第 13 章

存储过程的应用

学习指引

存储过程（Stored Procedure）是在大型数据库系统中，一组为了完成特定功能的 SQL 语句集，存储过程是数据库中的一个重要对象，它代替了传统的逐条执行 SQL 语句的方式。本章就来介绍数据库的存储过程，主要内容包括创建、调用、查看、修改、删除存储过程等。

重点导读

- 了解什么是存储过程。
- 掌握创建存储过程的方法。
- 掌握调用存储过程的方法。
- 掌握查看存储过程的方法。
- 掌握修改存储过程的方法。
- 掌握删除存储过程的方法。

13.1　存储过程概述

存储过程可以重复调用，当存储过程执行一次后，可以将语句缓存，这样下次执行的时候直接使用缓存中的语句，就可以提高存储过程的性能。

13.1.1　什么是存储过程

存储过程是一组为了完成特定功能的 SQL 语句的集合，经编译后存储在数据库中，用户通过指定存储过程的名称并给出参数来执行。存储过程中可以包含逻辑控制语句和数据操作语句，它可以接受参数、输出参数、返回单个或多个结果集以及返回值。

由于存储过程在创建时即在数据库服务器上进行了编译并存储在数据库中，所以存储过程运行比单个

的 SQL 语句块要快。同时由于在调用时只需要提供存储过程名和必要的参数信息，所以在一定程度上也可以减少网络流量、减轻网络负担。

13.1.2 存储过程的优点

相对于直接使用 SQL 语句，在应用程序中直接调用存储过程具有以下好处。

1．存储过程允许标准组件式编程

存储过程创建后可以在程序中被多次调用执行，而不必重新编写该存储过程的 SQL 语句。而且数据库专业人员可以随时对存储过程进行修改，但对应用程序源代码却毫无影响，从而极大地提高了程序的可移植性。

2．存储过程能够实现较快的执行速度

如果操作包含大量的 SQL 语句代码，分别被多次执行，那么存储过程要比批处理的执行速度快得多。因为存储过程是预编译的，在首次运行一个存储过程时，查询优化器对其进行分析、优化，并给出最终被存在系统表中的存储计划。而批处理的 SQL 语句每次运行都需要预编译和优化，所以速度就要慢一些。

3．存储过程减轻网络流量

对于同一个针对数据库对象的操作，如果这一操作所涉及的 SQL 语句被组织成一存储过程，那么当在客户机上调用该存储过程时，网络中传递的只是该调用语句，否则将会是多条 SQL 语句，从而减轻了网络流量，降低了网络负载。

4．存储过程可被作为一种安全机制来充分利用

系统管理员可以对执行的某一个存储过程进行权限限制，从而能够实现对某些数据访问的限制，避免非授权用户对数据的访问，保证数据的安全。

13.1.3 存储过程的缺点

任何一个事物都不是完美的，存储过程也不例外，除一些优点外，存储过程还具有如下缺点。

- 数据库移植不方便，存储过程依赖于数据库管理系统，SQL Server 存储过程中封装的操作代码不能直接移植到其他的数据库管理系统中。
- 不支持面向对象的设计，无法采用面向对象的方式将逻辑业务进行封装，甚至形成通用的可支持服务的业务逻辑框架。
- 代码可读性差、不易维护。
- 不支持集群。

13.2 存储过程的类型

在 SQL Server 中，存储过程主要分为自定义存储过程、扩展存储过程和系统存储过程，在存储过程中可以声明变量、执行条件判断语句等其他编程功能。

13.2.1　系统存储过程

系统存储过程是由 SQL Server 系统自身提供的存储过程，可以作为命令执行各种操作。例如，sp_rename 系统存储过程可以更改当前数据库中用户创建对象的名称；sp_helptext 存储过程可以显示规则、默认值或视图的文本信息等。

SQL Server 服务器中许多的管理工作都是通过执行系统存储过程来完成的，许多系统信息也可以通过执行系统存储过程来获得。系统存储过程位于数据库服务器中，并且以 sp_开头。系统存储过程定义在系统定义和用户定义的数据库中，在调用时不必在存储过程前加数据库限定名。

系统存储过程创建并存放于系统数据库 master 中，一些系统存储过程只能由系统管理员使用，而有些系统存储过程通过授权可以被其他用户所使用。

13.2.2　自定义存储过程

自定义存储过程即用户为了实现某一特定业务需求，在用户数据库中编写的 SQL 语句集合。用户存储过程可以接受输入参数，向客户端返回结果和信息，返回输出参数等。

创建自定义存储过程时，存储过程名前面加上“##”表示创建了一个全局的临时存储过程；存储过程名前面加上“#”时，表示创建局部临时存储过程。局部临时存储过程只能在创建它的会话中使用，会话结束时将被删除。这两种存储过程都存储在系统数据库 tempdb 之中。

用户定义存储过程可以分为两类：Transact-SQL 和 CLR。

- Transact-SQL 存储过程是指保存的 Transact-SQL 语句集合，可以接受和返回用户提供的参数。存储过程也可能从数据库向客户端应用程序返回数据。
- CLR 存储过程是指引用 Microsoft .NET Framework 公共语言方法的存储过程，可以接受和返回用户提供的参数，它们在.NET Framework 程序集中是作为类的公共静态方法实现的。

13.2.3　扩展存储过程

扩展存储过程是以在 SQL Server 环境外执行的动态链接（DLL 文件）来实现的，可以加载到 SQL Server 实例运行的地址空间中执行，扩展存储过程可以用 SQL Server 扩展存储过程 API 编程，扩展存储过程以前缀“xp_”来标识，对于用户来说，扩展存储过程和普通存储过程一样，可以用相同的方法来执行。

13.3　创建存储过程

存储过程是在数据库服务器端执行的一组 SQL 语句集合，经编译后存放在数据库服务器中，本节就来介绍如何创建存储过程。

13.3.1　在 SSMS 中创建存储过程

在 SSMS 中可以使用向导创建存储过程，具体操作步骤如下。

步骤 1：启动 SSMS 并连接到 SQL Server 数据库之中，打开 SSMS 窗口，选择“数据库”→mydbase→“可编程性”结点。在“可编程性”结点下，右击“存储过程”结点，在弹出的快捷菜单中选择“新建”→

“存储过程”菜单命令，如图 13-1 所示。

步骤 2：打开创建存储过程的代码模板，这里显示了 CREATE PROCEDURE 语句模板，可以修改要创建的存储过程的名称，然后在存储过程中的 BEGIN END 代码块中添加需要的 SQL 语句，最后单击“执行”按钮即可创建一个存储过程，如图 13-2 所示。

图 13-1　选择“新建”→“存储过程”菜单命令

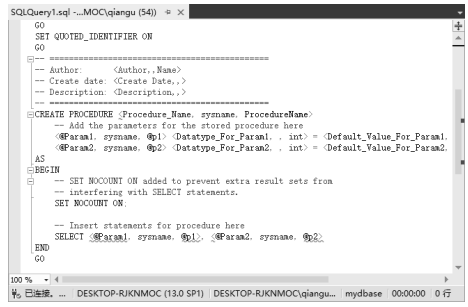


图 13-2　使用模板创建存储过程

【例 13-1】创建一个名称为 Proc_emp 的存储过程，要求该存储过程实现的功能为：在 employee 表中查询男员工的姓名、当前职位与基本工资，具体操作步骤如下。

步骤 1：在创建存储过程的窗口中选择“查询”→“指定模板参数的值”菜单命令，如图 13-3 所示。

步骤 2：弹出“指定模板参数的值”对话框，将 Procedure_Name 参数对应的名称修改为“Proc_emp”，单击“确定”按钮，即可关闭此对话框，如图 13-4 所示。

图 13-3 “指定模板参数的值”菜单命令

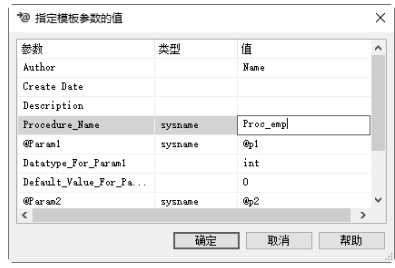


图 13-4 “指定模板参数的值”对话框

步骤 3：在创建存储过程的窗口中，将对应的 SELECT 语句修改为以下语句，如图 13-5 所示。

```
SELECT e_name,e_job,e_salary
FROM employee
WHERE e_gender='男';
```

步骤 4：单击“执行”按钮，即可完成存储过程的创建操作，执行结果如图 13-6 所示。

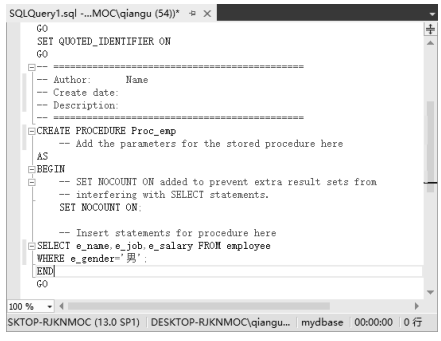


图 13-5　修改 SELECT 语句

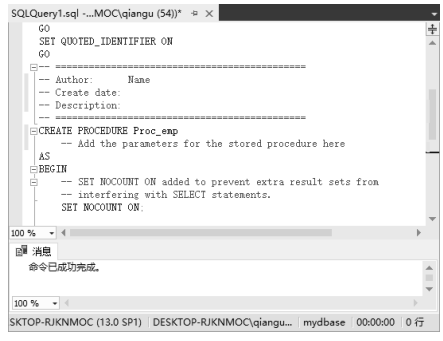


图 13-6　创建存储过程

13.3.2　创建存储过程的语法格式

使用 CREATE PROCEDURE 语句可以创建存储过程，语法格式如下：

```
CREATE PROCEDURE [schema name.] procedure name [ ; number ]
{ @parameter data_type }
[ VARYING ] [ = default ] [ OUT | OUTPUT ] [READONLY]
[ WITH  <ENCRYPTION ]|[ RECOMPILE ]|[ EXECUTE AS Clause ]> ]
[ FOR REPLICATION ]
AS  <sql_statement>
```

主要参数介绍如下。

- procedure_name：新存储过程的名称，并且在架构中必须唯一。可在 procedure_name 前面使用一个#字符号（#procedure_name）来创建局部临时过程，使用两个#字符号 （##procedure_name）来创建全局临时过程。对于 CLR 存储过程，不能指定临时名称。
- number：是可选整数，用于对同名的过程分组。使用一个 DROP PROCEDURE 语句可将这些分组过程一起删除。例如，称为 orders 的应用程序可能使用名为 orderproc;1、orderproc;2 等的过程。DROP PROCEDURE orderproc 语句将删除整个组。如果名称中包含分隔标识符，则数字不应包含在标识符中；只应在 procedure_name 前后使用适当的分隔符。
- @ parameter：存储过程中的参数。在 CREATE PROCEDURE 语句中可以声明一个或多个参数。除非定义了参数的默认值或者将参数设置为等于另一个参数，否则用户必须在调用过程时为每个声明的参数提供值。存储过程最多可以有 2100 个参数。如果过程包含表值参数，并且该参数在调用中缺失，则传入空表默认值。通过将 at 符号（@）用作第一个字符来指定参数名称。每个过程的参数仅用于该过程本身；其他过程中可以使用相同的参数名称。默认情况下，参数只能代替常量表达式，而不能用于代替表名、列名或其他数据库对象的名称。如果指定了 FOR REPLICATION，则无法声明参数。
- date_type：指定参数的数据类型，所有数据类型都可以用作 Transact-SQL 存储过程的参数。可以使用用户定义表类型来声明表值参数作为 Transact-SQL 存储过程的参数。只能将表值参数指定为输入参数，这些参数必须带有 READONLY 关键字。cursor 数据类型只能用于 OUTPUT 参数。如果指定了 cursor 数据类型，则还必须指定 VARYING 和 OUTPUT 关键字。可以为 cursor 数据类型指定多个输出参数。对于 CLR 存储过程，不能指定 char、varchar、text、ntext、image、cursor、用户定义表类型和 table 作为参数。
- default：存储过程中参数的默认值。如果定义了 default 值，则无须指定此参数的值即可执行过程。默认值必须是常量或 NULL。如果过程使用带 LIKE 关键字的参数，则可包含下列通配符：%、_、[] 和[^]。
- OUTPUT：指示参数是输出参数。此选项的值可以返回给调用 EXECUTE 的语句。使用 OUTPUT 参数将值返回给过程的调用方。除非是 CLR 过程，否则 text、ntext 和 image 参数不能用作 OUTPUT 参数。使用 OUTPUT 关键字的输出参数可以为游标占位符，CLR 过程除外。不能将用户定义表类型指定为存储过程的 OUTPUT 参数。
- READONLY：指示不能在过程的主体中更新或修改参数。如果参数类型为用户定义的表类型，则必须指定 READONLY。
- RECOMPILE：表明 SQL Server 2016 不会保存该存储过程的执行计划，该存储过程每执行一次都要重新编译。在使用非典型值或临时值而不希望覆盖保存在内存中的执行计划时，就可以使用 RECOMPILE 选项。

- ENCRYPTION：表示 SQL Server 2016 加密后的 syscomments 表，该表的 text 字段是包含 CREATE PROCEDURE 语句的存储过程文本。使用 ENCRYPTION 关键字无法通过查看 syscomments 表来查看存储过程的内容。
- FOR REPLICATION：用于指定不能在订阅服务器上执行为复制创建的存储过程。使用此选项创建的存储过程可用作存储过程筛选，且只能在复制过程中执行。本选项不能和 WITH RECOMPILE 选项一起使用。
- AS：用于指定该存储过程要进行的操作。
- sql_statement：是存储过程中要包含的任意数目和类型的 Transact-SQL 语句。但有一些限制。

13.3.3 创建不带参数的存储过程

最简单的一种自定义存储过程就是不带参数的存储过程，下面介绍如何创建一个不带参数的存储过程。

【例 13-2】创建查看 mydbase 数据库中 employee 表的存储过程，SQL 语句如下：

```
USE mydbase;
GO
CREATE PROCEDURE Proc emp 01
AS
SELECT * FROM employee;
GO
```

单击“执行”按钮，即可完成存储过程的创建操作，执行结果如图 13-7 所示。

另外，存储过程可以是很多语句的复杂组合，其本身也可以调用其他函数，来组成更加复杂的操作。

【例 13-3】创建一个获取 employee 表记录条数的存储过程，名称为 Count_Proc，SQL 语句如下：

```
USE mydbase;
GO
CREATE PROCEDURE Count Proc
AS
SELECT COUNT(*) AS 总数 FROM employee;
GO
```

输入完成之后，单击“执行”按钮，即可完成存储过程的创建操作，执行结果如图 13-8 所示。

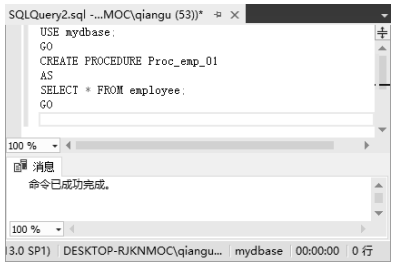


图 13-7 创建不带参数的存储过程

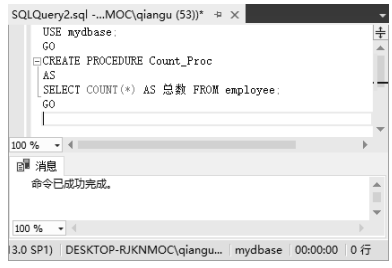


图 13-8 创建存储过程 Count_Proc

13.3.4 创建带输入参数的存储过程

在设计数据库应用系统时，可能会需要根据用户的输入信息产生对应的查询结果，这时就需要把用户的输入信息作为参数传递给存储过程，即开发者需要创建带输入参数的存储过程。

【例 13-4】创建存储过程 Proc_emp_02，根据输入的员工编号，查询员工的相关信息，如姓名、所在职位与基本工资，SQL 语句如下：

```
USE mydbase;
GO
CREATE PROCEDURE Proc emp 02 @sID INT
AS
SELECT * FROM employee WHERE e_no=@sID;
GO
```

输入完成之后，单击“执行”按钮，即可完成存储过程的创建操作，该段代码创建一个名为 Proc_emp_02 的存储过程，使用一个整数类型的参数@sID 来执行存储过程，如图 13-9 所示。

【例 13-5】创建带默认参数的存储过程 Proc_emp_03，输入语句如下：

```
USE mydbase;
GO
CREATE PROCEDURE Proc emp 03 @sID INT=101
AS
SELECT * FROM employee WHERE e_no=@sID;
GO
```

输入完成之后，单击“执行”按钮，即可完成带默认输入参数存储过程的创建操作，该段代码创建的存储过程在调用时即使不指定参数值也可以返回一个默认的结果集，如图 13-10 所示。

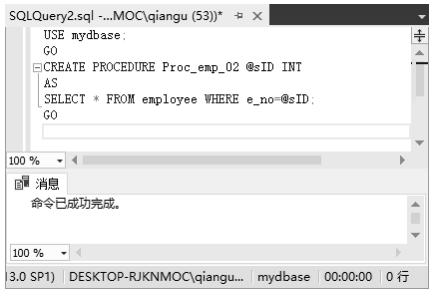


图 13-9 创建存储过程 Proc_emp_02

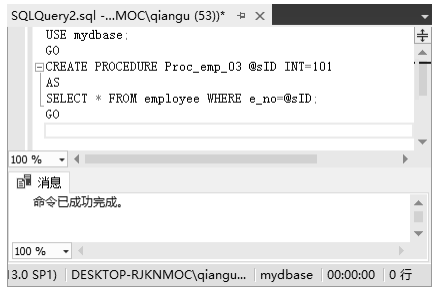


图 13-10 创建存储过程 Proc_emp_03

13.3.5 创建带输出参数的存储过程

存储过程中的默认参数类型是输入参数，如果要为存储过程指定输出参数，还要在参数类型后面加上 OUTPUT 关键字。

【例 13-6】定义存储过程 Proc_emp_04，根据用户输入的部门编号，返回该部门中员工的个数，SQL 语句如下：

```
USE mydbase;
GO
CREATE PROCEDURE Proc_emp_04
@sID INT=1,
@employeecount INT OUTPUT
AS
SELECT @employeecount=COUNT(employee.dept_no)  FROM employee WHERE dept_no=@sID;
GO
```

输入完成之后，单击“执行”按钮，即可完成带输出参数存储过程的创建操作。该段代码将创建一个名称为 Proc_emp_04 的存储过程，该存储过程中有两个参数，@sID 为输出参数，指定要查询的员工部门编号的 id，默认值为 1；@employeecount 为输出参数，用来返回该部门中员工的个数，如图 13-11 所示。

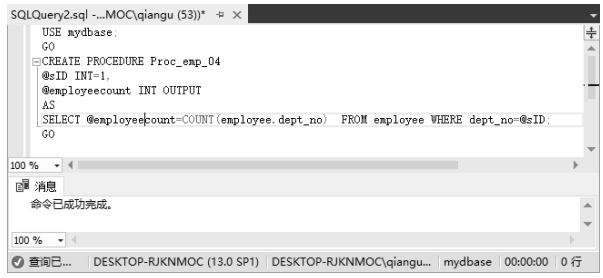


图 13-11　定义存储过程 Proc_emp_04

13.3.6　创建带加密选项的存储过程

所谓加密选项并不是对存储过程中查询出来的内容加密，而是将创建存储过程本身的语句加密，通过对创建存储过程的语句加密，可以在一定程度上保护存储过程中用到的表信息，同时也能提高数据的安全性。带加密选项的存储过程使用的是 with encryption。

【例 13-7】定义带加密选项的存储过程 Proc_emp_05，查询员工的姓名、当前职位与基本工资信息，SQL 语句如下：

```
USE mydbase;
CREATE PROCEDURE Proc_emp_05
WITH ENCRYPTION
AS
BEGIN
SELECT e_name,e_job,e_salary  FROM employee;
EMD
```

输入完成之后，单击“执行”按钮，即可完成带加密选项存储过程的创建操作，执行结果如图 13-12 所示。

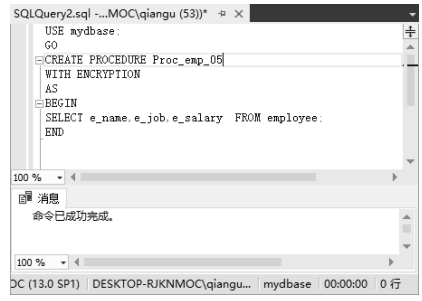


图 13-12　创建带加密选项的存储过程

13.4　执行存储过程

当存储过程创建完毕后，下面就可以执行存储过程了，本节就来介绍执行存储过程的方法。

13.4.1　执行存储过程的语法格式

在 SQL Server 2016 中执行存储过程时，需要使用 EXECUTE 语句，如果存储过程是批处理中的第一条语句，那么不使用 EXECUTE 关键字也可以执行该存储过程，EXECUTE 语法格式如下：

```
[ { EXEC | EXECUTE } ]
   {
     [ @return_status = ]
     { module name [ ;number ] | @module name var }
      [ [ @parameter = ] { value | @variable [ OUTPUT ]  | [ DEFAULT ]   }  ]
      [ ,...n ]
     [ WITH RECOMPILE ]
   }
```

主要参数介绍如下。

- @return_status：可选的整型变量，存储模块的返回状态。这个变量在用于 EXECUTE 语句前，必

须在批处理、存储过程或函数中声明过。在用于调用标量值用户定义函数时，@return_status 变量可以为任意标量数据类型。

- module_name：是要调用的存储过程的完全限定或者不完全限定名称。用户可以执行在另一数据库中创建的模块，只要运行模块的用户拥有此模块或具有在该数据库中执行该模块的适当权限。
- number：可选整数，用于对同名的过程分组。该参数不能用于扩展存储过程。
- @module_name_var：是局部定义的变量名，代表模块名称。
- @parameter：存储过程中使用的参数，与在模块中定义的相同。参数名称前必须加上符号@。在与 @parameter_name=value 格式一起使用时，参数名和常量不必按它们在模块中定义的顺序提供。但是，如果对任何参数使用了 @parameter_name=value 格式，则对所有后续参数都必须使用此格式。默认情况下，参数可为空值。
- value：传递给模块或传递命令的参数值。如果参数名称没有指定，参数值必须以在模块中定义的顺序提供。
- @variable：是用来存储参数或返回参数的变量。
- OUTPUT：指定模块或命令字符串返回一个参数。该模块或命令字符串中的匹配参数也必须使用关键字 OUTPUT 创建。使用游标变量作为参数时使用该关键字。
- DEFAULT：根据模块的定义，提供参数的默认值。当模块需要的参数值没有定义默认值并且缺少参数或指定了 DEFAULT 关键字时，会出现错误。
- WITH RECOMPILE：执行模块后，强制编译、使用和放弃新计划。如果该模块存在现有查询计划，则该计划将保留在缓存中。如果所提供的参数为非典型参数或者数据有很大的改变，使用该选项。该选项不能用于扩展存储过程。建议尽量少使用该选项，因为它消耗较多的系统资源。

13.4.2　执行不带参数的存储过程

存储过程创建完成后，可以通过 EXECUTE 语句来执行创建的存储过程，该命令可以简写为 EXEC。

【例 13-8】执行不带参数的存储过程 Proc_emp_01，来查看员工信息，SQL 语句如下：

```
USE mydbase;
GO
EXEC Proc_emp_01;
```

单击“执行”按钮，即可完成执行不带参数存储过程的操作，这里是查询员工信息表，执行结果如图 13-13 所示。

图 13-13　执行不带参数的存储过程

提示：EXECUTE 语句的执行是不需要任何权限的，但是操作 EXECUTE 字符串内引用的对象是需要相应的权限的，例如，如果要使用 DELETE 语句执行删除操作，则调用 EXECUTE 语句执行存储过程的用户必须具有 DELETE 权限。

13.4.3　执行带输入参数的存储过程

执行带输入参数的存储过程时，SQL Server 提供了如下两种传递参数的方式。

（1）直接给出参数的值，当有多个参数时，给出的参数的顺序与创建存储过程的语句中的参数的顺序一致，即参数传递的顺序就是定义的顺序。

（2）使用“参数名=参数值”的形式给出参数值，这种传递参数的方式的好处是，参数可以按任意的顺序给出。

【例 13-9】执行带输入参数的存储过程 Proc_emp_02，根据输入的员工编号，查询员工信息，这里员工编号可以自行定义，如这里定义的员工编号为 102，SQL 语句如下：

```
USE mydbase;
GO
EXECUTE Proc_emp_02 102;
```

单击“执行”按钮，即可完成执行带输入参数存储过程的操作，执行结果如图 13-14 所示。

【例 13-10】执行带输入参数的存储过程 Proc_emp_03，根据输入的员工编号，查询员工信息，这里员工编号可以自行定义，如这里定义的员工编号为 103，SQL 语句如下：

```
USE mydbase;
GO
EXECUTE Proc_emp_02 @sID=103;
```

单击“执行”按钮，即可完成执行带输入参数存储过程的操作，执行结果如图 13-15 所示。

图 13-14　执行带输入参数的存储过程

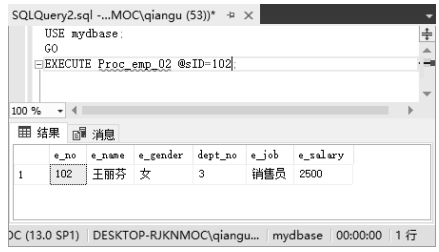


图 13-15　执行带输入参数的存储过程

提示：执行带有输入参数的存储过程时需要指定参数，如果没有指定参数，系统会提示错误，如果希望不给出参数时存储过程也能正常运行，或者希望为用户提供一个默认的返回结果，可以通过设置参数的默认值来实现。

13.4.4　执行带输出参数的存储过程

执行带输出参数的存储过程，既然有一个返回值，为了接收这一返回值，需要一个变量来存放返回参数的值，同时，在执行这个存储过程时，该变量必须加上 OUTPUT 关键字来声明。

【例 13-11】执行带输出参数的存储过程 Proc_emp_04，并将返回结果保存到@employeecount 变量中。

```
USE mydbase;
GO
DECLARE @employeecount INT;
DECLARE @sID INT =1;
EXEC Proc_emp_04 @sID, @employeecount OUTPUT
SELECT '该部门一共有' +LTRIM(STR(@employeecount)) + '员工'
GO
```

单击“执行”按钮，即可完成执行带输出参数存储过程的操作，执行结果如图 13-16 所示。

图 13-16　执行带输出参数的存储过程

13.4.5　在 SSMS 中执行存储过程

除了使用 SQL 语句执行存储过程之外，还可以在 SSMS 中以界面方式执行存储过程，具体步骤如下。

步骤 1：右击要执行的存储过程，这里选择名称为 Proc_emp_04 的存储过程。在弹出快捷菜单中选择“执行存储过程”菜单命令，如图 13-17 所示。

步骤 2：打开“执行过程”窗口，在“值”列中输入参数值：@sID=2，如图 13-18 所示。

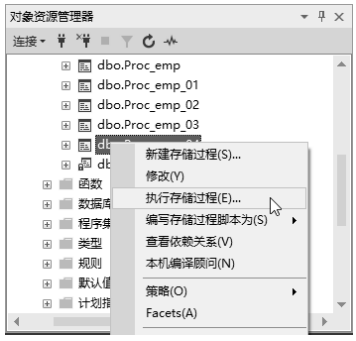


图 13-17　选择“执行存储过程”菜单命令

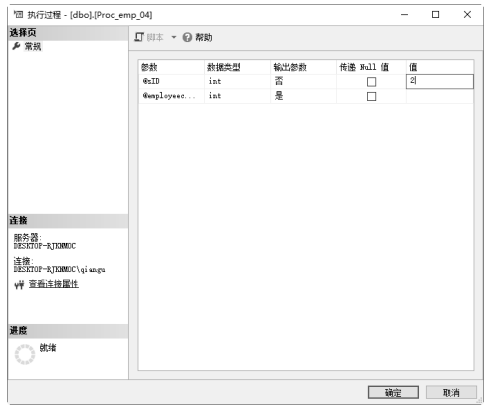


图 13-18　“执行过程”窗口

步骤 3：单击“确定”按钮执行带输入参数的存储过程，执行结果如图 13-19 所示。

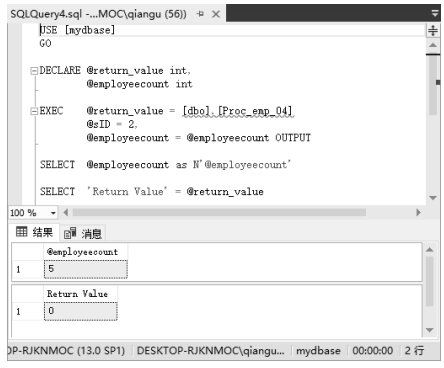


图 13-19　存储过程执行结果

13.5　修改存储过程

修改存储过程可以改变存储过程当中的参数或者语句，可以通过 SQL 语句中的 ALTER PROCEDURE 语句来实现，还可以在 SSMS 中以界面方式修改存储过程。

13.5.1　修改存储过程的语法格式

使用 ALTER PROCEDURE 语句可以修改存储过程，在修改存储过程时，SQL Server 会覆盖以前定义的存储过程，语法格式如下：

```
ALTER PROCEDURE [schema_name.] procedure_name [ ; number ]
{ @parameter data type }
[ VARYING ] [ = default ] [ OUT | OUTPUT ] [READONLY]
[ WITH  <ENCRYPTION ]|[ RECOMPILE ]|[ EXECUTE AS Clause ]> ]
[ FOR REPLICATION ]
AS  <sql_statement>
```

提示：除了 ALTER 关键字之外，这里其他的参数与 CREATE PROCEDURE 中的参数作用相同。

13.5.2 使用 SQL 语句修改存储过程

使用 SQL 语句可以修改存储过程，下面给出一个实例，来介绍使用 SQL 语句修改存储过程的方法。

【例 13-12】通过 ALTER PROCEDURE 语句修改名为 Count_Proc 存储过程，具体操作步骤如下。

步骤 1：打开 SSMS，并连接到 SQL Server 中的数据库，然后选择存储过程所在的数据库，如这里选择 mydbase，如图 13-20 所示。

步骤 2：单击工具栏中的“新建查询”按钮 新建查询(N)，新建查询编辑器，并输入以下 SQL 语句，将 SELECT 语句查询的结果按部门编号 dept_no 进行分组。

```
USE mydbase
GO
SET ANSI NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[Count Proc]
AS
SELECT dept_no,COUNT(*) AS 总数 FROM employee GROUP BY dept_no;
```

步骤 3：单击“执行”按钮，即可完成修改存储过程的操作，如图 13-21 所示。

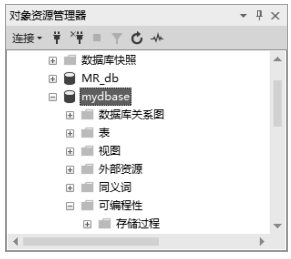


图 13-20 选择 mydbase

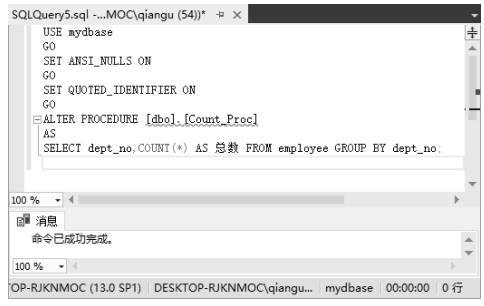


图 13-21 修改存储过程

步骤 4：下面执行修改后的 Count_Proc 存储过程，SQL 语句如下：

```
USE mydbase;
GO
EXEC Count_Proc;
```

单击“执行”按钮，即可完成存储过程的执行操作，执行结果如图 13-22 所示。

图 13-22 执行修改后的存储过程

13.5.3 在 SSMS 中修改存储过程

在 SSMS 中可以以界面方式修改存储过程，具体的操作步骤如下。

步骤 1：登录 SQL Server 服务器之后，在 SSMS 中打开“对象资源管理器”窗口，选择“数据库”结点下创建存储过程的数据库，选择“可编程性”→“存储过程”结点，右击要修改的存储过程，在弹出的快捷菜单中选择“修改”菜单命令，如图 13-23 所示。

步骤 2：打开存储过程的修改窗口，用户即可修改存储过程，然后单击“保存”按钮即可，如图 13-24 所示。

注意：ALTER PROCEDURE 语句只能修改一个单一的存储过程，如果过程调用了其他存储过程，嵌套的存储过程不受影响。

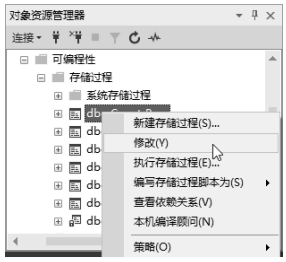


图 13-23　选择“修改”菜单命令

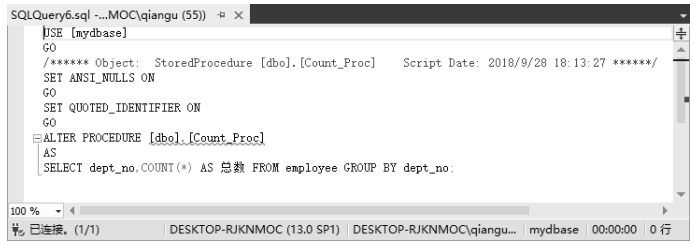


图 13-24　修改存储过程窗口

13.6　重命名存储过程

重命名存储过程可以修改存储过程的名称，这样可以将不符合命名规则的存储过程的名称根据统一的命名规则进行更改。

13.6.1　在 SSMS 中重命名存储过程

重命名存储过程可以在 SSMS 中以界面方式来轻松地完成，具体操作步骤如下。

步骤 1：选择需要重命名的存储过程，右击鼠标，并在弹出的快捷菜单中选择“重命名”菜单命令，如图 13-25 所示。

步骤 2：在显示的文本框中输入要修改的新的存储过程的名称，这里输入“dbo.Count_Proc_01”，按 Enter 键确认即可，如图 13-26 所示。

图 13-25　选择“重命名”菜单命令

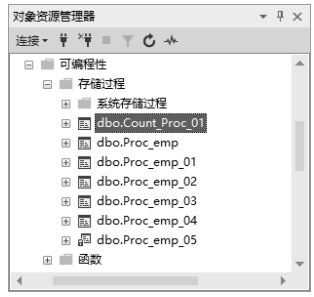


图 13-26　输入新的名称

注意：输入新名称之后，在对象资源管理器中的空白地方单击鼠标，或者直接按回车键确认，即可完成修改操作。也可以在选择一个存储过程之后，间隔一小段时间，再次单击该存储过程；或者选择存储过程之后，直接按 F2 键。这几种方法都可以完成存储过程名称的修改。

13.6.2　使用 sp_name 系统存储过程重命名

使用系统存储过程 sp_rename 也可以重命名存储过程，语法格式如下：

```
sp_rename oldObjectName,newObjectName
```

主要参数介绍如下。

- oldObjectName：存储过程的旧名称。
- newObjectName：存储过程的新名称。

【例 13-13】重命名存储过程 Count_Proc_01 为“CountProc”，SQL 语句如下：

```
sp_rename Count_Proc_01,CountProc
```

单击“执行”按钮，即可完成存储过程的重命名操作，执行结果如图 13-27 所示。

图 13-27　重命名存储过程

13.7　查看存储过程

创建完存储过程之后，需要查看修改后的存储过程的内容，查询存储过程有两种方法，一种是使用 SSMS 对象资源管理器查看，另一种是使用 T-SQL 语句查看。

13.7.1　使用 SSMS 查看存储过程信息

在 SSMS 中可以以界面方式查看存储过程信息，具体的操作步骤如下。

步骤 1：登录 SQL Server 服务器之后，在 SSMS 中打开“对象资源管理器”窗口，选择“数据库”结点下创建存储过程的数据库，选择“可编程性”→“存储过程”结点，右击要修改的存储过程，在弹出的快捷菜单中选择“属性”菜单命令，如图 13-28 所示。

步骤 2：弹出“存储过程属性”窗口，用户即可查看存储过程的具体属性，如图 13-29 所示。

图 13-28　选择“属性”菜单命令

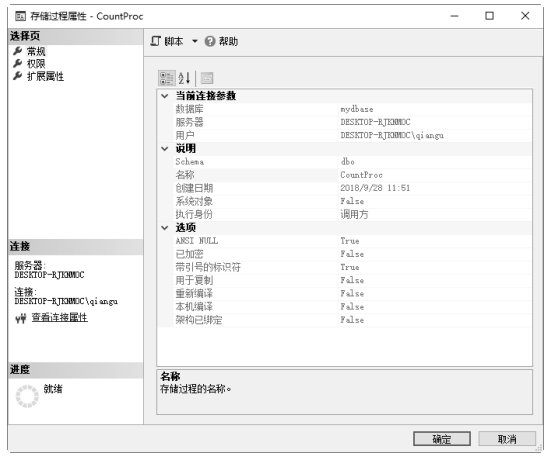


图 13-29 “存储过程属性”窗口

13.7.2　使用系统存储过程查看信息

许多系统存储过程、系统函数和目录视图都提供有关存储过程的信息，可以使用这些系统存储过程来查看存储过程的定义，即用于创建存储过程的 T-SQL 语句。可以通过下面三种系统存储过程和目录视图查看存储过程。

1. 使用 sys.sql_modules 查看存储过程的定义

sys.sql_modules 为系统视图，通过该视图可以查看数据库中的存储过程。

【例 13-14】查看存储过程 CountProc 相关信息，SQL 语句如下：

```
select * from sys.sql_modules
```

单击“执行”按钮，即可完成查看 sys.sql_modules 系统视图的操作，执行结果如图 13-30 所示。

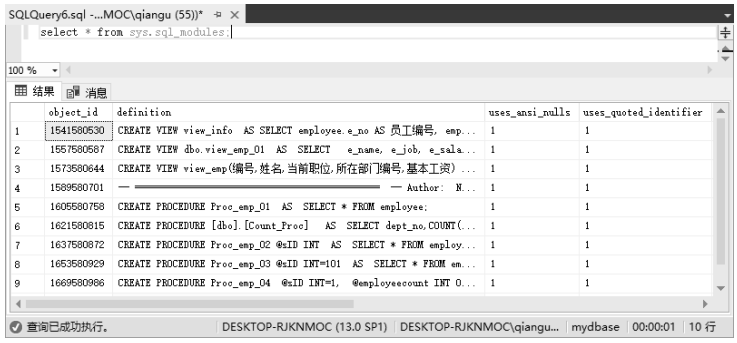


图 13-30　查看存储过程的信息

2. 使用 OBJECT_DEFINITION 查看存储过程的定义

返回指定对象定义的 T-SQL 源文本，语法格式如下：

```
SELECT OBJECT_DEFINITION(OBJECT_ID);
```

主要参数 OBJECT_ID 为要使用的对象的 ID，object_id 的数据类型为 int，并假定表示当前数据库上下文中的对象。

【例 13-15】使用 OBJECT_DEFINITION 查看存储过程的定义，SQL 语句如下：

```
USE mydbase;
GO
SELECT OBJECT_DEFINITION(OBJECT_ID('CountProc'));
```

单击“执行”按钮，即可完成使用 OBJECT_DEFINITION 查看存储过程定义的操作，执行结果如图 13-31 所示。

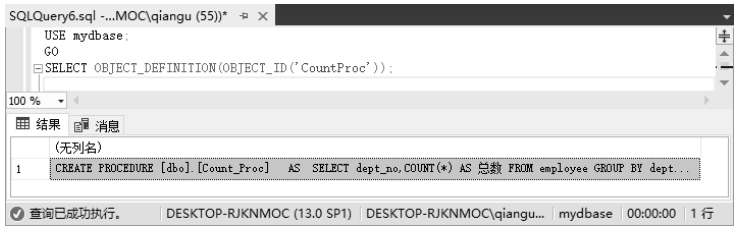


图 13-31　查看存储过程的定义

3. 使用 sp_helptext 查看存储过程的定义

显示用户定义规则的定义、默认值、未加密的 T-SQL 存储过程、用户定义 T-SQL 函数、触发器、计算列、CHECK 约束、视图或系统对象，语法格式如下：

```
sp_helptext[@objname=]'name'[,[@columnname=]computed_column_name]
```

主要参数介绍如下。

- [@objname=]'name'：架构范围内的用户定义对象的限定名称和非限定名称。
- [@columnname=]computed_column_name]：要显示定义信息的计算列的名称，必须将包含列的表指

定为 name。column_name 的数据类型为 sysname，无默认值。

【例 13-16】通过 sp_helptext 系统存储过程查看名为 CountProc 的相关定义信息，SQL 语句如下：

```
USE mydbase;
GO
EXEC sp_helptext CountProc
```

单击“执行”按钮，即可完成通过 sp_helptext 查看存储过程的相关定义信息，执行结果如图 13-32 所示。

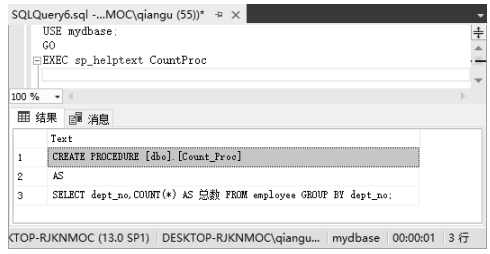


图 13-32　使用 sp_helptext 查看存储过程的定义

13.8　删除存储过程

不需要的存储过程可以删除，删除存储过程有两种方法，一种是通过图形化工具删除，另一种是使用 T-SQL 语句删除。

13.8.1　在 SSMS 中删除存储过程

删除存储过程可以在对象资源管理器中轻松地完成。具体操作步骤如下。

步骤 1：选择需要删除的存储过程，右击鼠标，在弹出的快捷菜单中选择“删除”菜单命令，如图 13-33 所示。

步骤 2：打开“删除对象”窗口，单击“确定”按钮，完成存储过程的删除，如图 13-34 所示。

图 13-33　选择“删除”命令

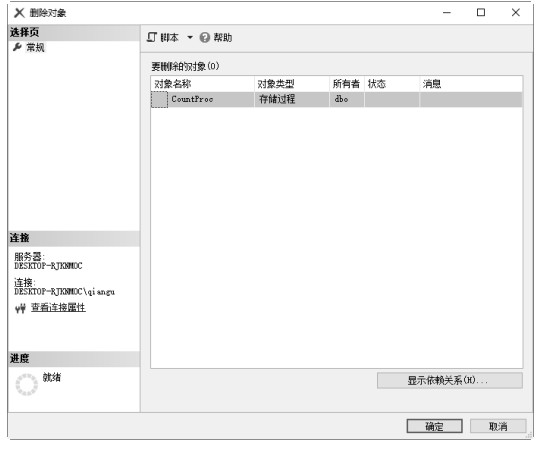


图 13-34　“删除对象”窗口

提示：该方法一次只能删除一个存储过程。

13.8.2　使用 SQL 语句删除存储过程

使用 DROP PROCEDURE 语句可以删除存储过程，该语句可以从当前数据库中删除一个或多个存储过程，语法格式如下：

```
DROP { PROC | PROCEDURE } { [ schema_name. ] procedure } [ ,···n ]
```

- schema_name：存储过程所属架构的名称。不能指定服务器名称或数据库名称。
- procedure：要删除的存储过程或存储过程组的名称。

【例 13-17】删除存储过程 CountProc，SQL 语句如下：

```
USE mydbase;
GO
DROP PROCEDURE dbo.CountProc
```

输入完成之后，单击“执行”命令，即可删除名称为 CountProc 的存储过程，如图 13-35 所示。删除之后，可以刷新“存储过程”结点，即可查看删除结果，可以看到名称为 CountProc 的存储过程不存在了，如图 13-36 所示。

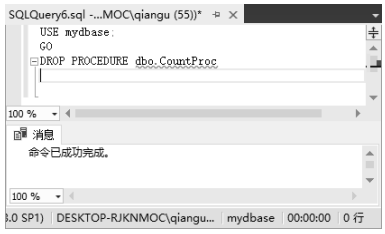


图 13-35　删除存储过程 CountProc

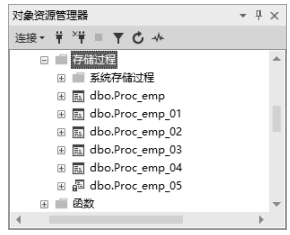


图 13-36　“对象资源管理器”窗口

13.9　扩展存储过程

扩展存储过程使用户能够在编程语言（如 C、C++）中创建自己的外部程序。扩展存储过程的显示方式和执行方式与常规存储过程一样，可以将参数传递给扩展存储过程，且扩展存储过程也可以返回结果和状态。

扩展存储过程是 SQL Server 实例可以动态加载和运行的 DLL，使用 SQL Server 扩展存储过程 API 编写的，可直接在 SQL Server 实例的地址空间中运行。SQL Server 中常规扩展存储过程如表 13-1 所示。

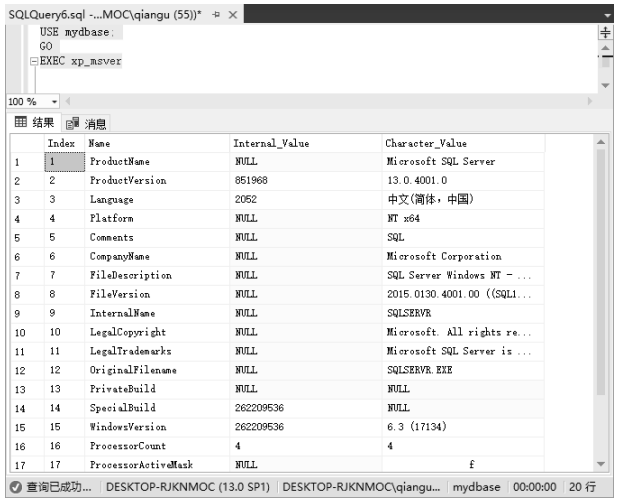
表 13-1　常规扩展过程

名　称	说　明
xp_enumgroups	提供 Windows 本地组列表或在指定 Windows 域中定义的全局组列表
xp_findnextmsg	接受输入的邮件 ID 并返回输出的邮件 ID，需要与 xp_processmail 配合使用
xp_grantlogin	授予 Windows 组或用户对 SQL Server 2016 的访问权限
xp_logevent	将用户定义消息记入 SQL Server 2016 日志文件和 Windows 事件查看器
xp_loginconfig	报告 SQL Server 2016 实例在 Windows 上运行时的登录安全配置
xp_logininfo	报告账户、账户类型、账户的特权级别、账户的映射登录名和账户访问 SQL Server 2016 的权限路径
xp_msver	返回有关 SQL Server 2016 的版本信息
xp_revokelogin	撤销 Windows 组或用户对 SQL Server 2016 的访问权限
xp_sprintf	设置一系列字符和值的格式并将其存储到字符串输出参数值。每个格式参数都用相应的参数替换
xp_sqlmaint	用包含 SQLMaint 开关的字符串调用 SQLMaint 实用工具，在一个或多个数据库上执行一系列维护操作
xp_sscanf	将数据从字符串读入每个格式参数所指定的参数位置
xp_availablemedia	查看系统上可用的磁盘驱动器的空间信息
xp_dirtree	查看某个目录下子目录的结构

【例 13-18】执行 xp_msver 扩展存储过程，查看系统版本信息，在查询编辑窗口中输入语句如下。

```
USE mydbase;
GO
EXEC xp_msver
```

单击“执行”按钮，即可完成使用扩展过程查看系统版本信息的操作，这里返回的信息包含数据库的产品信息、产品编号、运行平台、操作系统的版本号以及处理器类型信息等，执行结果如图 13-37 所示。



	Index	Name	Internal_Value	Character_Value
1	1	ProductName	NULL	Microsoft SQL Server
2	2	ProductVersion	851968	13.0.4001.0
3	3	Language	2052	中文(简体，中国)
4	4	Platform	NULL	NT x64
5	5	Comments	NULL	SQL
6	6	CompanyName	NULL	Microsoft Corporation
7	7	FileDescription	NULL	SQL Server Windows NT - ...
8	8	FileVersion	NULL	2015.0130.4001.00 ((SQL1...
9	9	InternalName	NULL	SQLSERVR
10	10	LegalCopyright	NULL	Microsoft. All rights re...
11	11	LegalTrademarks	NULL	Microsoft SQL Server is ...
12	12	OriginalFilename	NULL	SQLSERVR.EXE
13	13	PrivateBuild	NULL	NULL
14	14	SpecialBuild	262209536	NULL
15	15	WindowsVersion	262209536	6.3 (17134)
16	16	ProcessorCount	4	4
17	17	ProcessorActiveMask	NULL	f

图 13-37　查询数据库系统信息

13.10　就业面试技巧与解析

面试官：删除存储过程需要注意什么问题？

应聘者：存储过程之间可以相互调用，如果删除被调用的存储过程，那么重新编译时调用者会出现错误，所以在进行删除操作时，最好要分清各个存储过程之间的关系。