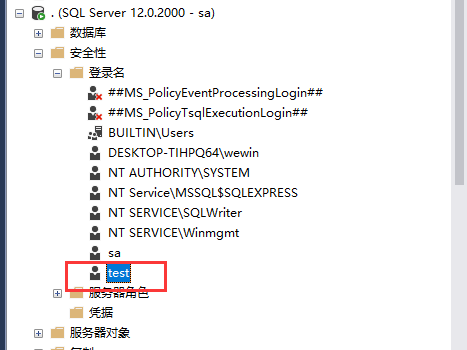
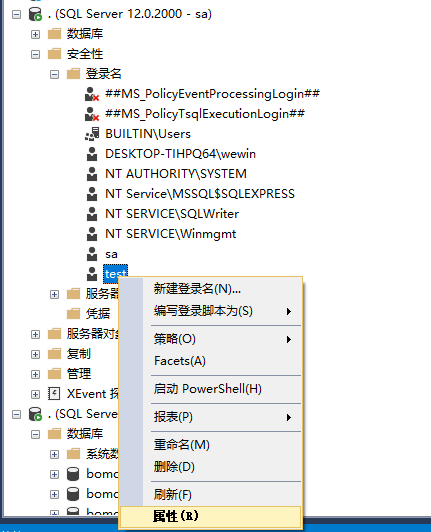
MSSQLServer中架构（模式）的总结笔记

1、查询表中的dbo，是数据库表的架构类型（Schema Type）,我们简称为数据库模式，即 Schema。他是针对用户来设置的。



2、点击用户（属性），来进行设置

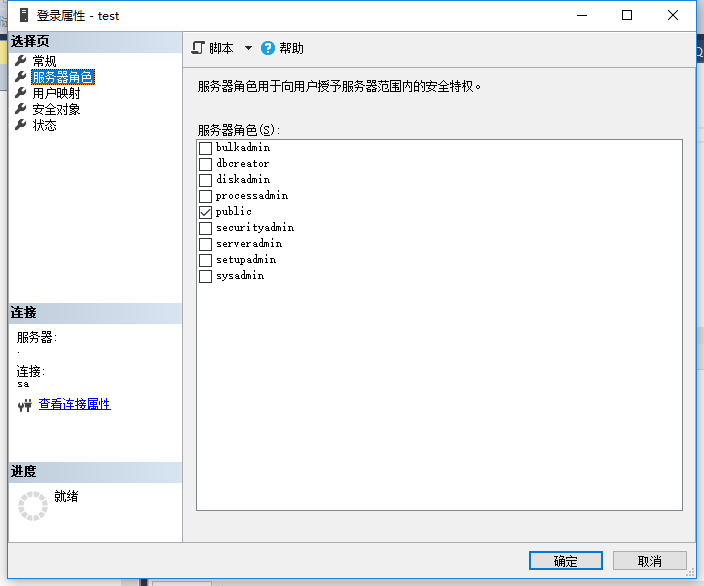


3、我们首先来设置一下“服务器的角色”，用户操作数据库服务的权限。

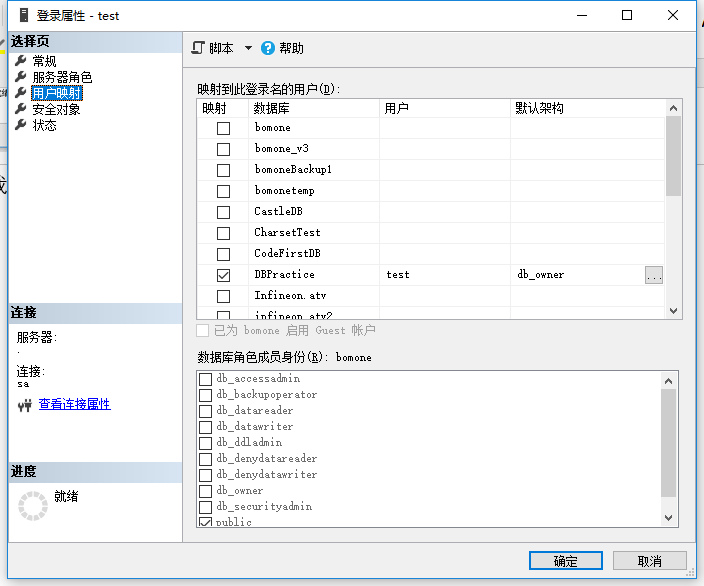
注意1：sysadmin是超级管理员角色，拥有所有的权限。

注意2：public 是否公开自己的用户信息，即：别的用户可以看到当前设置用户的存在。

注意3：其他的权限根据英文意思，直接翻译就是其对应的操作权限。

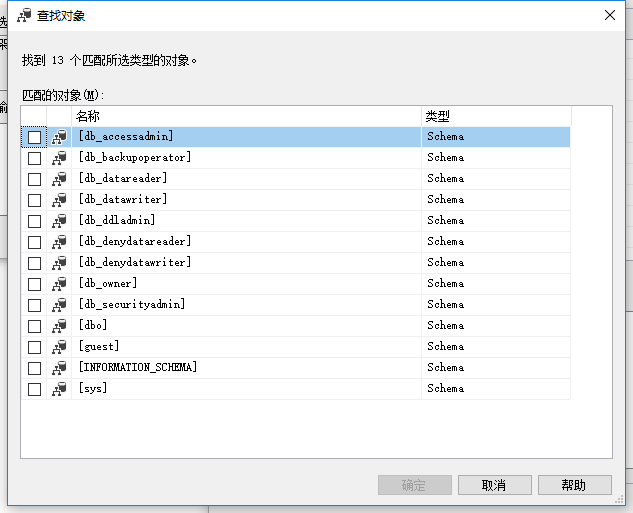


4、我们再来看一下“用户映射”，用户操作数据库的权限分配：



1. 我们给特定的用户开放 指定的数据库，并且分配 对数据库的操作权限（即：数据库角色成员身份）
2. 在上一步分配角色权限的时候，分配了一个基础的权限（public），当前就是要基于权限（public）增加额外的权限。如果一开始分配的就是sysadmin，那么这里就不用再分配了，超级管理用拥有所有的权限。
3. 接下来再说说“架构”的事情。我们在为用户指定查看数据的权限时候，每个数据库后面的都有两列（用户 和 默认架构）。系统的默认架构dbo（database owner的缩写）。但是现有的系统架构里面还有一个 架构(db\_owner)，这个是用户级别的私人拥有。

5、数据库中的架构，默认就是这几种。在实际的场景中，每个开发者都想拥有数据库的基本操作权限，否则很不爽。但是又不能影响系统的基本运行，那么就给不同的角色成员分配不同的架构（模式，Schema）



1. 例如：给用户test，分配了一个默认架构（db\_owner）,又分配了数据库定义（db\_ddladmin）的权限。就可以创建表结构了。用户test创建表，eg

create table TestTable1(

Id int primary key identity(1,1),

[Name] nvarchar(20)

)

1. 此时创建的表为：db\_owner.TestTable1
2. 当前我们没有分配查看表 和 以及写入的操作权限，所以不能查询、新增、删除、修改等操作。
3. 从表的结构上就可以知道，这张表示拥有数据库权限的人创建的。目的：是为了防止拥有权限的用户随意增加表结构，扰乱数据库的正常的结构。
4. 当然维护数据库的不仅仅是拥有数据库权限的人，也可以是一般的浏览用户，他们有时也需要创建一些供参考的表结构，此时我们可以给他开放数据库的操作权限，我们只需要修改他的默认架构（db\_reader），这样，当他创建一些表结构的时候，管理员是可以通过表的架构（模式）可以看到。而当前的这个用户在操作的时候也不需要加上这个架构前缀，直接使用表名就可以了。但是非该架构（模式）下的用户查询或其他操作表，都是要加上表的架构（模式）的。

例如：

--test 用户查数据：

select \* from TestTable1

--sa 超级管理员

select \* from db\_owner.TestTable1

--就是这样的差别，我们就知道，某个角色的管理员都干了点啥

6、其实对于小型的项目来说，这个压根不需要。大家都是用一个用户来登录的。使用的都是超级管理员角色的账号。使用的系统默认架构（模式）是dbo，不需要那么的费事。了解即可。