

专题6

存储过程

6.1

T-SQL语言

6.2

存储过程的创建与执行

6.3

存储过程修改与删除

6.4

存储过程应用举例

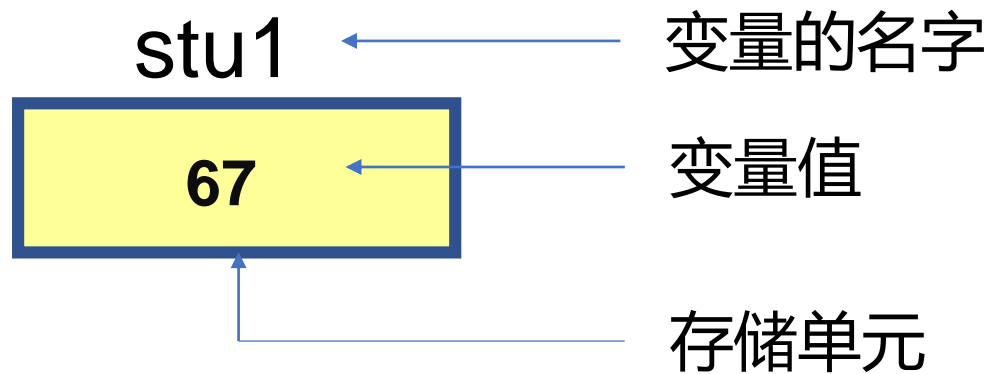
6.1

T-SQL语言

- ➔ 变量
- ➔ 注释语句
- ➔ 控制语句
- ➔ 批处理命令

01 变量

- 变量有名字、类型和值三个要素



- 变量的分类:

局部变量

• 由用户定义和赋值

全局变量

• 由系统定义和维护

01 局部变量

◆ 变量的定义语句

DECLARE @变量名称 变量类型 [, ...n]

```
DECLARE @sportavg int
```

```
DECLARE @name char(20), @birthday datetime
```

注意： 在局部变量没有被赋值之前，它的值是NULL。

01 局部变量

◆ 变量的赋值语句

SET @局部变量 = 变量值

SELECT @局部变量 = 变量值[, ...n]

```
DECLARE @sportavg int
```

```
DECLARE @name char(20), @birthday datetime
```

```
SET @Name = '张强'
```

```
SELECT @sportavg = 82, @birthday = '1999-5-8'
```

01 局部变量

- ◆ **例1** 创建一个名为@sex的局部变量，并在 SELECT 语句中使用该局部变量查找所有女同学的学号、姓名。

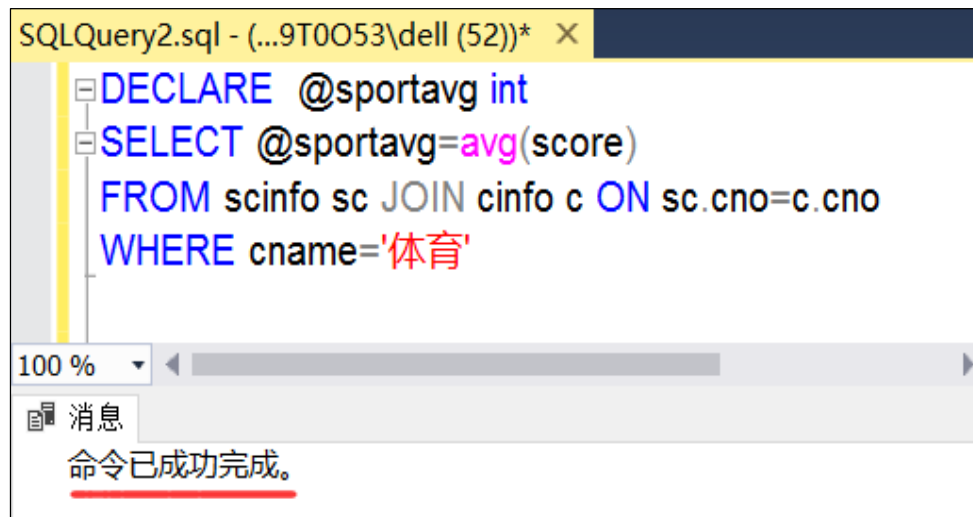
```
DECLARE @sex char(2)
SET @sex='女'
SELECT stid,stname
FROM stinfo
WHERE stsex=@sex
```

结果 消息		
	stid	stname
1	0603170211	曾莉娟
2	2001160115	邓红艳
3	2001160206	金萍
4	2001170205	张好然
5	2001170206	李娜
6	2201150204	宋羽佳
7	2602170105	杨平娟
8	2602170107	刘小玲
9	2701170101	孙洋
10	2701180101	李玉霞

01 局部变量

- ◆ **例2** 创建一个名为@sportavg的局部变量，用于保存所查询出来的体育课的平均分。

```
DECLARE @sportavg int
SELECT @sportavg=avg(score)
FROM scinfo sc JOIN cinfo c ON sc.cno=c.cno
WHERE cname='体育'
```



The screenshot shows a SQL query execution window titled "SQLQuery2.sql - (...9T0053\dell (52))*". The query text is as follows:

```
DECLARE @sportavg int
SELECT @sportavg=avg(score)
FROM scinfo sc JOIN cinfo c ON sc.cno=c.cno
WHERE cname='体育'
```

Below the query text, there is a progress bar showing 100% completion. At the bottom, a message box displays the text "命令已成功完成。" (Command completed successfully.)

01 局部变量

变量的输出语句

- PRINT @变量名
- SELECT @变量名

```
DECLARE @sportavg int
SELECT @sportavg=avg(score)
FROM scinfo sc JOIN cinfo c
      ON sc.cno=c.cno
WHERE cname='体育'

SELECT @sportavg
PRINT @sportavg
```

PRINT输出的结果 SELECT输出的结果



(无列名)	
1	87

区别:

- PRINT 是以文本方式在消息栏显示结果
- SELECT是以表格形式在结果栏显示结果

02 全局变量

- ◆ 全局变量不需要用户声明，是服务器级定义的，作用范围是任何程序。
- ◆ 全局变量以@@开头，分为两大类,记录了系统的活动状态或系统内部信息。

变 量	含 义
@@ERROR	最后一个T-SQL错误的错误号
@@IDENTITY	最后一次插入的标识值
@@ROWCOUNT	受上一个SQL语句影响的行数
@@SERVERNAME	本地服务器的名称

03 注释语句

- 在T-SQL语言中可使用两种注释符：行注释和块注释。
- 注释不参与代码执行。

-- 行注释用于描述性文字较少的场合

/**

块注释用于描述性文字较多的场合

.....

.....

**/

```
DECLARE @stn char(10), @sex char(2) -- 定义了两个局部变量
```

```
SELECT @stn=stname, @sex=StSex
```

```
FROM stinfo
```

```
WHERE StID='2001160308'
```

```
SELECT @stn '姓名:', @sex '性别' /** 输出学号为2001160308
```

```
学生的姓名和性别**/
```

04 控制语句

- ◆ 条件语句
- ◆ 循环语句
- ◆ 分支语句

04 控制语句

- ◆ 语句块：在流程控制语句中作为一个整体来执行

BEGIN

<SQL语句或语句块>

END

注意： **BEGIN** 和 **END** 语句必须成对使用

04 控制语句

◆ 条件语句

```
IF <条件表达式>      --进行条件判断
    <SQL语句1或语句块1>
[ELSE [条件表达式]
    <SQL语句2或语句块2>
]
```

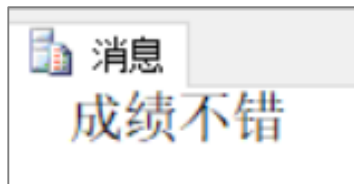
```
--定义变量
DECLARE @x int, @y int
SET @x=1    --给变量赋值
SET @y=4
IF @x>@y
    PRINT 'x>y'
ELSE
    PRINT 'x<y'
```

04 控制语句

条件分支语句

◆ **例3** 计算体育课的平均成绩，如果体育课程平均成绩在80分以上则显示“考试成绩不错” 否则显示“需要继续努力”

分析： 1) 计算体育课的平均成绩
2) 对@sportavg的值进行判断，
大于80分就显示考试成绩不错，
否则显示继续努力



```
DECLARE @sportavg int
SELECT @sportavg=avg(score)
FROM scinfo sc JOIN cinfo c
      ON sc.cno=c.cno
WHERE cname='体育'

IF @sportavg>=80
    PRINT'成绩不错'
ELSE
    PRINT'需要继续努力'
```

04 控制语句

◆ 循环语句

WHILE <循环条件表达式>

BEGIN

< SQL语句或语句块>

[BREAK]

[CONTINUE]

[SQL语句或语句块]

END

04 控制语句

循环语句

◆ 例4 分析以下代码功能

```
DECLARE @num int
SET @num=80
WHILE @num<100
BEGIN
    SELECT '平均分大于'+ str(@num)+'学生: '
    SELECT stdid,AVG(score)
    FROM SCInfo
    GROUP BY StID
    HAVING AVG(score)> @num
    SET @num=@num+10
END
```

```
SELECT stdid,AVG(score)
FROM SCInfo
GROUP BY StID
HAVING AVG(score)> @num
```

@num

80

90

100

查询出平均分超过 @num值的学生的学号及平均分

04 控制语句

例4 分析以下代码功能

```
DECLARE @num int
SET @num=80
WHILE @num<100
BEGIN
    SELECT '平均分大于'+ str(@num)+'学生: '
    SELECT stdid,AVG(score)
    FROM SCInfo
    GROUP BY StID
    HAVING AVG(score)> @num
    SET @num=@num+10
END
```

(无列名)		
1	平均分大于	80学生:
	stdid	(无列名)
2	0603170212	92
3	2001160115	89
4	2001160206	91
5	2201150101	88
6	2602170106	97
7	2602170107	92
8	2602170108	83
9	2701180101	90
(无列名)		
1	平均分大于	90学生:
	stdid	(无列名)
1	0603170211	92
2	0603170212	92
3	2001160206	91
4	2602170106	97
5	2602170107	92

04 控制语句

◆ 分支语句

```
CASE <表达式>
    WHEN <表达式值1> THEN <结果表达式1>
    .....
    WHEN <表达式值n > THEN <结果表达式n>
    [ELSE <结果表达式n+1>]
END
```

注意：CASE语句不同于其它的T_SQL语句，不能作为独立的语句来执行，而是需要作为其它语句的一部分来执行。

04 控制语句

例5 分析以下代码功能

```
SELECT std,cno,score=
CASE score/10
    WHEN 6 then '及格'
    WHEN 7 then '中'
    WHEN 8 then '良'
    WHEN 9 then '优'
    WHEN 10 then '优'
    ELSE '不及格'
END
FROM scinfo
```

	std	cno	score
1	0603170108	9710021	不及格
2	0603170108	9710041	及格
3	0603170109	9710041	中
4	0603170110	9710041	不及格
5	0603170211	9710041	优
6	0603170211	1805012	良
7	2001160115	9710011	良
8	2001160115	9720013	优
9	2001160206	9710011	良
10	2001160206	9720013	优
11	2001160307	9710011	中
12	2001160307	9720013	中

05 批处理

- 在SQL Sever中，可以一次执行多个T-SQL语句，这些多个T-SQL语句称为批处理语句。
- “GO” 就是批处理的标志

```
USE sems
```

```
GO --批处理标志
```

```
SELECT * FROM stinfo
```

```
SELECT * FROM scinfo
```

```
GO --批处理标志
```

06 RETURN语句

- 语法： RETURN [整型表达式]
- 无条件地从存储过程、批处理或语句块中退出，在RETURN之后的其他语句不会被执行。
- 存储过程返回0值表示成功，除非特别指明，返回非零值表示失败。

07 小结

- ◆ 变量
- ◆ 流程控制语句
- ◆ 其他语句

专题6

存储过程

6.1

T-SQL语言

6.2

存储过程的创建与执行

6.3

存储过程修改与删除

6.4

存储过程应用举例

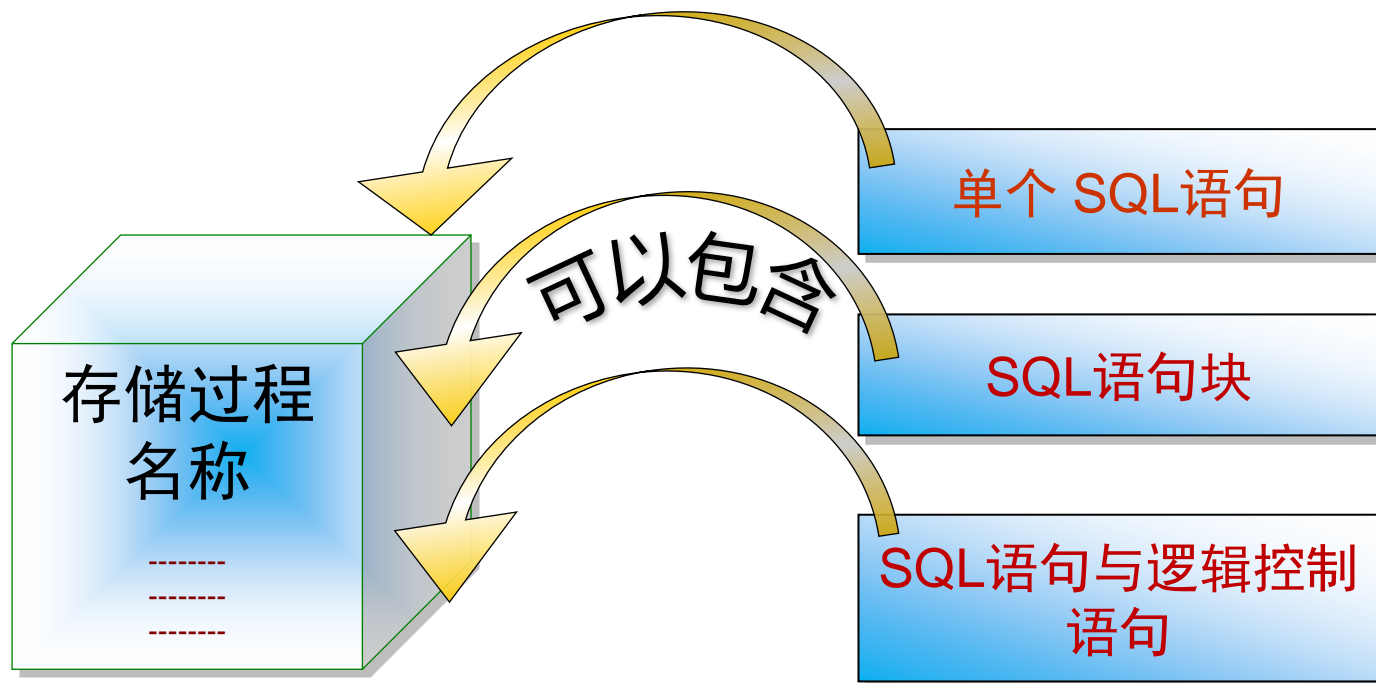
6.2

存储过程的创建与执行

- ➡ 存储过程概述
- ➡ 存储过程的创建和执行

01 存储过程概述

- 存储过程是预编译好的一组完成特定功能的SQL语句集合，它与特定的数据库相关联，存储在SQL Server服务器上。
- 存储过程以一个名称存储并作为一个单元处理，可以通过调用的方法执行存储过程，以实现它定义的功能。

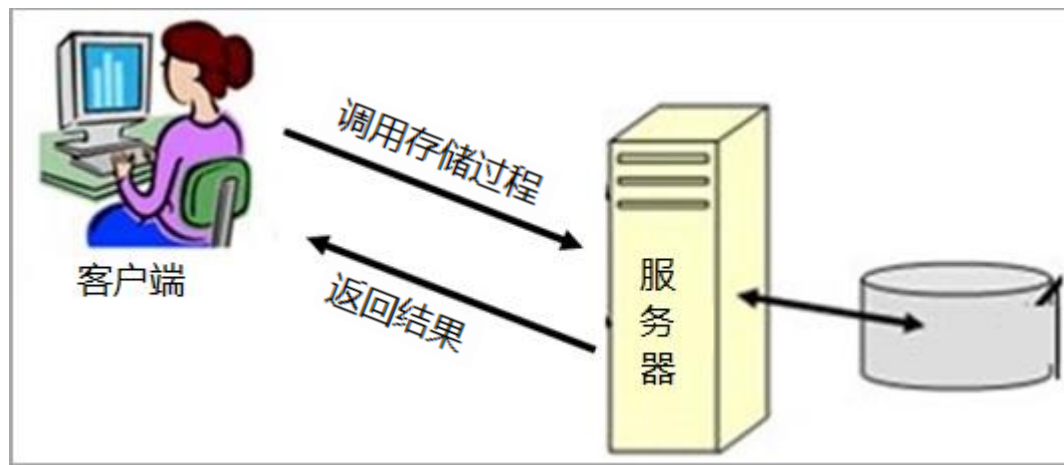


01 存储过程概述

存储过程的特点

存储过程是在服务器端运行，执行速度快，能有效提高数据使用效率。

- 改善系统运行性能
- 实现代码重用
- 减少网络流量
- 增强安全性



01 存储过程概述

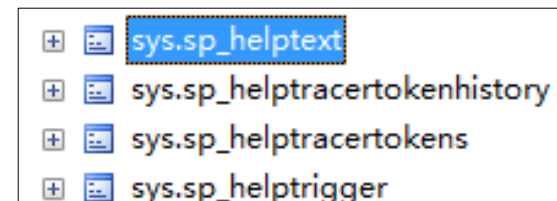
存储过程的类型

- 系统存储过程 (sp_):

由数据库系统自身创建，存储在master数据库中，以 “sp_” 前缀标识。

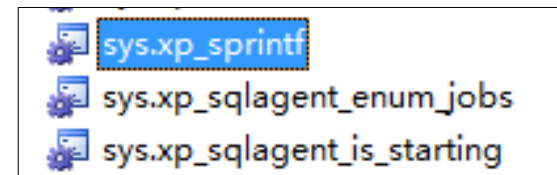
- 用户定义存储过程 (本地存储过程):

由用户创建，存放在用户数据库中。



- 扩展存储过程:

以动态链接库 (DLL) 的形式实现。以 “xp_” 为前缀，只能添加到master数据库中，使用方法与系统存储过程一样。



02 存储过程的创建和执行

◆ CREATE PROCEDURE语句创建存储过程

语法格式:

```
CREATE PROC[EDURE] 存储过程名  
    [{ @参数名 数据类型} [=default] [ OUTPUT ] ] [...n]  
AS  
    SQL语句体 [...n]
```

◆ EXECUTE (或 EXEC) 命令执行存储过程

```
[ EXEC[UTE] ] 存储过程名 [实参 [,OUTPUT] ] [...n] ]
```

注意: 如果存储过程是批处理中的第一条语句, EXECUTE命令可以省略

02 存储过程的创建和执行

- ◆ 不带参数的存储过程
- ◆ 带输入参数的存储过程
- ◆ 带输出参数的存储过程



不带参数的存储过程

- 创建语句语法格式：

```
CREATE PROC[EDURE] 存储过程名  
  
AS  
  
SQL语句
```

- 执行语句语法格式：

```
[ EXEC[UTE] ] 存储过程名
```



不带参数的存储过程的创建

例1 在sems数据库中创建一个名为pStu的存储过程，查询法学1603班学生的姓名、性别、电话。

USE sems

GO

--存储过程的内容如下:

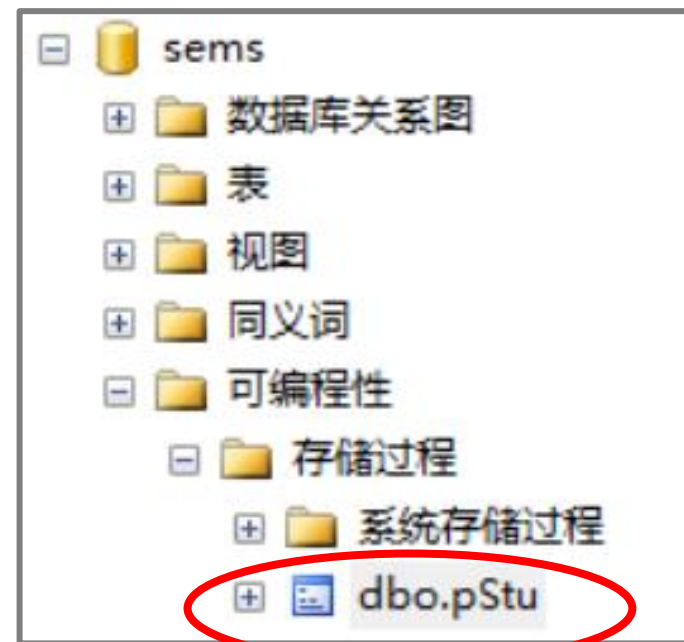
CREATE PROCEDURE pStu

AS

SELECT Stname,StSex,Telephone

FROM StInfo

WHERE CName='法学1603'



运行CREATE PROCEDURE语句，在服务器上创建存储过程



不带参数的存储过程

例1 在sems数据库中创建一个名为pStu的存储过程，查询法学1603班学生的姓名、性别、电话。

```
CREATE PROCEDURE pStu
AS
SELECT Stname, StSex, Telephone
FROM St_Info
WHERE CName='法学1603'
```

运行存储过程:

在查询设计器窗口，输入命令：

EXEC pStu



SQLQuery24.sql - ...ACHER\admin (58))* SQLQuery23.sql - ...ACHER\

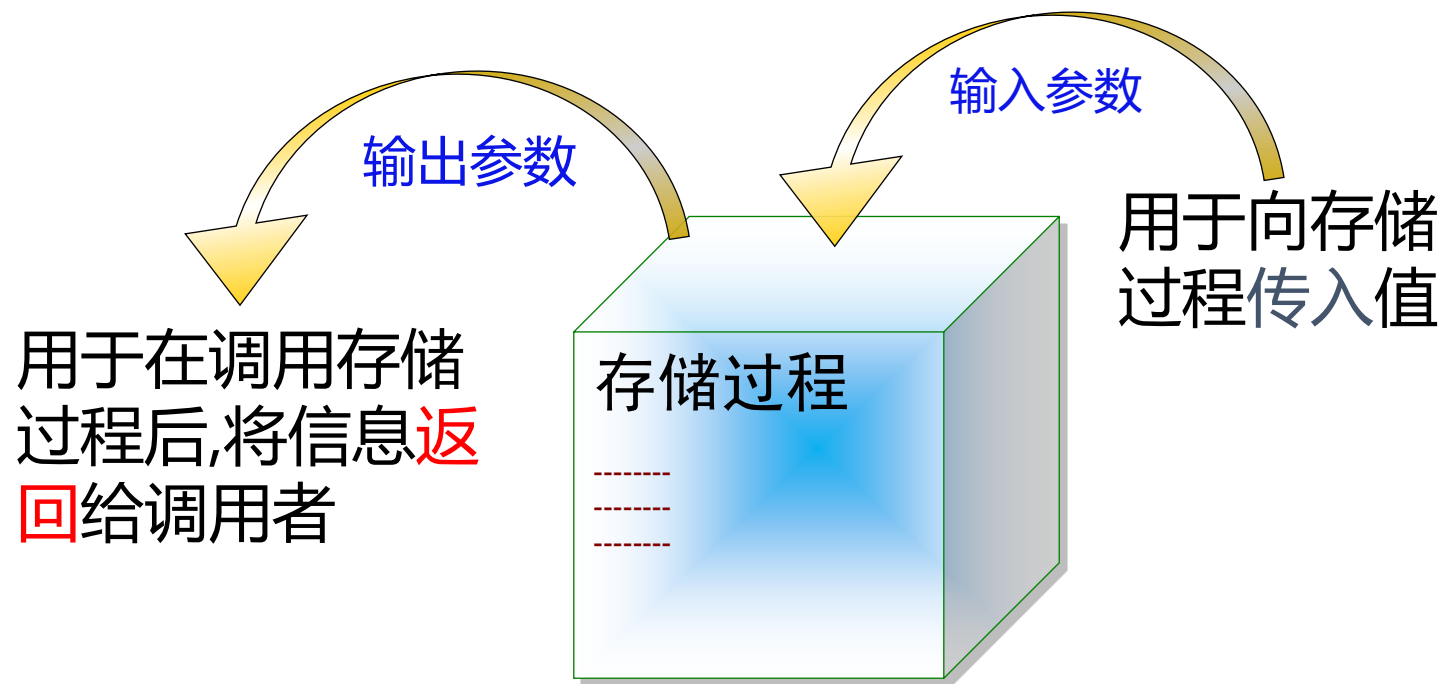
EXEC pstu

	StID	stname	StSex	Telephone
1	2001160307	吴中华	男	0310_45687959
2	2001160308	王铭	男	0371_64586366



带参数的存储过程

- 存储过程参数: “输入” 和 “输出” 参数





带输入参数的存储过程的创建与执行

- 创建语句语法格式:

```
CREATE PROC[EDURE]  存储过程名  
  { @输入参数名1 数据类型 [=default] }  
  [,...n]  
AS  
    SQL语句 [...n]
```

Default表示参数的默认值，如果定义了默认值，则在执行存储过程时，可以不必指定该参数的值

- 执行语句的语法格式:

```
EXEC 过程名 [参数]
```

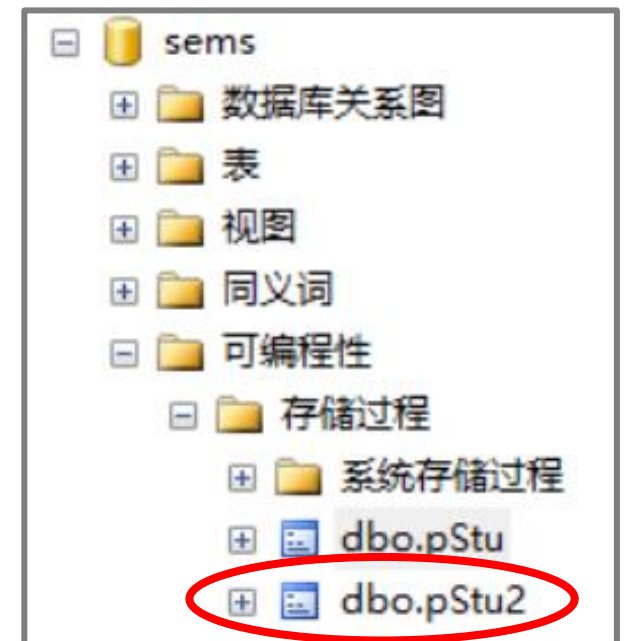
实参，必须具有确定的值



带输入参数的存储过程的创建与执行

例2 创建带一个输入参数的存储过程pstu2，从StInfo表中查询指定班级的学生姓名、性别、电话。

```
CREATE PROCEDURE pstu2  
    @class varchar(30)  
AS  
    SELECT Stname, StSex, Telephone  
    FROM StInfo  
    WHERE clname = @class
```





带输入参数的存储过程的执行

例2 创建带一个输入参数的存储过程pstu2, 从St_Info表中查询指定班级的学生姓名、性别、电话

```
CREATE PROCEDURE pstu2 @class varchar(20)
AS
SELECT StID,stname,StSex,Telephone
FROM StInfo
WHERE CName=@class
```



SQLQuery5.sql - (L...ACHER\admin (52))* SQLQuery3.sql - 未连接

EXEC pstu2 '材料科学1702'

结果 消息

	StID	stname	StSex	Telephone
1	0603170211	曾莉娟	女	0710_65234237
2	0603170212	张红飞	男	0370_54586336

带输入参数的存储过程的执行

```
EXECUTE pstu2 '材料科学1702'
```

如果没有给出具体的班级名,就查询法学1603班的学生相关信息,应该如何解决?



带默认值的输入参数的存储过程的执行

例3 创建带一个输入参数的存储过程pstu3，从StInfo表中查询指定班级的学生姓名、性别、电话，若没有指定班级名，则默认班级为“法学1603”。

```
CREATE PROCEDURE pstu3  
    @class varchar(20) = '法学1603'  
AS  
SELECT StID, stname, StSex, Telephone  
FROM StInfo  
WHERE CName=@class
```

带输入参数的存储过程的执行

```
EXEC pstu3 '材料科学1702'  
EXEC pstu3
```

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL query: `EXEC pstu3 '材料科学1702'`. The bottom pane shows the results of the query, which is a table with 5 columns: StID, stname, StSex, and Telephone. The results are divided into two sections, each containing 2 rows of data.

	StID	stname	StSex	Telephone
1	0603170211	曾莉娟	女	0710_65234237
2	0603170212	张红飞	男	0370_54586336

	StID	stname	StSex	Telephone
1	2001160307	吴中华	男	0310_45687959
2	2001160308	王铭	男	0371_64586366



带多个输入参数的存储过程的执行

例4 创建带两个输入参数的存储过程 **ScoreInfo**，查询某个学生某门课程的考试成绩，要求显示学生名、课程名及课程分数。

```
CREATE PROCEDURE ScoreInfo
```

```
    @stname varchar(20), @course varchar(30)
```

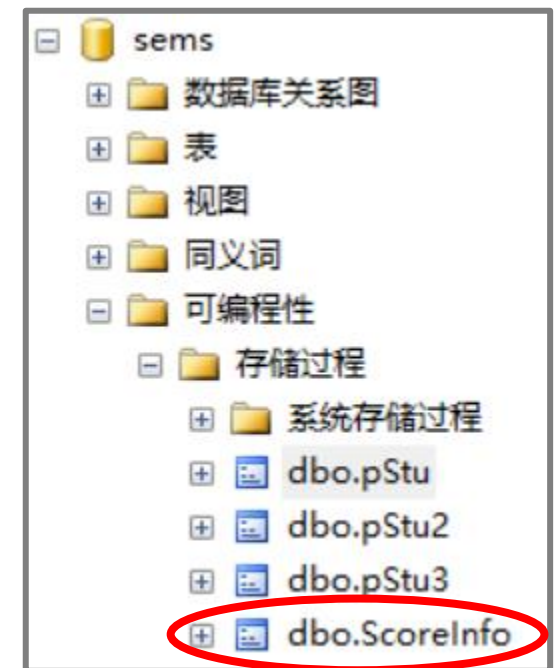
```
AS
```

```
SELECT StName, CName, Score
```

```
FROM StInfo s JOIN SCInfo sc ON s.stid=sc.stid
```

```
    JOIN CInfo c ON sc.cno=c.cno
```

```
WHERE StName = @stname AND Cname = @course
```





带多个输入参数的存储过程的执行

- 点击“新建查询”按钮，在“查询设计器”窗格中输入并运行如下命令：

EXEC ScoreInfo '李玉霞', '网络技术与应用'

输出结果如下图所示。

StName	CName	Score
李玉霞	网络技术与应用	90



存储过程的参数传递方式

执行带多个参数的存储过程时，两种传递参数的方式：

- **按位置传递参数：** 以参数的定义顺序列出提供值

EXEC ScoreInfo '李玉霞', '网络技术与应用'

- **按参数名传递：** 当通过参数名传递值时，可以以任何顺序指定参数值，

EXEC ScoreInfo @stname = '杨平娟', @course = '体育'



带输出参数的存储过程的创建与执行

- 创建语句的语法格式:

```
CREATE PROC[EDURE]    存储过程名  
    { @输出参数名 数据类型 } OUTPUT  
    [...n]  
    AS  
    SQL语句 [...n]
```

在创建和执行存储过程时都使用OUTPUT关键词

- 执行语句的语法格式:

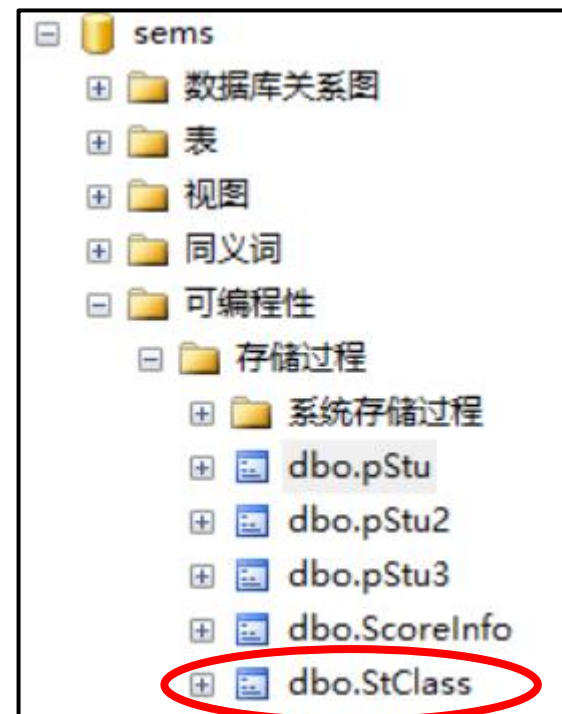
```
[ EXEC[UTE] ]    存储过程名 输出参数 OUTPUT
```



带输出参数的存储过程的创建

例5 创建带一个输入参数和一个输出参数的存储过程StClass，查询指定学号的学生，以输出参数的形式返回学生所在的班级名称。

```
CREATE PROCEDURE  StClass  
    @stid char(10), @classname char(20) OUTPUT  
AS  
    SELECT  @classname = cname  
    FROM StInfo  
    WHERE StInfo.StID = @stid
```





带输出参数的存储过程的执行

```
CREATE PROC StClass @stid char(10), @classname char(20) OUTPUT  
AS  
    SELECT @classname = cname  
    FROM StInfo  
    WHERE StID = @stid
```

执行该存储过程的语句如下：

```
DECLARE @getcname char(20)  
EXEC StClass '0603170109', @getcname OUTPUT  
SELECT @getcname
```

	(无列名)
1	材料科学1701

03 小结

存储过程与特定的数据库相关联，是存储在SQL Server服务器上的预编译好的一组完成特定功能的SQL语句集。

存储过程一旦创建成功，就可在需要时多次调用。

- 创建存储过程语句：

```
CREATE PROC[EDURE] 存储过程名  
    [{ @参数名 数据类型} [=default] [ OUTPUT ] ] [...n]  
AS  
    SQL语句体 [...n]
```

- 执行存储过程语句：

```
[ EXEC[UTE] ] 存储过程名 [实参 [,OUTPUT] ] [...n] ]
```

专题6

存储过程

6.1

T-SQL语言

6.2

存储过程的创建与执行

6.3

存储过程修改与删除

6.4

存储过程应用举例

6.3

存储过程的修改与删除

- ➡ 查看存储过程
- ➡ 修改存储过程
- ➡ 删除存储过程

01 存储过程的查看

使用对象资源管理器查看存储过程

对象资源管理器

连接(O) ▼

(local) (SQL Server 10.0.1600 - TEACHER\admin)

- 数据库
 - 系统数据库
 - 数据库快照
 - ReportServer
 - ReportServerTempDB
 - sems
 - 数据库关系图
 - 表
 - 视图
 - 同义词
 - 可编程性
 - 存储过程
 - 系统存储过程
 - dbo.pStu
 - dbo.pStu
 - dbo.pStu
 - dbo.Sco
 - dbo.StCl
 - 函数
 - 数据库触发器
 - 程序集
 - 类型
 - 规则
 - 默认值
 - 计划指南

新建存储过程(N)...

修改(Y)

执行存储过程(F)...

编写存储过程脚本为(S)

查看依赖关系(V)

策略(O)

方面(A)

启动 PowerShell(H)

报表(P)

CREATE 到(C)

ALTER 到(A)

DROP 到(D)

DROP 和 CREATE 到(R)

SELECT 到(S)

新查询编辑器窗口

文件...

剪贴板

代理作业...

```
GO

CREATE PROCEDURE [dbo].[pStu]
AS
SELECT StID, stname, StSex, Telephone
FROM StInfo
where CName='法学1603'
```

01 存储过程的查看

◆ 使用系统存储过程查看存储过程

系统存储过程	作 用
sp_helptext 存储过程名	查看存储过程的文本信息
sp_depends 存储过程名	查看存储过程的相关性
sp_help 存储过程名	查看存储过程的一般信息

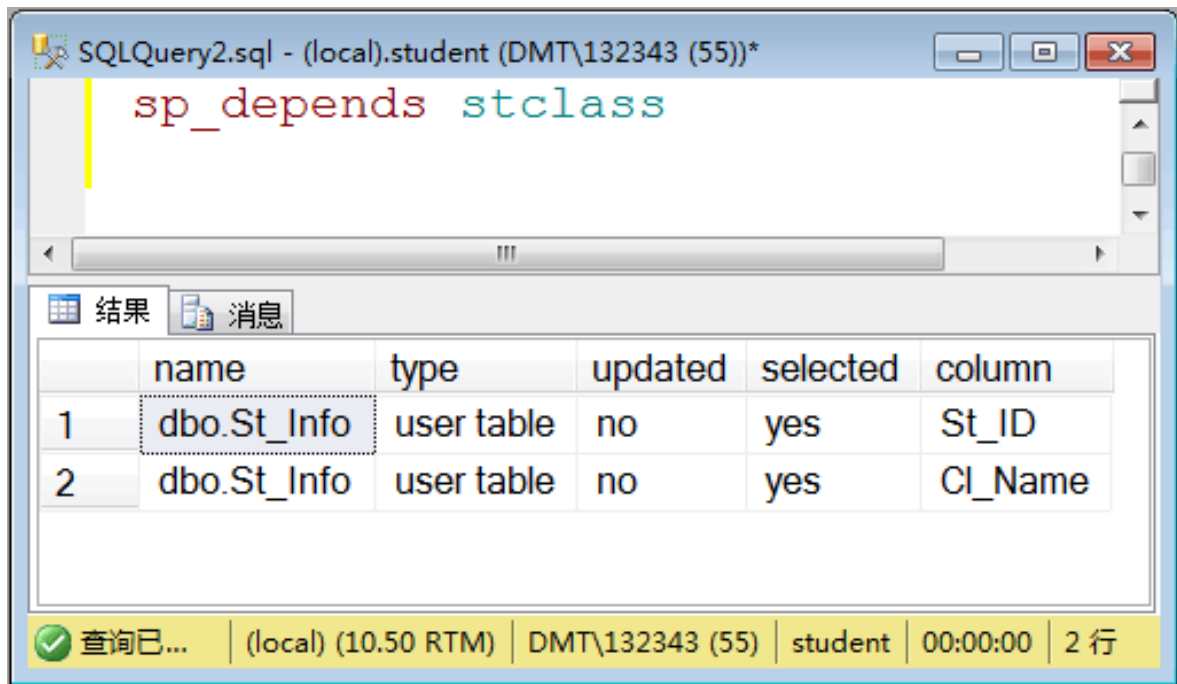
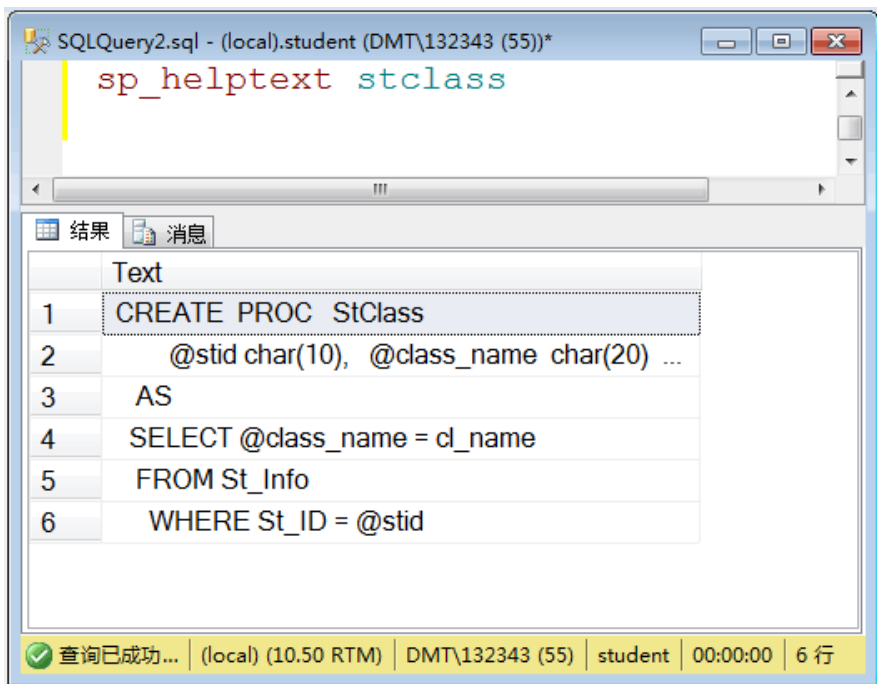
执行系统存储过程的语句格式：

[EXEC[UTE]] 系统存储过程名 需查看的存储过程名

01 存储过程的查看

使用系统存储过程查看存储过程

系统存储过程	作 用
sp_helptext 存储过程名	查看存储过程的文本信息
sp_depends 存储过程名	查看存储过程的相关性
sp_help 存储过程名	查看存储过程的一般信息



01 存储过程的查看

◆ 使用系统存储过程查看存储过程

系统存储过程	作 用
sp_helptext 存储过程名	查看存储过程的文本信息
sp_depends 存储过程名	查看存储过程的相关性
sp_help 存储过程名	查看存储过程的一般信息

SQLQuery10.sql - A...PJLY\wxling (55))*

sp_help stclass

结果 消息

	Name	Owner	Type	Created_datetime
1	StClass	dbo	stored procedure	2013-07-23 16:18:25.407

	Parameter_na...	Type	Len...	Prec	Scale	Param_or...	Collation
1	@stid	char	10	10	NULL	1	Chinese_PRC_CI_AS
2	@class_name	char	20	20	NULL	2	Chinese_PRC_CI_AS



常用的系统存储过程

系统存储过程	说明
sp_help	用于查看对象信息
sp_helpdb	用于查询数据库的信息
sp_helptext	查看规则、默认值、未加密的存储过程、用户定义函数、触发器或视图的文本
sp_helpindex	查看某个表的索引
sp_databases	用于显示所有数据库的信息，如数据库名和数据大小。
sp_renamedb	更改数据库的名称
Sp_rename	用于在当前数据库更改用户创建的对象名称，如数据表、字段、索引等
sp_tables	返回当前数据库中数据表和视图
sp_password	添加或修改登录帐户的密码

02 存储过程的修改

◆ 使用ALTER PROCEDURE语句修改存储过程

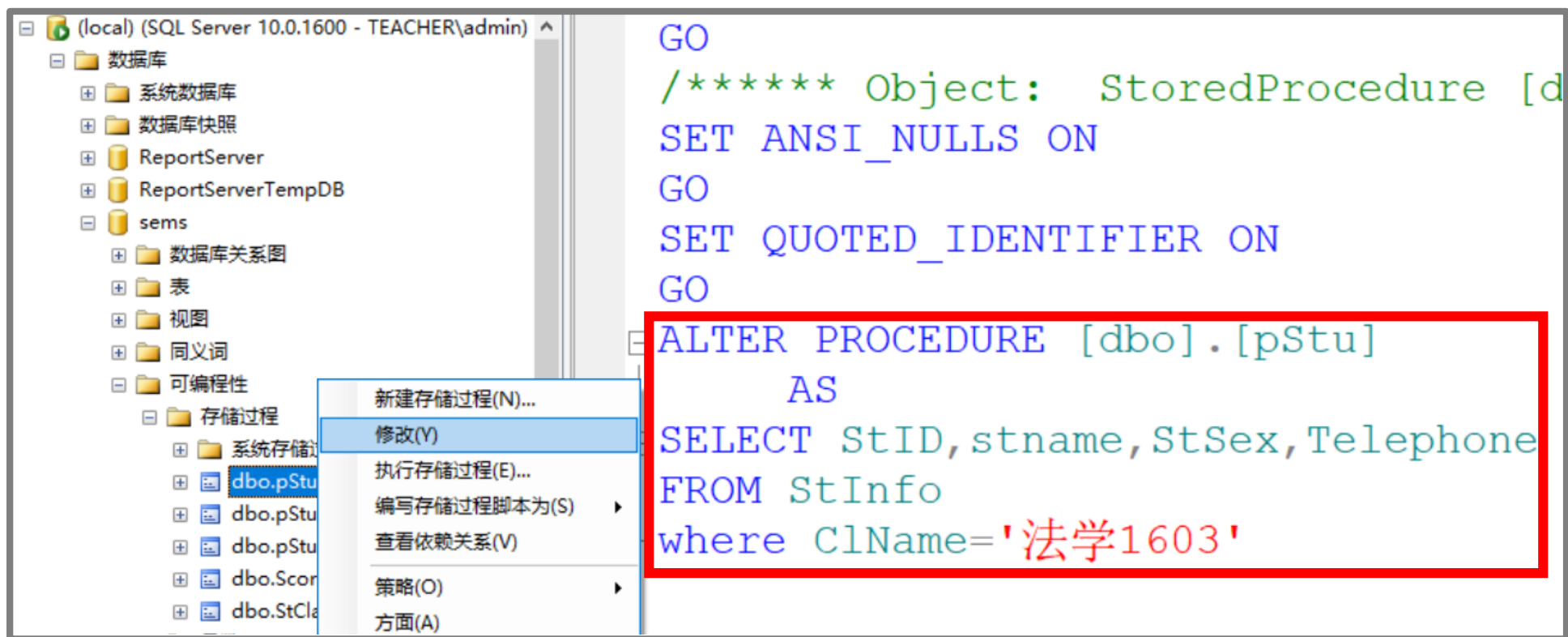
- 语法格式:

```
ALTER PROC[EDURE]      存储过程名  
[ { @参数名 数据类型} [=default ] [ OUTPUT ]  
] [...n]  
    AS  
    SQL语句 [...n]
```

按“执行”按钮，执行修改语句

02 存储过程的修改

使用对象资源管理器修改存储过程



注意：修改完以后，一定要按“执行”按钮使修改生效

02 存储过程的修改

◆ 存储过程的重命名

可使用系统存储过程 **sp_rename**，语法格式：

sp_rename ' 原存储过程名' , '新存储过程名'

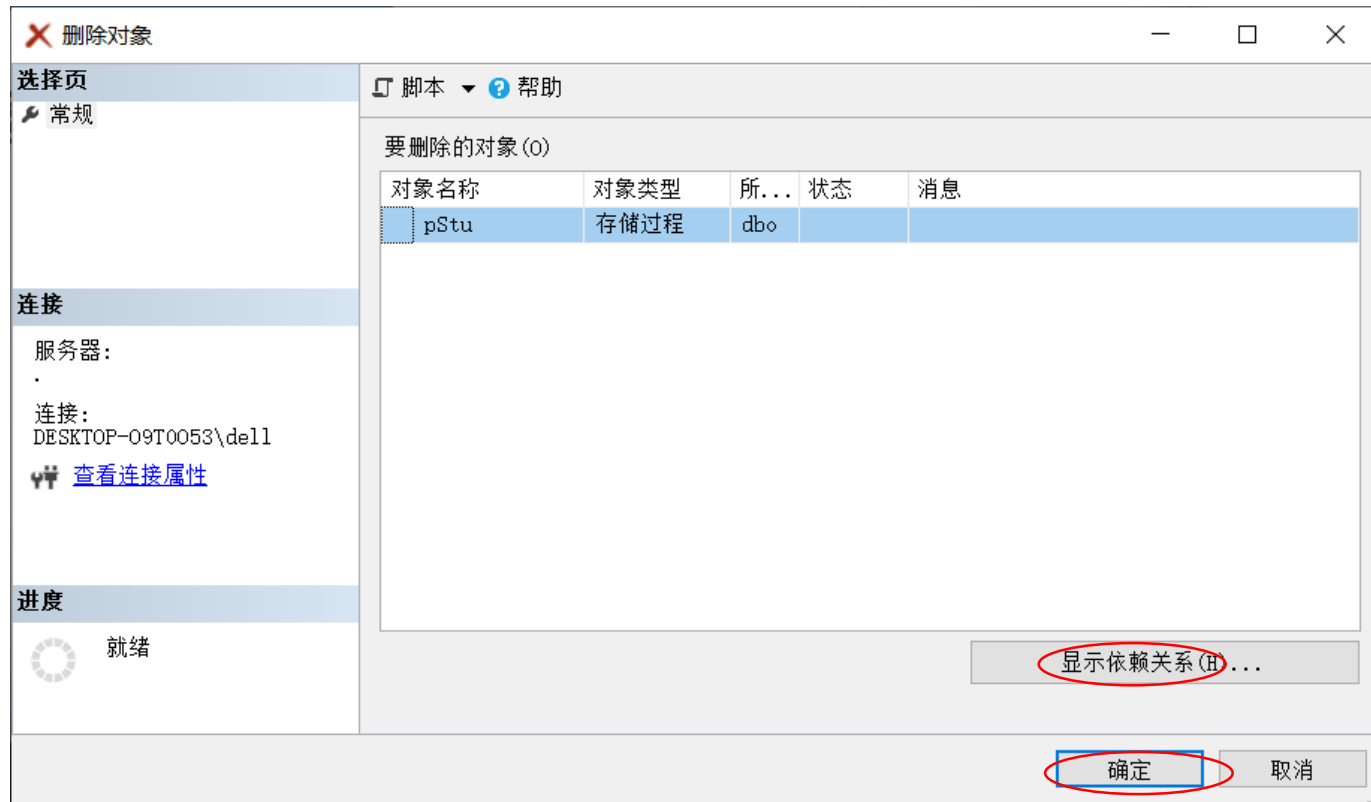
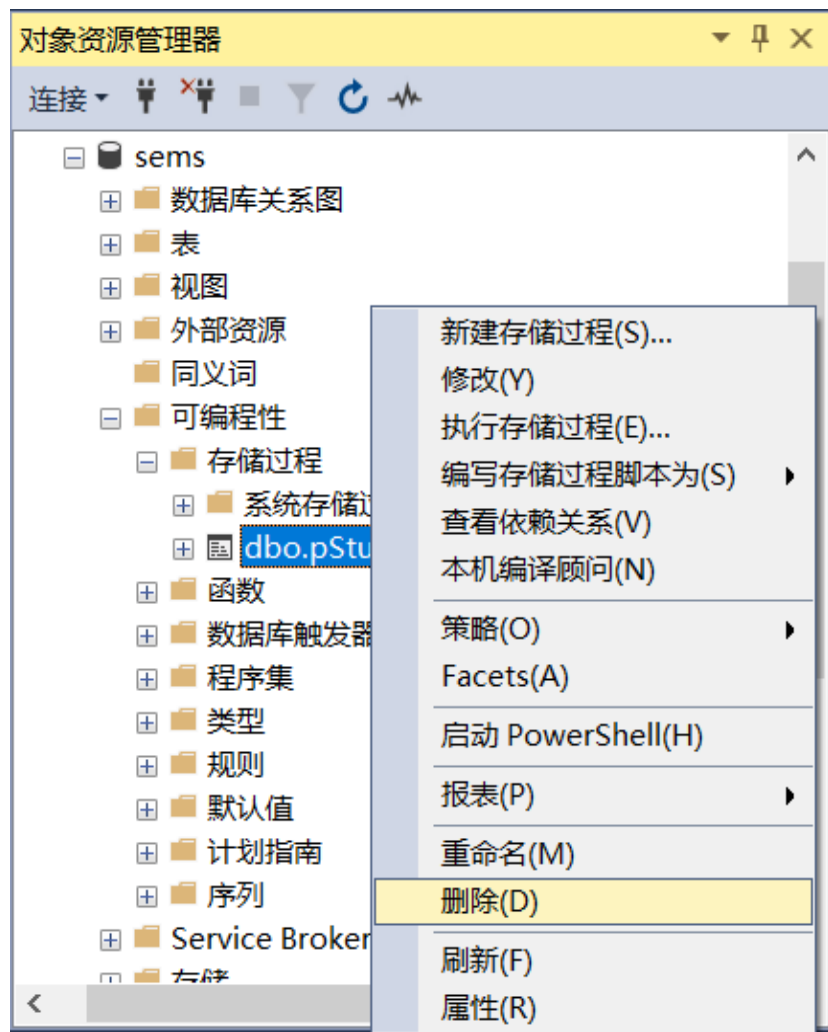
例1 将已创建的存储过程pStu更名为Studentproc。

```
sp_rename 'pStu', ' Studentproc'
```

注意：通过对象资源管理器也可以修改存储过程的名称

03 存储过程的删除

使用对象资源管理器删除存储过程



注意：在删除存储过程之前，单击“显示依赖关系”按钮，检查是否有对象依赖此存储过程，在没有其它对象依赖的情况下，删除此存储过程

03 存储过程的删除

- 使用DROP PROCEDURE语句删除存储过程

语法格式：

```
DROP PROC[EDURE] { procedure_name } [ ,...n ]
```

注意：在删除存储过程之前，执行系统存储过程sp_depends检查是否有对象依赖此存储过程

例2 删除已创建的存储过程pstu

```
DROP PROC pstu
```


03 存储过程的删除

例3 删除存储过程Average_Score

```
USE sems
```

```
GO
```

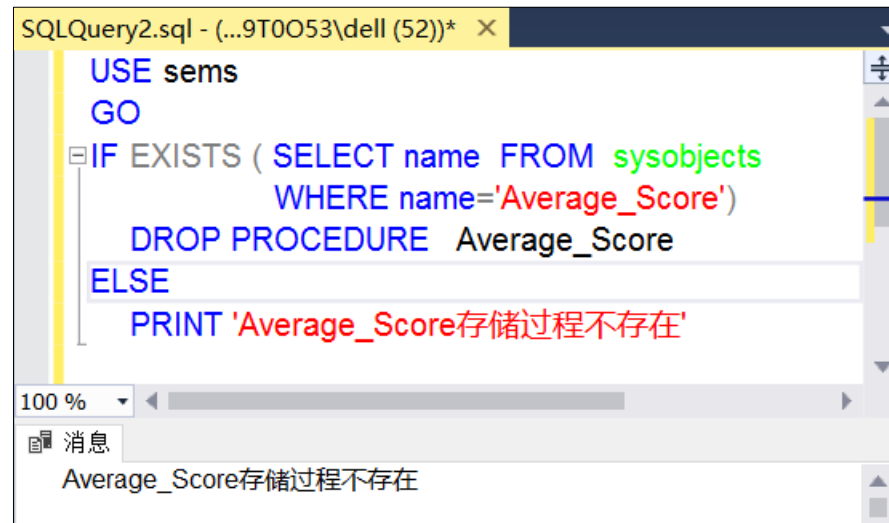
```
IF EXISTS ( SELECT name FROM sysobjects  
            WHERE name='Average_Score' )
```

```
    DROP PROCEDURE Average_Score
```

```
ELSE
```

```
    PRINT 'Average_Score存储过程不存在'
```

在系统视图下可以找到系统对象表 **sysobjects** 归属于sys架构，它保存当前数据库的对象，如约束、默认值、日志、规则、存储过程等



04 小结

- 查看存储过程
- 修改存储过程
- 删除存储过程

专题6

存储过程

6.1

T-SQL语言

6.2

存储过程的创建与执行

6.3

存储过程修改与删除

6.4

存储过程应用举例

6.4 存储过程应用举例

- ➔ 数据查询
- ➔ 数据统计
- ➔ 数据更新
- ➔ 综合应用举例

01 存储过程应用举例

例1 查询指定班级、指定性别的学生中来自指定年份的学生详细信息，班级的默认值是‘法学1701’。

分析：

- 定义输入参数@class -----所指定的班级
- 输入参数@sex ----- 所指定的性别
- 输入参数@syear-----所指定的年份

```
CREATE PROCEDURE pex1
    @class varchar(20) = '法学1701' ,@sex char(2),@syear int
AS
SELECT *
FROM StInfo
WHERE CName=@class AND stsex=@sex AND year(birthdate)=@syear
```

01 存储过程应用举例

例2 统计指定课程（课程号）的平均成绩和选课人数，并将统计结果用输出参数返回。

分析：

定义输入参数@courseno	-----	所指定的课程号
输出参数@cavg	-----	所统计的课程平均分
输出参数@num	-----	所统计的选课人数

```
CREATE PROCEDURE pex2
  @courseno char(10), @cavg int OUTPUT, @num int OUTPUT
AS
SELECT @cavg=AVG(score), @num=COUNT(cno)
FROM ScInfo
WHERE CNo= @courseno
```

01 存储过程应用举例

例3 将指定课程（课程号）的学分改为指定值，要求指定值在1~8之间，否则不与修改。

分析： 定义输入参数@courseno ----- 所指定的课程号
 输入参数@credit ----- 所指定的学分

```
CREATE PROCEDURE pex3
@courseno char(10), @credit tinyint
AS
IF @credit BETWEEN 1 AND 8
    UPDATE CInfo
    SET CCredit= @credit
    FROM cInfo
    WHERE CNo= @courseno
```

若指定值在1到8之间，则将指定课程号的课程学分改为指定值，

01 存储过程应用举例

例4 创建存储过程pstname，该存储过程需要在应用程序中调用；按学号查询学生姓名，分3种情况返回给用户信息：

如果输入空的学号，则返回 “没有输入学号” ；

如果在StInfo表中不存在指定学号的学生，则返回 “没有这个学号的学生” ；

否则（即找到了指定学号的学生），则返回学生的姓名信息。

分析：

- 1.存储过程的参数及其执行状态的处理
- 2.调用执行语句中对存储过程的返回值的处理

01 存储过程应用举例

例4创建存储过程pstname，该存储过程需要在应用程序中调用；按学号查询学生姓名，分3种情况返回给用户信息：

如果输入空的学号，则返回 “没有输入学号” ；

如果在StInfo表中不存在指定学号的学生，则返回 “没有这个学号的学生” ；

否则（即找到了指定学号的学生），则返回学生的姓名信息。

分析： 定义输入参数@stid -----所指定的学号
 定义输出参数@stname-----学生姓名

```
CREATE PROCEDURE pstname  
@stid char(10), @class_name char(20) OUTPUT  
AS
```

对于不同的查询状况，反馈给用户不同的信息

01 存储过程应用举例

例4创建存储过程pstname，该存储过程需要在应用程序中调用；按学号查询学生姓名，分3种情况返回给用户信息：

如果输入空的学号，则返回 “没有输入学号” ；

如果在StInfo表中不存在指定学号的学生，则返回 “没有这个学号的学生” ；

否则（即找到了指定学号的学生），则返回学生的姓名信息。

分析：定义输入参数@stid -----所指定的学号
定义输出参数@stname-----学生姓名

```
CREATE PROCEDURE  pstname  
@stid char(10), @class_name char(20) OUTPUT  
AS
```

对于不同的查询状况，反馈给用户不同的信息

RETURN语句

- 用于无条件地终止一个查询、存储过程或者批处理，此时位于RETURN语句之后的程序将不会被执行。
- 无条件返回，同时可以返回一个整数状态值（返回码）
- 返回码为0表示执行成功，返回其它非零整数，表示执行失败。

01 存储过程应用举例

例4创建存储过程pstname，该存储过程需要在应用程序中调用；按学号查询学生姓名，分3种情况返回给用户信息：

如果输入空的学号，则返回 “没有输入学号” ；

如果在StInfo表中不存在指定学号的学生，则返回 “没有这个学号的学生” ；

否则（即找到了指定学号的学生），则返回学生的姓名信息。

分析：

```
CREATE PROCEDURE pstname
```

```
@stid char(10)=null , @class_name char(20) OUTPUT
```

```
As
```

如果输入空的学号参数值，则返回执行状态 “-1” ；

如果在St_Info表中不存在指定学号的学生，则返回执行状态 “-2” ；

如果找到了指定学号的学生，则返回执行状态 “0” 表示执行正常。

01 存储过程应用举例

如果输入空的学号参数值，则返回执行状态“-1”；

如果在Stinfo表中不存在指定学号的学生，则返回执行状态“-2”；

如果找到了指定学号的学生，则返回执行状态“0”表示执行正常。

分析：

```
CREATE PROCEDURE pstname
@stid char(10)=null , @stname char(20) OUTPUT
AS
IF @stid is null                --没有输入学号
    RETURN -1
SELECT @stname = stname
FROM StInfo WHERE StInfo.StID = @stid
IF @stname IS NULL             --没有查询到指定学号的学生
    RETURN -2
RETURN 0                        --除去前两种情况，查询成功
```

01 存储过程应用举例

例4 创建存储过程pstname，该存储过程需要在应用程序中调用，并把相关信息反馈给调用者；按学号查询学生姓名，分3种情况返回给用户信息：

如果输入空的学号，则返回 “没有输入学号” ；

如果在St_Info表中不存在指定学号的学生，则返回 “没有这个学号的学生” ；

否则（即找到了指定学号的学生），则返回学生的姓名信息。

分析：

1.存储过程的参数及其执行状态的处理

2.调用执行语句中对存储过程的返回值的处理

01 存储过程应用举例

例4 创建存储过程pstname，该存储过程需要在应用程序中调用，并把相关信息反馈给调用者；按学号查询学生姓名，分3种情况返回给用户信息：

如果输入空的学号，则返回 “没有输入学号” ；

如果在St_Info表中不存在指定学号的学生，则返回 “没有这个学号的学生” ；

否则（即找到了指定学号的学生），则返回学生的姓名信息。

分析：

- 1.存储过程的参数及其执行状态的处理

- 2.调用执行语句中对存储过程的返回值的处理

01 存储过程应用举例

例4 创建存储过程pstname，该存储过程需要在应用程序中调用，并把相关信息反馈给调用者；按学号查询学生姓名，分3种情况返回给用户信息：

如果输入空的学号，则返回 “没有输入学号” ；

如果在St_Info表中不存在指定学号的学生，则返回 “没有这个学号的学生” ；

否则（即找到了指定学号的学生），则返回学生的姓名信息。

调用存储过程pstname，并把相关信息反馈给调用者；

如果返回执行状态 “-1” ，则输出 “没有输入学号” ；

如果返回执行状态 “-2” ，则输出 “没有该学号的学生” ；

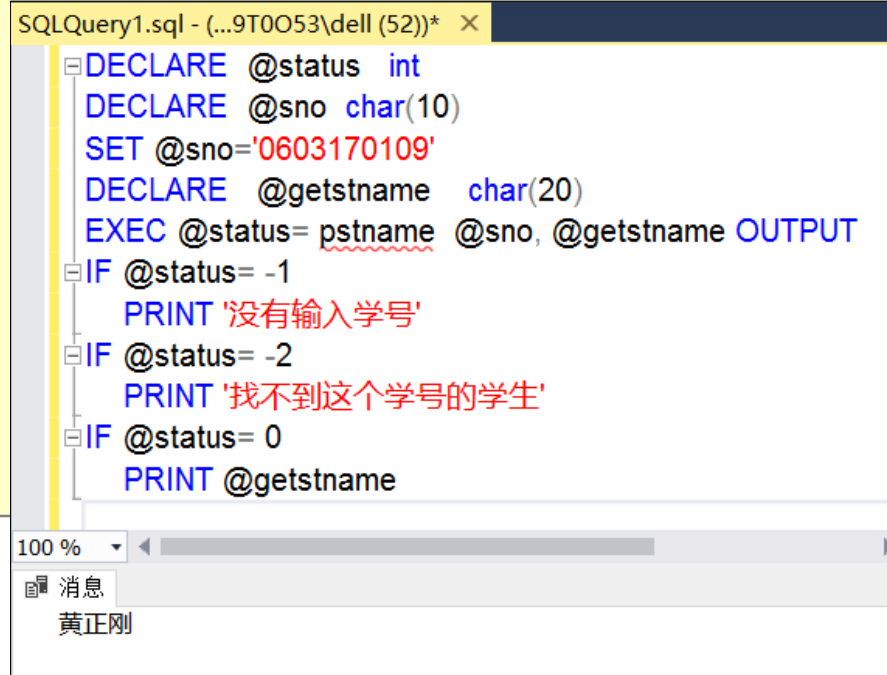
如果返回执行状态 “0” ，则输出该学生的姓名。

```
IF @status= -1  
    PRINT '没有输入学号'  
IF @status= -2  
    PRINT '找不到这个学号的学生'  
IF @status= 0  
    PRINT @getstname
```

01 存储过程应用举例

- 调用psname, 根据返回值进行相应输出

```
DECLARE @status int
DECLARE @sno char(10)
SET @sno= '0603170109'
DECLARE @getstname char(20)
EXEC @status= psname @sno, @getstname OUTPUT
IF @status= -1
    PRINT '没有输入学号'
IF @status= -2
    PRINT '找不到这个学号的学生'
IF @status= 0
    PRINT @getstname
```



The screenshot shows a SQL query editor window titled "SQLQuery1.sql - (...9T0O53\dell (52))*". The query text is as follows:


```
DECLARE @status int
DECLARE @sno char(10)
SET @sno='0603170109'
DECLARE @getstname char(20)
EXEC @status= psname @sno, @getstname OUTPUT
IF @status= -1
    PRINT '没有输入学号'
IF @status= -2
    PRINT '找不到这个学号的学生'
IF @status= 0
    PRINT @getstname
```

The editor interface includes a zoom level of 100% and a message pane at the bottom showing the message "黄正刚".

01 存储过程应用举例

- 调用pstname, 根据返回值进行相应输出

```
DECLARE @status int
DECLARE @sno char(10)
SET @sno= '0603170109'
DECLARE @getstname char(20)
EXEC @status= pstname @sno, @getstname OUTPUT
IF @status= -1
    PRINT '没有输入学号'
IF @status= -2
    PRINT '找不到这个学号的学生'
IF @status= 0
    PRINT @getstname
```



The screenshot shows a SQL query editor window titled "SQLQuery1.sql - (...9T0053\dell (52))*". The code in the editor is identical to the one in the slide. Below the code editor, there is a "消息" (Messages) pane. It shows a single message: "没有输入学号" (No student number entered), which corresponds to the output of the first IF statement in the code when @status is -1.

```
SQLQuery1.sql - (...9T0053\dell (52))* ×
DECLARE @status int
DECLARE @sno char(10)

DECLARE @getstname char(20)
EXEC @status= pstname @sno, @getstname OUTPUT
IF @status= -1
    PRINT '没有输入学号'
IF @status= -2
    PRINT '找不到这个学号的学生'
IF @status= 0
    PRINT @getstname
```

100 %

消息

没有输入学号

02 小结

- 举例
- 综合应用举例