SQL Server 安全验证

验证是检验主体的过程。主体需要唯一标识，那样 SQL Server 可以确定主体有哪些权限。正确的验证是提供安全访问数据库对象的必要的第一步。

SQL Server 支持两种验证：Windows 身份验证和 SQL Server 身份验证。你使用的方式依赖于网络环境，将访问数据库的应用程序类型，以及这些应用程序的用户类型。

Windows 身份验证：这种形式的身份验证依赖于 Windows 来做验证身份。访问 SQL Server 对象的权限被分配给 Windows 登录名。这种类型的验证仅适用于 SQL Server 运行在支持 Windows NT 或 Kerberos 身份验证的 Windows 版本(Windows 2000 及之后)。

SQL Server 身份验证：SQL Server 可以完全依靠自己进行验证。在这种情况下，你可以创建独特的用户名称(SQL Server 称做登录名)和密码。用户或应用程序连接到 SQL Server 和访问所提供的凭据。权限直接给登录名或通过角色分配。

SQL Server 身份验证配置并不是一个简单地非此即彼的选择。你可以配置身份验证中的任何一种：

混合身份验证模式：服务器支持 SQL Server 和 Windows 身份验证。

Windows 身份验证模式：服务器只支持 Windows 身份验证。

微软强烈建议使用 Windows 身份验证。Windows 具有强大的身份验证选项，包括密码策略，但在实际应用中，Windows 验证并不总是可行的。SQL Server 身份验证可以引用一些 Windows 身份验证功能，但它是不安全的。

Windows 身份验证

如果你配置你的 SQL Server 在 Windows 身份验证模式操作，SQL Server 假设和 Windows 服务器的信任关系。当用户登录到 Windows 时，Windows 将验证用户的身份。SQL Server 检查用户帐户(任何 Windows 组，和任何 SQL Server 角色的用户)决定是否允许用户操作 SQL Server 对象。

Windows 身份验证比 SQL Server 身份验证的几个优势，包括：

->由用户单一登录，所以它不用分开登录到 SQL Server

->审计特点

->简化登录管理

->密码策略(在 Windows Server 2003 和之后版本)

Windows 身份验证的另一大优势是你对 Windows 用户和组的任何更改都会自动反映在 SQL Server，所以你不必分开管理。然而，如果你改变一个已经连接到 SQL Server 的 Windows 用户，这些更改不会生效，直到下一次用户连接到 SQL Server。

配置 SQL Server 的安全设置

当你安装 SQL Server，你可以选择服务器实例将允许身份验证模式。你也可以在 SSMS 的服务器属性对话框更改设置。这些设置适用于 SQL Server 实例中的所有数据库和其他对象。所以如果你需要对任何数据库使用 SQL Server 身份验证，你需要为服务器设置混合模式。

图 2.1 显示了服务器属性对话框安全性页面。要打开这个对话框，在对象资源管理器中右键单击服务器实例名称，然后从弹出的菜单中选择“属性”，然后转到安全性页面。你可以通过单击适当的按钮来更改身份验证模式，然后单击“确定”提交更改。

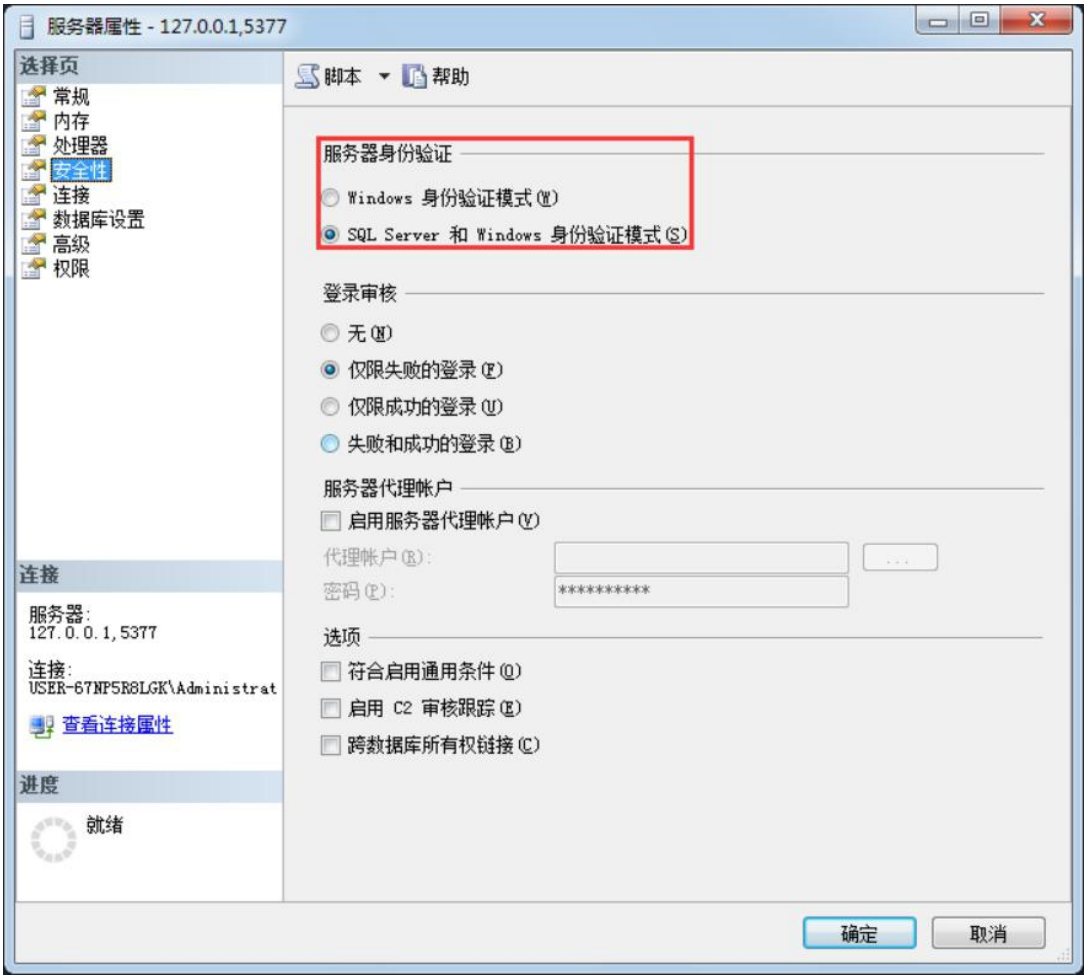

图 2.1 配置 SQL Server 实例的验证模式

添加 Windows 登录

为了使用 Windows 身份验证，你首先需要一个有效的 Windows 登录帐户。然后你可以将权限授予一个 Windows 组连接到 SQL Server，或者你可以对单个 Windows 用户授予权限，如果你不想给集体权限。

一个关于使用 Management Studio 管理安全的好处，你可以设置登录的同时提供数据库访问。为了使 Windows 登录访问 SQL Server 和 AdventureWorks2012 数据库，使用以下步骤，假设本地机器上已经定义 ClearFile 登录。

1、打开 SSMS 确保对象资源管理器窗口是可见的，并且你已经连接到一个 SQL Server 实例

2、展开服务器对象，然后展开安全性。你会看到几个子节点，如图 2.2 所示

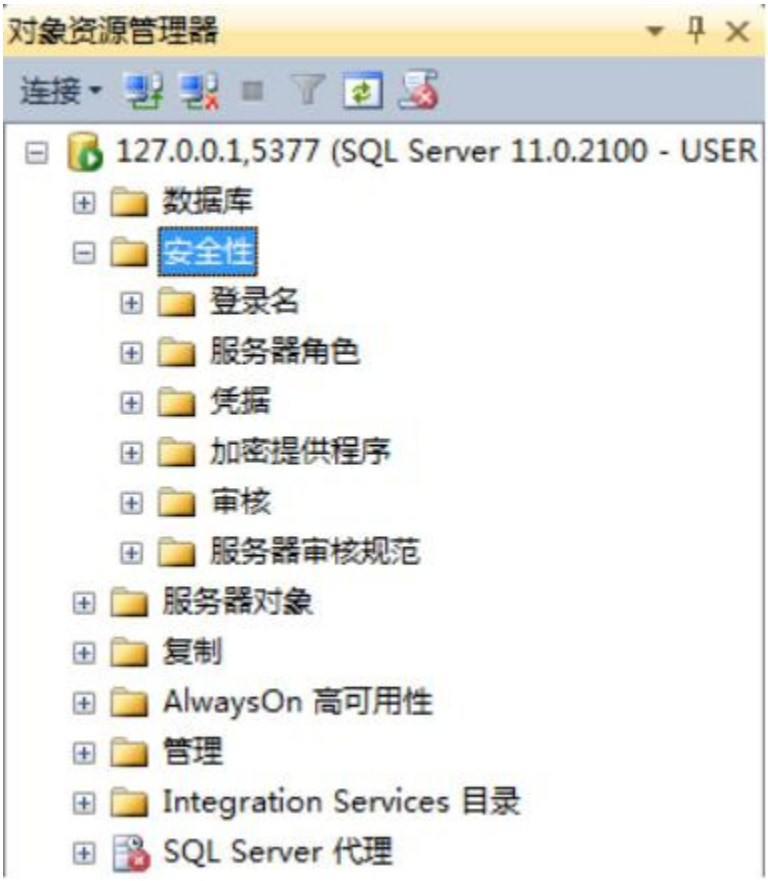

图 2.2 对象资源管理器下的安全性

3、右键单击登录名节点，从弹出菜单选择新建登录名，打开登录名-新建对话框

4、确保选择 Windows 身份验证

5、有两种方式选择 Windows 登录名。第一种方式是直接输入域或计算机名称，然后一个反斜杠和 Windows 登录名。第二种方法是单击“搜索”按钮打开“选择用户或组”对话框中。键入用户名，然后单击“检查名称”按钮以找到确切的名称。如果用户被发现，将出现在该框中，如图 2.3 所示。单击“确定”选择该用户。

图 2.3 SQL Server 查找 Windows 登录名

6、退回到登录名-新建对话框，设置 AdventureWorks2012 数据库为默认数据库。这是当用户连接到服务器时并且没有指定数据库的情况下使用的数据库。它不限制用户只访问这一个数据库。图 2.4 显示了我本地设置的 Windows 登录名。

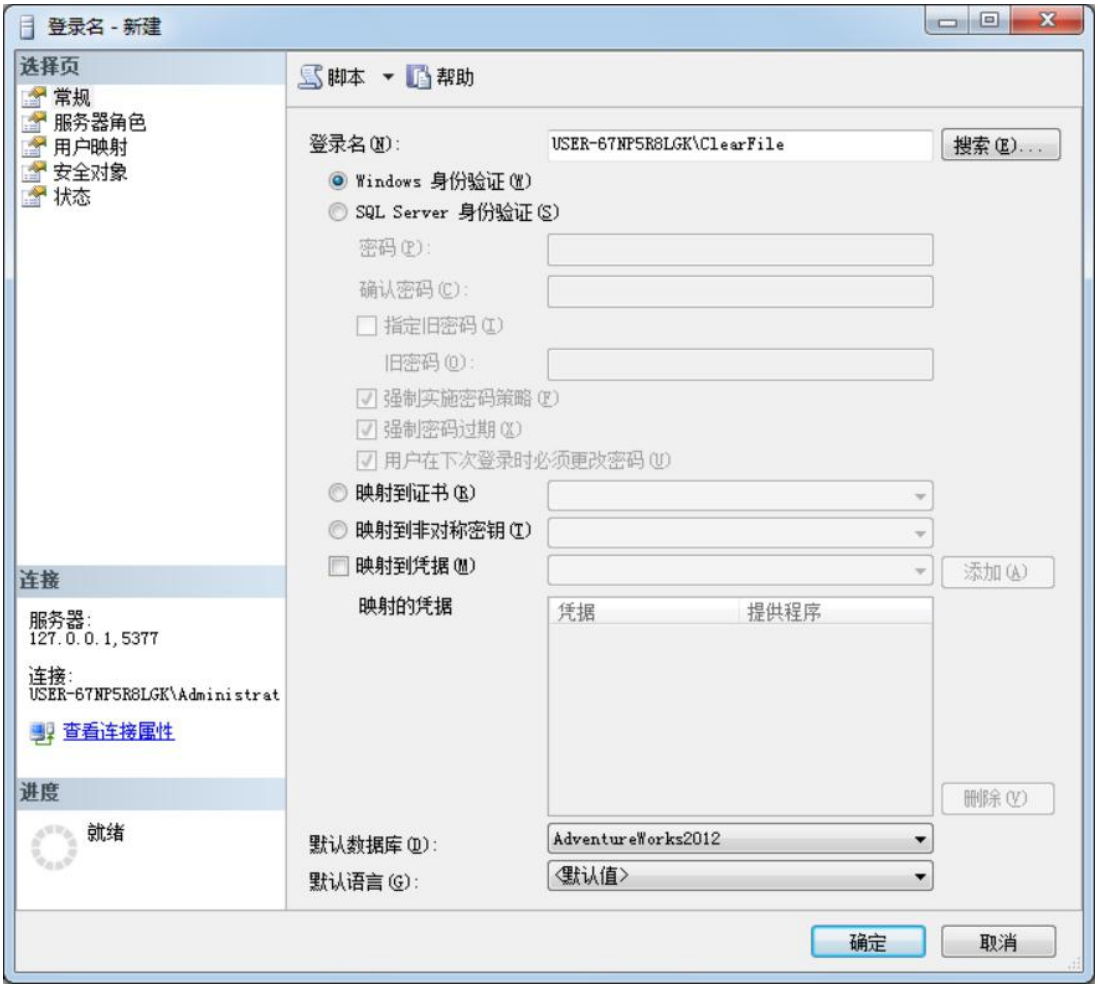

图 2.4 新建 Windows 身份验证登录

注意：永远不要把默认数据库设置为 master。我以痛苦的经验告知：太容易连接到一个服务器，忘记改变数据库。如果你运行一个脚本，在 master 数据库中创建数百个数据库对象，手动删除这些对象以清理 master 数据库，将会是一项繁琐的工作。

7、下一步，让用户访问数据库。从对话框左侧的列表中选择用户映射。通过数据库旁边的映射授予用户访问 AdventureWorks2012 数据库。SQL Server 会自动在数据库下创建一同名用户，并与之映射。你可以看到在表中的第三列，当然你也可以更改用户名。分配 Sales 作为用户的默认架构 ,通过键入或单击省略号(...)按钮从列表中选择。对话框应该看起来像图 2.5。

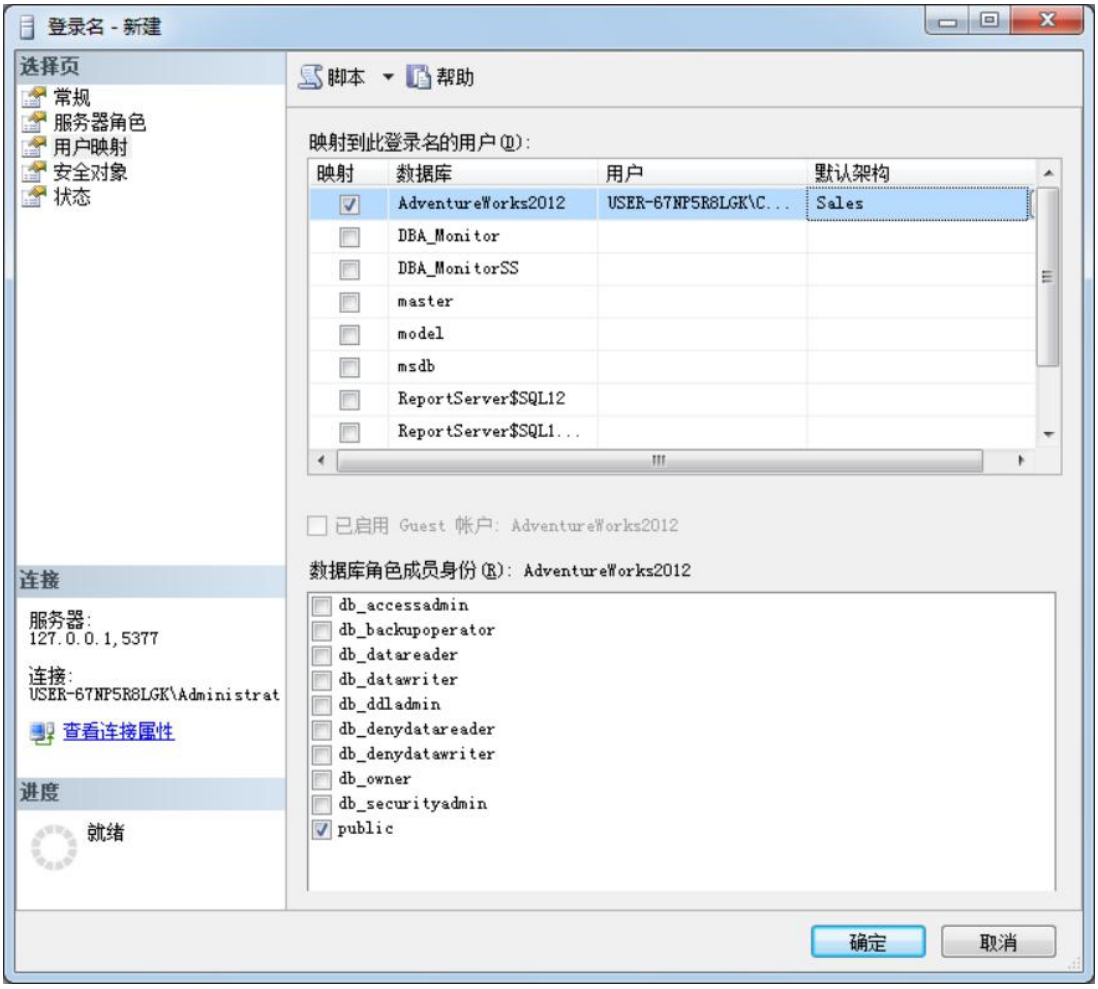

图 2.5 给 Windows 登录名 AdventureWorks2012 访问数据库

注意：为登录名设置默认数据库和授权访问数据库是有区别的。默认的数据库仅仅意味着当用户登录时没有指定数据库，SQL Server 试图改变上下文到默认数据库。但这并没有授予任何在数据库中做任何事情的权限，甚至允许访问数据库。这意味着，它有可能分配一个默认的数据库，但用户不能访问。为用户在访问数据库时做任何有用的事，你需要显式地授予用户权限。

8、默认情况下，新的 Windows 登录名能访问服务器。但是，如果你想显式拒绝登录访问服务器，在登录名-新建对话框左侧选择状态，选择“拒绝”按钮。你还可以通过选择禁用按钮来暂时禁用该登录名。图 2.6 显示了这些选项。

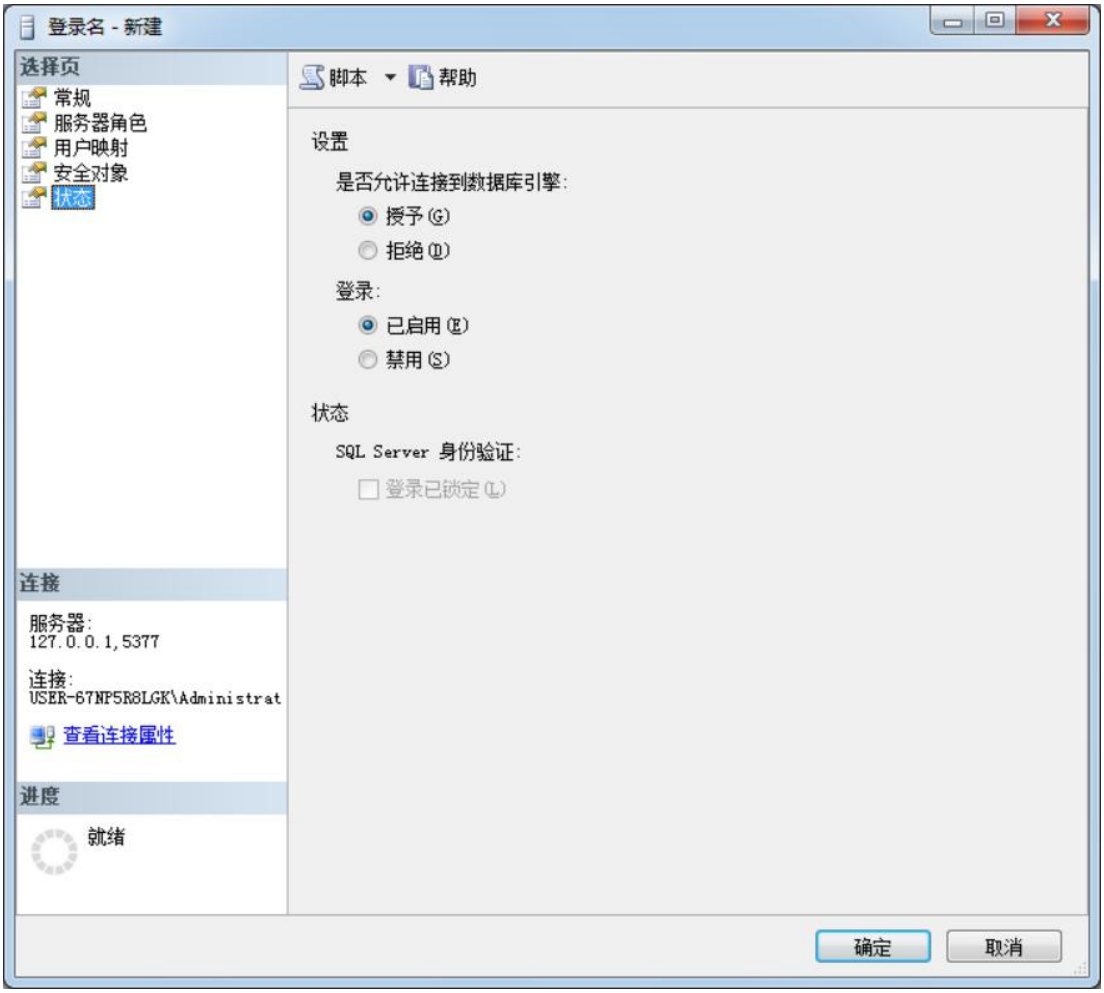

图 2.6 授予/拒绝连接到数据库引擎或禁用登录名

9、点击确定创建用户

你还可以以同样的方式添加一个 Windows 组。在这种情况下，该组中的任何成员都能访问数据库服务器，只要你授予了组访问数据库对象的权限。

SQL Server 身份验证

当你使用 SQL Server 登录认证，客户端应用程序必须提供一个有效的用户名和密码以连接到数据库。这些 SQL Server 登录名保存在 SQL Server。在登录时，如果没有帐户匹配用户名和密码，SQL Server 将抛出错误，并且用户无法访问 SQL Server。

尽管 Windows 身份验证更安全，在某些情况下你可以选择使用 SQL Server

登录。对于不需要特别安全需求的简单应用,SQL Server 身份验证更容易管理，它可以让你避免纠缠于 Windows 安全。如果客户端是在旧版本的 Windows 上运行(比 Windows 2000 以前)或非 Windows 操作系统，你必须使用 SQL Server 登录。

为了创建一个 SQL Server 登录名，使用与 Windows 登录名相同的对话框，登录名-新建。输入一个独特的登录名(没有域名或机器名)，并提供密码。例如，图 2.7 显示了如何创建一个新的 SQL Server 登录 Casper ,并将 AdventureWorks2012 设置为默认数据库。

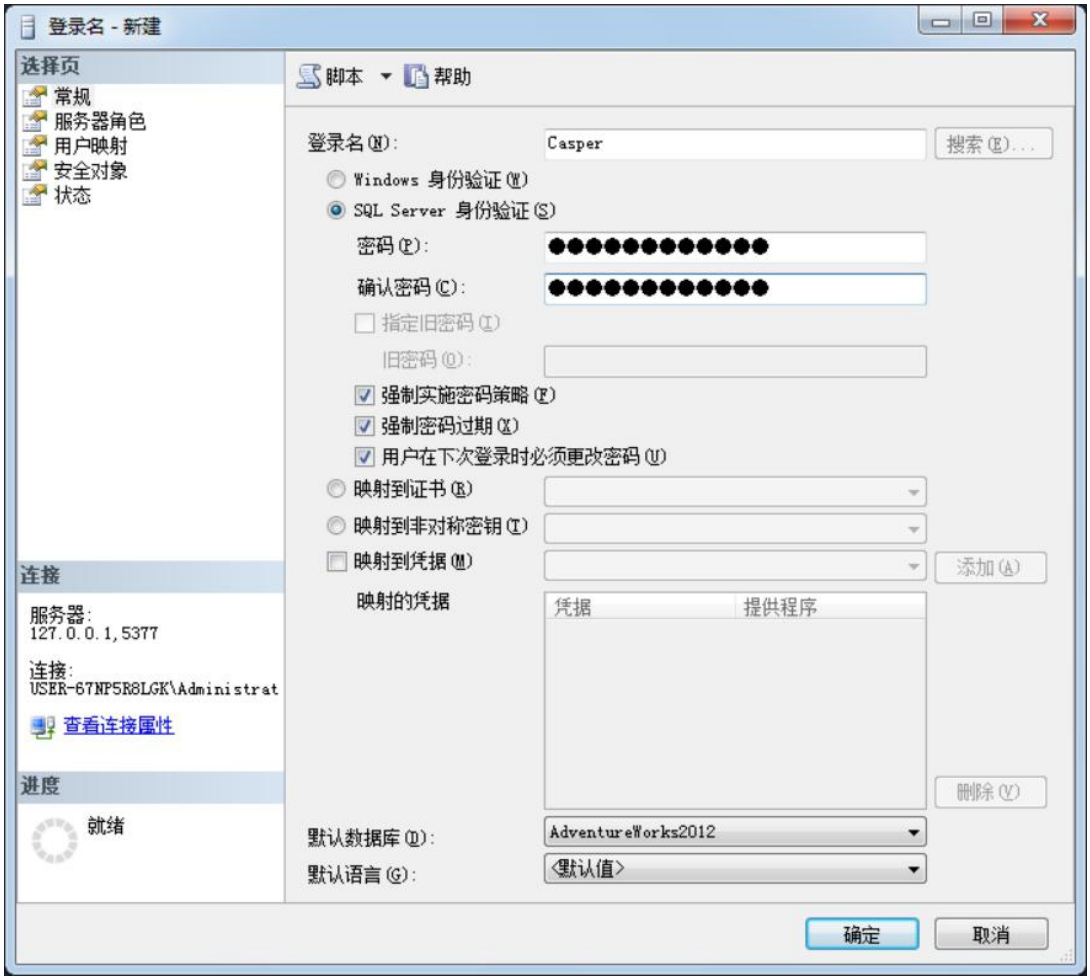

图 2.7 创建 SQL Server 登录名

SQL Server 登录名的用户映射和状态选项和 Windows 登录名一样。

通过 T-SQL 语句创建 SQL Server 登录名

你还可以使用 T-SQL 代码来执行同样的操作 ,代码 2.1 使用 CREATE LOGIN 创建了一个名为 Topaz 的 SQL Server 登录名

```
CREATE LOGIN Topaz WITH PASSWORD = 'yBqyZIPT8}b]b[{5al0v';GO
```

代码 2.1 T-SQL 语句创建 SQL Server 登录名

然后给 Topaz 访问 AdventureWorks2012 数据库的权限，使用 CREATE USER 语句并分配默认架构，如代码 2.2 所示

```
USE AdventureWorks2012;GOCREATE USER Topaz FOR LOGIN Topaz
    WITH DEFAULT_SCHEMA = HumanResources;GO
```

代码 2.2 为登录名创建数据库用户

类似 Windows 登录名，你可以将登录名映射给不同名的数据库用户。代码 2.3 中将 Topaz 登录名映射给 AdventureWorks2012 数据库的 TopazD 用户

```
DROP USER Topaz;GOCREATE USER TopazD FOR LOGIN Topaz WITH
DEFAULT_SCHEMA = HumanResources;GO
```

代码 2.3 删除已存在的数据库用户，然后创建与登录不同名的数据库用户

当心 sa 登录名

如果你配置你的 SQL Server 支持 SQL Server 登录，你需要当心内置的 SQL Server 登录——sa 登录，你可能已经注意到对象资源管理器登录节点。包含 sa，或系统管理员登录主要是对 SQL Server 旧版本的向后兼容性。sa 登录是映射到 sysadmin 固定服务器角色，任何以 sa 登录到 SQL Server 是一个全系统管理员，可以访问整个 SQL Server 实例和所有的数据库。

你不能修改或删除 sa 登录名。在你安装 SQL Server 实例的时候如果你选择混合模式身份验证，系统会提示你为 sa 用户设置密码。如果没有设置密码，任何人都可以用 sa 登录，玩“让我们管理服务器”。不用说，这是你最不想让你的

用户做的最后一件事。如果其他系统管理员不可用或已经忘记了他们的 windows 密码，才会用 sa 作为后门登录。如果发生这种情况，你可能需要新的管理员！

永远不要在应用程序使用 sa 登录名访问数据库。这样做可能让黑客控制你的数据库服务器，如果黑客能够控制应用程序。在遥远的过去，这已经是攻击服务器的简单方法，是一个可怕的做法。相反，无论是设置了自定义 Windows 或 SQL Server 登录名，给登录名绝对最低权限运行应用程序(实现最小特权原则)。

注意：事实上，你应该考虑禁用 sa 登录名。这样攻击者就无法使用这个强大的登录名来控制你的服务器实例，无论你是否有一个强大的密码。

强制密码策略

在 SQL Server 2005 之前的版本，系统管理员没有简单的方法强制密码策略以帮助系统更安全。例如，SQL Server 没有办法强迫用户创建一个强密码：最小长度和混合字母数字和其他字符。如果有人想创建一个登录用一个字母作为密码，你不能配置 SQL Server 来阻止它。同样地，没有办法使密码定期过期，如每三个月。有些人理所当然地认为这是不使用 SQL Server 登录的主要原因。

SQL Server 的最近版本可以使用 Windows Server 2003 ,Windows Vista(或之后版本)的密码策略。密码仍然存储在 SQL Server，但 SQL Server 调用 Windows API 的 NetValidatePasswordPolicy()方法，首次在 Windows Server 2003 推出。这个 API 功能适用于 Windows 密码策略给 SQL Server 登录名并返回一个值，指示密码是否有效。当用户创建、设置、或重置密码时，SQL Server 调用这个函数。

你可以通过控制面板>管理工具>本地安全策略(secpol.msc)>密码策略，来定义你的 Windows 密码策略。图 2.8 中显示了密码策略部分默认设置。图 2.9 显示了帐户锁定策略，当用户多次登录不成功时，会被锁定。默认情况下，在一个新

安装的 Windows 系统中锁定策略是被禁用的。

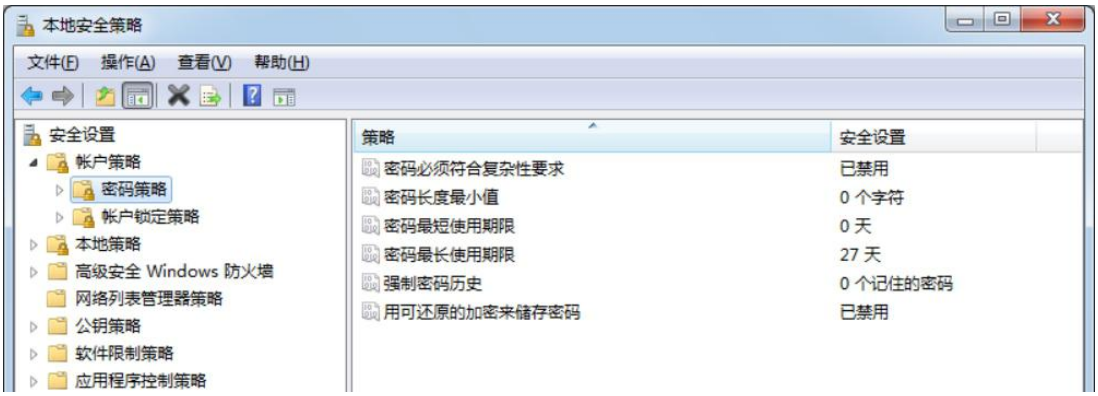

图 2.8 密码策略

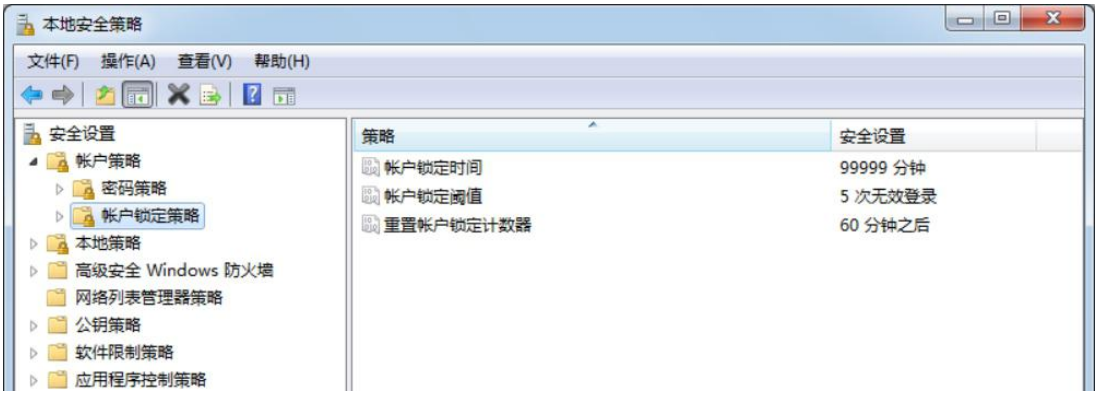

图 2.9 帐户锁定策略

表 2.1 罗列了部分密码策略，默认值以及相应的说明

类别	策略	默认值	说明
密码策略	强制密码历史	0个记住的密码	此策略使管理员能够通过确保旧密码不被连续重新使用来增强安全性
	密码长度最小值	0个字符	此安全设置确定用户帐户密码包含的最少字符数
	密码必须符合复杂性要求	禁用	此安全设置确定密码是否必须符合复杂性要求
密码过期	密码最长使用期限	42天	此安全设置确定在系统要求用户更改某个密码之前可以使用该密码的期间
	密码最短使用期限	0天	此安全设置确定在用户更改某个密码之前必须使用该密码一段时间
帐户锁定策略	帐户锁定时间	无	此安全设置确定锁定帐户在自动解锁之前保持锁定的分钟数
	帐户锁定阈值	0次无效登录	此安全设置确定导致用户帐户被锁定的登录尝试失败的次数
	重置帐户锁定计数器	无	此安全设置确定在某次登录尝试失败之后将登录尝试失败计数器重置为 0 次错误登录尝试之前需要的时间

表 2.1 Windows 密码策略设置

当你创建登录时，可以启用或禁用密码策略强制执行。当你创建一个 SQL Server 登录时，登录名-新建对话框，在登录名的下面有一个选项，如图 2.10 所示

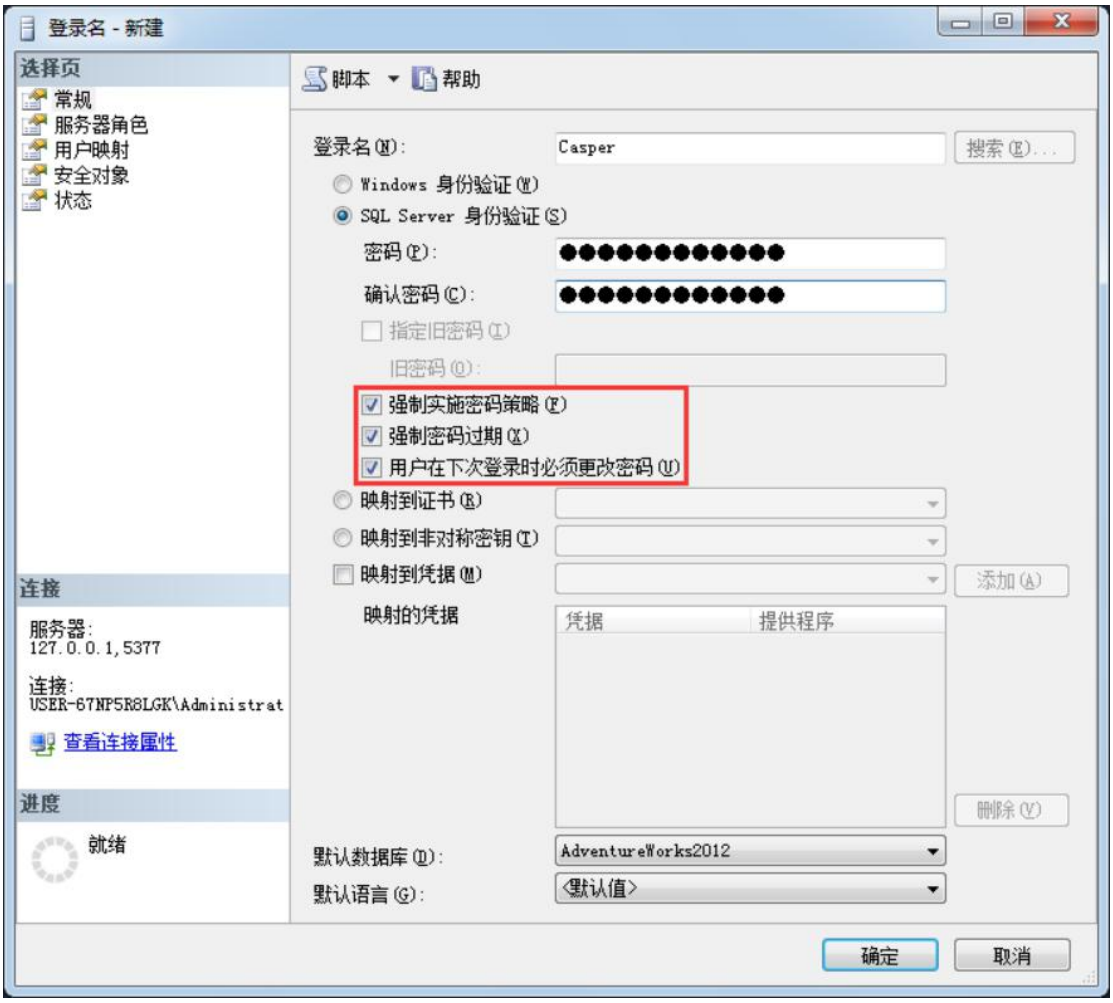

图 2.10 为新登录强制密码策略

当你使用 T-SQL 创建登录名时也会应用密码策略。例如，如果你在 Windows 2003 Server(或之后版本)运行 SQL Server，并且开启密码策略，代码 2.4 将失败

```
USE master;GOCREATE LOGIN SIMPLEPWD WITH PASSWORD = 'SIMPLEPWD';GO
```

代码 2.4 尝试违背密码策略创建数据库登录名

原因就是密码不能和用户名相同

你可以在创建或修改登录名的时候控制策略。代码 2.5 将关闭该检查过期和强制策略选项。

```
ALTER LOGIN Topaz WITH PASSWORD = 'yBqyZIPT8}b]b[{5al0v',
    CHECK_EXPIRATION = OFF, CHECK_POLICY = OFF;
```

代码 2.5 修改登录禁用密码策略

CHECK_EXPIRATION 选项控制 SQL Server 检查密码是否过期；CHECK_POLICY 选项适用于其他策略。MUST_CHANGE 选项，强迫使用户在下次登录时更改密码。

如果用户有多次不成功的登录，超过了帐户锁定策略中的数目，管理员可以使用解锁选项重置帐户，如代码 2.6 所示。

```
ALTER LOGIN Topaz WITH PASSWORD = 'yBqyZIPT8}b]b[{5al0v' UNLOCK
```

代码 2.6 解锁登录

在 Windows Server 2003 前，你可以开启 SQL Server 使用强制密码策略。但 SQL Server 使用默认设置的最小长度为 6 个字符，检查密码不匹配的登录名的全部或任何部分，是一个混合大写字母、小写字母、数字、和其他字符。你不能更改这些默认值。但我希望你不是这样一个旧版本的 Windows 上运行 SQL Server。