验七 用户和安全性管理

一、实验目的

1、了解 SQL Server 的安全性管理策略并大致能举一反三到其它数据库系统

2、理解用户和角色的含义

3、掌握如何创建用户和角色

二、预备知识

对任何企业组织来说，数据的安全性最为重要。安全性主要是指允许那些具有相应的数据访问权限的用户能够登录到SQL Server，并访问数据以及对数据库对象实施各种权限范围内的操作。但是要拒绝所有的非授权用户的非法操作。因此安全性管理与用户管理是密不可分的。SQL Server提供了内置的安全性和数据保护，主要提供了创建和管理用户账号，实现和管理安全性。

SQL Server的安全性管理是建立在认证和访问许可两者机制上的。认证是指来确定登录SQL Server 的用户的登录账号和密码是否正确，以此来验证其是否具有连接SQL Server 的权限。但是通过认证阶段并不代表能够访问SQL Server 中的数据，用户只有在获取访问数据库的权限之后才能够对服务器上的数据库进行权限许可下的各种操作。主要是针对数据库对象（如表、视图、存储过程等）。这种用户访问数据库权限的设置是通过用户账号来实现的。同时在SQL Server 中角色作为用户组的代名词大大地简化了安全性管理。

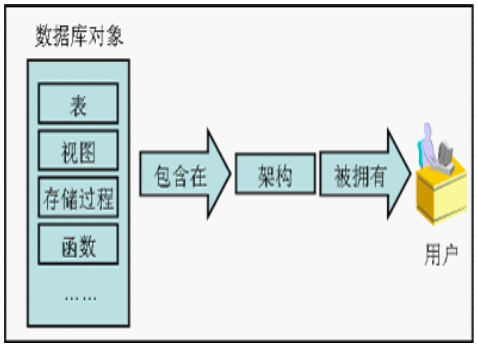
三、实验示例

1、SQL Server 的登录认证模式

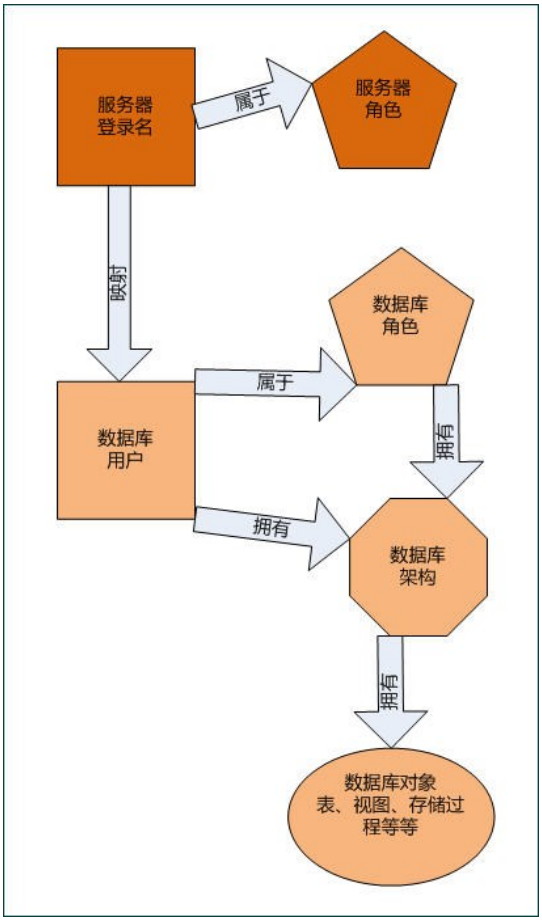
微软的SQL Server 能在两种安全模式下运行：

- Windows 认证模式：SQL Server 数据库系统通常运行在NT 服务器平台或基于NT构架的Windows 2000以上。而NT作为网络操作系统本身就具备管理登录验证用户合法性的能力。所以Windows认证模式正是利用这一用户安全性和账号管理的机制允许SQL Server 也可以使用NT的用户名和口令。在该模式下，用户只要通过Windows的认证就可连接到SQL Server。 而SQL Server本身就没有必要管理一套登录数据。自己可以试着操作在操作系统环境下建立用户，测试之。
- 混合模式：在混合认证模式下，Windows认证和SQL Server认证这两种认证模式都是可用的。NT的用户既可以使用NT认证，也可以使用SQL Server 认证。在SQL Server 认证模式下，用户在连接SQL Server 时必须提供登录名和登录密码。这些登录信息存储在系统表syslogins中，与NT的登录账号无关。SQL Server自己执行认证处理。如果输入的登录信息与系统表syslogins中的某条记录相匹配，则表明登录成功。

- 用户，架构，数据库对象之间关系：数据库中的对象由谁所有？如果由用户所有，那么当用户被删除时，其所拥有的对象怎么办呢？数据库对象可以成为没有所有者的"孤儿"吗？在Microsoft SQL Server 2008 系统中，这个问题是通过用户和架构分离来解决的。在该系统中，用户并不拥有数据库对象，架构可以拥有数据库对象。用户通过架构来使用数据库对象。这种机制使得删除用户时不必修改数据库对象的所有者，提高了数据库对象的可管理性。数据库对象、架构和用户之间的这种关系如下图所示。



服务器登录名，指有权限登录到某服务器的用户；
服务器角色，指一组固定的服务器用户，默认有9组；
登录名一定属于某些角色，默认为public
服务器角色不容许更改
登录后也不一定有权限操作数据库
数据库用户，指有权限能操作数据库的用户；
数据库角色，指一组固定的有某些权限的数据库角色；
数据库架构，指数据库对象的容器；
数据库用户对应于服务器登录名以便登录者可以操作数据库
数据库角色可以添加，可以定制不同权限
数据库架构，类似于数据库对象的命名空间，用户通过架构访问数据库对象
而通过下图可以让这些概念清晰一些：再如下图：



即：

服务器登录名属于某组服务器角色；

服务器登录名需要于数据库的用户映射后才拥有操作数据库的权限

数据库用户属于某组数据库角色以获取操作数据库的权限

数据库角色拥有对应的数据库架构，数据库用户可以通过角色直接拥有架构

数据库用户有默认架构，写SQL语句可以直接以“对象名”访问

非默认架构则要以“架构名.对象名”访问

【例7-1】SQL Server 认证模式的设置

在对登录进行增加删除等操作前，必须首先设置SQL Server的认证模式。通过SQL Server Enterprise Manager 来进行认证模式的设置。主要执行以下步骤：

（1） 启动企业管理器，选择要进行认证模式设置的“服务器”。

（2） 右击该服务器，在弹出菜单中选择“属性”，SQL Server 将弹出SQL Server 属性对话框。

（3）在SQL Server 属性对话框中选择安全性选项。如图7-1 所示

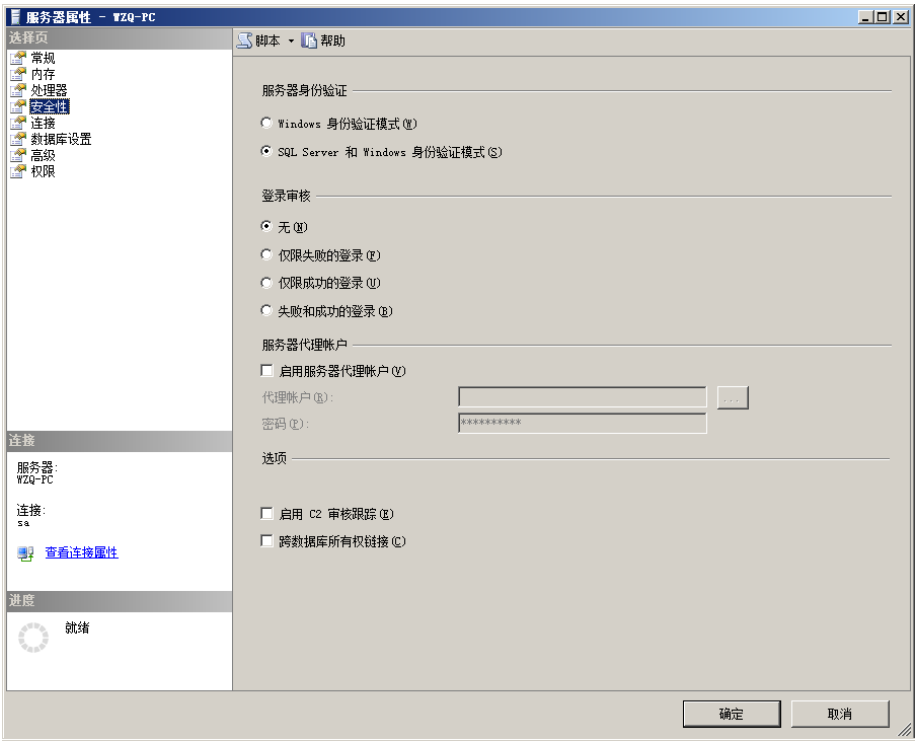


图 7-1 安全性设置对话框

（4） 在安全性选项栏的身份验证处选择要设置的认证模式。同时可以在审核级别处选择任意一个单选按钮来决定跟踪记录用户登录时的哪种信息，例如登录成功或失败的信息。

【例7-2】管理SQL Server 登录

在SQL Server 中通过SQL Server企业管理器中执行以下步骤来管理SQL Server 登录

（1） 启动SQL Server企业管理器中，单击登录服务器紧邻的+ 标志

（2） 单击安全性文件夹旁边的+ 标志

（3）右击登录名图标，从弹出菜单中选择新建登录选项，弹出新建登录对话框，如图7-2 所示

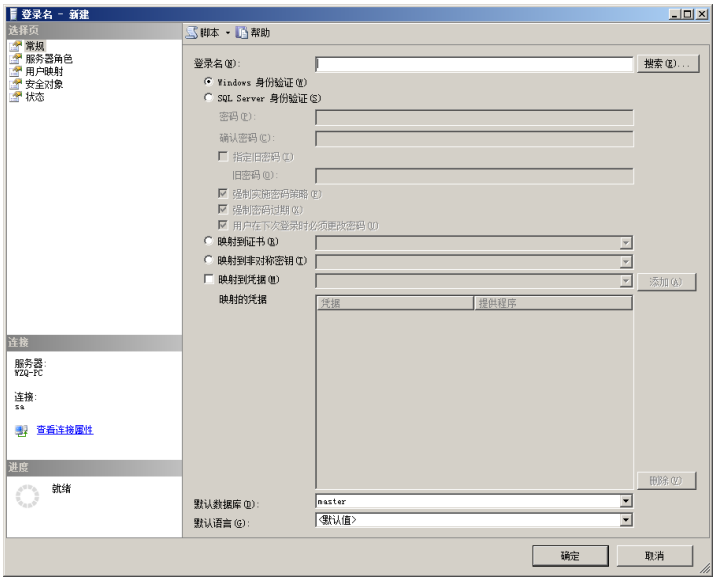


图 7-2 新建登录属性对话框

（4） 在名称编辑框中输入登录名。

（5） 在身份验证下的选项栏中，选择身份认证模式。如果使用SQL Server认证模式，那么在选择SQL Server身份验证单选按钮之后就必须在密码栏中输入密码。如果使用Windows 认证模式，那么在选择Windows 身份认证单选按钮之后，则必须在域中输入域名（SQL Server 2000下）。

（6） 在默认设置的两个选项框中，给出用户在登录时的默认数据库以及默认的语言。

（7） 单击确定按钮创建登录

2、数据库用户

数据库用户用来指出哪一个人可以访问哪一个数据库。在一个数据库中，用户ID唯一标识一个用户，用户对数据的访问权限以及对数据库对象的所有关系都是通过用户账号来控制的。用户账号总是基于数据库的，即两个不同数据库中可以有两个相同的用户账号。

在数据库中，用户账号与登录账号是两个不同的概念。一个合法的登录账号只表明该账号通过了NT 认证或SQL Server 认证，但不能表明其可以对数据库数据和数据对象进行某种或某些操作，所以一个登录账号总是与一个或多个数据库用户账号（这些账号必须分别存在相异的数据库中）相对应，这样才可以访问数据库。例如登录账号**sa自动与每一个数据库用户dbo** 相关联。

通常而言数据库用户账号总是与某一登录账号相关联，但有一个例外，那就是guest用户。在安装系统时，guest 用户被加入到master、pubs、tempdb和Northwind数据中。guest 用户主要是让那些没有属于自己的用户账号的SQL Server 登录者把其作为缺省的用户，从而使该登录者能够访问具有guest 用户的数据库。

关于登录名和数据库用户名区别（俗语化），比作工作人员进入大楼

没有登录名,进不了大楼--有了登录名,就能进大楼

没有用户名,进不了房间--有了用户名,才能进房间

大楼的钥匙可以在大楼内建立房间/删除房间，以及配置整个大楼的安保等功能。而用户只能对自己的房间进行收拾。

如果两者关联，就好比登录名进入房间，收拾房间。

【例 7-3】 使用“企业管理器”建立登陆帐号 User1 和用户 User1

打开“企业管理器”，选择“安全性”→“登录”→右键选“新建登录”，进入如图 7-3 所示的界面。名称填写“User1”，身份验证选择“SQL Server”，默认数据库选择“Stu_Cou”。默认的服务器角色是 public

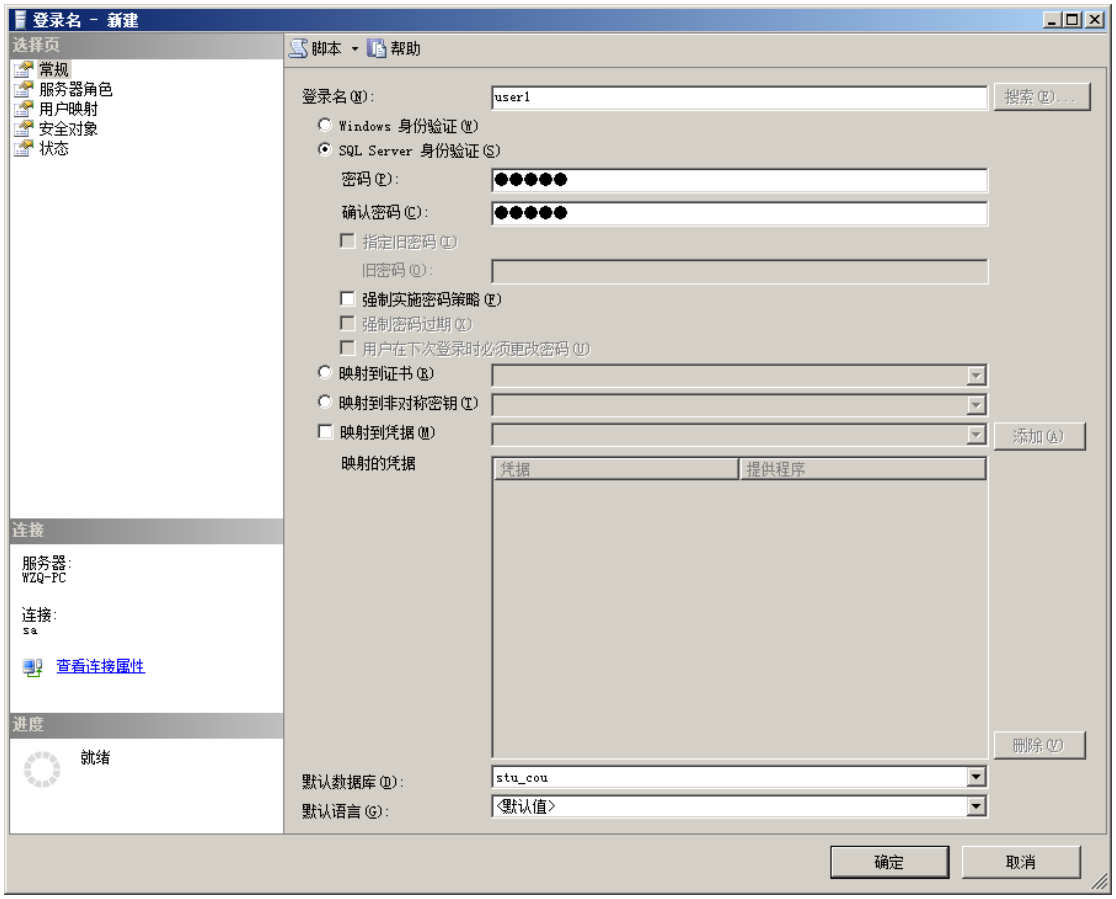


图 7-3 登录窗口（一）

然后在点击 stu_cou 数据库，点击安全性，再右键用户—>新建，进入数据库用户—新建，如图 7-4 所示，用户名为 User1。经确定后建立了一个 SQL Server 身份验证的登陆名“User1”。并且数据库 Stu_Cou 的用户名中新添了用户 “User1”。

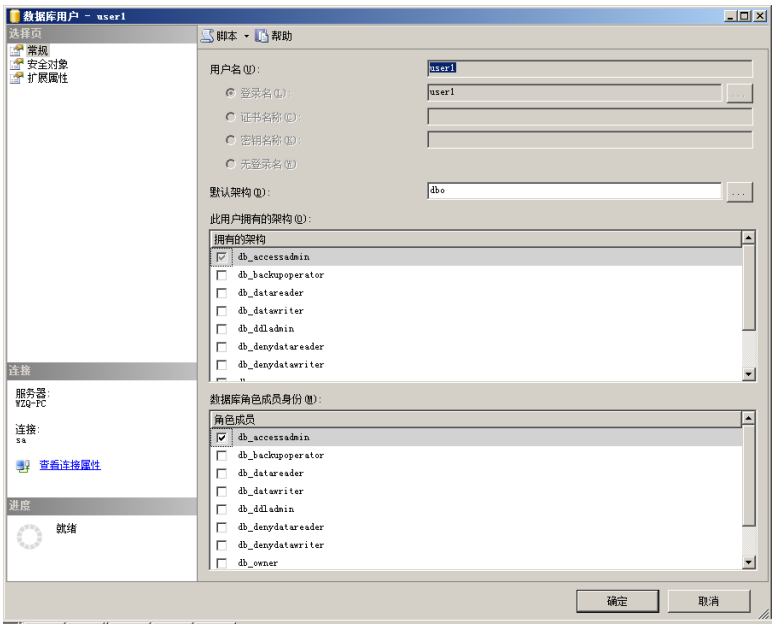


图 7-4　登录窗口（一）

默认的架构是“dbo”，点击【确定】创建完成，该用户出现在 Stu_Cou 数据库的【安全性】-【用户】节点中。这时，可以先用登录名访问数据库等

在图 7-4　登录窗口（一）中【数据库用户－新建】对话框的“选择页”中选择“安全对象”，进入权限设置页面（即“安全对象”页面），如图 7-4　登录窗口（二）所示。“安全对象页面”主要用于设置数据库用户拥有的能够访问的数据库对象以及相应的访问权限。单击“搜索”按钮为该用户添加数据库对象，并为添加的对象添加显示权限。

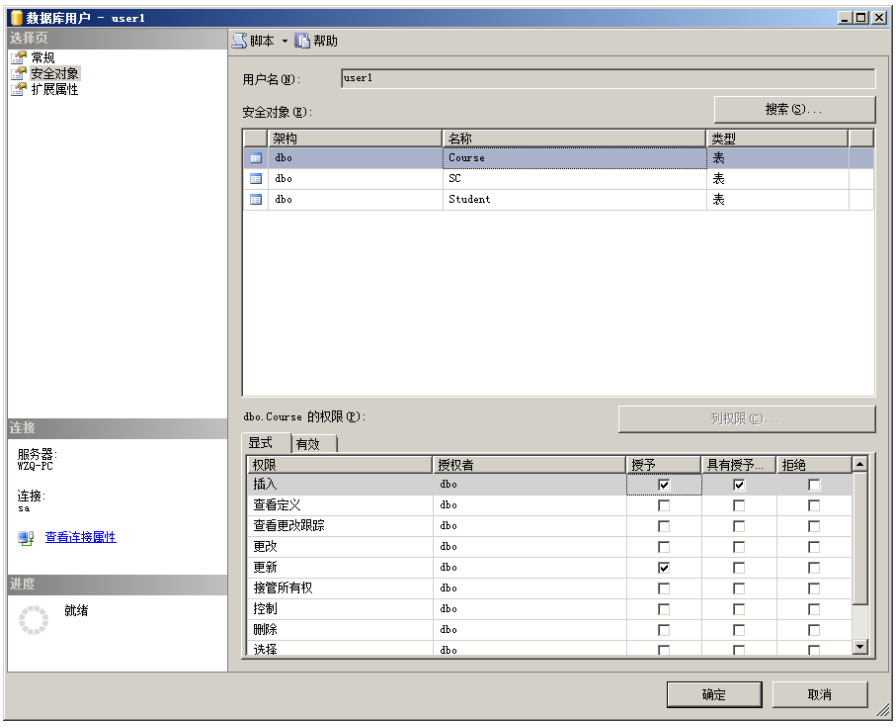


图 7-4　登录窗口（二）

请按照上述方式依次建立 SQL Server 身份验证的登录名 User2、User3、T1、T2、T3 以及数据库 Stu_Cou 中的新用户 User2、User3、T1、T2、T3。注意登录名和用户名本质上不

一样的。

【例7-4】 查看数据库用户

在企业管理器中选中User 图标，则在右面的窗格中显示当前数据库的所有用户，如图7-5 所示。

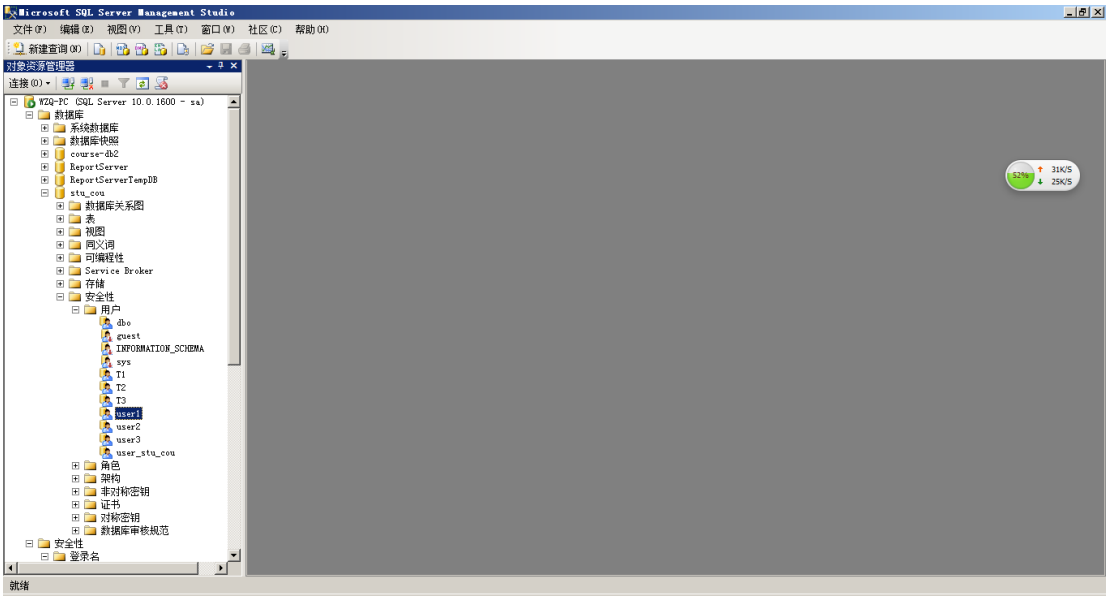


图 7-5 查看数据库用户

【例7-5】 删除数据库用户

选中User 图标后，在右面窗格中右击想要删除的数据库用户，则会弹出选项菜单，然后选择删除， 则会从当前数据库中删除该数据库用户，见图7-6。

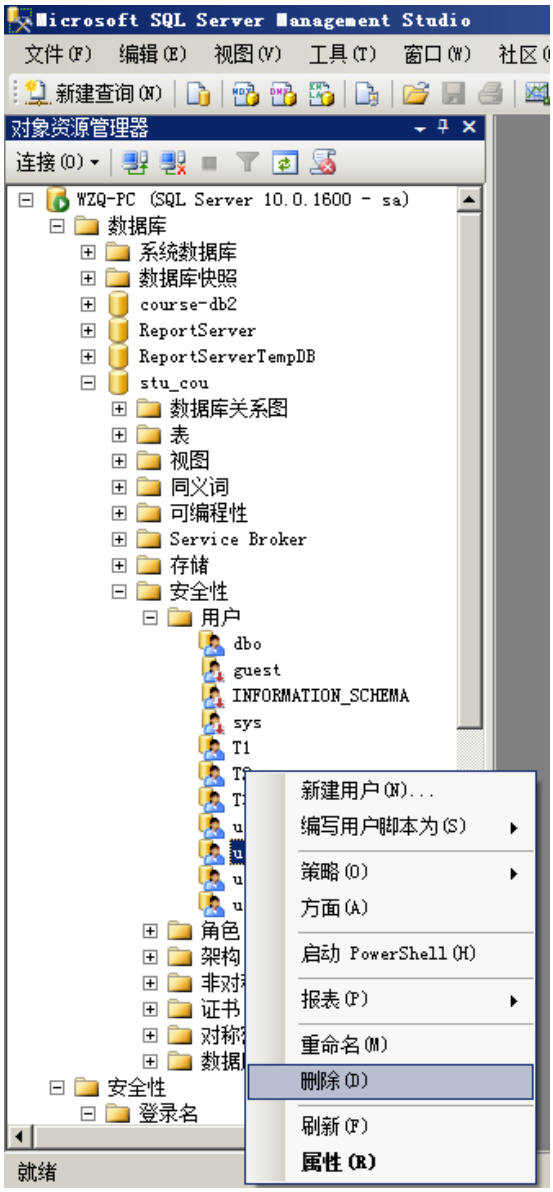


图 7-6 删除数据库用户

3、权限管理

用户在登录到SQL Server 之后，其用户账号所归属的NT组或角色所被授予的权限决定了该用户能够对哪些数据库对象执行哪种操作，以及能够访问、修改哪些数据。在SQL Server中包括两种类型的权限：即对象权限和语句权限。

（1）对象权限：对象权限总是针对表、视图、存储过程而言。它决定了能对表、视图、存储过程执行哪些操作（如UPDATE、DELETE、INSERT和EXECUTE）。如果用户想要对某一对象进行操作，其必须具有相应的操作的权限。例如，当用户要成功修改表中数据时，则前提条件是他已经被授予表的UPDATE权限。权限设置方法如图7-7。（见图7-4　登录窗口（二））

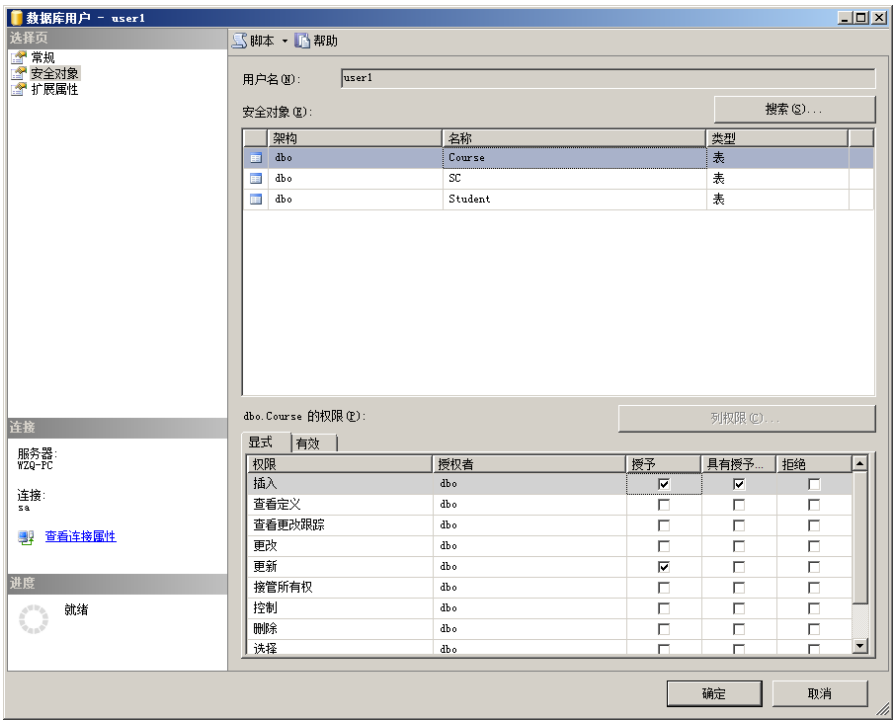


图 7-7 权限设置对话框

（2）语句权限：语句权限主要指用户是否具有权限来执行某一语句，这些语句通常是一些具有管理性的操作，如创建数据库、表、存储过程等。这种语句虽然仍包含有操作，如CREATE的对象，但这些对象在执行该语句之前并不存在于数据库中。如创建一个表，在CREATE TABLE 语句未成功执行前数据库中没有该表，所以将其归为语句权限范畴。下面是所有的语句权限。

- CREATE DATABASE 创建数据库
- CREATE TABLE 创建表
- CREATE VIEW 创建视图
- CREATE RULE 创建规则
- CREATE DEFAULT 创建缺省
- CREATE PROCEDURE 创建存储过程
- BACKUP DATABASE 备份数据库
- BACKUP LOG 备份事务日志

这些语句权限只有SQL Server的dbo用户才具有，一般的用户是不具有这些权限的。

（3）权限管理

在SQL Server中，使用GRANT、REVOKE和DENY 三种命令来管理权限。

- GRANT：用来把权限授予某一用户以允许该用户执行针对该对象的操作如UPDATE、SELECT、DELETE、EXECUTE等。
- REVOKE：取消用户对某一对象的权限，这些权限是经过GRANT 语句授予的不允许该用户执行针对数据库对象的某些操作，如UPDATE、SELECT、DELETE和EXECUTE等。
- DENY：用来禁止用户对某一对象的权限明确禁止其对某一用户对象执行某些操作。

4、角色管理

SQL Server管理者可以将某些(类)用户设置为某一角色，这样只对角色进行权限设置便可实现对所有用户权限的设置，大大减少了管理员的工作量。

SQL Server 2008 中角色分为 2 类，分别是：服务器级别的角色和数据库级别的角色。
服务器级别角色有已经定义好的 9 种，在【对象资源管理器】-【安全性】-【服务器】节点下查看。
数据库级别角色在具体的数据库的【安全性】-【角色】-【数据库角色】节点下查看。
【例 7-6】应用“企业管理器”建立角色 TEST
1、创建角色的步骤如下：
①选择数据库 Stu_Cou 的“角色”文件夹，右键在快捷菜单中选择“新建数据库角色”。
②如图 7-8 所示，输入角色名“TEST”，点击添加角色的用户 T1、T2、T3，其它默认，然后单击确定。

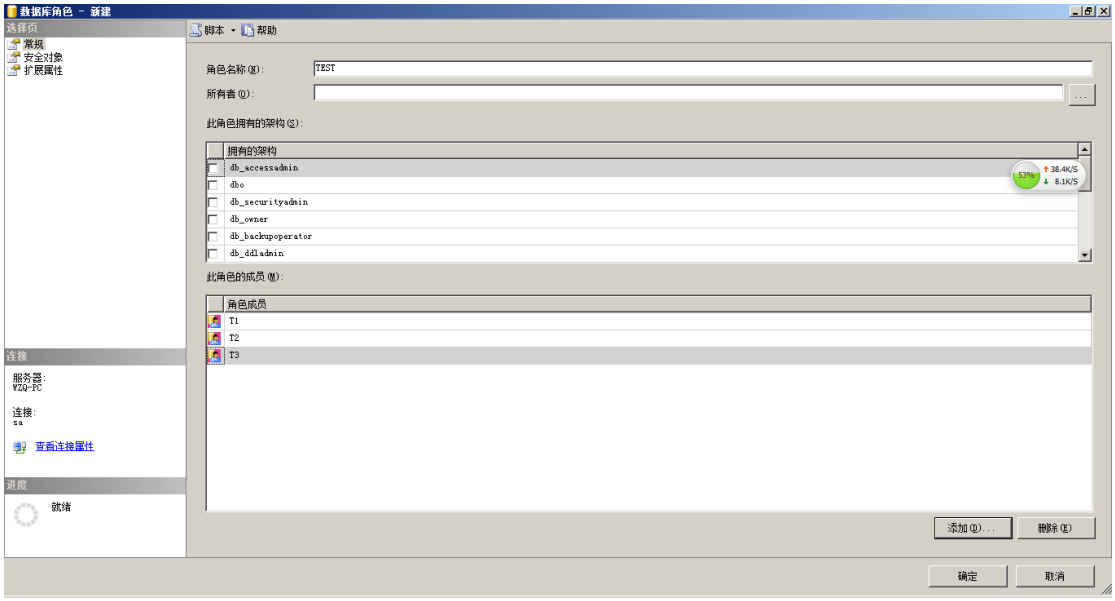
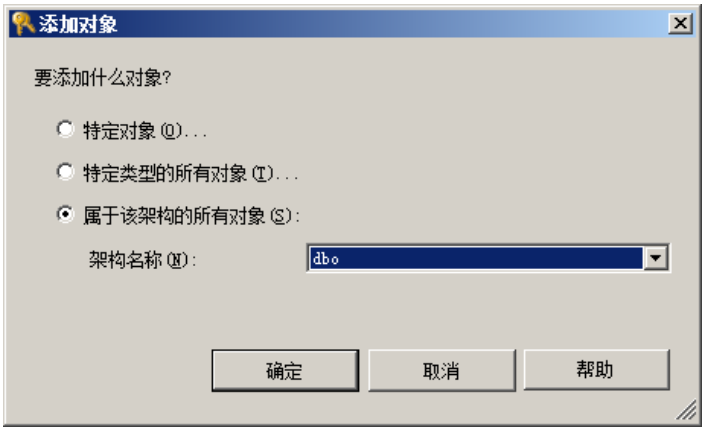


图 7-8　创建角色

2、选择角色的权限
①双击上步创建的角色“TEST”。
②单击“安全对象”按钮，弹出的对话框如图 7-9 所示。点击搜索按钮。此图如下设置，自己可以看看选择其它的区别，是对整个数据库等？



3 此窗口中选择表 Student 的权限 select、insert 和表 Course 的权限 select，单击确定即完成权限设定。

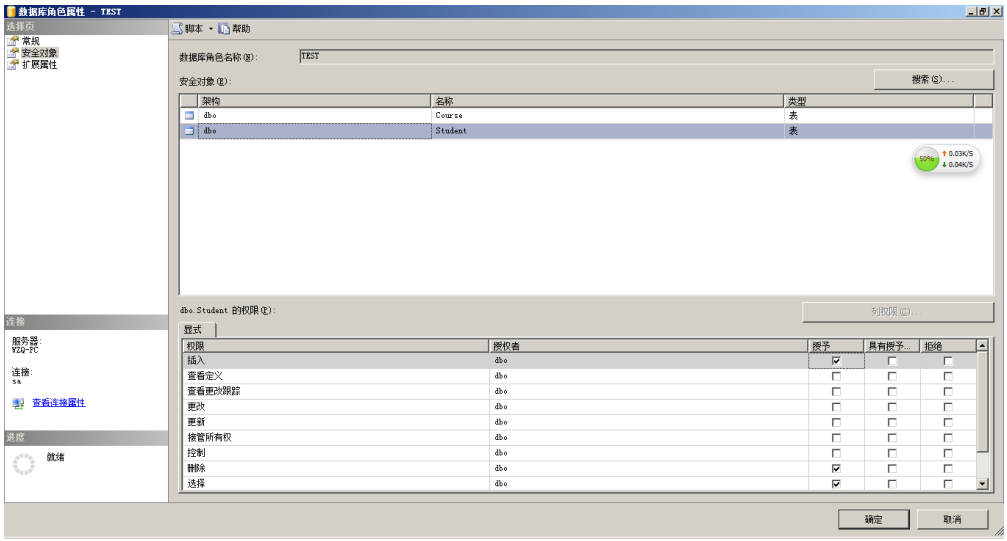


图 7-9　选择用户角色的权限

3、打开查询分析器，以 T1 身份连接数据库服务器，输入以下语句检查用户的权限

①查询 Student 表和 Course 表的内容，看结果如何？

```
use Stu_Cou
select * from Student
go
select * from Course
go
```

②在另一窗口中查询 SC 表的内容，看结果如何？

```
use Stu_Cou
select * from SC
go
```

实验八 权限控制

一、实验目的

1、掌握使用 grant 语句来为用户授权的方法

2、掌握使用 revoke 语句来为收回用户权限的方法

二、预备知识

1、创建用户的系统存储过程格式为：

sp_adduser User1

具体如下

CREATE USER 用户名

FOR LOGIN 登录名 WITH DEFAULT_SCHEMA=[dbo]

2、创建角色的系统存储过程格式为：

Create role….

sp_addrole Role1

3、grant 语句向用户授予操作权限，其一般格式为：（一定要记住）

grant <权限>[, <权限>]…

[on <对象类型> <对象名>]

to <用户> [, <用户>] …

[with grant option]

注：如果指定了 with grant option 子句，则获得某种权限的用户还可以把这种权限再授予其他用户。

4、revoke 语句表示可以由 DBA 或其他授权者收回权限。其一般格式为：

revoke <权限>[, <权限>]…

[on <对象类型> <对象名>]

from <用户> [, <用户>] …

3，4 必须记住

三、实验示例

【例 8-1】 打开 SQL Server Management Studio，以 SQL Server 认证登录－sa 用户(或者 Windows 身份连接数据库服务器登录)，把查询 Student 表的权限授权给用户 User1，并允许 User1 把这些权限授予别的用户

use Stu_Cou

grant select

on Student

to User1

with grant option

//自己通过视窗查看是否成功

【例 8-2】打开 SQL Server Management Studio，以 SQL Server 认证登录－sa 用户(或者 Windows 身份连接数据库服务器登录)，把对 Student 表的全部操作权限授予给用户 User2 和

User3
use Stu_Cou
grant SELECT,INSERT,UPDATE,DELETE
on Student
to User2 , User3

【例 8-3】打开 SQL Server Management Studio，以 SQL Server 认证登录－sa 用户(或者 Windows 身份连接数据库服务器登录)，把查询和修改 Student 表的权限授给用户 User4
use Stu_Cou
grant select , update
on Student
to User4

【例 8-4】打开 SQL Server Management Studio，以 SQL Server 认证登录－sa 用户(或者 Windows 身份连接数据库服务器登录)，把创建表的权限授给 User3
use Stu_Cou
grant create table
to User3
测试：然后以 User3 身份连接数据库服务器，并创建表 t1
use Stu_Cou
create table t1(sno char(8))
如果不行，在用户 user3 中添加数据库角色成员身份为 db_owner。在哪里？自己找。或者在 use Stu_Cou grant create table to User3 后加入 grant alter on schema :: dbo to User3）
【例 8-5】打开 SQL Server Management Studio，以 SQL Server 认证登录－sa 用户(或者 Windows 身份连接数据库服务器登录)，收回所有用户对表 SC 的查询权限
use Stu_Cou
revoke select
on student
from 相应的用户名，比如 user2
然后再以 User2 身份连接数据库服务器，并查询 student 表，结果如何？
use Stu_Cou
select * from student

【例 8-6】打开 SQL Server Management Studio，以 SQL Server 认证登录－sa 用户(或者 Windows 身份连接数据库服务器登录)，把用户 User4 修改学生学号的权限收回
use Stu_Cou
revoke update
on Student
from User4 cascade

注：cascade 收回用户拥有的这份特权