SQL Server 安全架构和安全

架构本质上是一个数据库对象，其他对象的一个容器，在复杂的数据库中它能够很容易的管理各组对象。架构具有重要的安全功能。

通常使用架构和对象名称来引用当前数据库下的对象。一个架构是一个对象集合，如表、代码模块，如图 5.1 所示。这种组织结构简化了用户管理，特别是当你需要改变对象的所有权时。但更重要的是为了安全，它简化了权限管理。

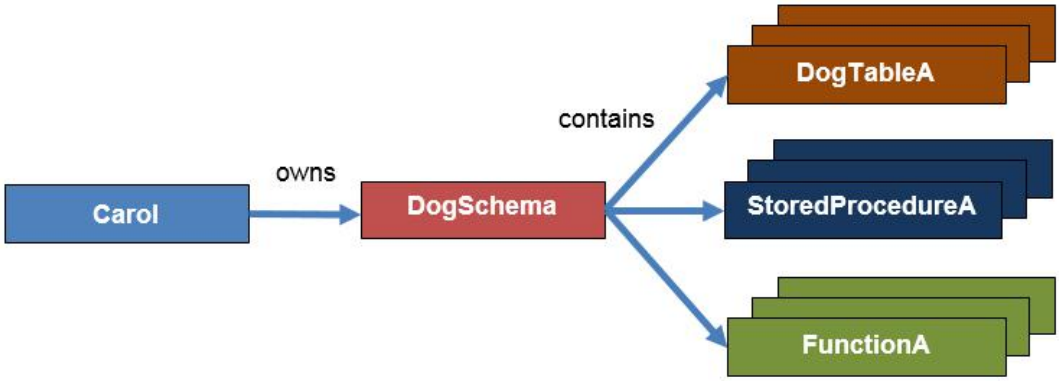

图 5.1 一个包含数据库对象的示例架构

你可以将权限分配给架构应用于架构下的所有对象。例如，如果你将 DogSchema 架构的 SELECT 权限授予给一个主体 ,这个主体将可以查询 DogSchema 架构下的所有表格。和所有用户定义的数据库对象一样，架构有一个所有者可以完全控制对象。

在架构中单独设置对象权限通常是一个选项 ,但如果你已经将数据库中的架构设计好，在某些功能类型的数据库中，你可以在架构上设置权限，并将其应用到对象。最棒的是，你在架构上分配的权限会自动应用到你添加到架构中的任何对象。继续 SELECT 的例子，如果一年后你添加表 DogTable4 到架构 DogSchema，所有对架构有 SELECT 权限的主体都能自动的 SELECT 新表。

多个用户和角色可以具有相同的默认架构，如果主体没有默认架构，SQL Server 会尝试在 dbo 架构下查找或创建对象。

现在 ,你将看到如何使用架构来分配对象的权限。使用以下步骤将 Purchasing 架构的查询、更新、删除、插入权限授予给 DataEntry 用户自定义角色。创建 DataEntry 角色，请执行代码 5.1

```
USE AdventureWorks2012;
GO
CREATE ROLE DataEntry AUTHORIZATION dbo;
```

代码 5.1 在 AdventureWorks2012 数据库创建 DataEntry 角色

然后按照下面步骤在 SSMS 使用图形工具来分配所需的权限

1、对象资源管理器->数据库->AdventureWorks2012->安全性->角色->数据库角色->DataEntry

2、右击 DataEntry，弹出菜单选择属性。在数据库角色属性对话框选择安全对象。

3、点击“搜索”按钮打开“添加对象”对话框

4、在添加对象对话框，选择“特定类型的所有对象”选项，如图 5.2 所示。点击确定，打开“选择对象类型”对话框

图 5.2 添加对象

5、在“选择对象类型”对话框中，向下滚动到“架构”项目，然后选择旁边的复选框。对话框应该看起来像图 5.3。单击“确定”以保存选择并关闭对话框

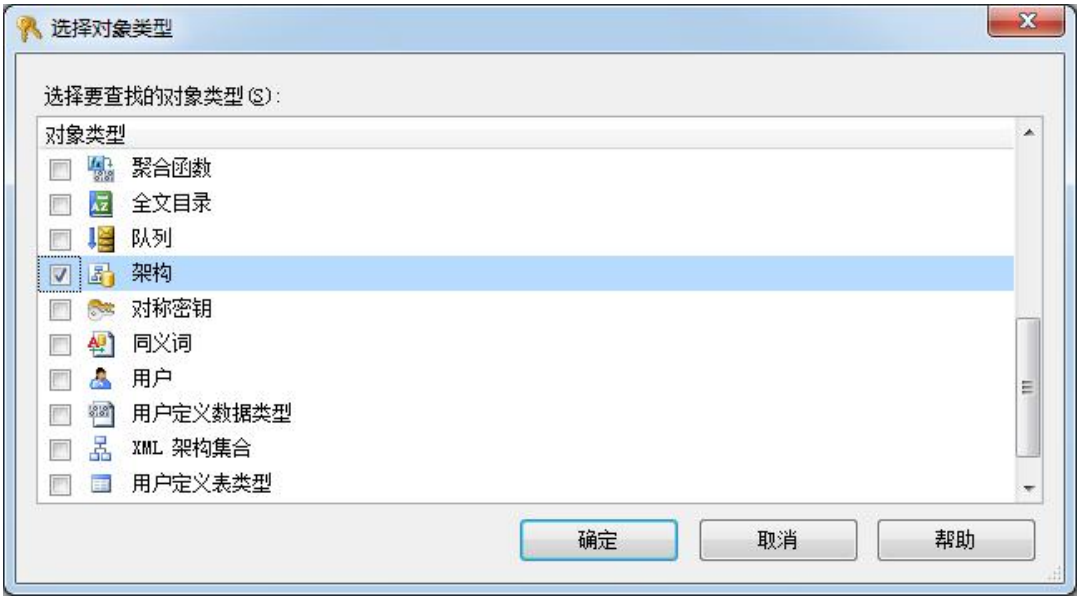

图 5.3 选择对象类型

6、返回到数据库角色属性对话框中，向下滚动安全对象列表然后点击 Purchasing 架构。页面的下部显示了可用的权限

7、显示页签为 Purchasing 架构勾选删除、插入、选择、更新授予复选框。数据库角色属性对话框如 5.4 所示

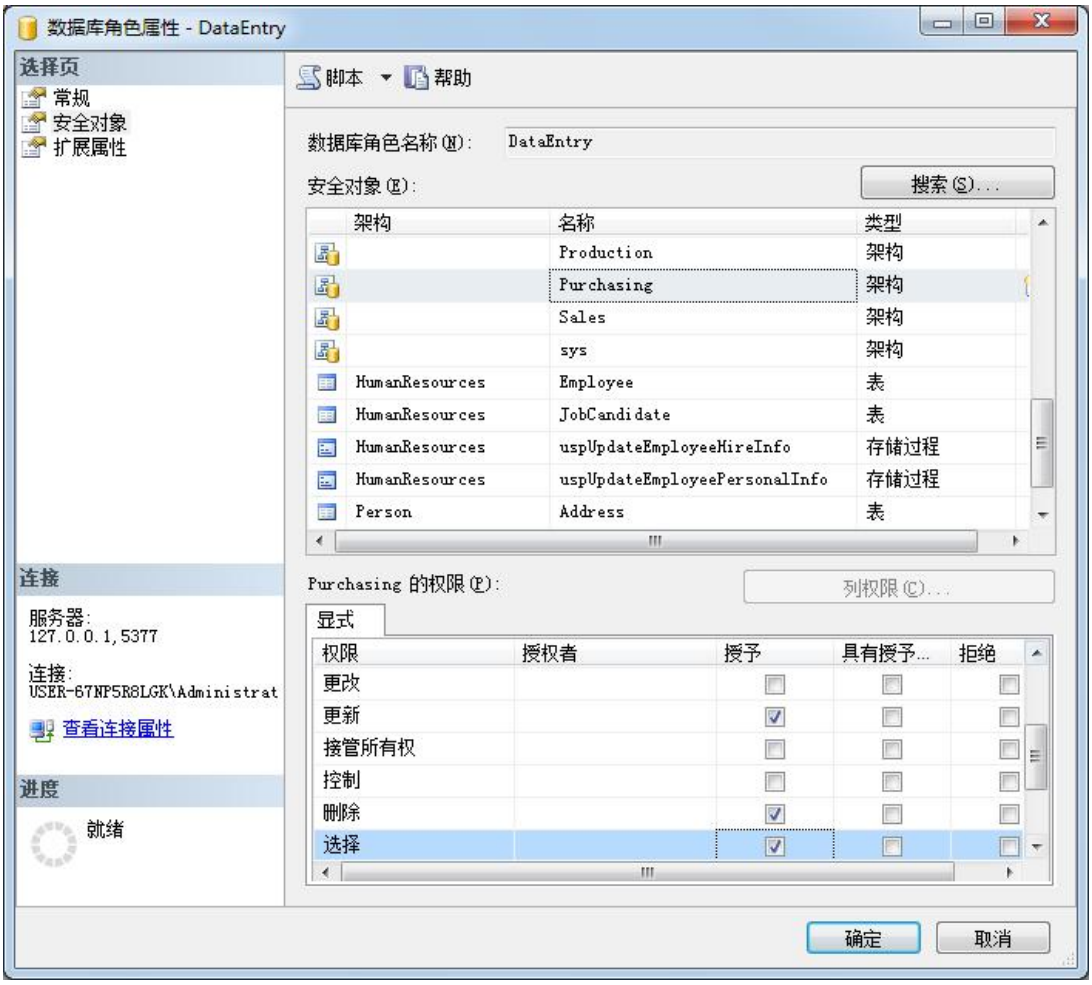

图 5.4 设置对 Purchasing 架构的访问权限

到此 DataEntry 角色下的所有成员对 AdventureWorks2012 库 Purchasing 架构下的所有表都有选择、更新、删除和插入权限。只有当该角色的成员被拒绝任何权限时，才会出现异常。DENY 阻止了他们通过角色继承的权限。

当然，你也可以使用 TSQL 脚本将架构上的权限授予给角色

```
GRANT DELETE ON SCHEMA::Purchasing TO DataEntry;
GRANT INSERT ON SCHEMA::Purchasing TO DataEntry;
GRANT SELECT ON SCHEMA::Purchasing TO DataEntry;
GRANT UPDATE ON SCHEMA::Purchasing TO DataEntry;
GO
```

代码 5.2 给 Purchasing 架构授权

这些技术表明你可以创建不同的架构，在每个架构中放置不同的对象，然后在架构上分配权限。这节省了单个表上分配权限的工作。如果你授予权限给角色，正如我们对 DataEntry 角色的操作，你可以高效地对许多主体分配权限。这让你分割数据库，简化你的设计并实现数据库的安全性。

默认架构

在 SQL-99 规范定义，架构本质上是一个数据库的对象容器。它可以由一个主体拥有 ,如图 5.5 所示(同图 5.1)。使用架构作为数据库对象容器的好处之一是：当 Carol 离开公司时，不需要改变 Carol 拥有的数百或数千个对象的所有权，管理员只需要改变这些架构的所有权，每个可能有成千上万的对象。这种方法是更简洁、更容易、更安全。

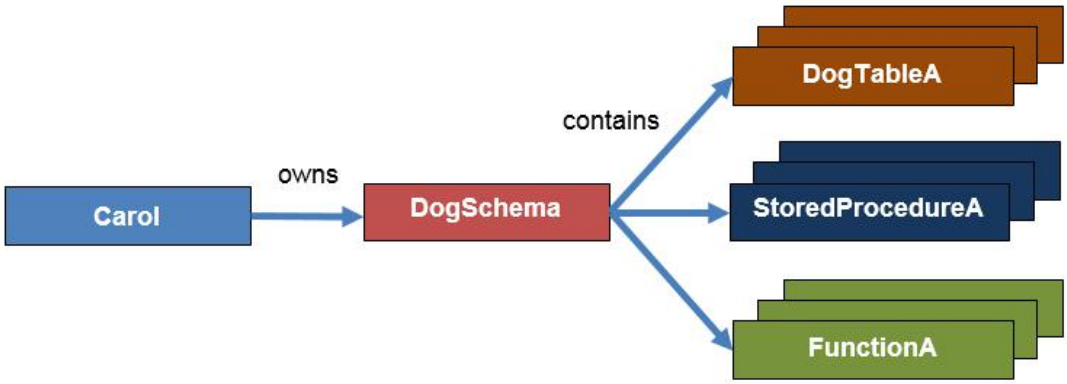

图 5.5 Carol 拥有的 DogSchema 架构

SQL Server 允许你为用户和组分配默认的架构。设置默认架构是一种方便，并且具有一些重要的操作好处。特别是 ,当命名和访问对象时它消除了一些歧义。

用户默认架构

SQL Server 不会自动创建具有相同名称的架构当你创建一个用户。你必须显式地创建一个架构，分配架构所有者，然后创建和添加对象到该架构。你可以(通

常应该)为用户分配一个默认架构，这样所有用户创建的对象，如果没有显式地分配给另一个架构，都会成为默认架构的一部分。

本篇中的代码显示了所有的操作,当用户没有一个设置默认架构时会发生什么情况。我会解释每一步会发生什么，但你可能想独自执行代码，以更好地了解发生了什么事。当然，你自己试试吧！如果你想看看发生了什么，执行每一个代码块。

代码 5.3 做了演示所需的一些设置。创建 DefaultSchema 数据库并设置为当前数据库，接着创建登录名 carol，并映射到数据库用户，并授予它创建表的权限。然后，它将执行上下文更改为用户 carol。

```
IF suser_sid('carol') IS NOT NULL DROP LOGIN carol;
GO
CREATE DATABASE DefaultSchema;
GO
USE DefaultSchema;
GO
CREATE LOGIN carol WITH PASSWORD = 'crolPWD123%%%';
CREATE USER carol FOR LOGIN carol;
GRANT CREATE TABLE TO carol;--user

EXECUTE AS LOGIN = 'carol';
GO
```

代码 5.3 创建 DefaultSchema 数据库并设置 carol 用户

接下来的代码试图创建一个新表 table1，如代码 5.4 所示。但之前的代码创建 carol 用户时并没有给它分配默认架构。SQL Server 尝试使用 dbo 架构，这是

默认的回退架构。但 carol 没有数据库的所有权，所以它不能在 dbo 架构中创建对象。

```
CREATE TABLE table1 (tID int);
```

代码 5.4 尝试以 carol 上下文创建表

因为 carol 没有必要的权限，创建表语句失败并返回错误信息。

指定的架构名称 "dbo" 不存在，或者您没有使用该名称的权限。

代码 5.5REVERT 到原来的管理员登录，然后创建一个架构，并将架构所有者设为 carol 用户。你会在 SQL Server 中看到很多 AUTHORIZATION 子句，因为它可以让你在创建或更改一个对象时分配所有权。

REVERT;

```
CREATE SCHEMA DogSchema AUTHORIZATION carol;
```

代码 5.5 创建 DogSchema 架构

再次更改执行上下文为 carol，然后试图再创建表 table1，但它失败了！现在的问题是，用户拥有一个架构并不意味着它是用户的默认架构。一个用户可以拥有成百上千个架构，SQL Server 没有责任挑选一个作为默认架构。最终在创建表的时候显示的包含架构才能通过。代码 5.6 中明确地在 DogSchema 架构创建表

```
EXECUTE AS LOGIN = 'carol';
GO
CREATE TABLE DogSchema.table1 (tID int);
```

代码 5.6 创建带有显示架构的表

如果 DogSchema 架构存在，第二种尝试创建表的方式，就是在创建用户的时候分配默认架构，或者后期修改用户默认架构。如代码 5.7 所示

```
   CREATE USER carol FOR LOGIN carol WITH DEFAULT_SCHEMA =
 DogSchema;

   -- or

   ALTER USER carol WITH DEFAULT_SCHEMA = DogSchema;--测试
可以在 carol 身份下执行
```

代码 5.7 给用户 carol 设置默认架构

如果你执行 ALTER USER 语句为 carol 设置默认架构 ,然后你可以执行代码 5.8 成功创建表而不需指定架构。创建表的语句将创建一个 DogSchema.table2 表 ,因为 DogSchema 是 carol 的默认架构。

```
   EXECUTE AS LOGIN = 'carol';

   GO

   CREATE TABLE table2 (tID int);

   GO

   SELECT * FROM table2;

   REVERT;
```

代码 5.8 创建表 table2 不需指定架构

另一个有趣的现象是，当你使用 REVERT 语句返回到自己的安全上下文，你不能执行代码 5.9。除非你设置自己的默认架构为 DogSchema，否则 SQL Server 将查找 dbo.table2

```
   SELECT * FROM table2;
```

代码 5.9 在 carol 安全上下文之外运行会出错

你需要显式地使用架构来识别你想从中读取数据的表，如代码 5.10。本代码成功，并返回 DogSchema.table2 中的内容。

```
SELECT * FROM DogSchema.table2;
```

代码 5.10 指定架构执行 SELECT 语句

SQL Server 用户与架构分离可以严格控制你的数据库和应用程序的安全结构。这使得它更容易管理一个数据库和 SQL Server。你不需要为每一个对象设置成 dbo 用户所有者，这在 2005 之前版本很常见。

组默认架构

用户的默认架构，在 SQL Server 2005 引进，解决了查询、创建对象以及其他操作在正确的架构下使用正确的对象。但这些默认架构的存在一个问题，你可以很容易的在数据库中创建大量的架构。Windows 组的默认架构，在 SQL Server 2012 引进，解决了这些问题。

使用下面的步骤来探索用户默认架构的潜在问题。步骤假定本地 Windows 有一个 DBAs 组，并且 ClearFile 用户是 DBAs 组中的成员。你需要更改示例代码中的机器名称。最后，DefaultSchema 数据库应该已经存在。

1、修改当前数据库为 DefaultSchema，代码 5.11 首先创建一个 Windows 组的登录名，然后创建 DataAdmins 用户映射到此登录名，接着创建 DBAs 角色并添加 DataAdmins 用户到角色中。

```
USE DefaultSchema
GO
```

```
    CREATE LOGIN [USER-67NP5R8LGK\DBAs] FROM WINDOWS;

    CREATE USER DataAdmins FROM LOGIN [USER-67NP5R8LGK\DBA
s];

    CREATE ROLE DBAs;

    ALTER ROLE DBAs ADD MEMBER DataAdmins;
```

代码 5.11 创建登录名、用户、角色

2、授予对 DogSchema 架构的创建表和控制权限给 DBAs 角色

```
    GRANT CREATE TABLE TO DBAs;

    GRANT CONTROL ON SCHEMA::DogSchema TO DBAs;
```

代码 5.12 授予权限给 DBAs 角色

3、用 ClearFile 身份运行另一个 SSMS。在 Windows 启动菜单按下 SHIFT 键，并且右击 SSMS。在弹出的菜单中选择以其他用户身份运行，键入 ClearFile 的用户名和密码

4、在连接到服务器对话框使用 Windows 身份验证，用户名为 ClearFile，如图 5.6 所示。点击连接，就以 ClearFile 身份运行 SSMS

图 5.6 以 ClearFile 身份登录 SSMS

5、在刚打开的 SSMS 新建查询，可用数据库切换到 DefaultSchema 数据库

6、执行代码 5.13，成功创建了表 table1，但它的架构是什么呢？

```
CREATE TABLE table1 (tID int)
```

代码 5.13 使用 ClearFile 创建表 table1

7、对象资源管理器下 DefaultSchema 数据库展开表、用户、架构。如图 5.7 所示，前面语句创建的表叫 USER-67NP5R8LGK\ClearFile.table1,数据库用户 USER-67NP5R8LGK\ClearFile，架构 USER-67NP5R8LGK\ClearFile

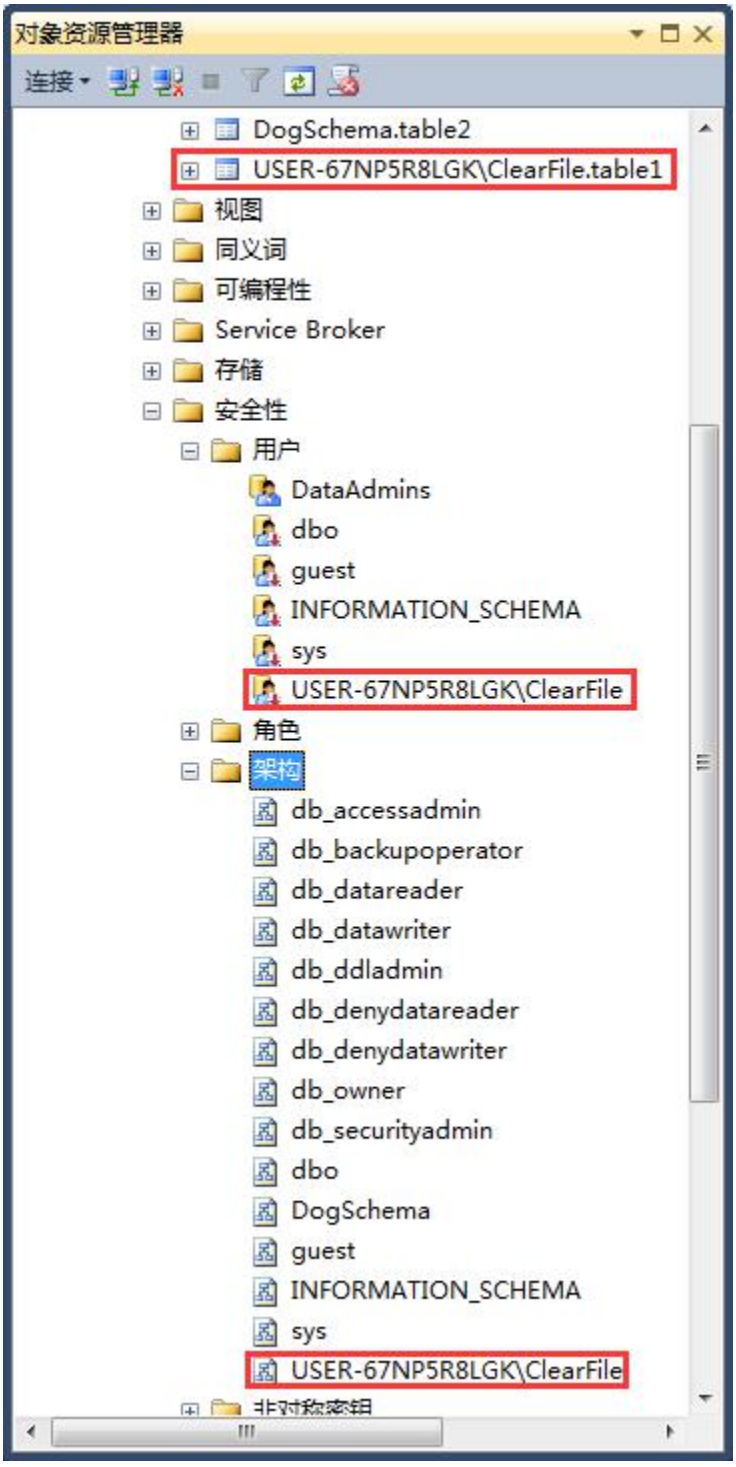

图 5.7 没有默认架构时创建表的结果

8、回到你以管理员登录的 SSMS，在对象资源管理器下按顺序删除刚才创建的表、架构和用户

9、还是在原 SSMS，执行代码 5.14 给用户 DataAdmins 设置默认架构 DogSchema

```
ALTER USER DataAdmins WITH DEFAULT_SCHEMA = DogSchema;
```

代码 5.14 为 DataAdmins 用户认置默认架构

10、返回到 ClearFile 登录的 SSMS，执行代码 5.15 创建新表 table3。这次代码创建的表叫做 DogSchema.table3，并且没有添加 ClearFile 数据库用户，也没有添加 ClearFile 架构

```
CREATE TABLE table3 (tID int)
```

代码 5.15 创建表 table3

你也可以在数据库用户对话框设置默认架构，如图 5.8 所示

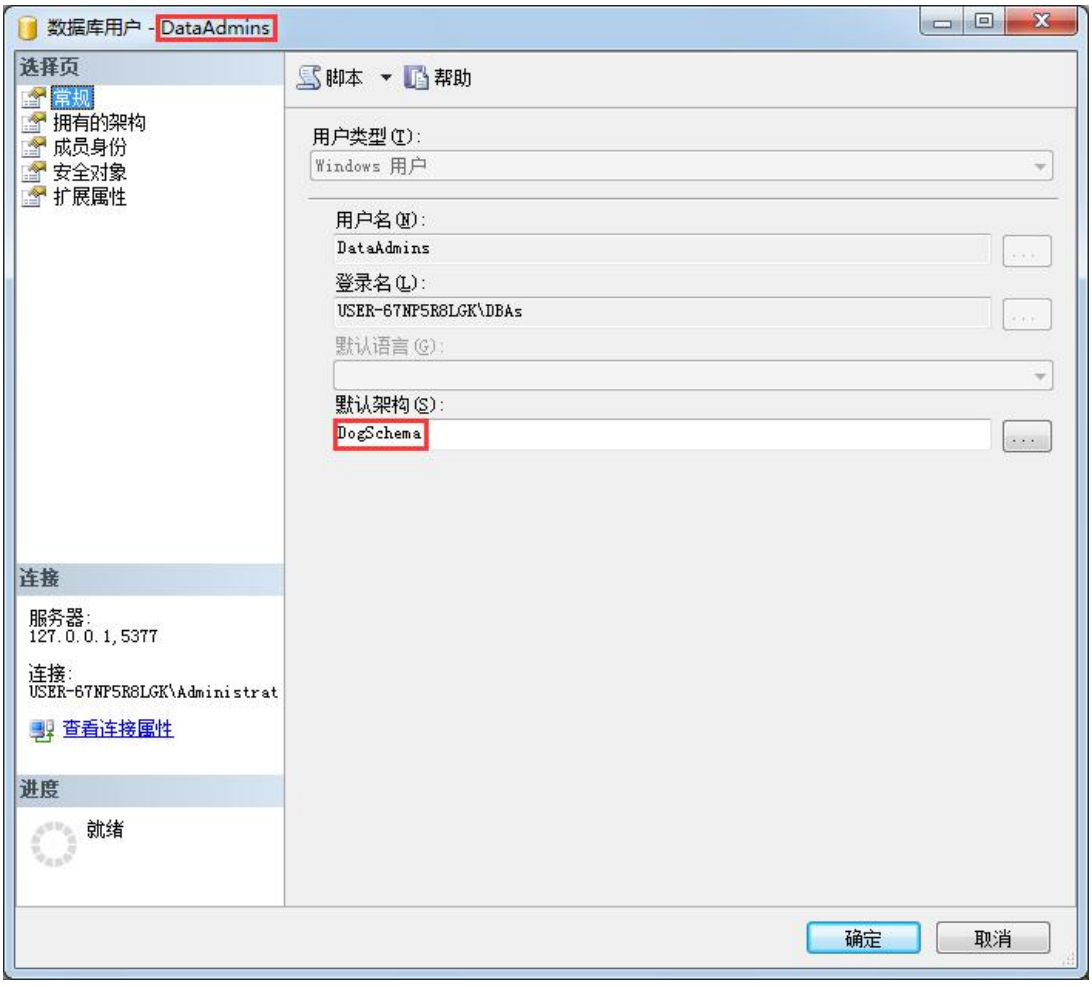

图 5.8 使用数据库用户对话框设置默认架构

SQL Server 2012 增加了给组设定默认架构的功能，类似于给用户设定默认架构解决问题，使安全管理更容易。对于同样的原因，你创建没有任何权限的用户，然后将它们添加到需要的权限组中，你可以为组指定默认架构，而不是每个用户。与用户一样，你可以通过 create user 或者 alter user 指定默认架构。