# **[SQL Server 存储过程](http://www.cnblogs.com/hoojo/archive/2011/07/19/2110862.html)**

Transact-SQL中的存储过程，非常类似于Java语言中的方法，它可以重复调用。当存储过程执行一次后，可以将语句缓存中，这样下次执行的时候直接使用缓存中的语句。这样就可以提高存储过程的性能。

Ø 存储过程的概念

    存储过程Procedure是一组为了完成特定功能的SQL语句集合，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纵语句，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，所以存储过程运行要比单个的SQL语句块要快。同时由于在调用时只需用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、简单网络负担。

    1、 存储过程的优点

        A、 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，从而极大的提高了程序的可移植性。

        B、 存储过程能够实现较快的执行速度

        如果某一操作包含大量的T-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T-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。

        C、 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到的T-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。

        D、 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。

Ø 系统存储过程

    系统存储过程是系统创建的存储过程，目的在于能够方便的从系统表中查询信息或完成与更新数据库表相关的管理任务或其他的系统管理任务。系统存储过程主要存储在master数据库中，以“sp”下划线开头的存储过程。尽管这些系统存储过程在master数据库中，但我们在其他数据库还是可以调用系统存储过程。有一些系统存储过程会在创建新的数据库的时候被自动创建在当前数据库中。

    常用系统存储过程有：

exec sp\_databases; --查看数据库  
exec sp\_tables; --查看表  
exec sp\_columns student;--查看列  
exec sp\_helpIndex student;--查看索引  
exec sp\_helpConstraint student;--约束  
exec sp\_stored\_procedures;  
exec sp\_helptext 'sp\_stored\_procedures';--查看存储过程创建、定义语句  
exec sp\_rename student, stuInfo;--修改表、索引、列的名称  
exec sp\_renamedb myTempDB, myDB;--更改数据库名称  
exec sp\_defaultdb 'master', 'myDB';--更改登录名的默认数据库  
exec sp\_helpdb;--数据库帮助，查询数据库信息  
exec sp\_helpdb master;

    系统存储过程示例：

--表重命名  
exec sp\_rename 'stu', 'stud';  
select \* from stud;  
--列重命名  
exec sp\_rename 'stud.name', 'sName', 'column';  
exec sp\_help 'stud';  
--重命名索引  
exec sp\_rename N'student.idx\_cid', N'idx\_cidd', N'index';  
exec sp\_help 'student';  
  
--查询所有存储过程  
select \* from sys.objects where type = 'P';  
select \* from sys.objects where type\_desc like '%pro%' and name like 'sp%';

Ø 用户自定义存储过程

   1、 创建语法

create proc | procedure pro\_name  
 [{@参数数据类型} [=默认值] [output],  
 {@参数数据类型} [=默认值] [output],  
 ....  
 ]  
as  
 SQL\_statements

   2、 创建不带参数存储过程

--创建存储过程  
if (exists (select \* from sys.objects where name = 'proc\_get\_student'))  
 drop proc proc\_get\_student  
go  
create proc proc\_get\_student  
as  
 select \* from student;  
  
--调用、执行存储过程  
exec proc\_get\_student;

   3、 修改存储过程

--修改存储过程  
alter proc proc\_get\_student  
as  
select \* from student;

   4、 带参存储过程

--带参存储过程  
if (object\_id('proc\_find\_stu', 'P') is not null)  
 drop proc proc\_find\_stu  
go  
create proc proc\_find\_stu(@startId int, @endId int)  
as  
 select \* from student where id between @startId and @endId  
go  
  
exec proc\_find\_stu 2, 4;

   5、 带通配符参数存储过程

--带通配符参数存储过程  
if (object\_id('proc\_findStudentByName', 'P') is not null)  
 drop proc proc\_findStudentByName  
go  
create proc proc\_findStudentByName(@name varchar(20) = '%j%', @nextName varchar(20) = '%')  
as  
 select \* from student where name like @name and name like @nextName;  
go  
  
exec proc\_findStudentByName;  
exec proc\_findStudentByName '%o%', 't%';

   6、 带输出参数存储过程

if (object\_id('proc\_getStudentRecord', 'P') is not null)  
 drop proc proc\_getStudentRecord  
go  
create proc proc\_getStudentRecord(  
 @id int, --默认输入参数  
 @name varchar(20) out, --输出参数  
 @age varchar(20) output--输入输出参数  
)  
as  
 select @name = name, @age = age from student where id = @id and sex = @age;  
go  
  
--   
declare @id int,  
 @name varchar(20),  
 @temp varchar(20);  
set @id = 7;   
set @temp = 1;  
exec proc\_getStudentRecord @id, @name out, @temp output;  
select @name, @temp;  
print @name + '#' + @temp;

   7、 不缓存存储过程

--WITH RECOMPILE 不缓存  
if (object\_id('proc\_temp', 'P') is not null)  
 drop proc proc\_temp  
go  
create proc proc\_temp  
with recompile  
as  
 select \* from student;  
go  
  
exec proc\_temp;

   8、 加密存储过程

--加密WITH ENCRYPTION   
if (object\_id('proc\_temp\_encryption', 'P') is not null)  
 drop proc proc\_temp\_encryption  
go  
create proc proc\_temp\_encryption  
with encryption  
as  
 select \* from student;  
go  
  
exec proc\_temp\_encryption;  
exec sp\_helptext 'proc\_temp';  
exec sp\_helptext 'proc\_temp\_encryption';

   9、 带游标参数存储过程

if (object\_id('proc\_cursor', 'P') is not null)  
 drop proc proc\_cursor  
go  
create proc proc\_cursor  
 @cur cursor varying output  
as  
 set @cur = cursor forward\_only static for  
 select id, name, age from student;  
 open @cur;  
go  
--调用  
declare @exec\_cur cursor;  
declare @id int,  
 @name varchar(20),  
 @age int;  
exec proc\_cursor @cur = @exec\_cur output;--调用存储过程  
fetch next from @exec\_cur into @id, @name, @age;  
while (@@fetch\_status = 0)  
begin  
 fetch next from @exec\_cur into @id, @name, @age;  
 print 'id: ' + convert(varchar, @id) + ', name: ' + @name + ', age: ' + convert(char, @age);  
end  
close @exec\_cur;  
deallocate @exec\_cur;--删除游标

   10、 分页存储过程

---存储过程、row\_number完成分页  
if (object\_id('pro\_page', 'P') is not null)  
 drop proc proc\_cursor  
go  
create proc pro\_page  
 @startIndex int,  
 @endIndex int  
as  
 select count(\*) from product  
;   
 select \* from (  
 select row\_number() over(order by pid) as rowId, \* from product   
 ) temp  
 where temp.rowId between @startIndex and @endIndex  
go  
--drop proc pro\_page  
exec pro\_page 1, 4  
--  
--分页存储过程  
if (object\_id('pro\_page', 'P') is not null)  
 drop proc pro\_stu  
go  
create procedure pro\_stu(  
 @pageIndex int,  
 @pageSize int  
)  
as  
 declare @startRow int, @endRow int  
 set @startRow = (@pageIndex - 1) \* @pageSize +1  
 set @endRow = @startRow + @pageSize -1  
 select \* from (  
 select \*, row\_number() over (order by id asc) as number from student   
 ) t  
 where t.number between @startRow and @endRow;  
  
exec pro\_stu 2, 2;

Ø Raiserror

Raiserror返回用户定义的错误信息，可以指定严重级别，设置系统变量记录所发生的错误。

   语法如下：

Raiserror({msg\_id | msg\_str | @local\_variable}  
 {, severity, state}  
 [,argument[,…n]]  
 [with option[,…n]]  
)

   # msg\_id:在sysmessages系统表中指定的用户定义错误信息

   # msg\_str:用户定义的信息，信息最大长度在2047个字符。

   # severity：用户定义与该消息关联的严重级别。当使用msg\_id引发使用sp\_addmessage创建的用户定义消息时，raiserror上指定严重性将覆盖sp\_addmessage中定义的严重性。

    任何用户可以指定0-18直接的严重级别。只有sysadmin固定服务器角色常用或具有alter trace权限的用户才能指定19-25直接的严重级别。19-25之间的安全级别需要使用with log选项。

   # state：介于1至127直接的任何整数。State默认值是1。

raiserror('is error', 16, 1);  
select \* from sys.messages;  
--使用sysmessages中定义的消息  
raiserror(33003, 16, 1);  
raiserror(33006, 16, 1);

http://www.cnblogs.com/hoojo/archive/2011/07/19/2110862.html