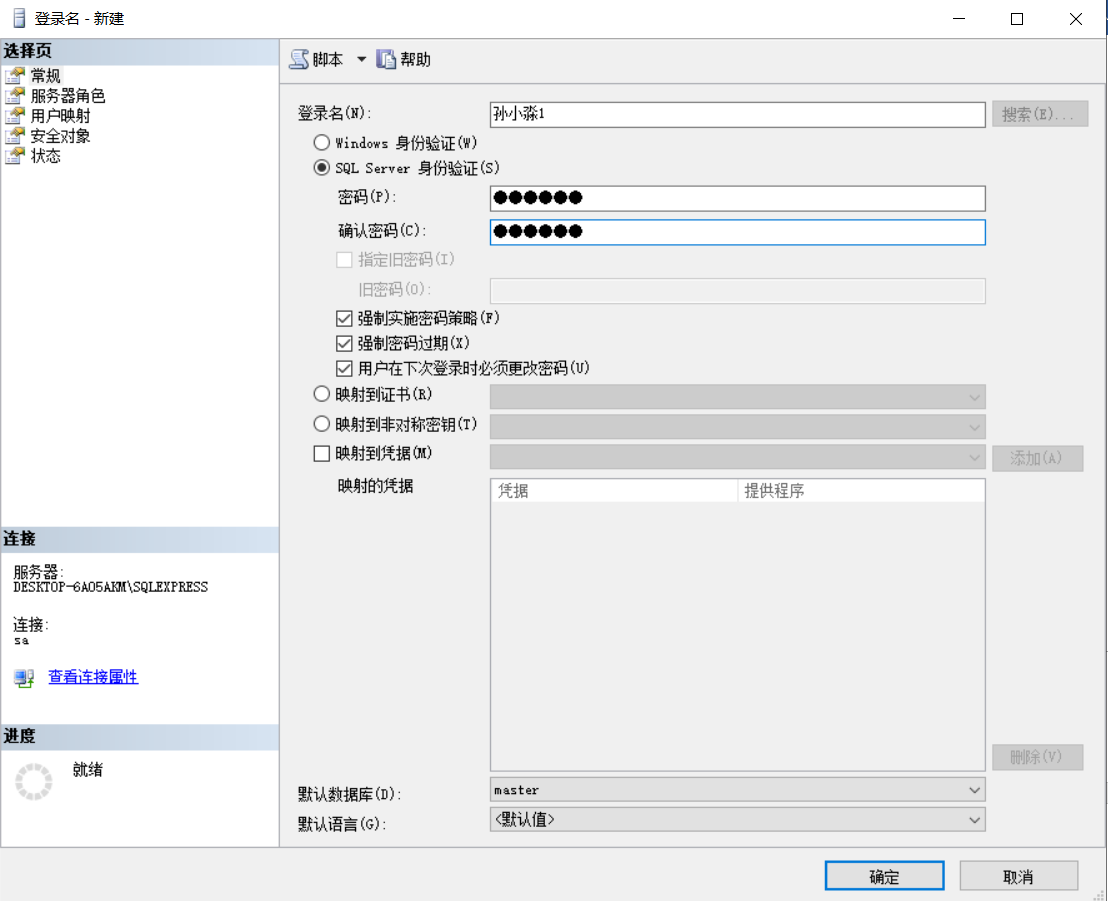
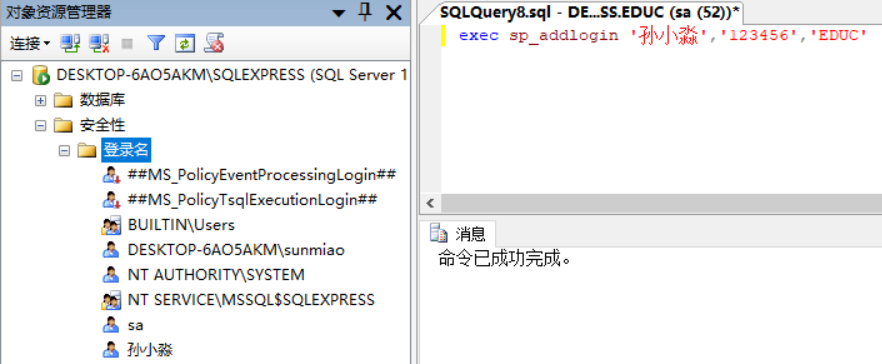
3.运行 p\_stu\_info3，得到了学号为20180007的学生所有的信息

4. 运行 p\_stu\_grade，得到了学号为20180001的学生的选课及成绩

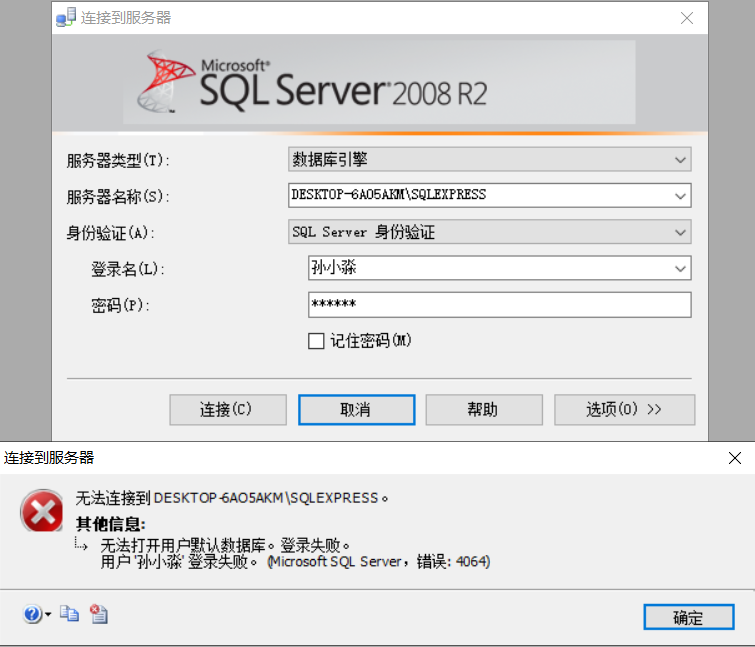


**五、实验步骤**

在查询分析器中完成实验内容。首先，

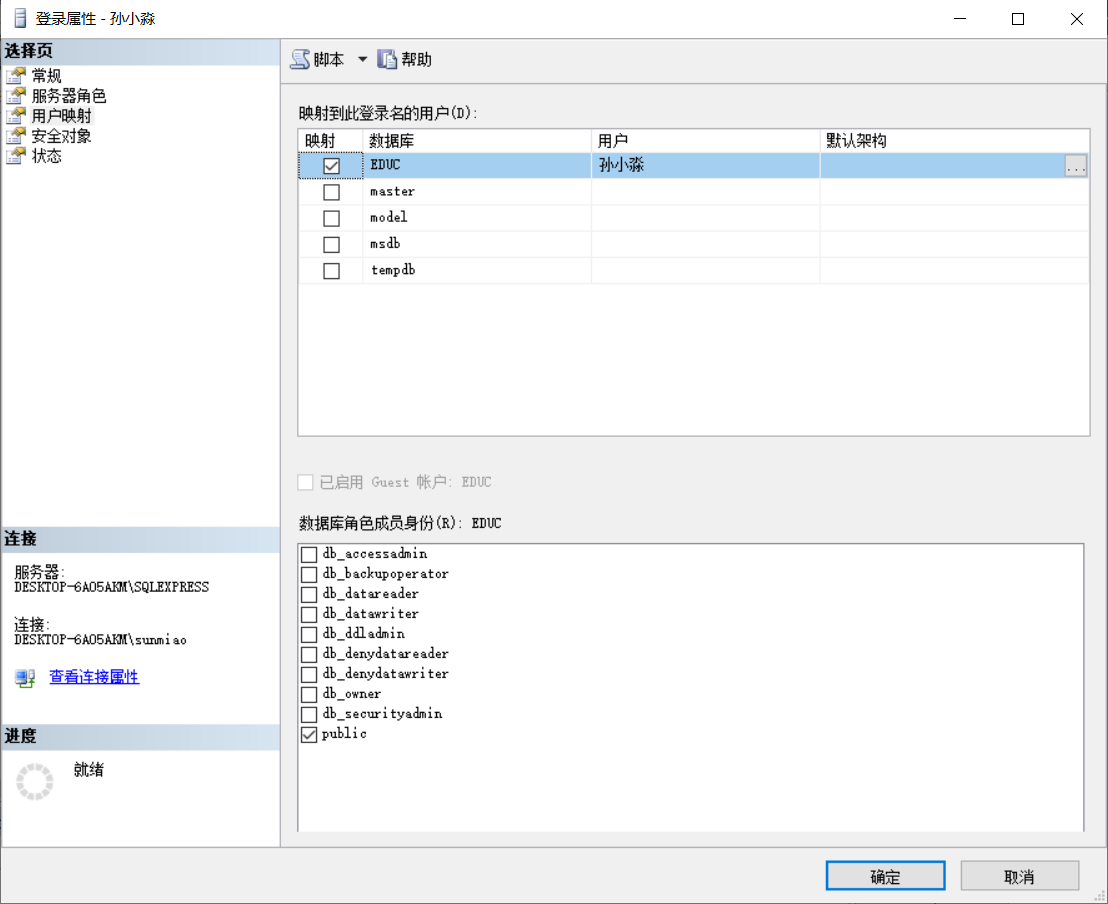


尝试登录用语句创建的用户“孙小淼”时，报错，查阅资料，通过

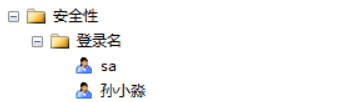
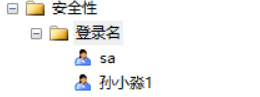


安全—登录名，双击孙小淼—用户映射，选择映射的数据库educ,再次登录即可进入

成功进入



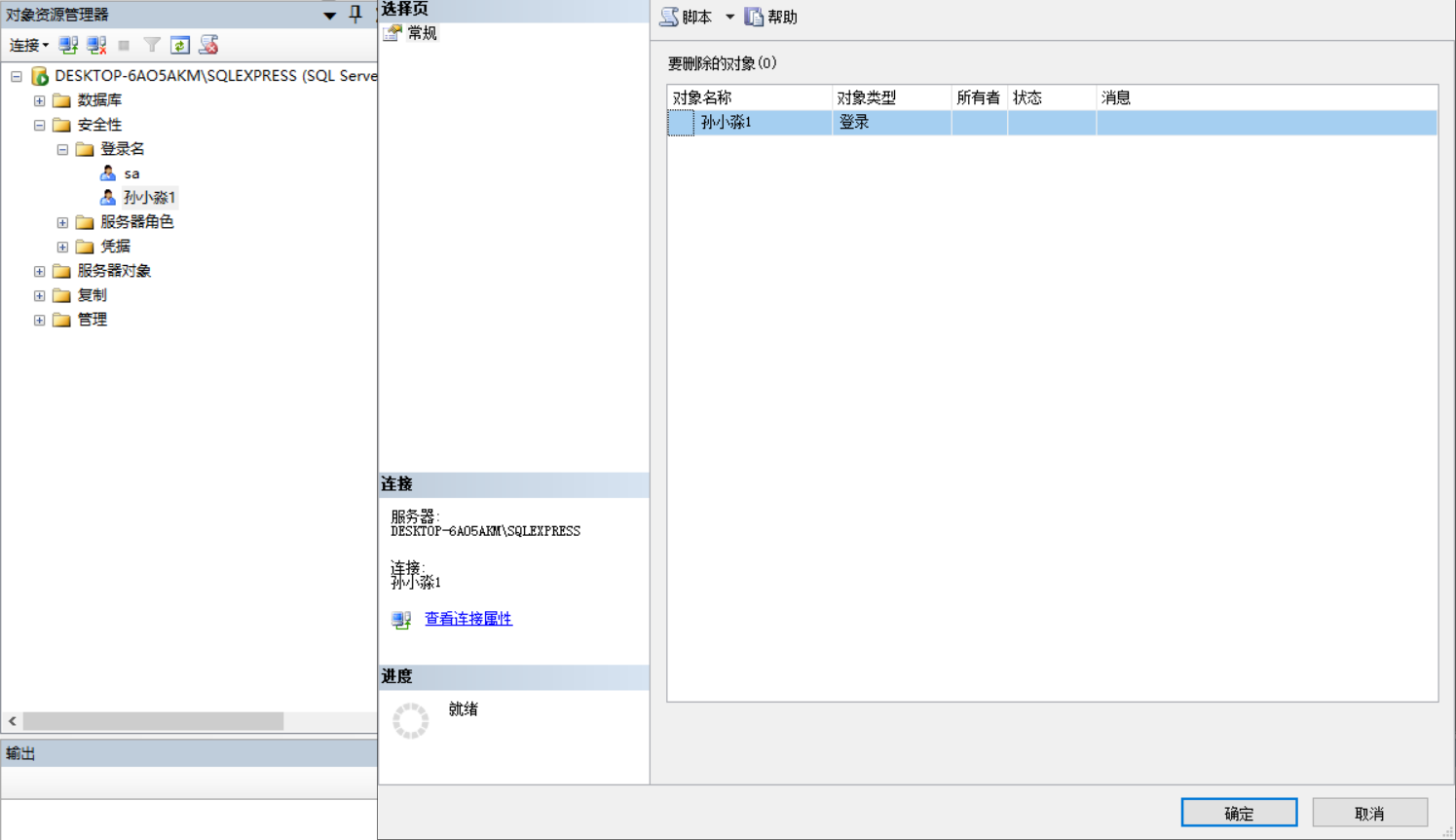
而尝试登录用向导创建的用户孙小淼1时，没有问题，顺利进入



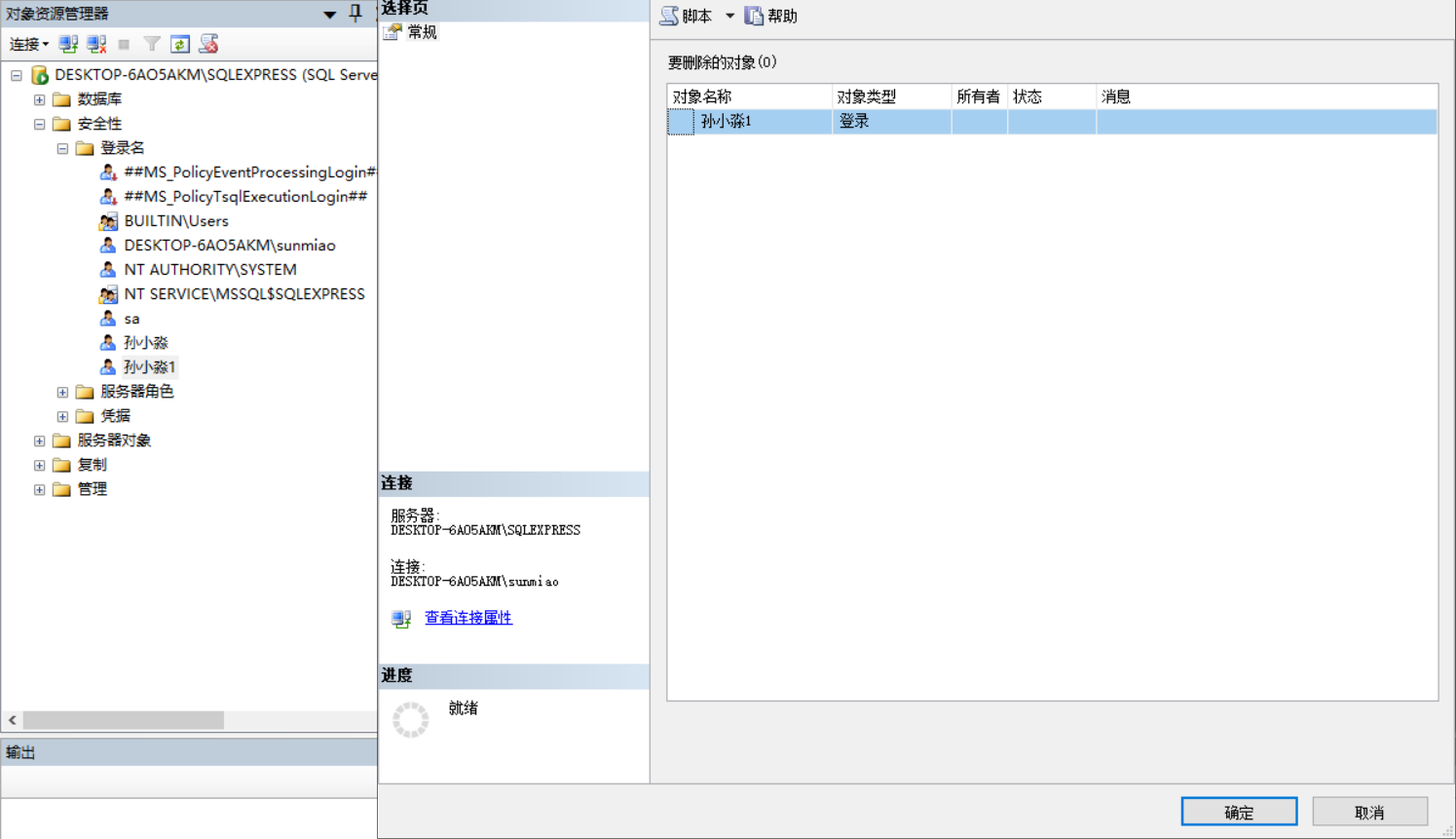
尝试用向导删除



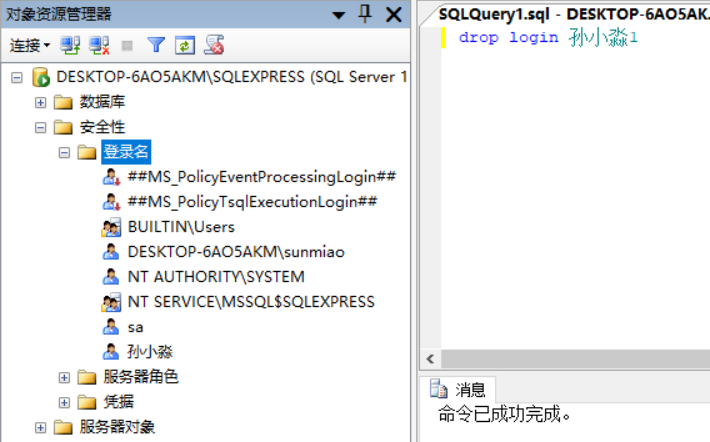
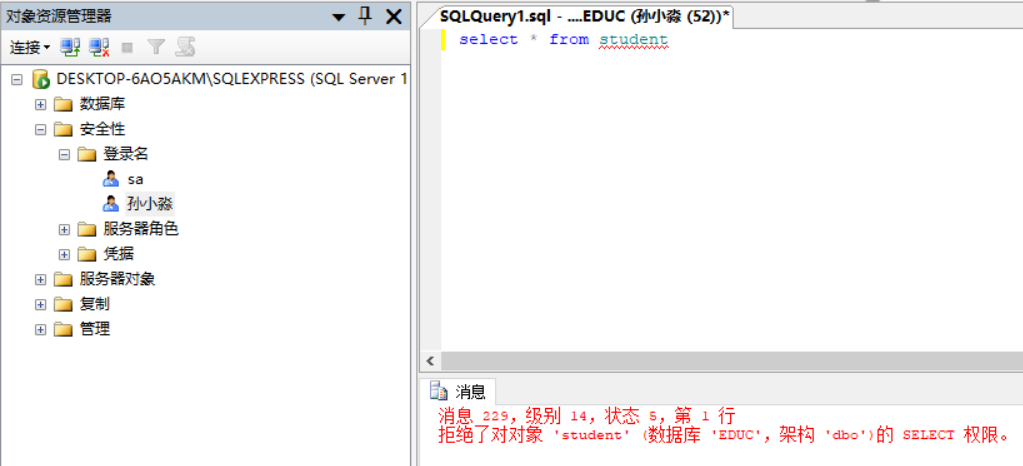
**尝试在windows模式里面删除**



**尝试用语句删除**



**刷新，左边登录名中孙小淼消失，删除成功**

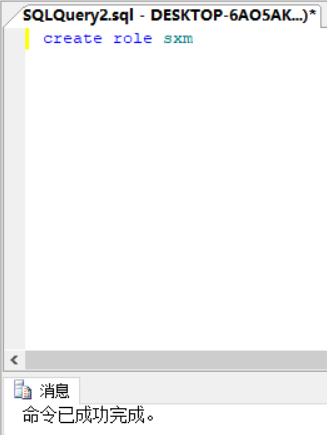


**接下来，我们尝试用孙小淼进行对EDUC数据库里面的student 表进行查询，提示没有select权限**

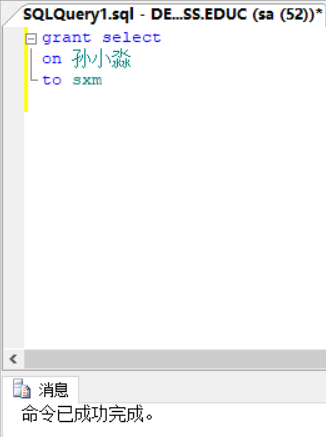
由于在创建登录账号孙小淼的时候，双击孙小淼—用户映射，选择了EDUC，故系统会默认的为登录账号孙小淼创建并绑定一个孙小淼用户，此时名字一样，**故此处我们不再创建用户了**

**接下来进行角色的创建：**

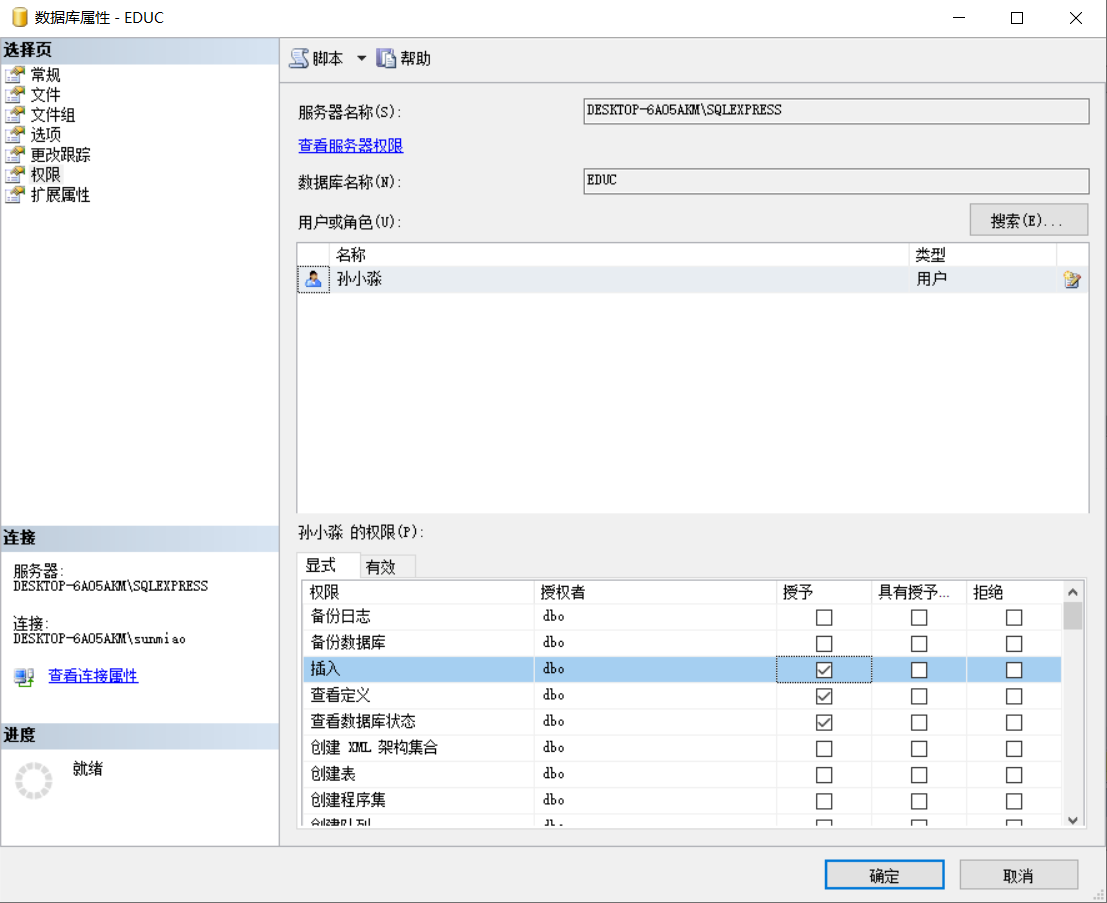
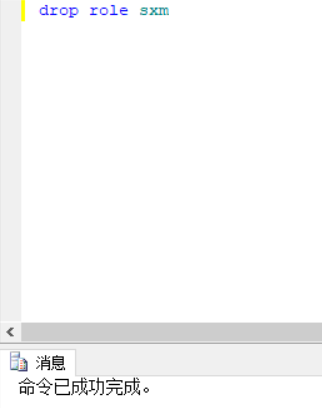
**然后给角色授权**



**删除角色**

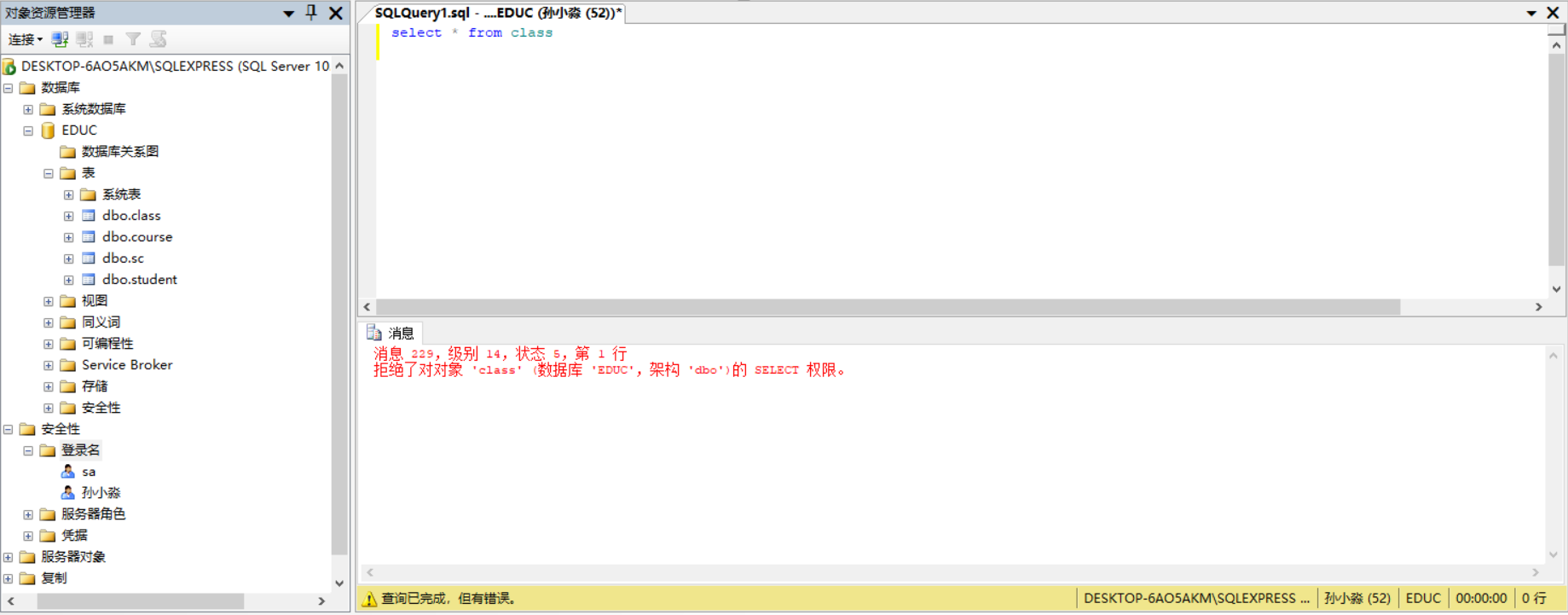
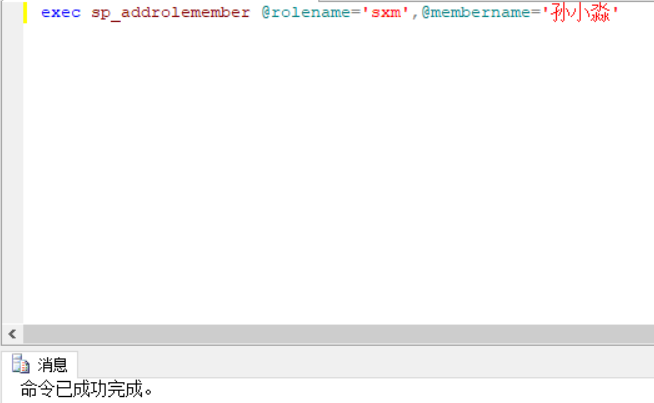


**使用向导给用户授权**



**也尝试了用语句授权**

**最后登录孙小淼，然后发现既可以查看定义，又可以插入数据，但是不能查询，和我们授予的权限一致**



**一、实验目的**

本实验的目的是使学生进一步掌握SQL Server用户角色及权限的使用方法，加深SQL用户管理的理解。SQL Server的特定数据库都有自己的用户和角色（用户组），该数据库只能由它的用户或角色访问，其他用户无权访问该数据库数据，数据库系统可以通过创建和管理特定的数据库的用户和角色来保证数据库不被非法用户访问。

**二、实验准备**

结合课堂教学内容，了解用户及角色权限的相关知识，掌握用户管理的定义，理解用户和角色的创建和分配，及有哪些权限。

**三、实验要求**

1.实现对SQL Server的用户和角色管理

2.设置和管理数据操作权限

**四、实验内容**

1、创建登录帐号

2、创建用户

3、创建角色

4、授权

5、验证授权操作

**五、实验步骤**（贴图仅供参考）

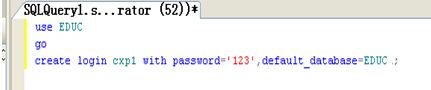
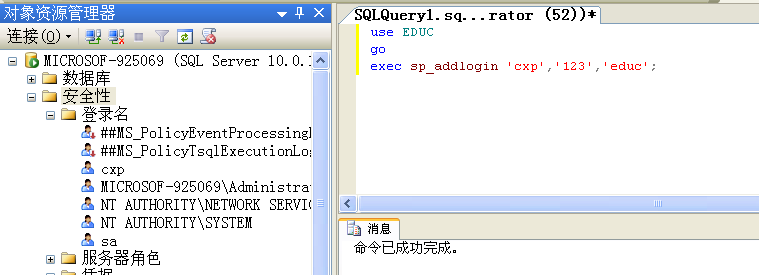
1. 创建登录帐号

--添加登录cxp,设置密码为123,默认数据库为EDUC

exec sp\_addlogin 'cxp','123',’ EDUC’

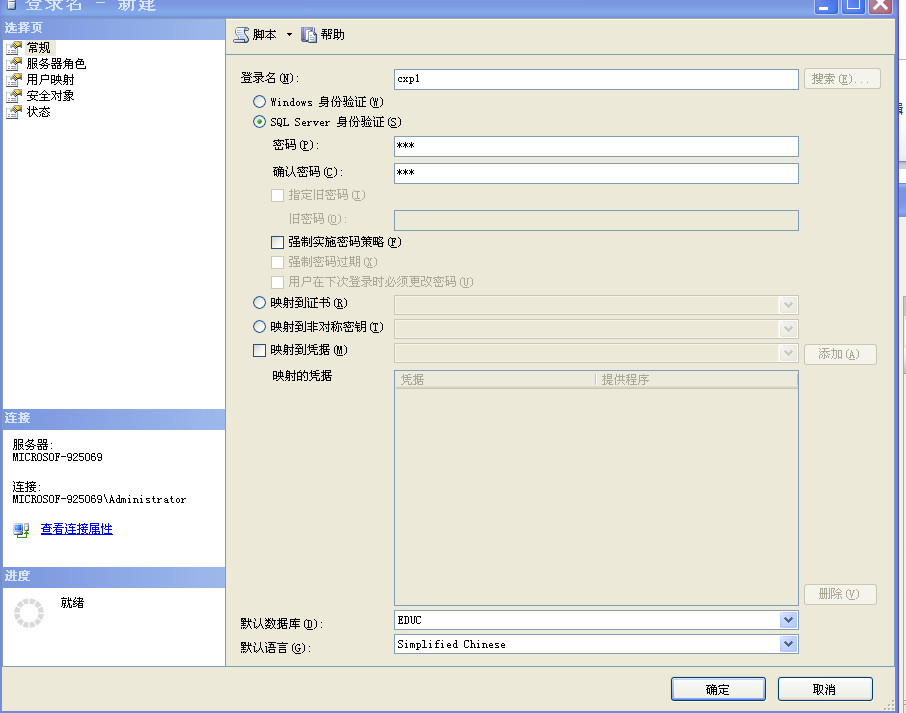
方法1.用语句创建：

或者



方法2.用向导创建：安全性-新建-登录

在弹出的如下窗口中输入相应的内容，如下所示，单击确定



服务器—属性，可以查看相应的内容



用创建的登录名登录：

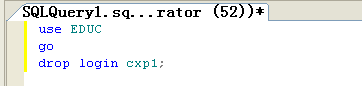
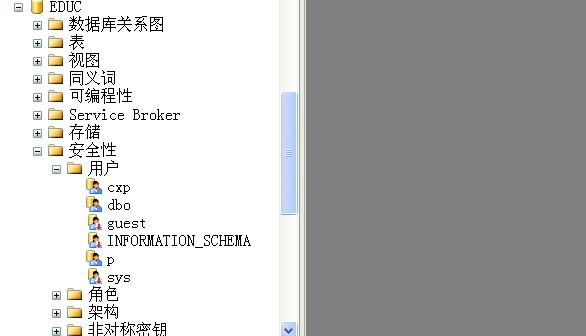
1）首先在配置管理器中将named pipeshe 和tcp/ip启动

2）断开连接，重新登录，选择sql身份验证的模式，将登录名和密码输入进去，

3）若无法登录，则先用windows模式进去，安全—登录名，双击cxp—用户映射，选择映射的数据库educ,即可进入

4）即使是为不同的数据库创建登录名，也不能使用相同的登录名；

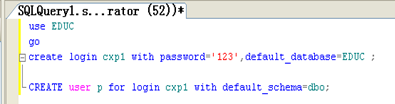
删除登录名：删除登录名，要用windows的模式进去，选中删除即可，也可以用语句删除，删除的时候要将相应数据库下面的安全性里的用户也删除，否则不能再次创建；



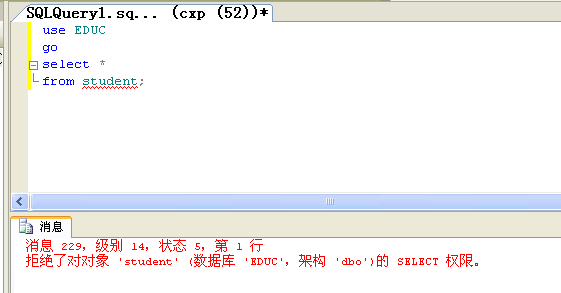
注意：1）如果没有选择用户映射选项卡，将该用户映射到相应的数据库，则即使在常规选项卡中指定可默认的数据库，该用户也不能正常的登录；

2）named pipes:是一种通讯协议，它一般用于局域网中，要求客户端必须具有访问服务器资源的权限；

2、创建用户



此时用户CXP没有一些操作的权限，要用创建的登录账号进入进行验证，如下：



注意：1）在创建登录账号的时候，若双击cxp—用户映射，选择了EDUC，则系统会默认的为登录账号CXP创建并绑定一个CXP用户，此时名字一样；

2）若在创建账号的时候，没有进行上一步，则可以利用上面的语句进行创建任意名称的用户；

3）登录名和用户名的区别：

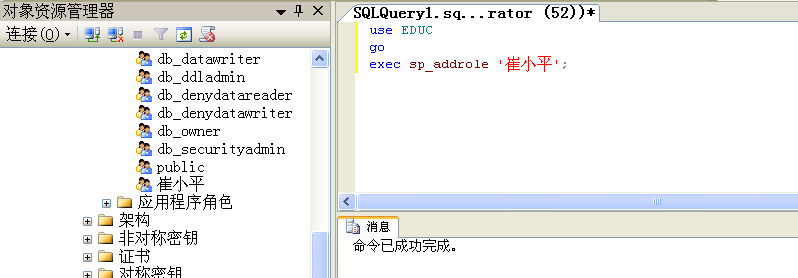
.登录名：服务器方的一个实体，使用一个登录名只能进入服务器，但是不能让用户访问服务器中的数据资源。

.用户名：一个或多个登录对象在数据库中的映射，可以对用户对象进行授权，以便为登录对象提供对数据库的访问权限。

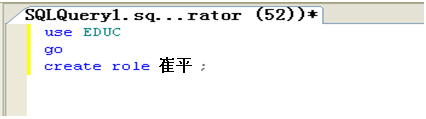
.sqlserver把登录名与用户名的关系称为映射，用登录名登录sserver后，在访问各个数据库时，sqlserver会自动查询此数据库中是否存在与此登录名关联的用户名，

3、创建角色

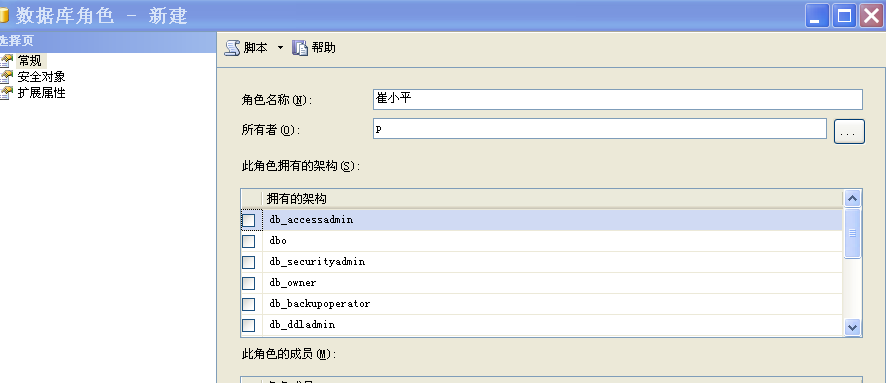
用语句创建角色，有两种方法1）调用存储过程进行创建：



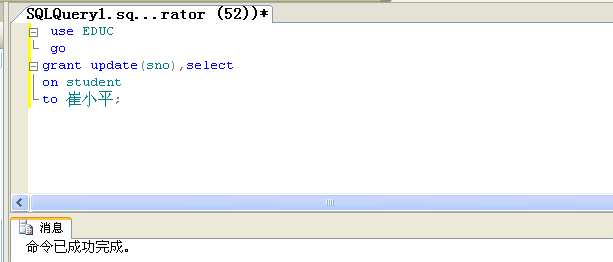
2）用create语句进行创建：（用崔平测试）



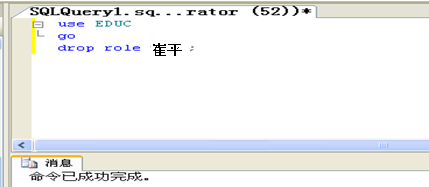
用向导创建：



给角色授权：



删除角色：（用崔平测试）



注意：1）数据库角色是被命名的一组与数据库操作相关的操作，角色是权限的集合；

2）可以为具有相同权限的用户创建一个角色，使用数据库角色来管理数据库的权限可以简化授权的操作；

3）sql中先用create role 语句创建角色，然后用grant语句给角色授权

Grant 权限（权限涉及到的列）[，权限（权限涉及到的列）]

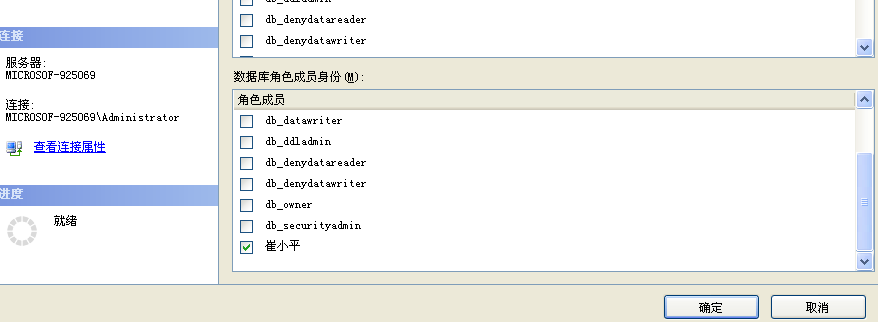
On <对象名>

To 角色[，角色]

4、授权

将角色所拥有的权限授予用户cxp

使用向导，eudc-安全性-用户，双击cxp,在弹出的窗口中选择数据库角色，如下图：

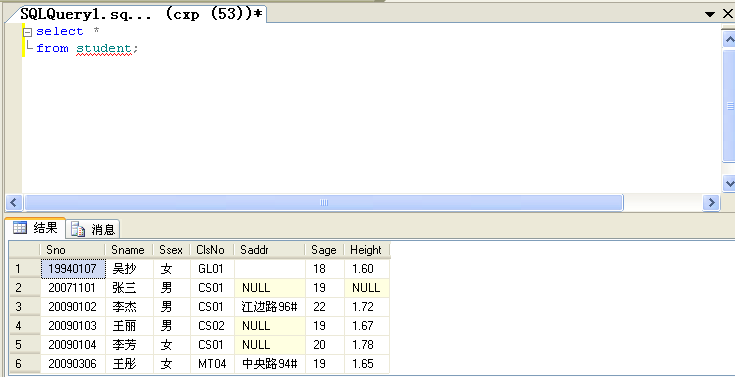


也可以用语句操作

（参考exec sp\_addrolemember @rolename='崔小平',@membername='cxp'或教材）

5、验证授权操作

修改数据库属性为混合登录模式后重新启动数据库，用cxp登录账号登录后，进行验证；

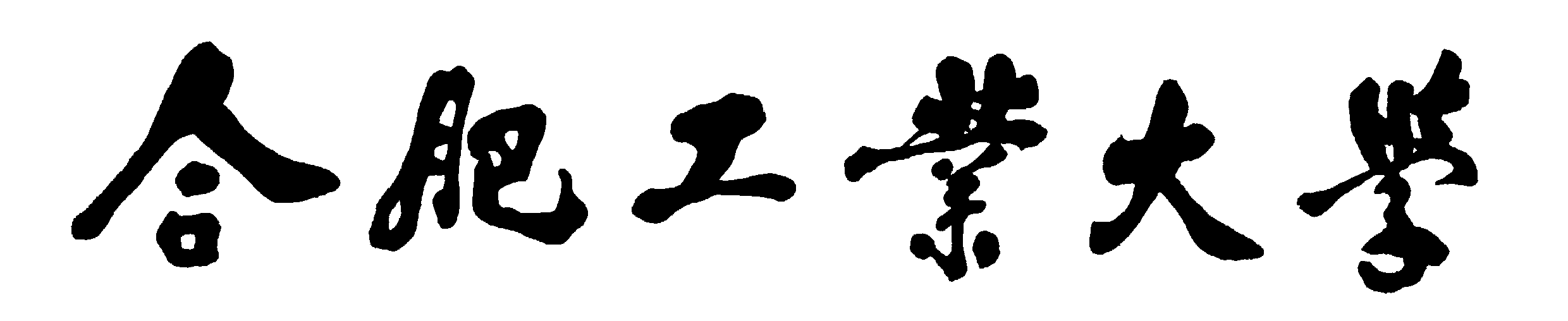


（在企业管理器中将崔小平角色授予p用户。右击Test数据库中的用户，打开p用户的属性，勾选数据库角色。

分别给相应的登录用户创建相应的用户名加入到相应的角色

exec sp\_adduser 'cxp','p','崔小平';）

【附录1 实验报告格式】



软件学院

实验报告

|  |  |
| --- | --- |
| 专 业 班 级 |  |
| 学生姓名及学号 |  |
| 课程教学班号 |  |
| 任 课 教 师 |  |
| 实验指导教师 |  |
| 实验地点 |  |
| 20 ~20 学年第 学期 | |

实验1 \*\*\*\*\*

1. **实验目的**

…….

1. **实验要求**

………

1. **实验内容**

…….

**四、实验步骤**

**五、实验总结**