第 6 章 chapter 6

存储过程及触发器

在数据库和数据处理中，经常用到存储过程和触发器，它们可以代替逐条执行 SQL 语句方式。存储过程是预编译 SQL 语句的集合，可以作为一个单元处理，其中可以包含查询、插入、删除、更新等操作。当存储过程被调用执行时，这些操作也会同时执行。触发器作为一种特殊的存储过程，其执行由事件触发。经常用于加强数据的完整性约束和业务规则等。通过本章的学习，可以掌握存储过程和触发器的创建及管理方法，从而优化查询和提高数据的管理能力，为后续数据库应用程序的开发奠定基础。

教学课件

*第 6 章 存储过程及触发器

教学目标

(1) 理解存储过程的基本概念及创建方法。

(2) 熟悉管理存储过程的常用方法。

(3) 理解并掌握触发器的基本概念。

(4) 熟悉管理触发器的具体应用方法。

6.1 存储过程概述

【案例 6-1】 在学生成绩管理系统中，需要提供对学生成绩进行查询的功能，T-SQL 语句中的 SELECT 子句可以实现这一功能，但并不是所有的人都会使用 SELECT 查询。要解决这个问题，可以把要执行的 T-SQL 语句做成一个相对固定的语句组，根据 SQL Server 2019 中所提供的存储过程的特点，在数据库服务器端先创建一个存储过程，再调用存储过程去实现查询并返回查询结果。这样就降低了用户的操作难度，同时也减少了网络传输的数据量，提高了系统性能，并且在多次查询时，直接调用存储过程的编译结果，可以使查询的速度更快。

6.1.1 存储过程的基本概念

存储过程是一组为了完成特定功能、可以接收和返回用户参数的 T-SQL 语句预编译集合，经过编译后存储在数据库中，以某个名称存储并作为一个单元处理。存储过程保存在数据库内，可由应用程序通过某个调用执行，而且允许用户声明变量、带参数执

行，以及具有其他强大的编程功能。存储过程是一种数据库对象，在第一次执行时会进行语法检查和编译，执行后它的执行计划就驻留在高速缓存中，用于后续调用。存储过程可以接收和输出参数、返回执行存储过程的状态值，还可以嵌套调用。

6.1.2 存储过程的特点和类型

1. 存储过程的特点

1）存储过程允许标准组件式编程

创建存储过程后，可以在程序中多次调用存储过程，而不必重新编写。所有的客户端都可以使用相同的存储过程来确保数据访问和修改的一致性。存储过程可以独立于应用程序而进行修改，这大大提高了程序的可移植性。

2）存储过程在服务器端运行，执行速度快

如果某一个操作包含大量的 T-SQL 代码或者被分别多次执行，那么存储过程相对批处理的执行速度要快很多。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，并给出最终被存放在系统表中的执行计划；而批处理的 T-SQL 语句在每次运行时都要进行编译和优化，因此速度相对要慢一些。

3）存储过程能够减少网络流量

对于同一个针对数据库对象的操作（如查询、修改），用户可以通过发送一条执行存储过程的语句来实现，而不需要在网络上发送众多的 T-SQL 语句，这样可以减少在服务器与客户端之间传递语句的数量，大大减少网络流量。

知识拓展

存储过程与自定义函数的区别

4）存储过程可被作为一种安全机制充分利用

存储过程支持用户需要执行的所有业务功能，SQL Server 2019 可以不授予用户直接访问表和视图的权限，而是授权用户执行某些存储过程。系统管理员通过对执行存储过程的权限进行限制，从而能够实现对相应数据访问权限的限制，避免非授权用户对数据的访问，保证数据的安全。

存储过程虽然既有参数又有返回值，但是它与函数不同。存储过程的返回值只指明执行是否成功，并且它不像函数那样被直接调用。也就是说，在调用存储过程时，在存储过程的名字前一定要有 EXEC 关键字。

2. 存储过程的分类

存储过程保留在服务器上，是一种有效的封装重复性工作的方法，可以带参数，以完成一个特定的任务。在 SQL Server 中有多种存储过程，可分为系统存储过程、用户自定义存储过程和扩展存储过程。

1）系统存储过程

系统存储过程主要存储在 master 数据库中，一般以 sp_为前缀。系统存储过程主要从系统表中获取信息，从而为系统管理员管理 SQL Server 提供支持。通过系统存储过程，SQL Server 中的许多管理性或信息性的活动都可以被有效地执行。尽管系统存储过

程被放在 master 数据库中,但是仍可以在其他数据库中对其进行调用,在调用时不必在存储过程名前加上数据库名。而且当创建一个新数据库时,一些系统存储过程会在新数据库中被自动创建。

2) 用户自定义存储过程

用户自定义存储过程是由用户创建并能够完成某些特定功能而编写的存储过程,它可以输入参数、向客户端返回表格或结果、消息等,也可以返回输出参数。在 SQL Server 2019 中,用户自定义存储过程又分为 T-SQL 存储过程和 CLR 存储过程两种。T-SQL 存储过程保存 T-SQL 语句的集合,可以接收和返回用户提供的参数。CLR 存储过程是针对微软公司的.NET Framework 公共语言运行时(CLR)方法的引用,可以接收和返回用户提供的参数。

SQL Server 2019 还支持临时存储过程,分为局部临时过程和全局临时过程。局部临时过程只能由创建该过程的连接使用。全局临时过程则可以由所有连接使用。局部临时过程在当前会话结束时自动删除,而全局临时过程在使用该过程的最后一个会话结束时才会被删除。

临时过程用#和##命名,可以由任何用户创建。创建临时存储过程后,局部临时过程的所有者是唯一可以使用该过程的用户。执行局部临时过程的权限不能授予其他用户。如果创建的是全局临时过程,则所有用户均可以访问该过程。

3) 扩展存储过程

扩展存储过程,是指用户可以使用外部程序设计语言编写的存储过程,通常以 xp_为前缀。它以动态连接库的形式存在,能让 SQL Server 动态装载和执行,它一定要存放在系统数据库 master 中。在 SQL Server 2019 版本中,仍然保留了该类型的存储过程,但一般不建议用户使用。

☺ 讨论思考

(1) 什么是存储过程?

(2) 存储过程有哪些特点?

(3) 存储过程的分类有哪些?

6.2 存储过程的常用操作

6.2.1 创建存储过程

1. 使用 SQL 语句创建存储过程

创建自定义存储过程可以直接使用 CREATE PROCEDURE 语句。在创建存储过程之前,需要考虑以下事项。

(1) CREATE PROCEDURE 语句不能与其他 SQL 语句在单个批处理中组合使用。

(2) 要创建存储过程,必须具有数据库的 CREATE PROCEDURE 权限,还必须具有对架构的 ALTER 权限。

(3) 存储过程是架构作用域内的对象,它们的名称必须遵守标识符的命令规则。

(4) 只能在当前数据库中创建存储过程。

(5) 如果用户不是存储过程的所有者,则在使用存储过程时,必须使用对象架构名称对存储过程内所有数据定义语言语句中使用的对象名进行限定。

创建存储过程的语法如下:

```
CREATE PROCEDURE|PROCprocedure_name [ ; number ]
  [ { @parameter data_type } [ VARYING ] [ =default ] [ OUTPUT ] ] [ ,…n ]
  [ WITH { RECOMPILE | ENCRYPTION | RECOMPILE , ENCRYPTION } ]
  [ FOR REPLICATION ]
  AS
  sql_statement [ …n ]
```

其中,仍然使用基本的 CREATE <Object Type><Object Name>语法,它是每个 CREATE 语句的主干。PROCEDURE 或 PROC 在使用时都可行,为防止出现例外情况,建议对所选择的选项保持一致。存储过程的名称必须遵循命名规则。

在对存储过程命名后,需要参数列表。各参数的详细解释如下。

知识拓展
存储过程中参数与变量的区别

(1) procedure_name 是存储过程的名称。要创建局部临时过程,可以在 procedure_name 前面加一个编号符(# procedure_name)。要创建全局临时过程,可以在 procedure_name 前面加两个编号符(# # procedure_name)。完整名称(包括 # 和 # #)不能超过 128 个字符。

(2) "; number"是可选的整数,用来对同名的过程分组,以便用一条 DROP PROCEDURE 语句将同组的过程一起删除。

(3) @parameter 是存储过程的参数。在 CREATE PROCEDURE 语句中可以声明一个或多个参数。除非定义了该参数的默认值,否则用户必须在执行过程时提供每个所声明参数的值。使用 @ 符号作为第一个字符来指定参数名称。参数名称必须符合标识符的命令规则。每个过程的参数仅用于该过程本身;相同的参数名称可以用在其他过程中。默认情况下,参数只能代替常量,而不能用于代替表名、列名或其他数据库对象的名称。

(4) data_type 是参数的数据类型。所有数据类型均可以用作存储过程的参数。其中,cursor 数据类型只能用于 OUTPUT 参数。

(5) VARYING 指定作为输出参数支持的结果集(由存储过程动态构造,内容可以变化)。该参数仅适用于游标参数。

(6) default 是参数的默认值。若定义默认值,不必指定该参数的值即可执行该过程。

(7) OUTPUT 表明该参数是返回参数,可将该参数的值返回给 EXEC[UTE]调用语句。使用此关键字的输出参数可以是游标占位符。

(8) RECOMPILE 表明 SQL Server 不会缓存该过程,该存储过程将在运行时被重新编译。

(9) ENCRYPTION,表示 SQL Server 加密存储过程的文本,防止用户使用系统存储过程来读取该存储过程的文本定义。

(10) FOR REPLICATION,指定不能在订阅服务器上执行为复制创建的存储过程。本选项不能和 WITH RECOMPILE 选项一起使用。

(11) AS 指定过程要执行的操作。

(12) sql_statement。过程中要包含的任意数目和类型的 T-SQL 语句。

【案例 6-2】 在 XSCJ 数据库中,使用 CREATE PROCEDURE 语句创建一个存储过程,用来根据学号查询学生信息。具体代码如下:

```
CREATE PROCEDURE Proc_stu1
@Proc_Sno char(10)
AS
SELECT * FROM Student WHERE Sno= @Proc_Sno
```

运行结果如图 6-1 所示。

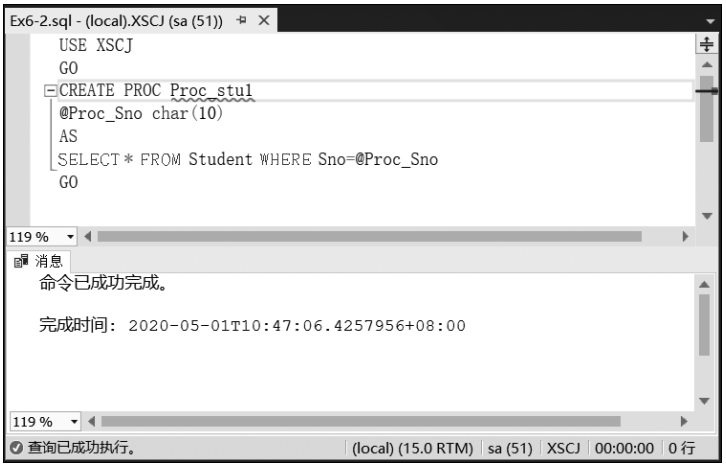

图 6-1 存储过程创建示例

2. 在 SQL Server Management Studio 中创建存储过程

【案例 6-3】 在 XSCJ 数据库中,创建存储过程实现对学生成绩进行查询。要求在查询时提供需要查询的学生姓名和课程名,存储过程根据用户提供的信息对数据进行查询,并显示成绩信息。

按照如下步骤用 SSMS 创建一个能够解决这一问题的存储过程。

(1) 启动 SQL Server Management Studio,登录服务器,在"对象资源管理器"窗格中,选择本地数据库实例→"数据库"→XSCJ→"可编程性"→"存储过程"选项。

(2) 右击"存储过程"选项,在弹出的快捷菜单中选择"存储过程"选项,如图 6-2 所示。

(3) 出现如图 6-3 所示的创建存储过程的查询编辑器窗格,其中已经加入了创建存储过程的代码。

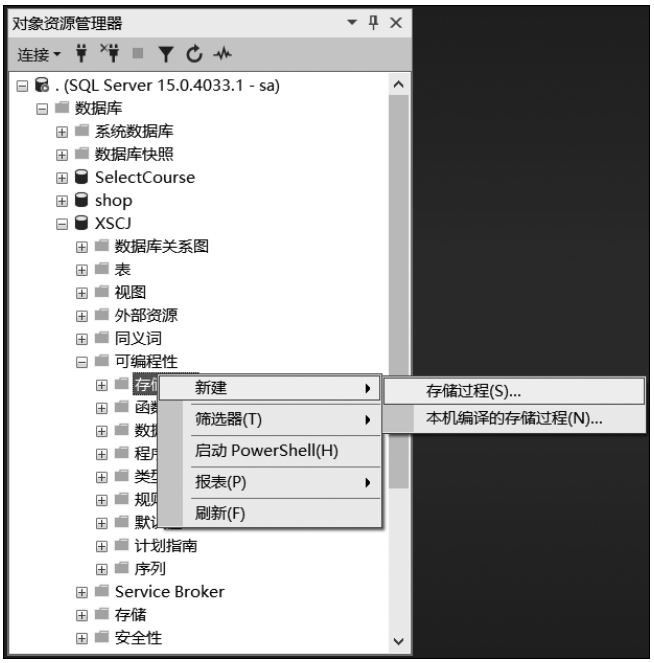

图 6-2 SQL Server Management Studio 中“存储过程”选项

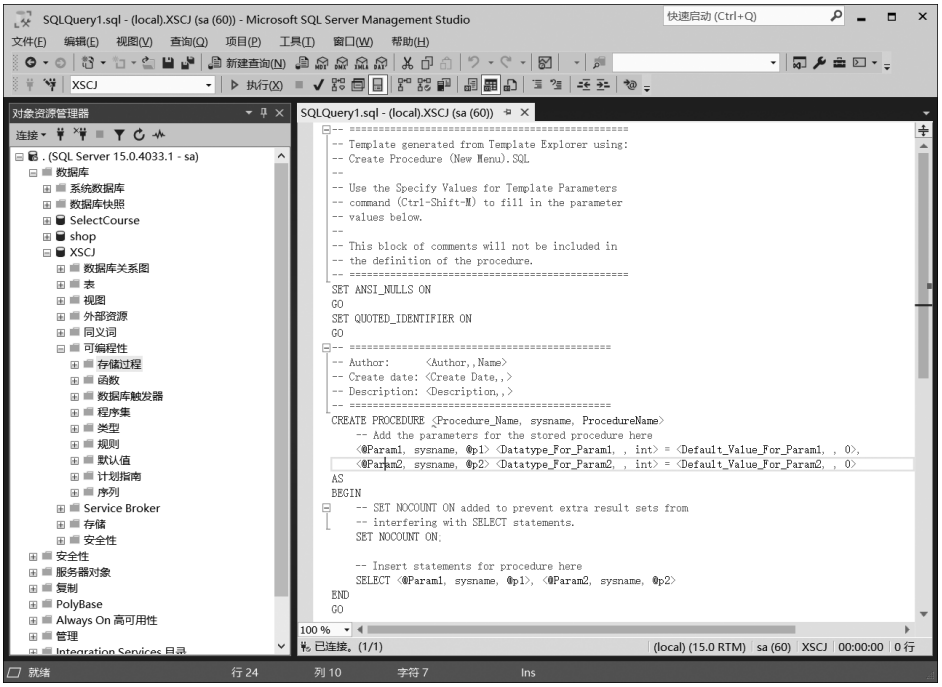

图 6-3 创建存储过程

(4) 单击菜单栏上的“查询”→“指定模板参数的值”选项，弹出如图 6-4 所示的对话框，其中 Author(作者)、Create Date(创建时间)、Description(说明)为可选项，内容可以

为空。Procedure_Name 为存储过程名,@Param1 为第一个输入参数名,Datetype_For_Param1 为第一个输入参数的类型,Default_Value_For_Param1 为第一个输入参数的默认值。后面为第二个输入参数的相关设置,这里不再赘述。

图 6-4 “指定模板参数的值”对话框

在本例中,将存储过程名设置为 Proc_Q_stugrade,第一个参数名为@stuname,类型为 varchar(10);第二个参数名为@kcname,类型为 varchar(50),其他内容设置为空。

(5) 设置完毕,单击“确定”按钮,返回创建存储过程的查询编辑器窗格,如图 6-5 所示,此时代码已经改变。

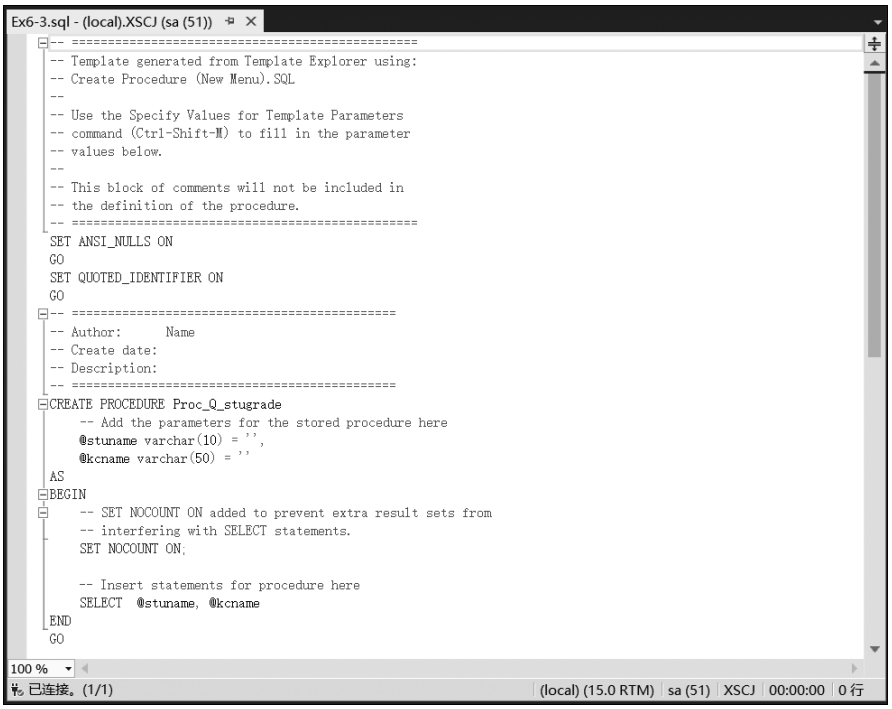

图 6-5 设置参数后的查询编辑器

(6) 在 Insert statements for procedure here 下面输入 T-SQL 代码，在本例中输入：

```
SELECT student.sname,course.cname,score.grade
FROM student INNER JOIN
    score ON student.sno=score.sno INNER JOIN
    course ON score.cno=course.cno
WHERE student.sname=@stuname AND course.cname=@kcname
```

(7) 单击“执行”按钮完成操作，设计完成的存储过程如图 6-6 所示。

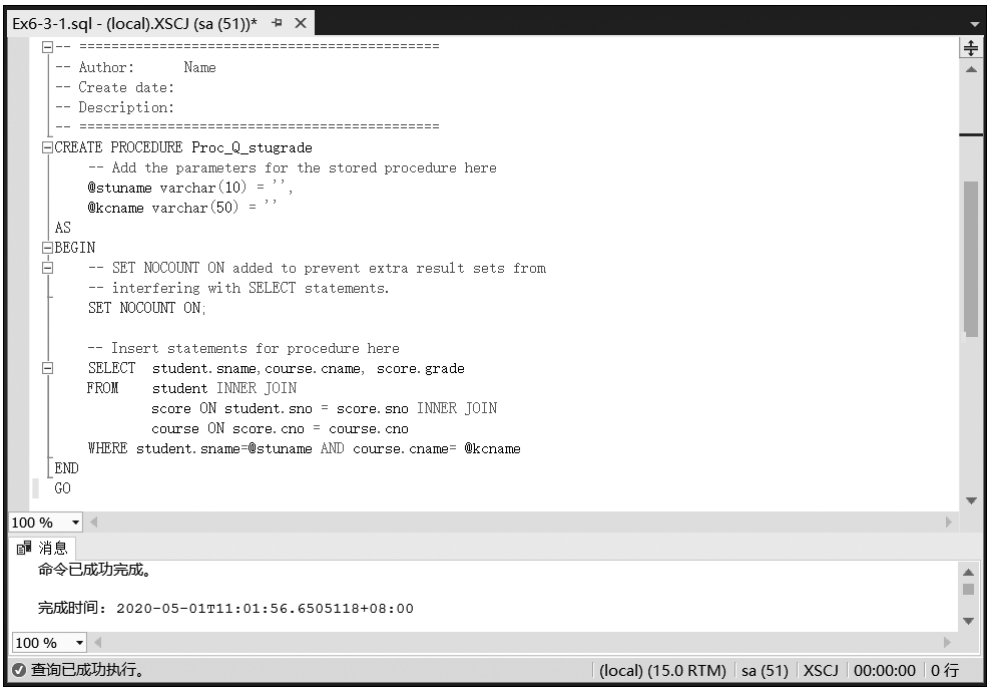

图 6-6 设计完成的存储过程

6.2.2 创建参数化存储过程

存储过程提供了一些过程式的能力，也提升了性能，但是如果存储过程没有接收一些数据，告诉其完成的任务，则在大多数情况下，存储过程不会有太多的帮助。例如，更新了表中的一些记录，并且想知道更新的数量，这时，就不太容易以记录集的形式获取该信息，而要使用输出参数。

知识拓展
参数化存储过程的优势

在存储过程的外部，可以通过位置或者引用传递参数。在存储过程的内部，由于它们使用同样的方式声明，不用关心参数传递的方式。

如果存储过程需要带参数，在编写时，直接在 CREATE PROCEDURE 语句后附加参数，不同于函数，存储过程的参数不需要用括号括起。在案例 6-2 中，SELECT 语句的 WHERE 条件的参考值为存储过程的参数@Proc_Sno，而这个@Proc_Sno 是 char 类型

的,代码如下:

```
CREATE Procedure Proc_stu1
@Proc_Sno char(10)
AS
SELECT * FROM Student WHERE Sno=@Proc_Sno
--调用存储过程
EXECUTE Proc_stu1 '1812010002'
GO
```

调用存储过程后执行结果如图 6-7 所示。

图 6-7　调用存储过程

执行存储过程的实例如下:

```
--值的顺序必须按照参数的顺序
EXEC 存储过程名 值[, 值 n…]
--值的顺序可以和参数顺序不同
EXEC 存储过程名 参数名=值[, 参数名 n=值 n…]
--使用已声明并赋值的变量作为值
EXEC 存储过程名 参数名=变量[, 参数名 n=变量 n…]
--调用具有 OUTPUT 参数的存储过程
EXEC 存储过程名 参数名 OUTPUT[, 参数名 n=值 n…]
```

上面是集中执行存储过程的方式,如果在执行存储过程时,执行语句是批处理中的第一个语句,则可不指定 EXECUTE 或 EXEC 关键字。

在以上的存储过程执行方式中,用 OUTPUT 修饰过的参数称为返回参数,就是存储过程用来向调用方返回值的。与 RETURN 只能返回数值类型的值不同,OUTPUT 类型返回参数可以返回其他类型的值,只要声明一个与存储过程标识了 OUTPUT 关键字的对应参数类型一致的变量,然后将这个变量作为值传递给存储过程中对应的返回参数即可。

6.2.3 查看及修改存储过程

在 SQL Server 2019 中，可用系统存储过程和目录视图查看有关存储过程的信息。

1. 使用 SSMS 查看存储过程

在 SQL Server Management Studio 中，首先找到要查看的存储过程，然后右击要查看的存储过程，弹出快捷菜单，如图 6-8 所示。

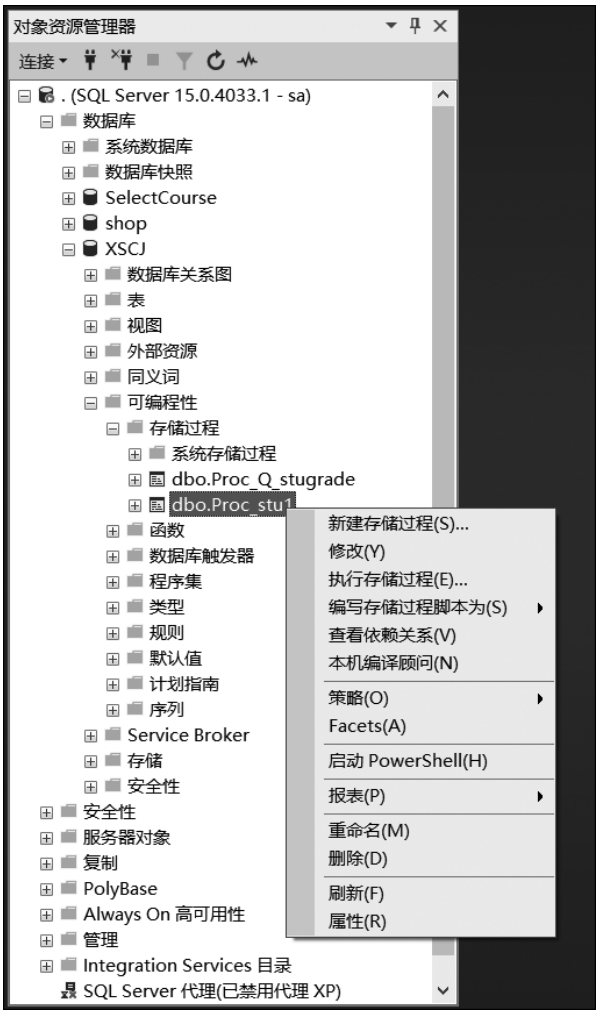

图 6-8 存储过程的快捷菜单选项

如果要查看存储过程的源代码，可以在弹出的快捷菜单中选择“修改”命令，即可在查询编辑器中查看该存储过程的定义文本，如图 6-9 所示。

如果要查看存储过程的相关性，在弹出的快捷菜单中选择“查看依赖关系”命令即可。