北京邮电大学

实验报告

课程名称：数据库系统原理

实验名称：数据库完整性与安全性实验

计算机系2015211312班 姓名刘佳鑫

计算机系2015211312班 姓名万诗婕

教师吴起凡 成绩\_\_\_\_\_\_\_\_\_

2018年5月3日

### 实验目的

1. 通过对完整性规则的定义实现，熟悉了解SQL Server数据库中实体完整性、参照完整性、断言等完整性保证的规则和实现方法，加深对数据完整性的理解。
2. 通过对安全性相关内容的定义，熟悉了解SQL Server数据库中安全性的内容和实现方法，加深对数据库安全性的理解。

### 实验环境

采用SQL Server数据库管理系统作为实验平台。SQL Server选用2012版本

### 实验内容

1. 添加语义约束
2. 主键/候选键约束验证
3. 外键约束验证
4. 缺省/默认值约束实验
5. 以DBA身份登陆系统，在图形界面下创建新登录，在安全性选择登录，然后新建登录用户，可以使用Windows认证模式和SQL Server认证模式。
6. 使用T-SQL命令create login login-name …和create login login-name from windows重复第1步的内容。
7. 理解服务器角色的概念。在服务器角色选项卡中赋予新建用户角色。
8. 对特定服务器对象设置权限，双击登录名以后选择安全对象，赋予该登录名对服务器内的某个对象的权限。

### 实验步骤

## (一)、 完整性实验与要求：

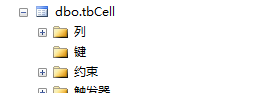
#### 一、 添加语义约束

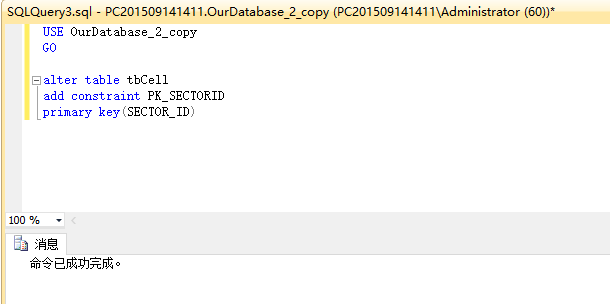
在前面完成的实验2、3中已建立了本实验所需的11张表。本实验将针对这11张表，采用alter table语句，添加主键、候选键、外键、check约束、缺省/默认值约束，并观察当用户对数据库进行增、删、改操作时，DBMS如何维护完整性约束。

建表阶段，在create table语句中添加各类完整性约束的内容请自行练习。

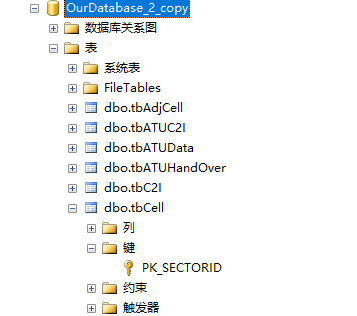
**（1）新建主键，找到未定义主键的表，利用下面的语句建立主键:**

**加之前：**





**加之后**

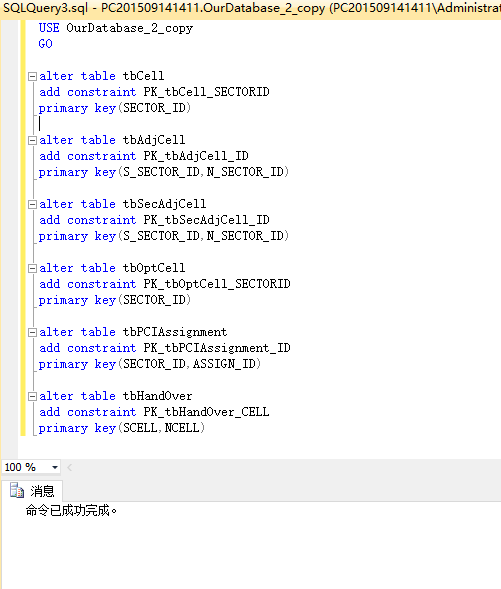


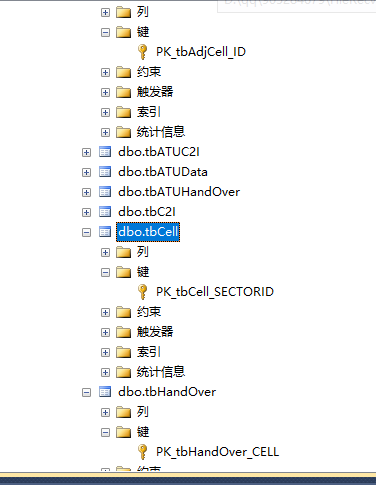
继续添加其余表的主键

开始所用主键名称和其他表重复，会报错。

计算机生成了可选文字:
一 alt e r t ab 1 e tbAdjCe11 
add constraint PK 
primary key S_SECTOR ID SECTOR ID 
一 alt e r t ab 1 e tbSecAdjCe11 
add constraint PK ID 
primary key S_SECTOR ID SECTOR ID 
一 alt e r t ab 1 e tbOptCe11 
add constraint PK SECTORID 
primary key SECTOR ID 
一 alt e r t ab 1 e tbPCIAssign-ment 
消 2 7 “ ， 另 刂 16 ， 状 态 5 ， 第 川 行 
數 掘 库 中 已 存 存 名 为 ' PK_ID' 的 时 象 。 
消 1750' 另 刂 16 ， 状 态 第 川 行 
无 法 创 津 约 束 。 清 阅 前 面 的 误 消 。 

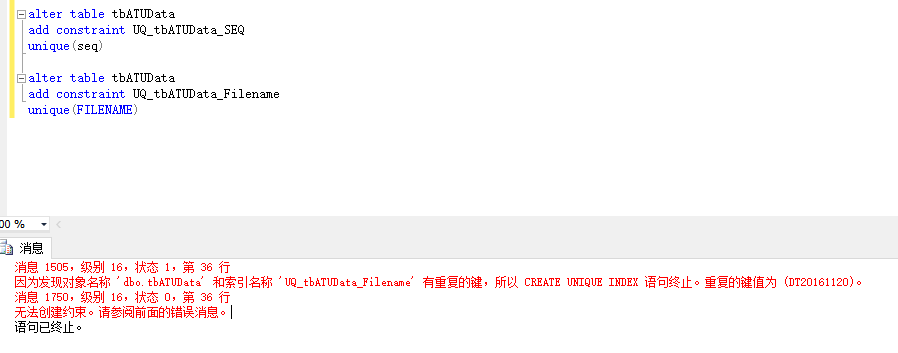
修改主键名称后，成功添加。



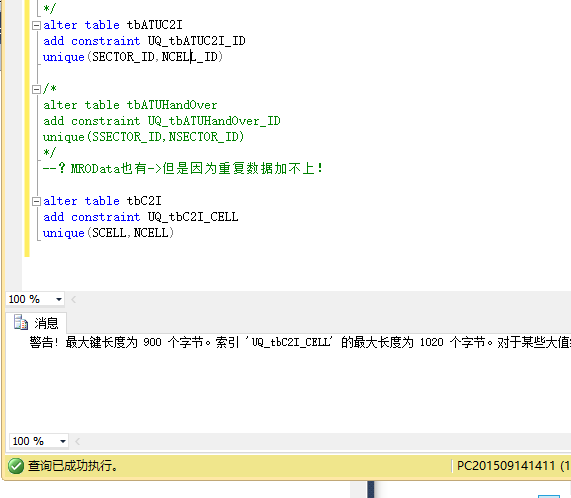


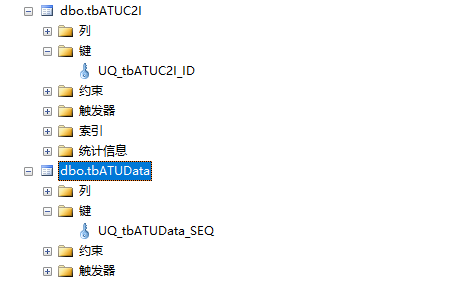
**（2）候选键，找到有非主键的其它候选键的表，利用下面的语句建立候选键:**

组合的候选键不能分开加，不然每一项都要求 unique。



修改后，添加成功

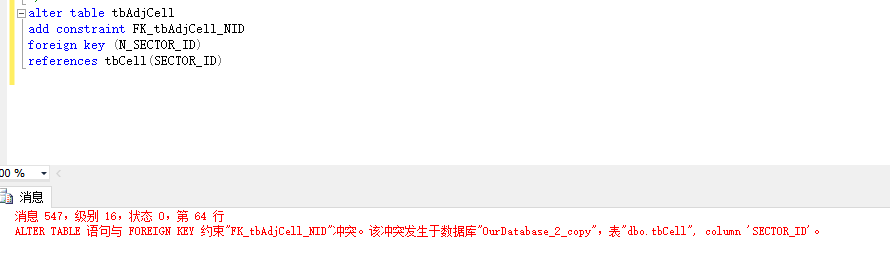




**（3）外键约束，找到拥有主外键关系的表，利用下面的语句新建外键：**

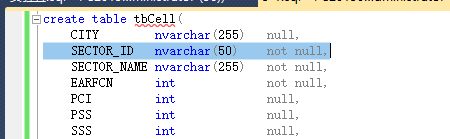
添加外键过程中，有以下几种报错：

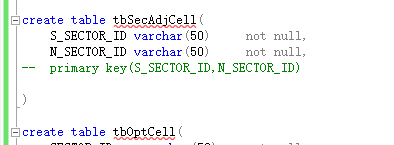
1. 因为存在tbAdjCell里的N\_SECTOR\_ID的值不在tbCell里的SECTOR\_ID里。

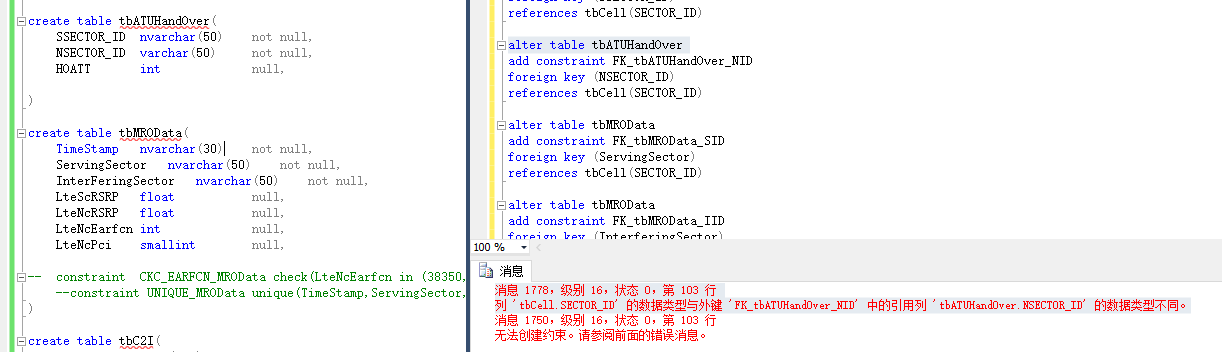


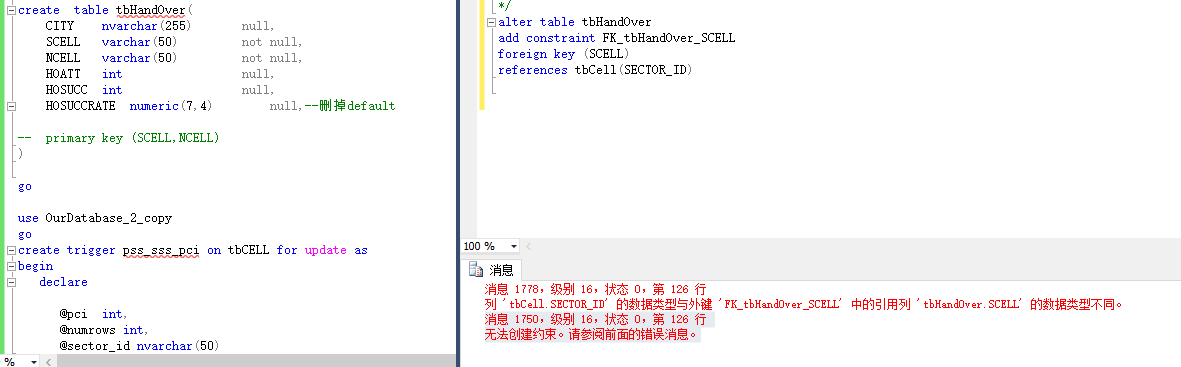
1. 两列数据类型不同



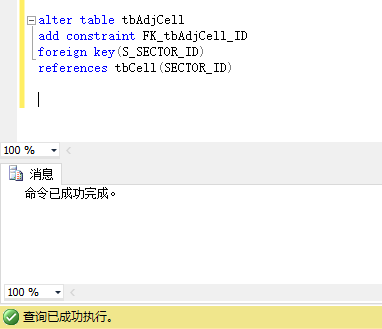


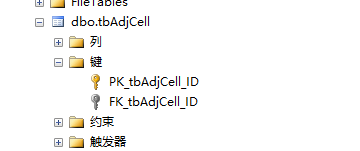


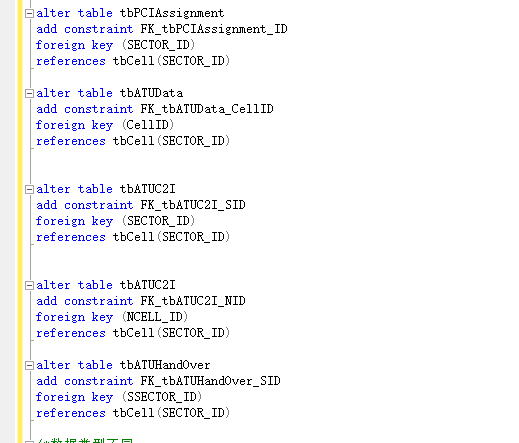




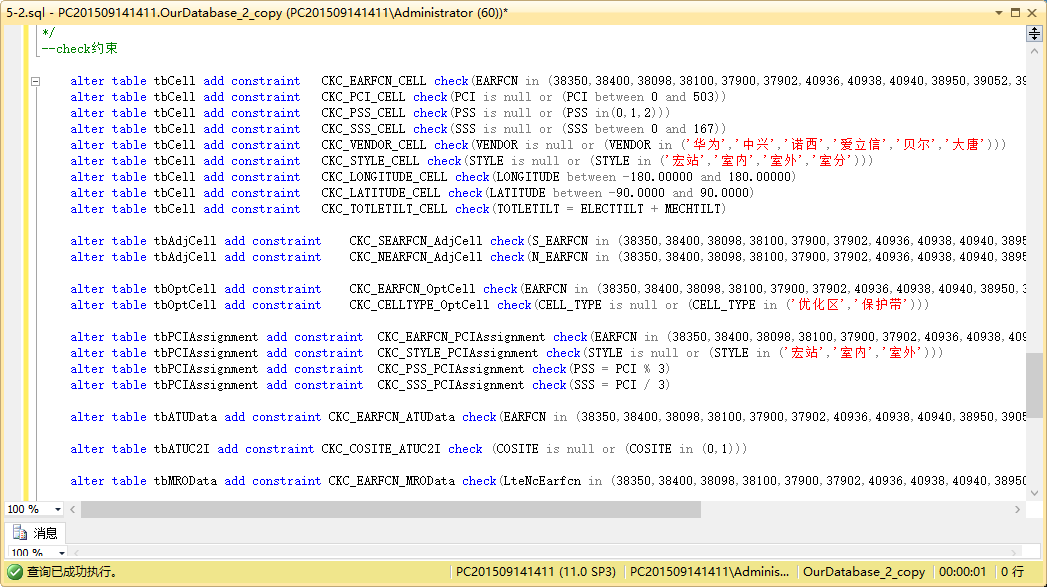
其余外键均添加成功。

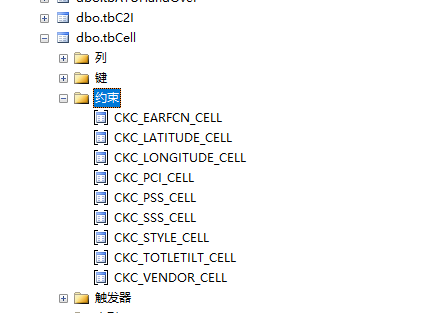




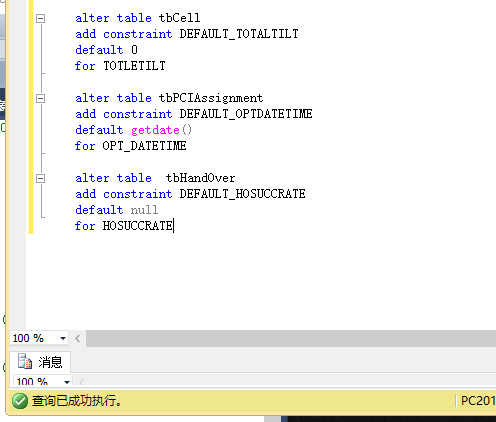


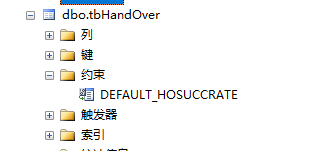
**（4）check约束，利用下面的语句给相关表建立约束**





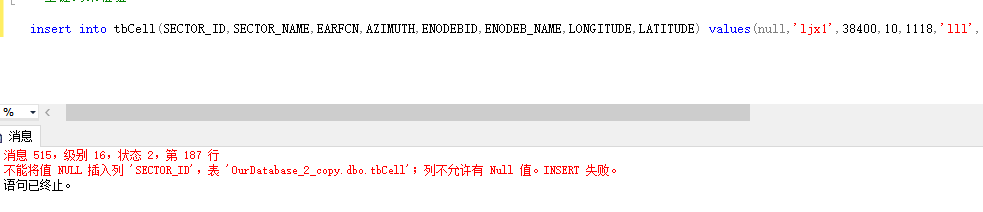
**（5）缺省/默认值约束**



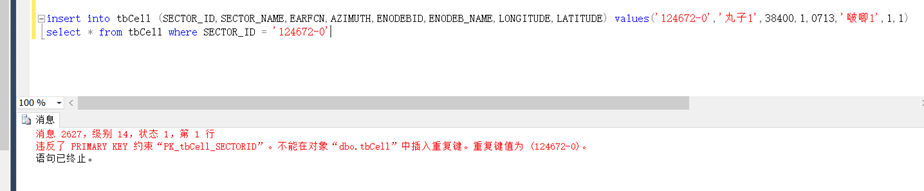


#### 二、主键/候选键约束验证

**1. 选取定义了主键的关系表，如tbCell、tbAdjCell、tbOptCell等，向该表插入在主属性上取值为空的元组，观察DBMS反应；**



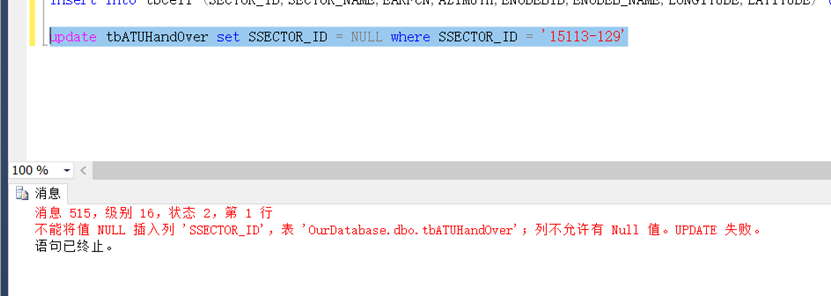
选取表中某些或某个元组，修改这些元组在主属性上的取值，或向表中插入新元组，使这些元组与表中已有其它元组的主属性取值相同，或者将选定的元组在主属性上的取值修改为null，观察系统反应。



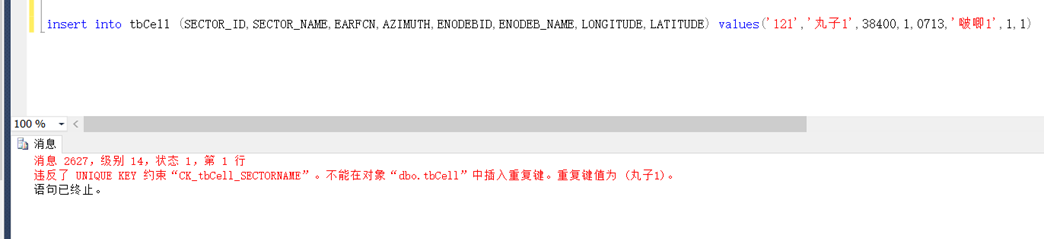
**2. 选取定义了候选键的关系表，向该表插入在候选键属性上取值为空的元组，观察DBMS的反应；**

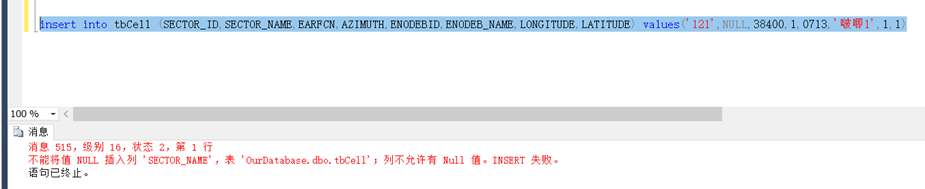
（1）将SSECTOR\_ID，NSECTOR\_ID设置为tbATUHandOver的候选键，并修改某一个元组的该属性值为空。

改值:将tbATUHandOver 表中SSECTOR\_ID 为 '15113-129'的记录中的SSECTOR\_ID修改为NULL。



比较在主键、候选键属性上插入null值或重复值时，DBMS的不同反应和处理方式。



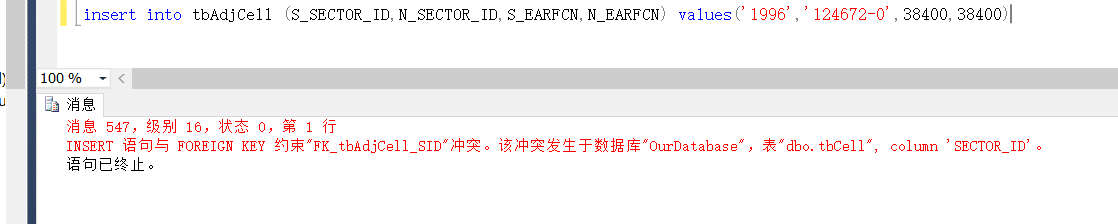


#### 三、外键约束验证

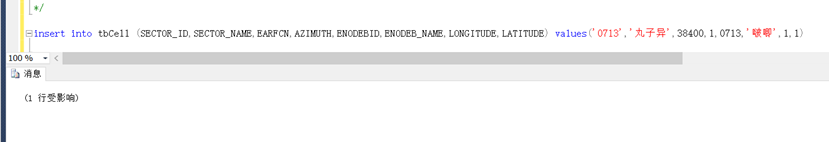
**1． 选取相互间定义了外键关联的一组表，分别在参照关系、被参照关系上，对表的主属性/外键属性作插入、删除、更新等操作，观察当其中1个表（如参照关系表、被参照关系表）在外键属性或主属性上的取值发生变化时，数据库管理系统DBMS对这些操作的反应，以及另外一个表（如被参照关系表、参照关系表）在主属性或外键属性上的取值的变化，并记录实验结果。**

上述插入、删除、更新操作操作分为违反约束和不违反约束两种情况。

**插入**

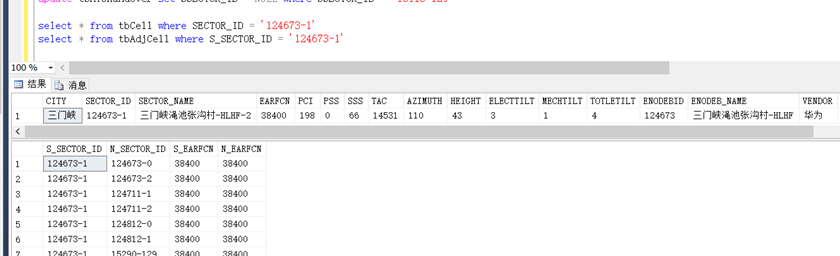
违反约束

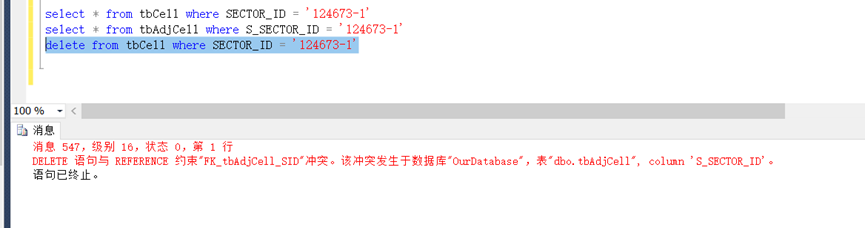
不违反约束



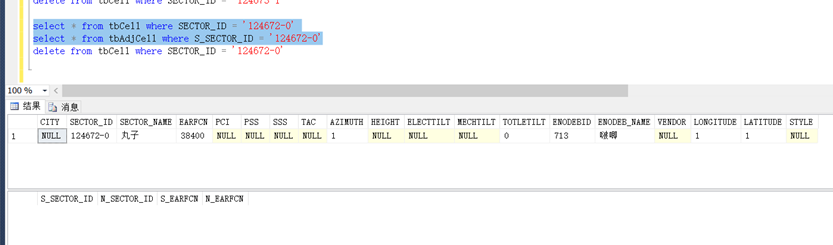
**删除**

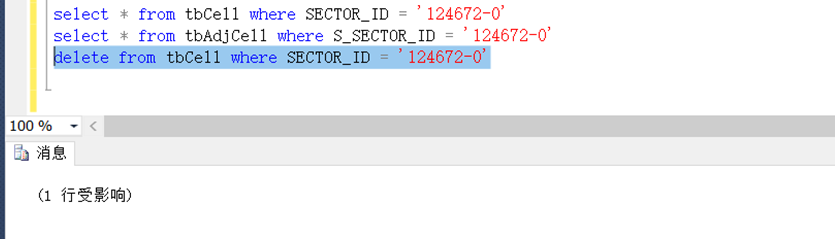
违反约束





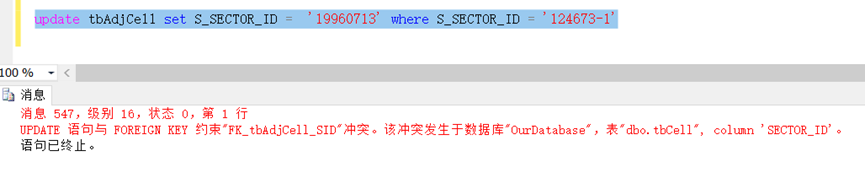
不违反约束



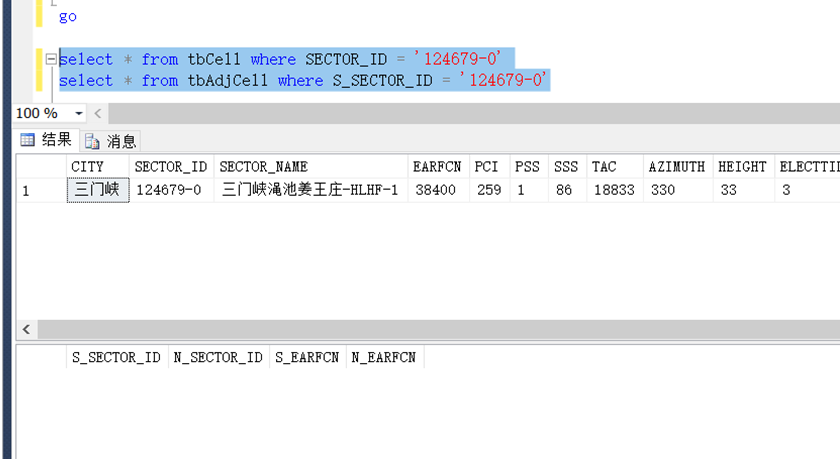


**更新**

违反约束



不违反约束



**2． 观察级联操作对外键约束的影响**

1）对相互间定义了外键关联关系的一组表，分别使用

foreign key－references

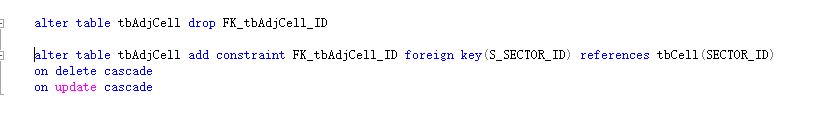
on delete cascade

on update cascade

语句创建级联外键约束(先删除之前创建的外键，使用Alter table 中的Drop constraint参数)。

然后重新创建级联外键（使用alter table 中的add constraint …

foreignkey … references … on delete/update/insert …参数）



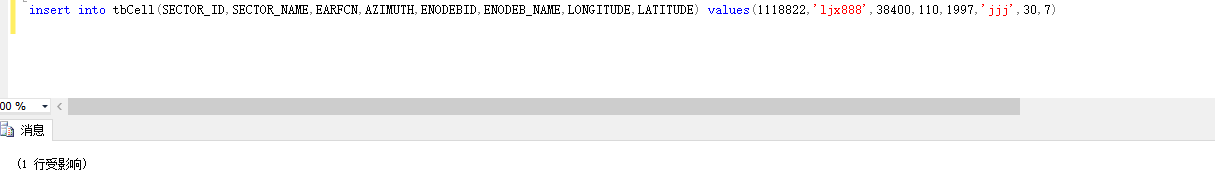
2) 分别在参照关系、被参照关系上，对表的主属性/外键属性作插入、删除、更新等操作，观察当1个表（如参照关系表、被参照关系表）在外键或主属性属性上的取值发生变化时，数据库管理系统对这些操作的反应，以及另外一个表（如被参照关系表、参照关系表）在主属性或外键属性上的取值的变化，并记录实验结果。

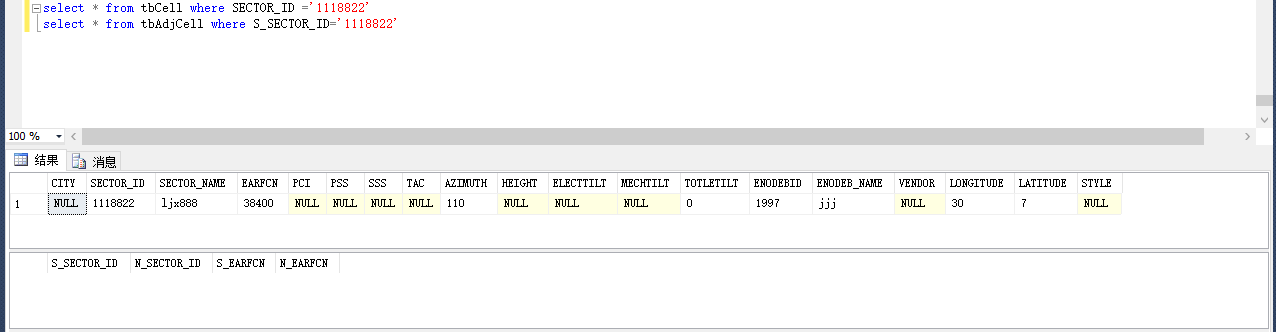
上述插入、删除、更新操作操作分为违反约束和不违反约束两种情况。

**对被参照表操作：**

**插入**

不违反约束





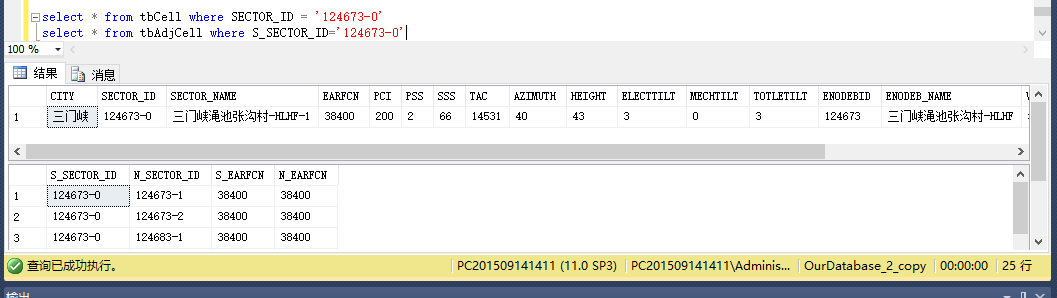
被参照表成功插入

**删除**

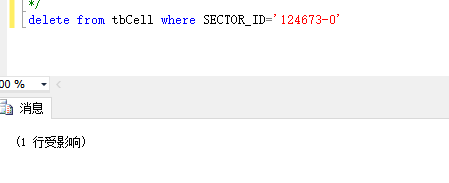
不违反约束

1. 参照和被参照表中都用了此值。

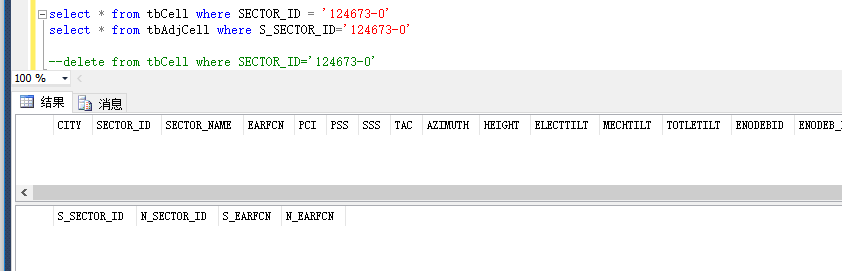
删除前



删除



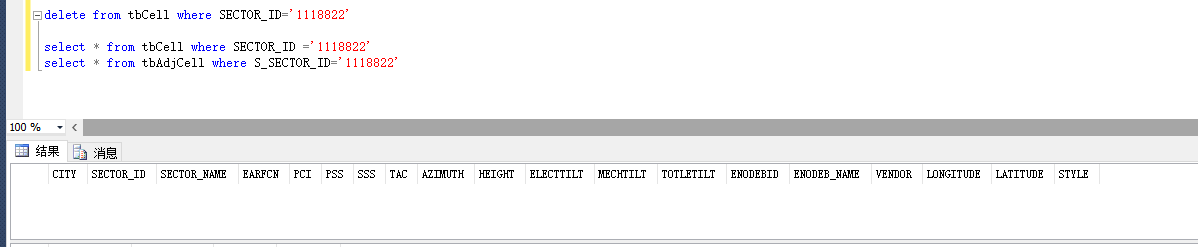
删除后



参照表和被参照表中值为该项的都被删除。

1. 此值仅在被参照表中存在

成功删除被参照表中元组

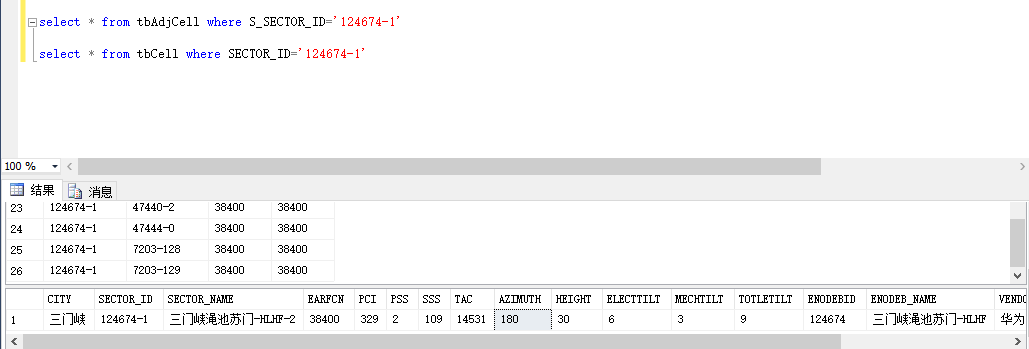


**更新**

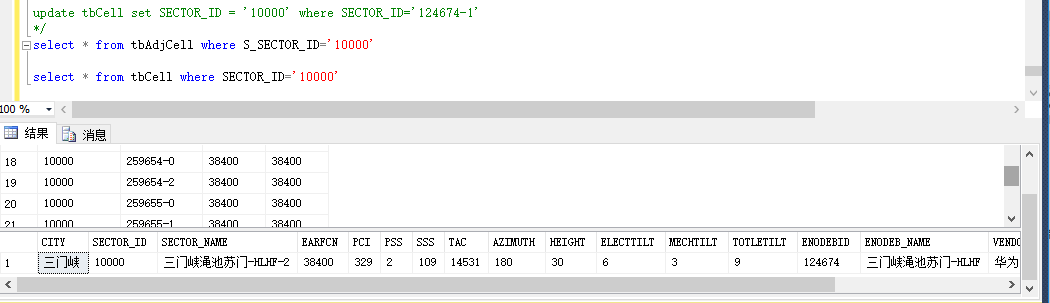
不违反约束

参照表和被参照表外键中都有此值

更新前



更新后，发现参照表和被参照表都被更新了。

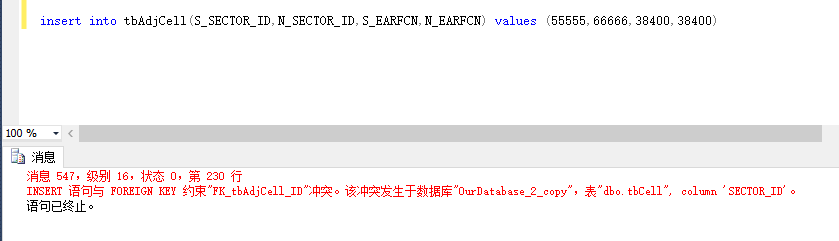


**对参照表操作**

**插入**

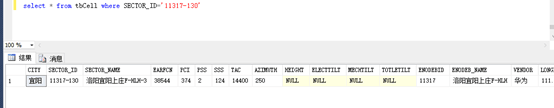
1. 违反约束

因为被参照表的主键没有这个值

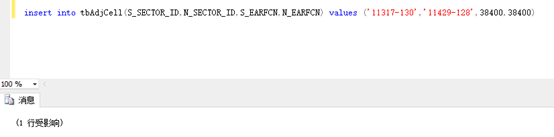


1. 不违反约束

被参照表的主键有这个值



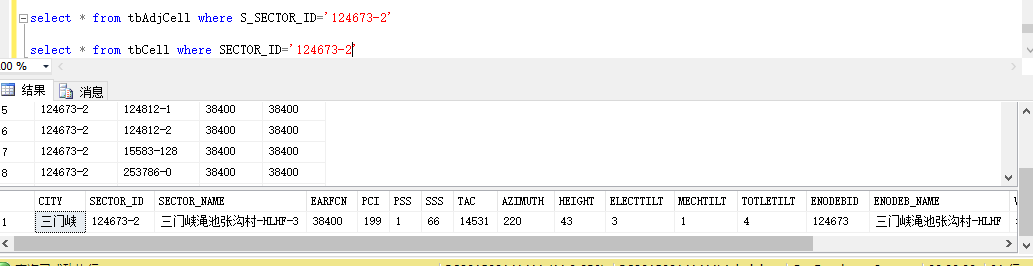
插入成功



**删除**

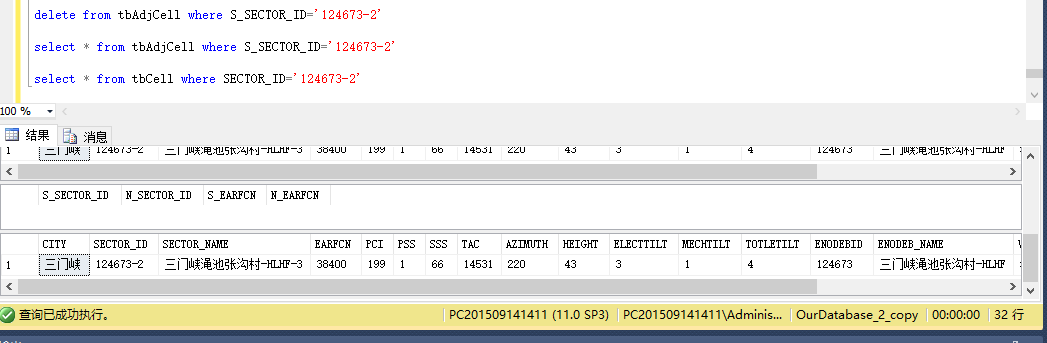
1. 违反约束：无
2. 不违反约束

删除前

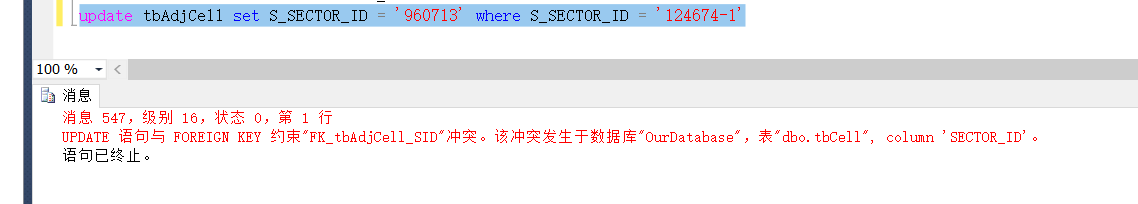
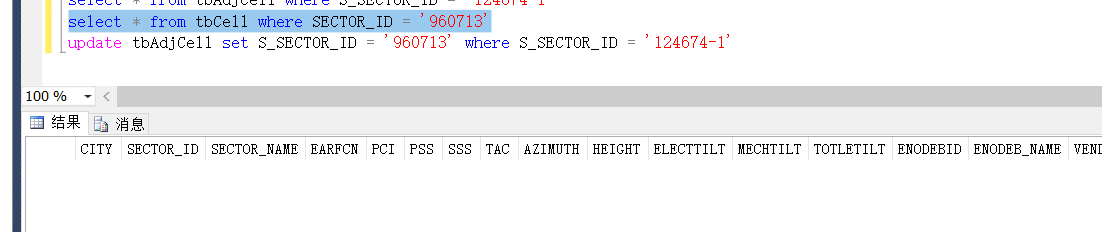


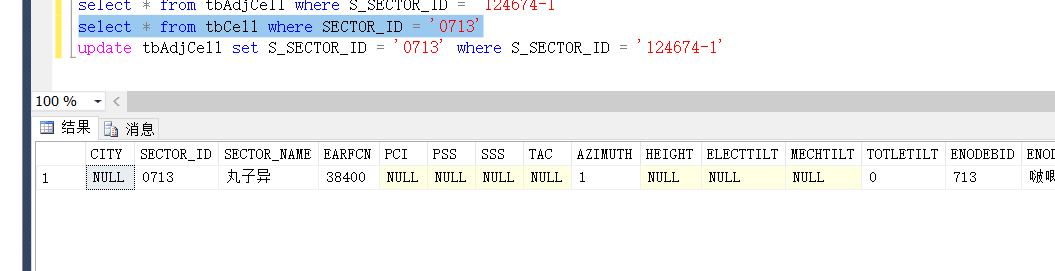
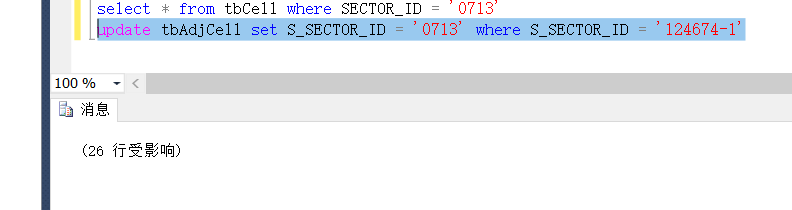
删除后：

被参照表中值相同的项并没有被删除。



更新：

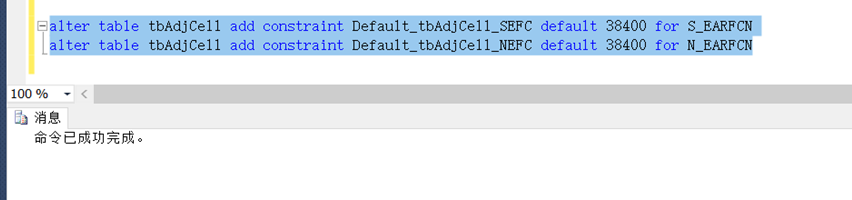
违反约束

不违反约束

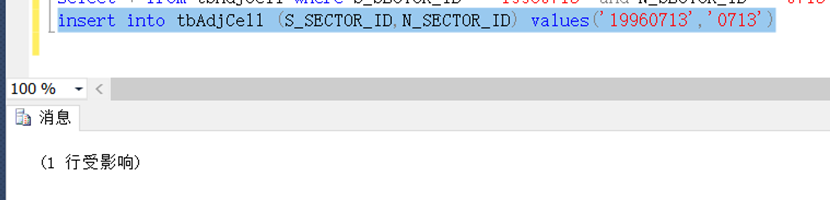
#### 四、缺省/默认值约束实验

以tbAdjCell为对象，针对S\_EARFCN、N\_EARFCN，建立默认/缺省值约束——将这2个频点的默认值设置为38400，然后向表中插入数据。

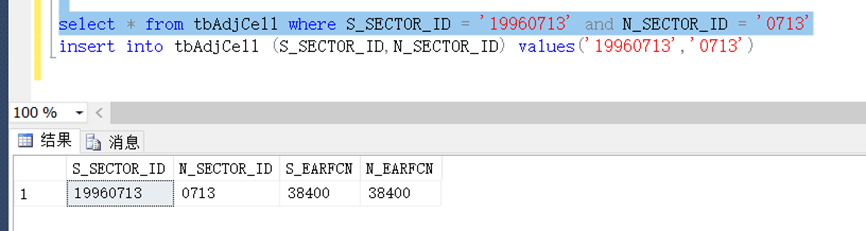
1. 在现有表中添加缺省/默认值约束



1. 插入一个新元组（缺少频点）



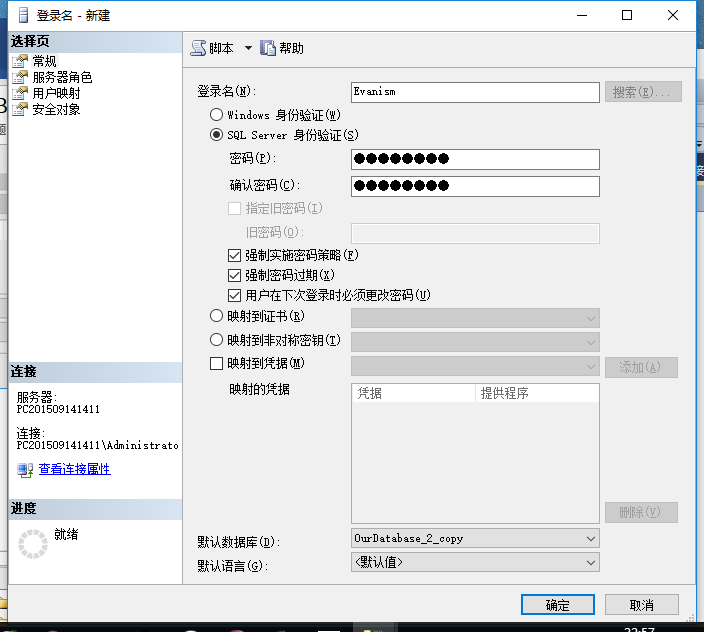
1. 检查新元组的频点是否为默认值

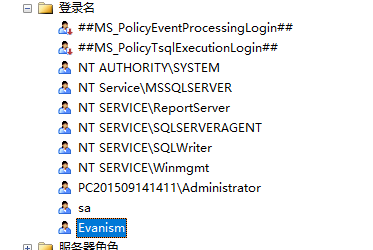


## (二)、 安全性实验内容与要求

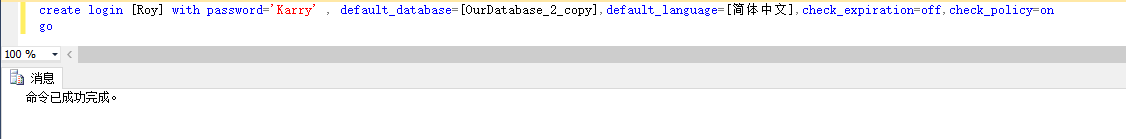
1. **以DBA身份（可以是SQL Server上的sa或者windows上的系统管理组的某个成员）登陆系统，在图形界面下创建新登录，在安全性选择登录，然后新建登录用户，可以使用Windows认证模式和SQL Server认证模式。选择Windows认证登录则必须添加已经存在的Windows本机或者域上的用户或者用户组。通常选择搜索，然后选择高级和立即查找，找到实际存在的Windows用户或者组。最后可以选择默认数据库为当前使用的数据库，而不是默认的系统数据库。**

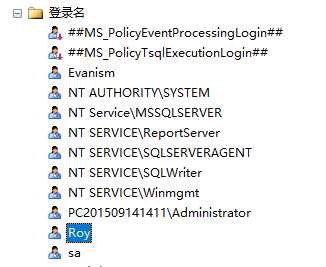
图形化新建登录用户





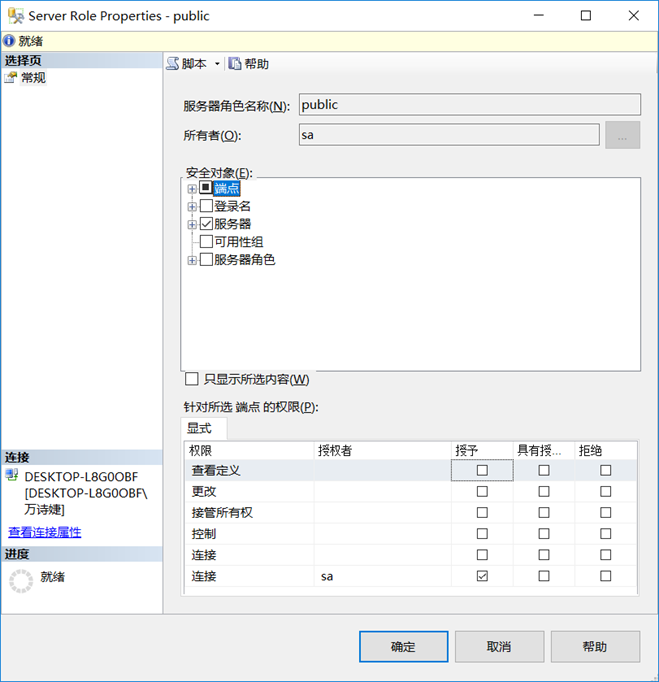
**(2) 使用T-SQL命令create login login-name …和create login login-name from windows重复第1步的内容。**



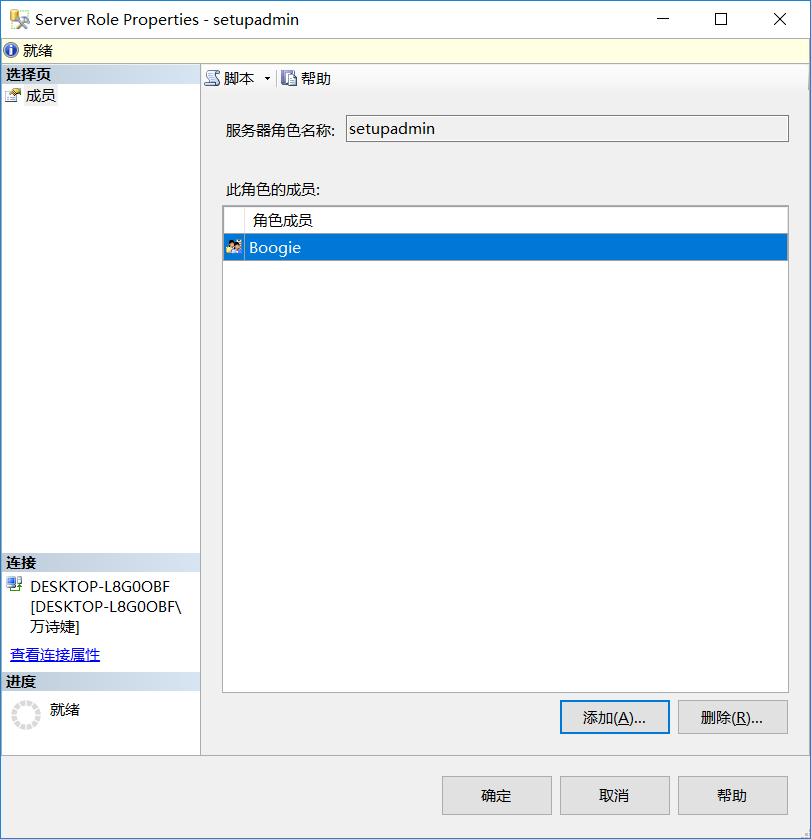


**(3) 理解服务器角色的概念。在服务器角色选项卡中赋予新建用户角色。完成后查找T-SQL中对应命令，在图形界面上撤消该权限后使用命令完成授予权限和收回权限的任务。**

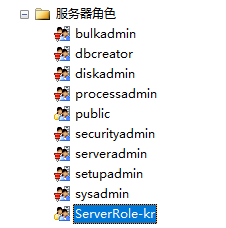
a) 首先查看现在存在的服务器角色，任选服务器角色下的一个实际角色，查看其属性。

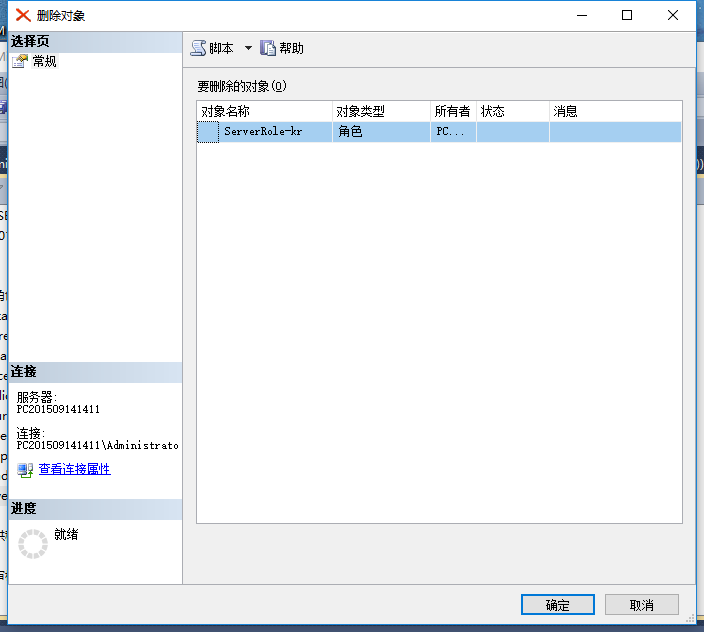


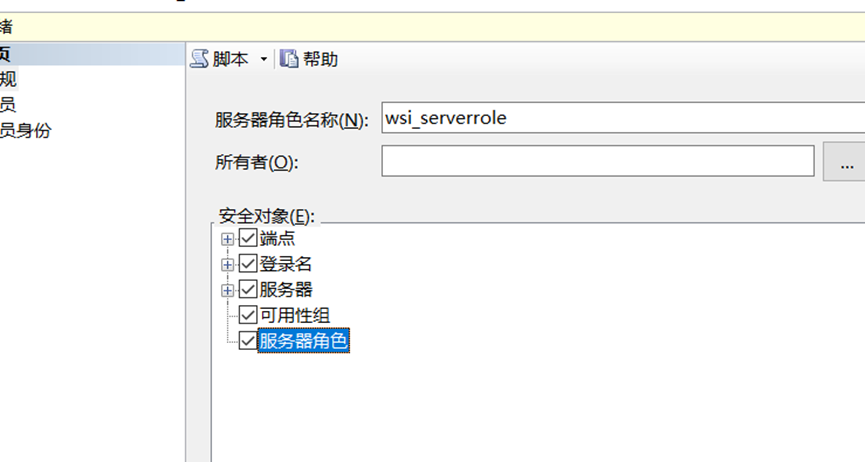
b) 选择登录名打开，将刚刚新建的Login添加上去并且删除掉。

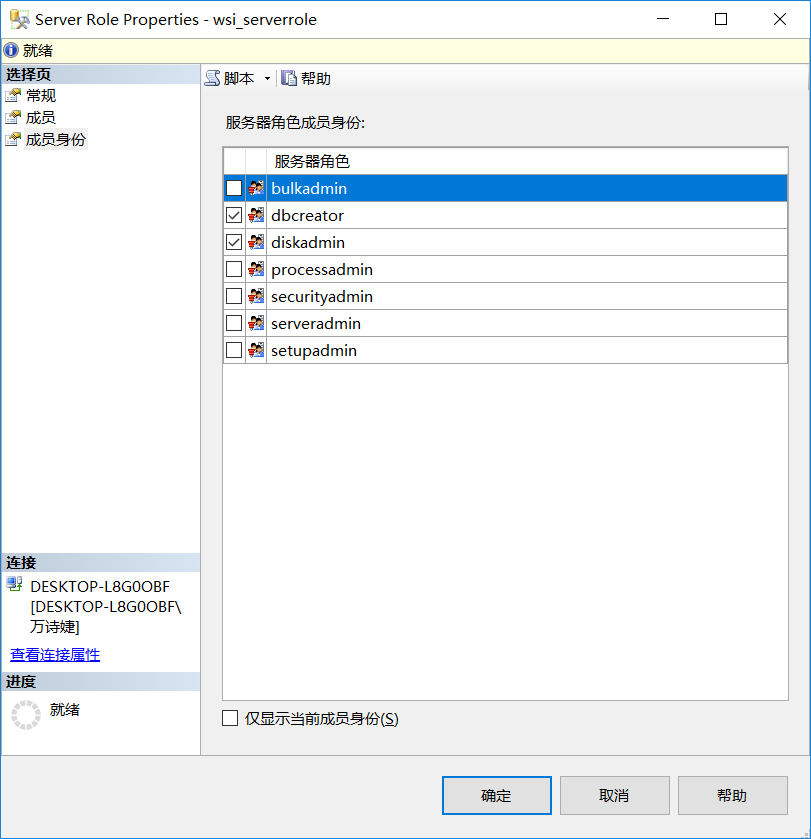


c) 增删新建服务器角色的权限。

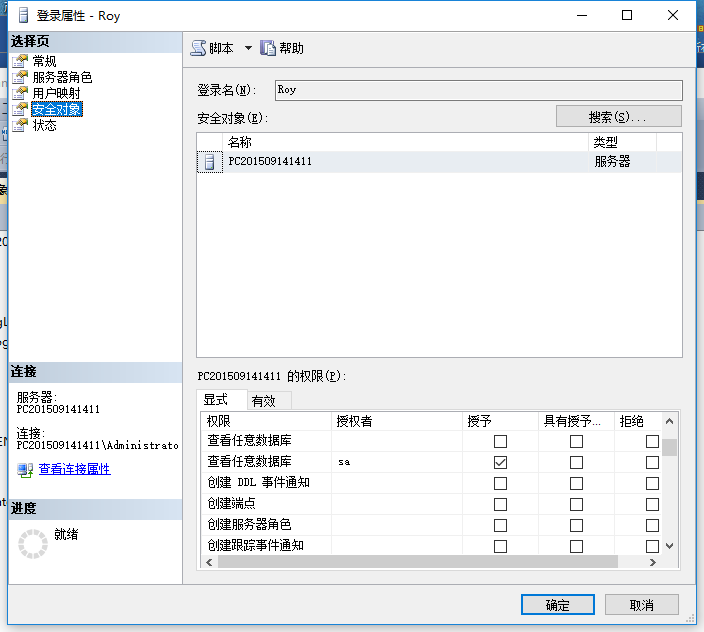


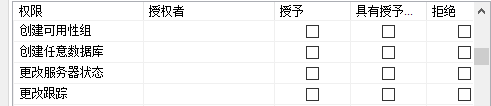


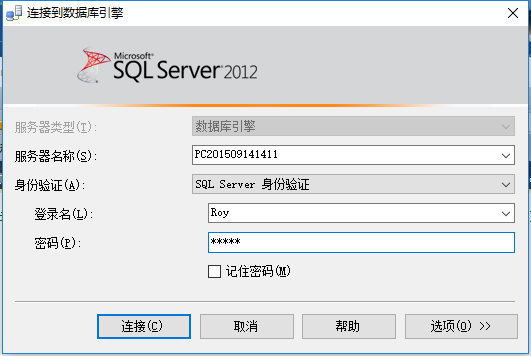


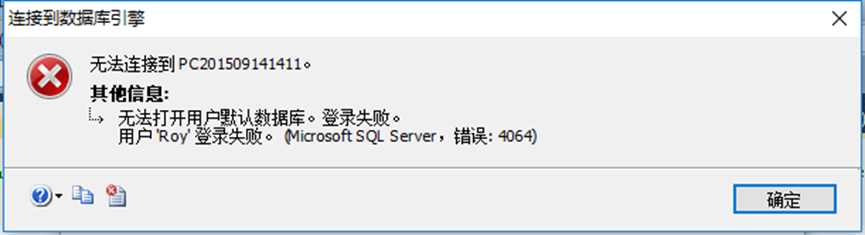


**(4) 对特定服务器对象设置权限，双击登录名以后选择安全对象，赋予该登录名对服务器内的某个对象的权限。执行相关SQL操作，检查该用户的权限。用户在权限范围内、超出权限范围访问相应数据对象，查看访问结果和DBMS的反映。**



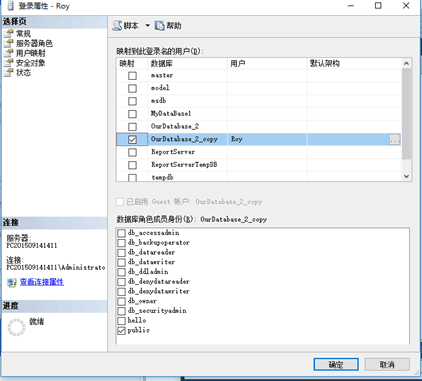




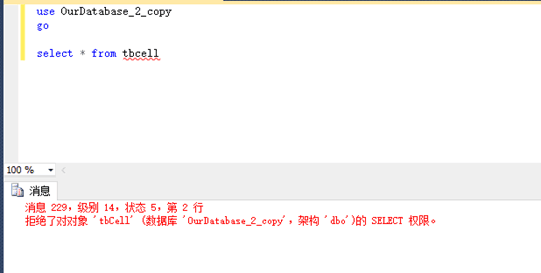


**解决方法：**

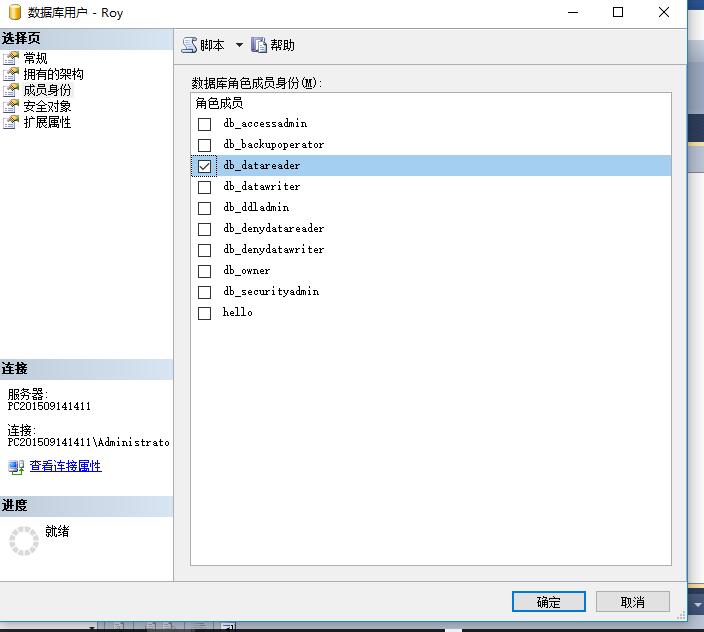
先要将用户映射到特定数据库，并对服务器用户授权

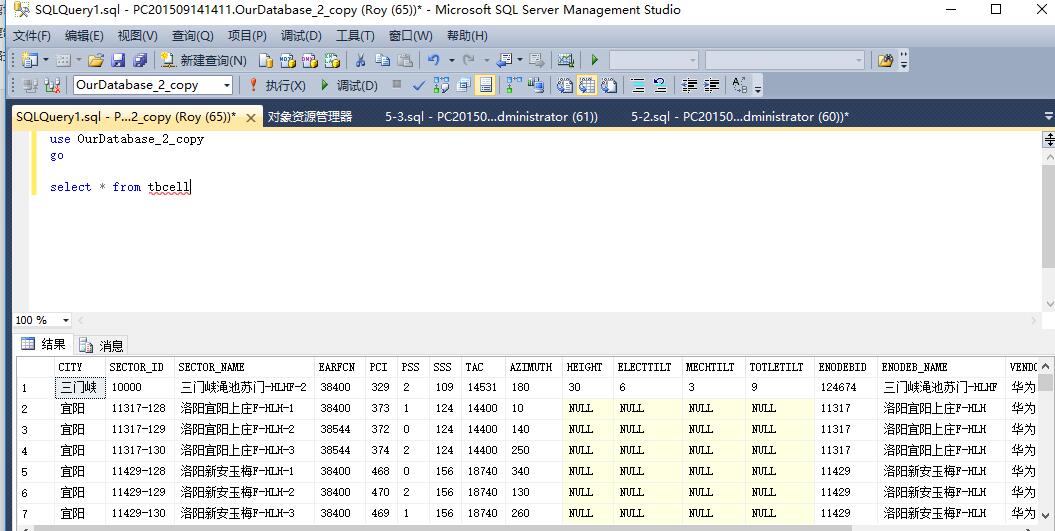


使用服务器用户登陆后，在查询时出现以下报错

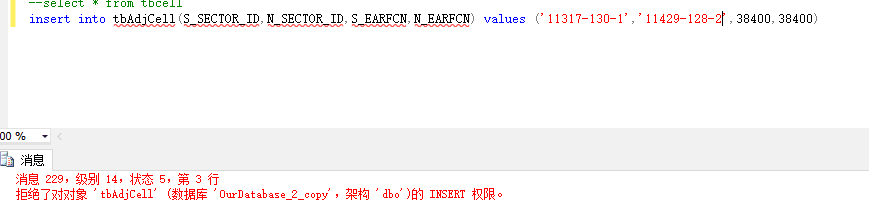


对映射的数据库用户授权后解决此报错



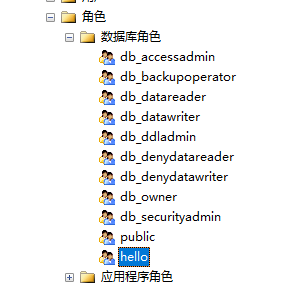


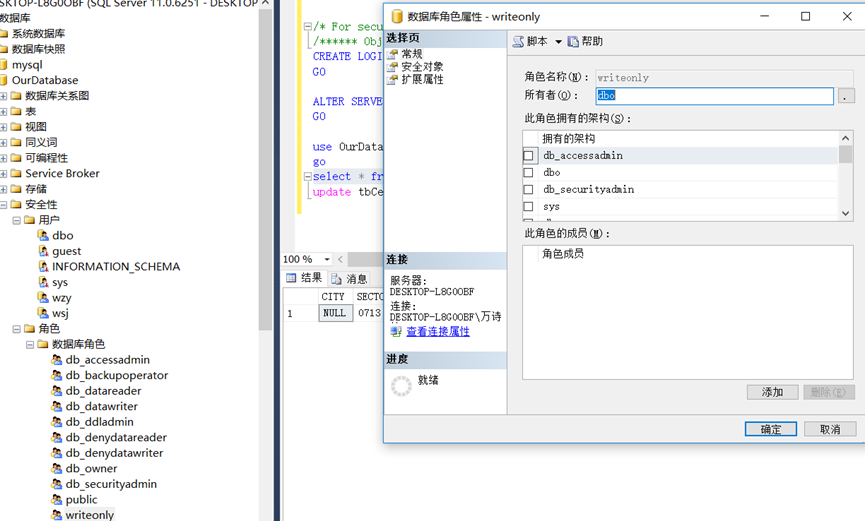
由于没有授权更改数据库的权限，所以DBMS会拒绝操作请求



**(5) 理解数据库角色的概念。**

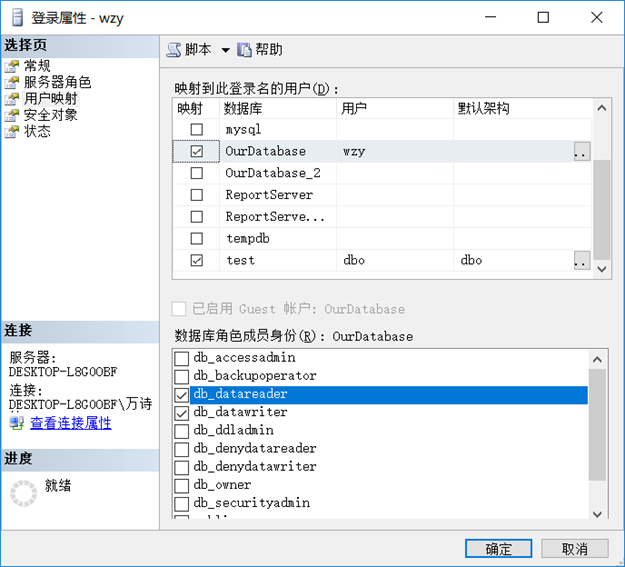
a) 新建一个数据库角色。在移动通信数据库中，选择安全性，角色，数据库角色，新建。

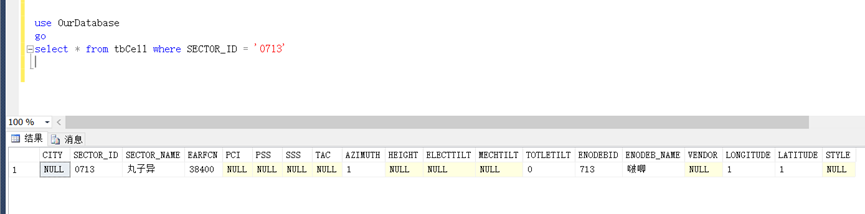


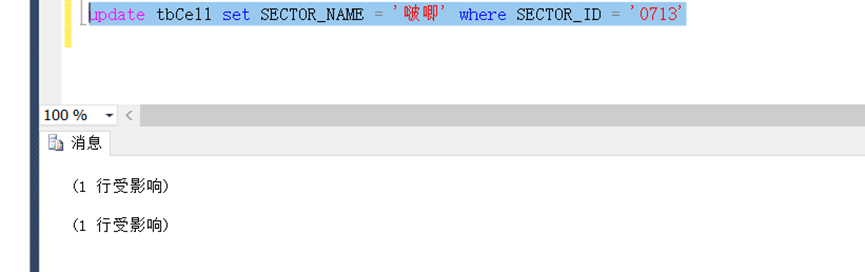


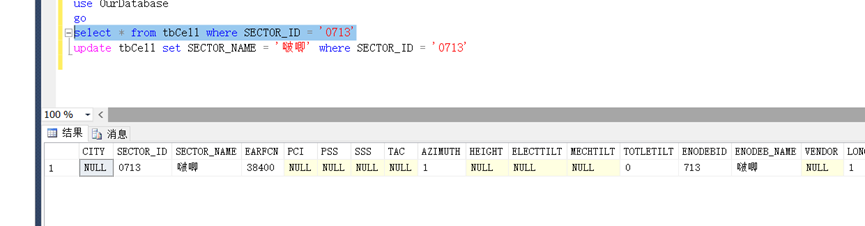
b) 授予访问移动通信数据库的读、写权限，并对其中的Cell表数据进行修改。

登录wzy用户 授予他读写的权限

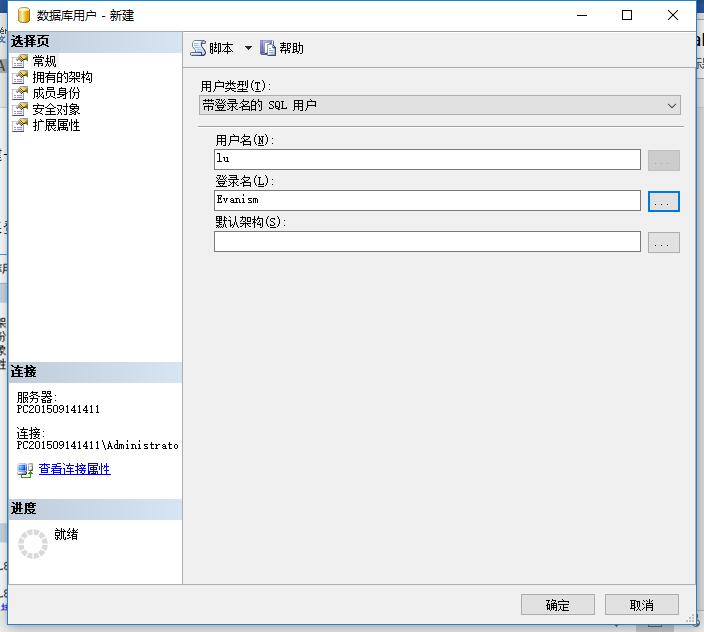




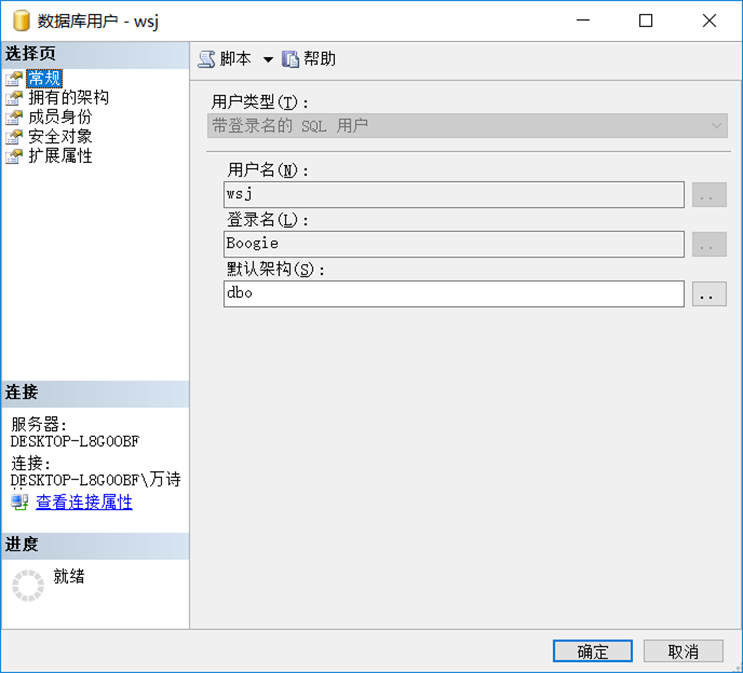




c) 新建一个数据库用户。



d) 将某登录映射到移动通信数据库某用户。



**(6) 根据实际操作，指出服务器角色和数据库角色有多少种，分别有什么权限？**

SQL服务器角色

sysadmin 可以在 SQL Server 中执行任何活动。

serveradmin 可以设置服务器范围的配置选项，关闭服务器。

setupadmin 可以管理链接服务器和启动过程。

securityadmin 可以管理登录和CREATE、DATABASE权限，还可以读取错误日志和更改密码。

processadmin 可以管理在 SQL Server 中运行的进程。

dbcreator 可以创建、更改和除去数据库。

diskadmin 可以管理磁盘文件。

bulkadmin 可以执行 BULK INSERT 语句。

public 它只拥有的权限是VIEW ANY DATABASE。

public角色有两大特点：

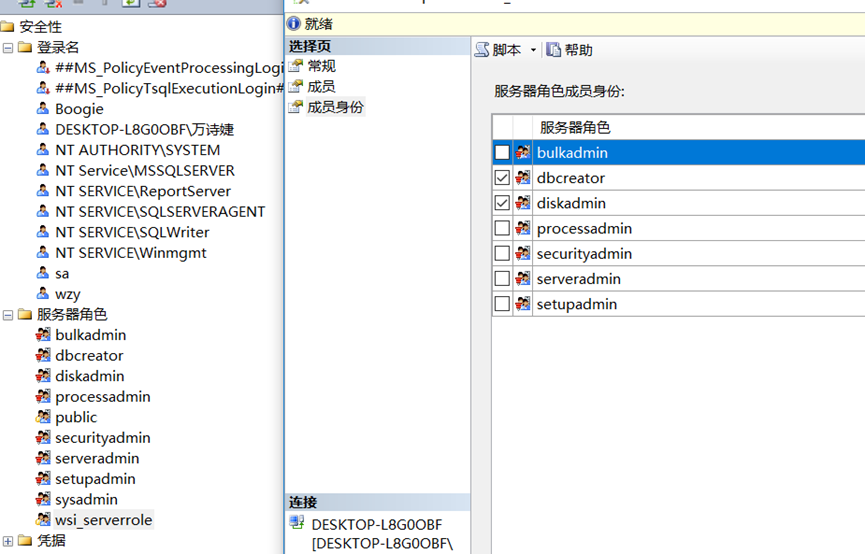
第一，初始状态时没有权限；

第二，所有的数据库用户都是它的成员，因此不能将用户、组或角色指派为public角色的成员，也不能删除public角色的成员。

public角色作用：可以通过对public设置权限从而为所有数据库设置相同的权限。

请勿为服务器 public 角色授予服务器权限。

wsj\_serverrole 可以创建，更改或除去数据库同时可以删除数据库



数据库角色

db\_owner 数据库中有全部权限。

db\_accessadmin 可以添加或删除用户 ID。

db\_securityadmin 可以管理全部权限、对象所有权、角色和角色成员资格。

db\_ddladmin 可以发出ALL DDL，但不能发出GRANT、REVOKE或DENY语句。

db\_backupoperator 可以发出 DBCC、CHECKPOINT 和 BACKUP 语句。

db\_datareader 可以选择数据库内任何用户表中的所有数据。

db\_datawriter 可以更改数据库内任何用户表中的所有数据。

db\_denydatareader 不能选择数据库内任何用户表中的任何数据。

db\_denydatawriter 不能更改数据库内任何用户表中的任何数据。

public

--public 角色是一个特殊的数据库角色，每个数据库用户都属于它。public 角色：

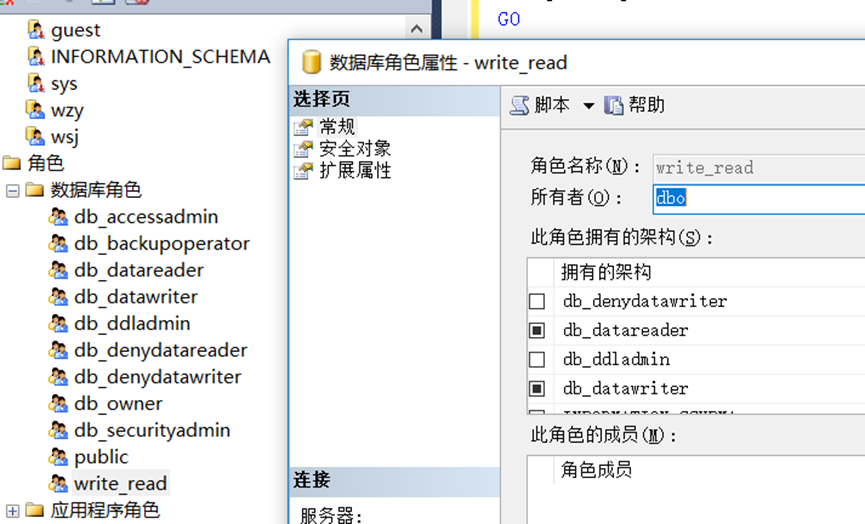
--捕获数据库中用户的所有默认权限。

--无法将用户、组或角色指派给它，因为默认情况下它们即属于该角色。

--含在每个数据库中，包括 master、msdb、tempdb、model 和所有用户数据库。

--无法除去。

write\_read 可以查看和更改数据库内任何用户表的所有数据



**(7) 创建数据库时，还会自动创建SYS、GUEST 和dbo组。通过帮助文档，了解这些组在数据库中起到的作用。**

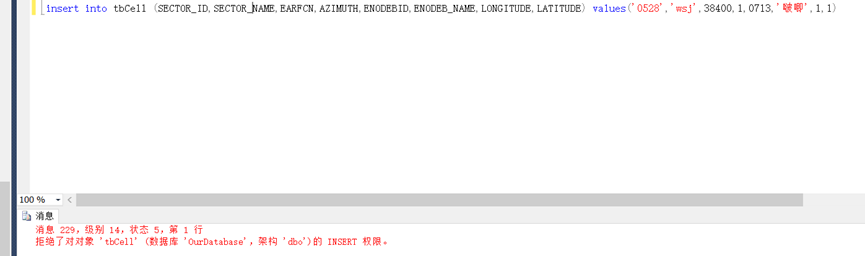
dbo。是数据库所有者用户。顾名思义，dbo用户对数据拥有所有权限，并且可以将这些权限授予给其他用户。在SQL Server 2012 中，创建数据库的用户默认就是数据库的所有者，从属于服务器角色sysadmin的登录名会自动被映射为dbo用户，因此sysadmin角色的成员就具有对数据库执行任何操作的权限

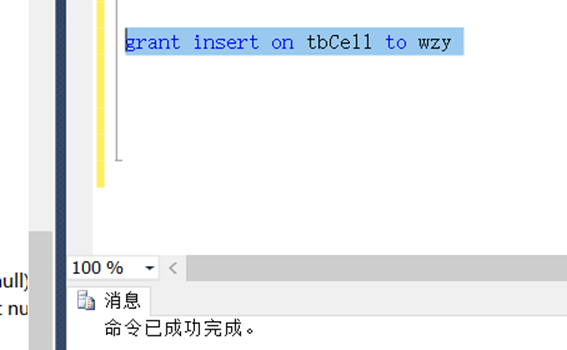
guest。是数据库客人的用户。当数据库中存在guest用户，则所有登录名，不管是否具有访问数据库的权限，都可以访问guest用户所在的数据库。因此。Guest用户的存在会降低系统的安全性。在用户数据库中guest用户默认处于关闭状态，而在master和tempdb数据库中出于系统运行的需要，guest用户是开启的。

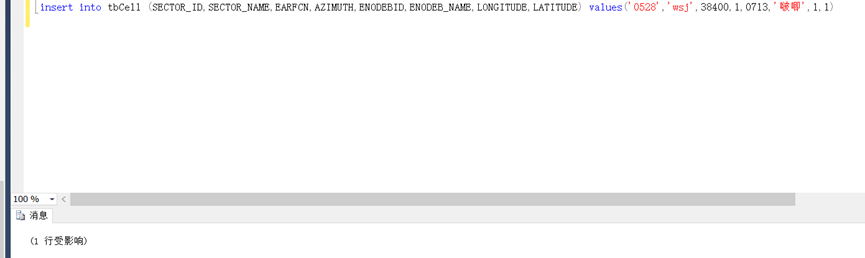
Sys是为使用sys架构的视图而创建的用户。为保证系统正常运行，建议不要修改该用户

**(8) 使用GRANT语句设置对某表的SELECT和INSERT权限，并进行相关数据访问或者修改操作。验证权限已经设置成功。执行超出权限范围访问相应数据对象的操作，查看访问结果和DBMS的反映。**

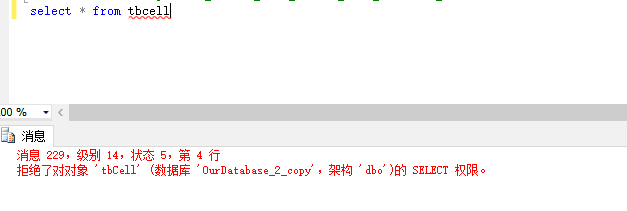
**insert**

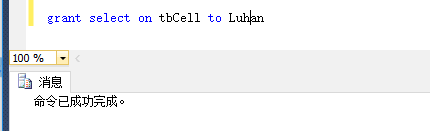


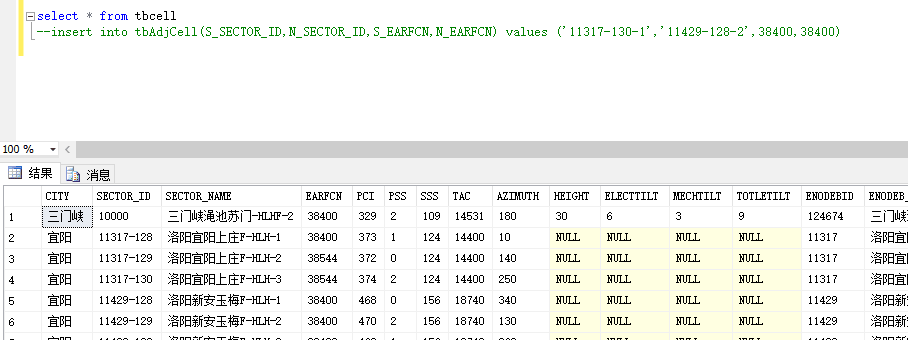




**select**

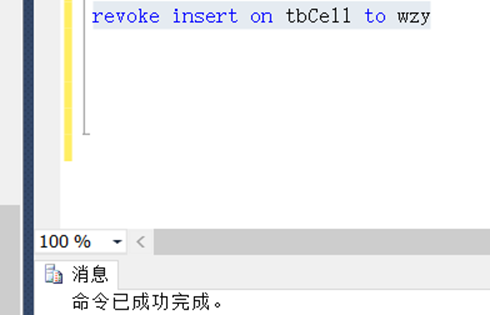


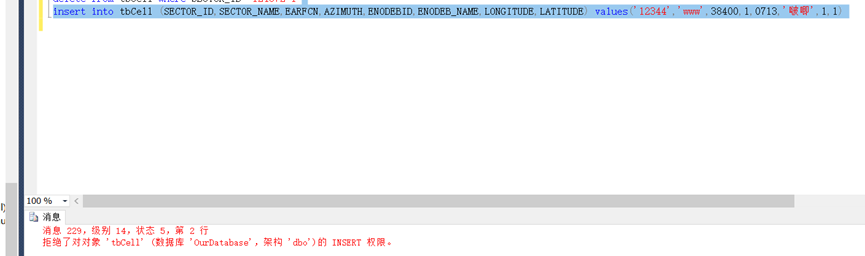




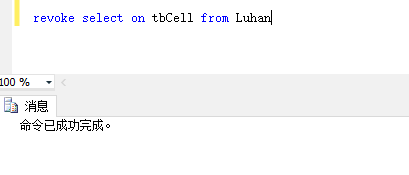
**(9) 分别在图形化界面、命令行中，撤销用户权限，命令行中使用REVOKE语句。验证原来能正常进行的操作现在是否仍能正常进行。**

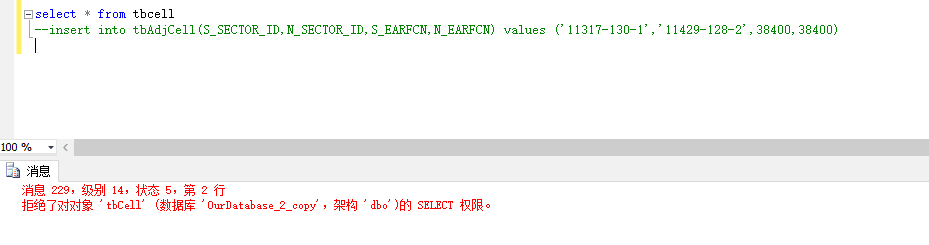
**insert**



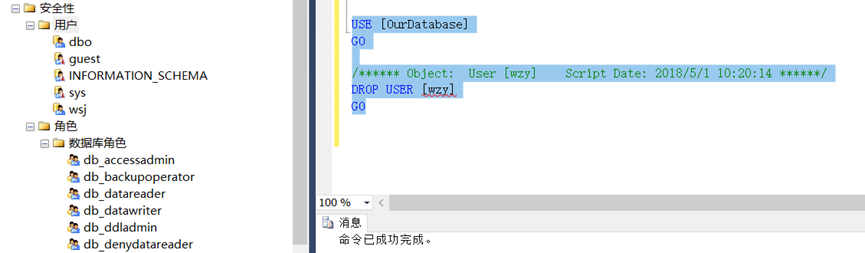


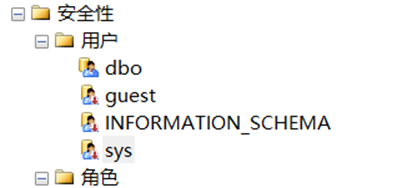
**select**





**(10) 分别在图形化界面、命令行中，删除某数据库用户。查看有什么结果。**





**(11) 分别采用Windows认证方式和SQL Server认证方式用不同的登录进行登录连接，重做上述内容，比较两种登录有不同吗？**

windows用户登录默认windows管理员为数据库的管理员，可以进行一切操作  
而sql身份验证创建的用户需要指定相应的权限，比如用sa用户登录也可以对数据库进行所有操作，如果是使用自己创建的用户登录，则只能执行该用户被授权的操作。如上述实验中创建的【wzy】由于被授权了可以创建数据库，而【Boogie】由于没有被授权所以无法新建数据库。

### 实验总结

对于服务器角色，登录名。数据库角色，数据库用户的概念理解不清晰。有一个很形象的比喻登录帐号是对于服务器而言的，数据库用户是针对特定数据库来讲的。就相当于一个房间里放着很多保险柜，你有房门钥匙了，必须得有每个保险柜的钥匙才能从保险柜里取东西。登录帐户是房门钥匙，数据库用户是保险柜钥匙。服务器角色和数据库角色我把它理解为权限，当被赋予这些角色之后，登录名或者用户就会拥有这些权限。

切换用户时，提示无法连接默认数据库，修改默认数据库后，应该对所需数据库做个映射。连接之后依旧无法完成授予权限的操作，是因为只对服务器用户授权，并未对数据库用户授权。

通过此实验，我们对于数据的完整性和安全性有了基本认识，通过观察什么时候破坏了完整性，才知道在之后对数据库的操作中，怎样做才能避免破坏完整性，产生报错。通过性实验，我们了解了用户角色和用户权限的概念和意义。