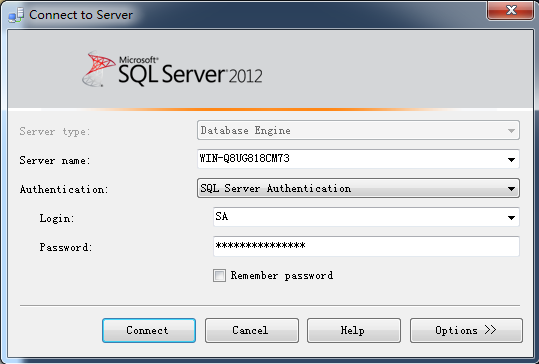
参考博文：<http://www.cnblogs.com/xwdreamer/archive/2012/06/25/2562828.html>

以超级管理员身份SA登录SQL Server。



超级管理员用户拥有数据库的全部权限，而普通用户，由SA创建，普通用户只拥有SA所分配的权限。SQL Server的安全和权限机制非常复杂，本次实验我们只需要简单了解。

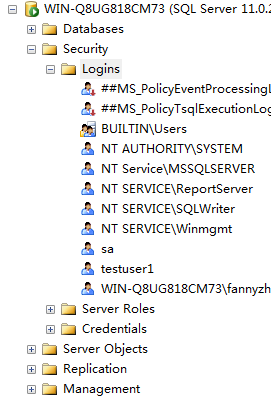
Step1：在SQL Server服务器级别，创建登录账户（create login）

-- 创建登录账户(create login)

create login testuser1 with password='123456',

default\_database=newtest

登录账户名为”testuser1”,默认连接到的数据库为”newtest”。



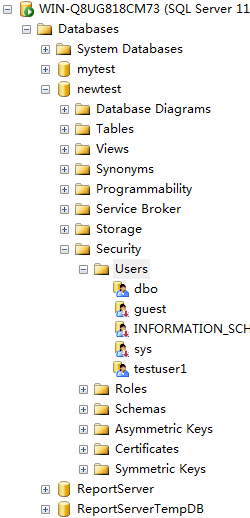
Step2：创建数据库用户（create user）

use newtest;

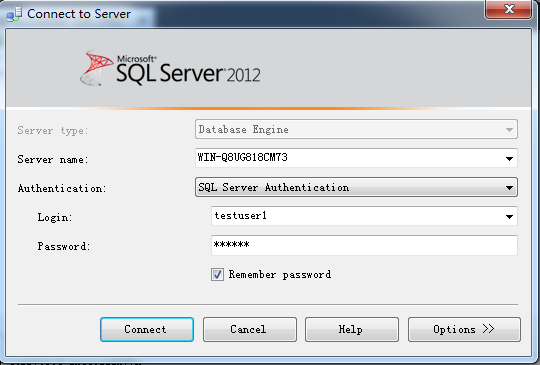
create user testuser1 for login testuser1 with

default\_schema=dbo

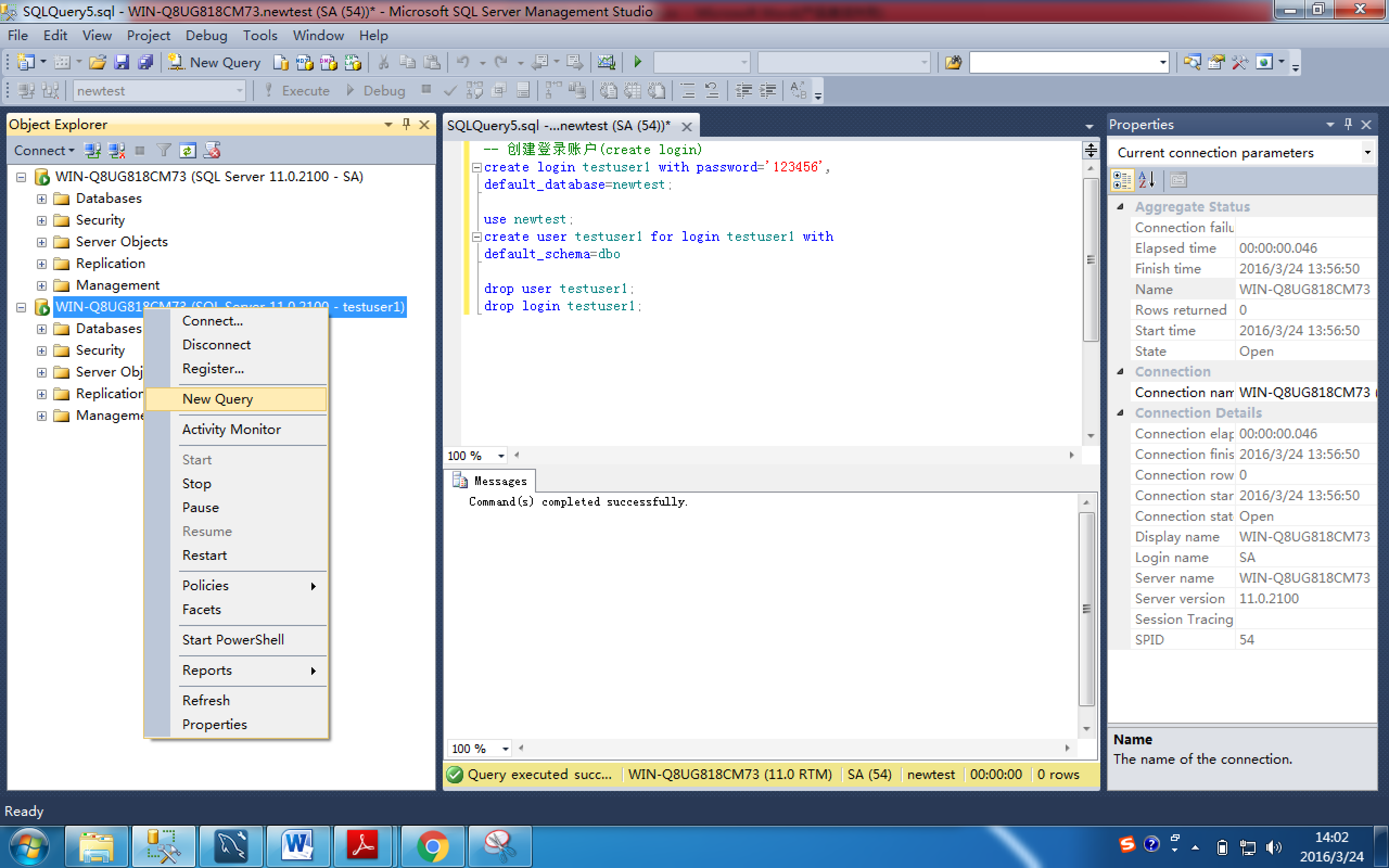
在数据库newtest内创建数据库用户名testuser1，并必须关联一个登录名（可不同名，建议同名）。默认schema是”dbo”。



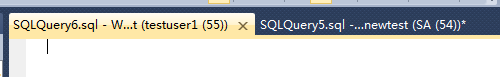
现在可以用testuser1登录了。



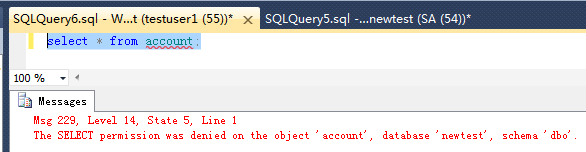
右键New Query。



注意两个SQLQuery登录名的不同。



这个新建的用户不能直接访问SA用户建立的表。



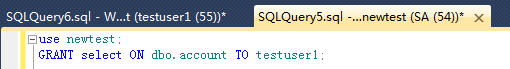
为此，需要回到SA身份，并为该用户授权。授权语句的语法：

GRANT priv\_type ON [object\_type] TO user;

现在为我们新建的用户授予数据库newtest下的表account的select权限。

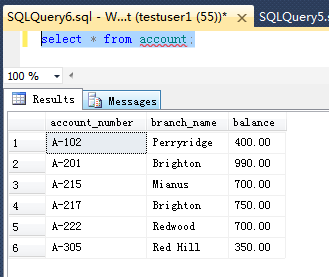
use newtest;

GRANT select ON dbo.account TO testuser1;



（有时候你可能没有意识到连接已经断开重连，所以对数据库、表做操作的过程中，如果出现了意料之外的报错，别忘了用use <database\_name>;语句试试。）

回到testuser1身份，现在testuser1也能查询SA用户建的表newtest.account了。



如果我们想回收权限，语法：

REVOKE priv\_type ON [object\_type] FROM user

我们回到SA身份，把表account的select权限回收：

REVOKE select ON dbo.account FROM testuser1;

此后再回到testuser1查询表account，发现访问被拒绝了。

1. **建立表，考察表的生成者拥有该表的哪些权限。**

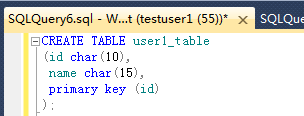
我们知道SA身份拥有所有的权限，所以我们想看看普通用户建立了表，他拥有该表的哪些权限。

Step3：通过加入数据库角色，赋予数据库用户”testuser1”权限

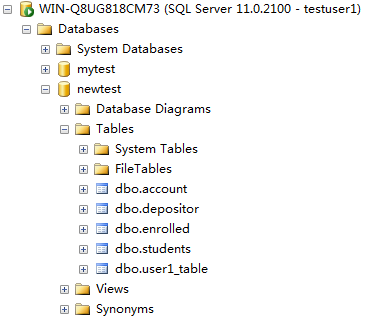
exec sp\_addrolemember 'db\_owner', 'testuser1';

此后，testuser1就可以全权管理数据库newtest中的对象了。

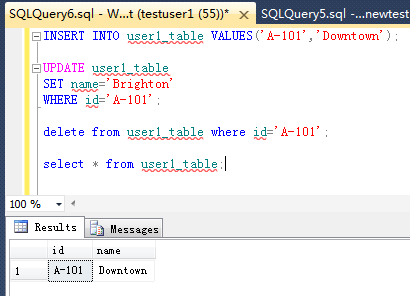
现在用testuser1身份建立表user1\_table，



在SA身份和testuser1身份刷新数据库，都能看到表user1\_table建立成功。



现在看看，testuser1对表user1\_table有哪些权限。



1. **使用SQL 的grant 和revoke命令对其他用户进行授权和权力回收，考察相应的作用。**

参见前面的内容介绍。

1. **建立视图，并把该视图的查询权限授予其他用户，考察通过视图进行权限控制的作用。**

简单举例：

SA用户建立了一张视图brighton\_account，现在把该视图的查询权限授予testuser1：

GRANT select ON dbo.brighton\_account TO testuser1;

用testuser1身份查询该视图：

