DBCC

Transact-SQL编程语言提供DBCC语句作为Microsoft&reg;SQLServer2000的数据库控制台命令。这些语句对数据库的物理和逻辑一致性进行检查。许多DBCC语句能够对检测到的问题进行修复。

数据库控制台命令语句被分为以下类别。

语句分类 执行

维护语句 对数据库、索引或文件组进行维护的任务。

杂项语句 诸如启用行级锁定或从内存中删除动态链接库(DLL)等杂项任务。

状态语句 状态检查。

验证语句 对数据库、表、索引、目录、文件组、系统表或数据库页的分配进行的验证操作。

SQLServer2000的DBCC语句使用输入参数和返回值。所有DBCC语句参数都可以接受Unicode和DBCS字面值。

使用DBCC结果集输出

许多DBCC命令可以产生表格格式的输出（使用WITHTABLERESULTS选项）。

该信息可装载到表中以便将来使用。以下显示一个示例脚本：

-- Create the table to accept the results

CREATE TABLE #tracestatus (   TraceFlag INT,   Status INT   )

  -- Execute the command, putting the results in the table

INSERT INTO #tracestatus

  EXEC ('DBCC TRACESTATUS (-1) WITH NO\_INFOMSGS')

-- Display the results

 SELECT \*   FROM #tracestatus

GO

维护语句

DBCC DBREINDEX

DBCC DBREPAIR

DBCC INDEXDEFRAG

DBCC SHRINKDATABASE

DBCC SHRINKFILE

DBCC UPDATEUSAGE

杂项语句

  DBCC dllname (FREE)

DBCC HELP

DBCC PINTABLE

DBCC ROWLOCK

DBCC TRACEOFF

DBCC TRACEON

DBCC UNPINTABLE

状态语句

DBCC PINTABLE

DBCC OPENTRAN

DBCC OUTPUTBUFFER

DBCC PROCCACHE

DBCC SHOWCONTIG

DBCC SHOW\_STATISTICS

DBCC SQLPERF

DBCC TRACESTATUS

DBCC USEROPTIONS

验证语句

DBCC CHECKALLOC

DBCC CHECKCATALOG

DBCC CHECKCONSTRAINTS

DBCC CHECKDB

DBCC CHECKFILEGROUP

DBCC CHECKIDENT

DBCC CHECKTABLE

DBCC NEWALLOC

 DUMP TRANSACTION IQC WITH NO\_LOG

DBCC SHRINKDATABASE(IQC)

Transact-SQL 参考

DBCC SHRINKDATABASE

收缩指定数据库中的数据文件大小

语法

DBCC SHRINKDATABASE

( database\_name [ , target\_percent ]

 [ , { NOTRUNCATE | TRUNCATEONLY } ] )

参数

database\_name

是要收缩的数据库名称。数据库名称必须符合标识符的规则。有关更多信息，请参见使用标识符。

target\_percent

是数据库收缩后的数据库文件中所要的剩余可用空间百分比。

NOTRUNCATE

导致在数据库文件中保留所释放的文件空间。如果未指定，将所释放的文件空间释放给操作系统。

TRUNCATEONLY

导致将数据文件中的任何未使用的空间释放给操作系统，并将文件收缩到上一次所分配的大小，从而减少文件大小，而不移动任何数据。不试图重新定位未分配页的行。使用 TRUNCATEONLY 时，忽略 target\_percentis。 注释

Microsoft SQL Server可收缩：

特定数据库的所有数据和日志文件。执行 DBCC SHRINKDATABASE。 一次一个特定数据库中的数据或日志文件。执行 DBCC SHRINKFILE。  DBCC SHRINKDATABASE 以每个文件为单位对数据文件进行收缩。然而，DBCC SHRINKDATABASE 在对日志文件进行收缩时，看起来好像所有的日志文件都存在于一个连续的日志池中。

假设名为 mydb 的数据库有两个数据文件和两个日志文件。这些数据文件和日志文件大小都为 10 MB。第一个数据文件包含 6 MB 数据。

对于每个文件，SQL Server 计算目标大小，即要收缩文件到的大小。当用 target\_percent 指定 DBCC SHRINKDATABASE 时，SQL Server 计算的目标大小是收缩后文件中的 target\_percent 可用空间大小。例如，如果指定按 target\_percent 为 25 收缩 mydb。SQL Server 计算的该文件的目标大小为 8 MB（6 MB 数据加上 2 MB 可用空间）。因此，SQL Server 将任何数据从数据文件的最后 2 MB 中移动到数据文件前 8 MB 的可用空间中，然后对该文件进行收缩。

假设 mydb 的第一个数据文件包含 7 MB 的数据。若将 target\_percent 指定为 30，则允许将该数据文件收缩到所要的可用百分比 30。然而，若将 target\_percent 指定为 40，就不会收缩该数据文件，因为 SQL Server 不能将文件收缩到比数据当前占有的空间还小的程度。也可以从另一个角度考虑这个问题：40% 要求的可用空间 + 70% 完整的数据文件（10 MB 中的 7 MB）大于 100%。因为所要求的可用百分比加上数据文件占用的当前百分比大于 100%（多出 10%），所以任何大于 30 的 target\_size 都不会收缩该数据文件。

对于日志文件，SQL Server 使用 target\_percent 以对整个日志的目标大小进行计算；因此，target\_percent 是收缩操作后日志中的可用空间大小。之后，整个日志的目标大小可以解释为每个日志文件的目标大小。DBCC SHRINKDATABASE 尝试立即将每个物理日志文件收缩到其目标大小。如果虚拟日志中的所有逻辑日志部分都没有超出日志文件的目标大小，则该文件将成功截断，DBCC

SHRINKDATABASE 完成且不显示任何消息。然而，如果虚拟日志中的逻辑日志部分超出其目标大小，则 SQL Server 释放尽可能多的空间并发出一条消息。该消息告诉您需要执行什么操作来移动文件末尾超出虚拟日志的逻辑日志部分。执行完该操作后，可以重新发出 DBCC SHRINKDATABASE 命令以释放剩余的空间。有关收缩事务日志的更多信息，请参见收缩事务日志。

因为日志文件只能收缩到虚拟日志文件边界，所以不可能将日志文件收缩到比虚拟日志文件更小（即使现在没有使用该文件）。例如，可以将数据库的 1 GB 日志文件收缩到只有 128 MB。有关截断的更多信息，请参见截断事务日志。有关确定虚拟日志文件大小的更多信息，请参见虚拟日志文件。

由 DBCC SHRINKDATABASE 计算的数据和日志文件的目标大小始终不小于文件的最小大小。文件的最小大小是在文件最初创建时指定的大小，或是上一次用更改文件大小的操作（如用带有 MODIFY FILE 选项的 ALTER DATABASE 或 DBCC SHRINKFILE）显式设置的大小。例如，如果在执行 CREATE DATABASE 时，将 mydb 的所有数据和日志文件大小都指定为 10 MB，则每个文件的最小大小都为 10 MB。DBCC SHRINKDATABASE 不能收缩任何小于 10 MB 的文件。如果使用带有 MODIFY FILE 选项的 ALTER DATABASE 将其中的一个文件显示增长到 20 MB，则该文件新的最小大小为 20 MB。若要将文件大小收缩到小于其最小大小，请使用 DBCC SHRINKFILE 并指定新大小。执行 DBCC SHRINKFILE 会将文件的最小大小更改为所指定的新大小。

在使用数据文件时，DBCC SHRINKDATABASE 有 NOTRUNCATE 和 TRUNCATEONLY 选项。如果为日志文件指定这两个选项，则将忽略这两个选项。DBCC SHRINKDATABASE 不带任何选项等价于 DBCC SHRINKDATABASE 带有 NOTRUNCATE 选项，后面还要跟着带有 TRUNCATEONLY 选项的 DBCC SHRINKDATABASE。

不论是否指定 target\_percent，NOTRUNCATE 选项都执行 DBCC SHRINKDATABASE 的实际数据移动操作（包括将分配页从文件的末尾移动到文件前面的未分配页）。然而，文件末尾的可用空间不会返回给操作系统，文件的物理大小也不会更改。因此，如果指定 NOTRUNCATE 选项，数据文件好像没有收缩。例如，假设再次使用 mydb 数据库，该 mydb 有两个数据文件和两个日志文件。第二个数据文件和第二个日志文件的大小都为 10 MB。执行 DBCC SHRINKDATABASE mydb NOTRUNCATE 时，Microsoft SQL Server 将数据从数据文件的后面几页移动到前面几页中。而文件的大小仍然为 10 MB。

TRUNCATEONLY 选项将文件末尾的所有可用空间收回给操作系统。然而，TRUNCATEONLY 在文件内不执行任何页移动操作。指定的文件只能收缩到上一次分配的大小。如果使用 TRUNCATEONLY 选项指定 target\_percent，则忽略后者。

所生成的数据库不能比 model 数据库更小。

要收缩的数据库不必在单用户模式下；其他的用户仍可以在数据库收缩时对其进行工作。这也包括系统数据库。 结果集 下表描述结果集内的列。 列名 描述  DbId SQL Server 试图收缩的文件的数据库标识号。  FileId SQL Server 试图收缩的文件的文件标识号。  CurrentSize 文件当前占用的 8KB 页数。

MinimumSize 文件可以占用的最小 8KB 页数。这与文件的最小大小或最初创建时的大小相对应。

UsedPages 文件当前使用的 8KB 页数。

EstimatedPages SQL Server 估计文件能够收缩到的 8KB 页数。  说明  SQL Server 并不显示未收缩的文件的行。 权限 DBCC SHRINKDATABASE 权限默认授予 sysadmin 固定服务器角色或 db\_owner 固定数据库角色的成员且不可转让。 示例 下例将 UserDB 用户数据库中的文件减小，以使 UserDB 中的文件有 10% 的可用空间。

DBCC SHRINKDATABASE (UserDB, 10)GO 请参见 ALTER DATABASE DBCC   SQL Server 2005：索引碎片整理脚本。

首先说明：该 SQL 索引碎片整理脚本，是从 SQL Server 2005 联机帮助上摘录下来，并且稍加整理而成的。

该 SQL 索引碎片整理脚本，首先从 SQL Server 2005 系统管理视图 sys.dm\_db\_index\_physical\_stats 中，找出索引碎片程度大于 10% 的索引，然后根据索引碎片程度，分别来采取不同的方法来整理索引碎片。小于 30% 的使用 alter index reorganize；大于等于 30% 的使用 alter index rebuild。其中 reorganize 相当于 dbcc indexdefrag()；rebuild 相当于 dbcc dbreindex()。

SQL 碎片整理后，索引数据页在数据库文件中排列的更紧凑，可以大幅提高一些 SQL 查询的效率。DBA 可以每周进行一次碎片整理。另外要注意的是，不要在收缩数据库（dbcc shrinkfile, dbcc shrinkdatabase）前整理索引碎片。

Using sys.dm\_db\_index\_physical\_stats in a script to rebuild or reorganize indexes  The following example automatically reorganizes or rebuilds all partitions in a database that have an average fragmentation over 10 percent. Executing this query requires the [view database state] permission.

----------------------------------------------------

-- ensure a USE statement has been executed first.

----------------------------------------------------

set nocount on

declare @objectid int,@indexid int,

@partitioncount bigint,@schemaname sysname,

@objectname sysname,@indexname sysname,

@partitionnum bigint,@partitions bigint,

@frag float,@command varchar(1000)

select objectid = object\_id,indexid = index\_id,

partitionnum = partition\_number,frag = avg\_fragmentation\_in\_percent

into #work\_to\_do

from sys.dm\_db\_index\_physical\_stats(db\_id(),null,null,null,'LIMITED')

where avg\_fragmentation\_in\_percent > 10.0 and index\_id > 0

-- declare the cursor for the list of partitions to be processed.

declare partitions cursor for select \* from #work\_to\_do

-- Open the cursor.

open partitions

-- Loop through the partitions.

fetch next from partitions into @objectid,@indexid,@partitionnum,@frag

while @@fetch\_status = 0

begin

select @objectname = o.name,@schemaname = s.name from sys.objects

as o inner join sys.schemas as s on s.schema\_id = o.schema\_id

where o.object\_id = @objectid

select @indexname = name from sys.indexes

where object\_id = @objectid and index\_id = @indexid

select @partitioncount = count (\*) from sys.partitions

where object\_id = @objectid and index\_id = @indexid

-- 30 is an arbitrary decision point at which to switch between reorganizing and rebuilding

if @frag < 30.0

begin

select @command = 'alter index '+@indexname+' on '+

@schemaname+'.'+@objectname+' reorganize'

if @partitioncount > 1

select @command = @command+' partition='+

convert(char,@partitionnum)

end

if @frag >= 30.0

begin

select @command = 'alter index '+@indexname+' on '+

@schemaname+'.'+@objectname+' rebuild'

if @partitioncount > 1

select @command = @command+' partition='+

convert(char, @partitionnum)

ENDd

-- exec (@command)

print 'Executed: ' + @command

fetch next from partitions into @objectid,@indexid,

@partitionnum,@frag

end

-- free resource

close partitions

deallocate partitions

drop table #work\_to\_do

DBCC是SQL Server提供的一组控制台命令，功能很强大，掌握一些必要的语句，对操作数据库有不少帮助，所以决定整理一下，发现已有不少类似的整理，减少了不少工作，归类如下：

一、DBCC 帮助类命令

\* DBCC HELP('?')

查询所有的DBCC命令

\* DBCC HELP('checktable')

查询指定的DBCC命令的语法说明

\* DBCC USEROPTIONS

返回当前连接的活动(设置)的SET选项

二、DBCC 检查验证类命令

\* DBCC CHECKALLOG ('数据库名称')

检查指定数据库的磁盘空间分配结构的一致性

\* DBCC CHECKCATALOG ('数据库名称')

检查指定数据库的系统表内和系统表间的一致性

\* DBCC CHECKCONSTAINTS ('tablename')

检查指定表上的指定约束或所有约束的完整性

\* DBCC CHECKDB

检查数据库中的所有对象的分配和结构完整性

\* DBCC CHECKFILEGROUP

检查指定文件组中所有表在当前数据库中的分配和结构完整性

\* DBCC CHECKTABLE

检查指定表或索引视图的数据、索引及test、ntest和image页的完整性

\* DBCC CHECKIDENT

检查指定的当前标识值

\* DBCC SQLPERF(UMSSTATS) undocumented in BOL

可以用来检查是否CPU使用达到瓶颈

最关键的一个参考数据num runnable,表明当前有多少个线程再等待运行

如果大于等于2,考虑CPU达到瓶颈

三、DBCC 维护类命令

\* DBCC CLEANTABLE ('db\_name','table\_name')

回收Alter table drop column语句删除可变长度列或text

\* DBCC DBREINDEX

重建指定数据库的一个或多个索引

\* DBCC INDEXDEFRAG

对表或视图上的索引和非聚集索引进行碎片整理

\* DBCC PINTABLE (db\_id,object\_id)

将表数据驻留在内存中

查看哪些表驻留在内存的方法是：

select objectproperty(object\_id('tablename'）,'tableispinned')

\* DBCC UNPINTABLE (db\_id,object\_id)

撤消驻留在内存中的表

\* DBCC SHRINKDATABASE(db\_id,int)

收缩指定数据库的数据文件和日志文件大小

\* DBCC SHRINKFILE(file\_name,int)

收缩相关数据库的指定数据文件和日志文件大小

四、DBCC 性能调节命令

\* DBCC dllname(FREE)

sp\_helpextendedproc 查看加载的扩展PROC

在内存中卸载指定的扩展过程动态链接库（dll)

\* DBCC DROPCLEANBUFFERS

从缓冲池中删除所有缓冲区

\* DBCC FREEPROCCACHE

从过程缓冲区删除所有元素

\* DBCC INPUTBUFFER

显示从客户机发送到服务器的最后一个语句

\* DBCC OPENTRAN (db\_name)

查询某个数据库执行时间最久的事务，由哪个程序拥有

\* DBCC SHOW\_STATISTICS

显示指定表上的指定目标的当前分布统计信息

\* DBCC SHOWCONTIG

显示指定表的数据和索引的碎片信息

\* DBCC SQLPERF

(logspace) 查看各个DB的日志情况

(iostats) 查看IO情况

(threads) 查看线程消耗情况

返回多种有用的统计信息

\* DBCC CACHESTATS

显示SQL Server 2000内存的统计信息

\* DBCC CURSORSTATS

显示SQL Server 2000游标的统计信息

\* DBCC MEMORYSTATS

显示SQL Server 2000内存是如何细分的

\* DBCC SQLMGRSTATS

显示缓冲中先读和预读准备的SQL语句

五、DBCC 未公开的命令

\* DBCC ERRLOG

初始化SQL Server 2000的错误日志文件

\* DBCC FLUSHPROCINDB (db\_id)

清除SQL Server 2000服务器内存中的某个数据库的存储过程缓存内容

\* DBCC BUFFER (db\_name,object\_name,int(缓冲区个数))

显示缓冲区的头部信息和页面信息

\* DBCC DBINFO (db\_name)

显示数据库的结构信息

\* DBCC DBTABLE

显示管理数据的表(数据字典)信息

\* DBCC IND (db\_name,table\_name,index\_id)

查看某个索引使用的页面信息

\* DBCC REBUILDLOG

重建SQL Server 2000事务日志文件

\* DBCC LOG (db\_name,3) (-1--4)

查看某个数据库使用的事物日志信息

\* DBCC PAGE

查看某个数据库数据页面信息

\* DBCC PROCBUF

显示过程缓冲池中的缓冲区头和存储过程头

\* DBCC PRTIPAGE

查看某个索引页面的每行指向的页面号

\* DBCC PSS (user,spid,1)

显示当前连接到SQL Server 2000服务器的进程信息

\* DBCC RESOURCE

显示服务器当前使用的资源情况

\* DBCC TAB (db\_id,object\_id)

显示数据页面的结构

六、DBCC跟踪标记

跟踪标记用于临时设置服务器的特定特征或关闭特定行为，常用于诊断性能问题或调试存储过程或复杂的计算机系统

\* DBCC TRACEON (3604)

打开跟踪标记

\* DBCC TRACEOFF

关闭跟踪标记

\* DBCC TRACESTATS

查看跟踪标记状态

七、使用 DBCC 结果集输出

　　许多 DBCC 命令可以产生表格格式的输出（使用 WITH TABLERESULTS 选项）。该信息可装载到表中以便将来使用。以下显示一个示例脚本：

　　CREATE TABLE DBCCResult (

　　DBCCFlag INT,

　　Result INT

　　)

　　INSERT INTO DBCCResult

　　EXEC ('DBCC TRACESTATUS (-1) WITH NO\_INFOMSGS')

　　SELECT \*

　　FROM DBCCResult

八、官方使用DBCC的建议

1、在系统使用率较低时运行 CHECKDB。

2、请确保未同时执行其它磁盘 I/O 操作，例如磁盘备份。

3、将 tempdb 放到单独的磁盘系统或快速磁盘子系统中。

4、允许 tempdb 在驱动器上有足够的扩展空间。 使用带有 ESTIMATE ONLY 的 DBCC

估计 tempdb 将需要多少空间。

5、避免运行占用大量 CPU 的查询或批处理作业。

6、在 DBCC 命令运行时，减少活动事务。

7、使用 NO\_INFOMSGS 选项显著减少处理和 tempdb 的使用。

8、考虑使用带有 PHYSICAL\_ONLY 选项的 DBCC CHECKDB 来检查页和记录首部的物理结构。当硬件导致的错误被置疑时，这个操作将执行快速检查。在发布,订阅复制时要用服务器实名时可以这样:

select \* from sysservers (可以找到原来服务器的名称)

exec sp\_dropserver 'jmsql9' (删除原来的服务器名)

exec sp\_addserver 'jmSQL9' ,LOCAL (改为新的服务器名)

ALTER DATABASE [jm] SET SINGLE\_USER (改为单用户模式)

DBCC CHECKDB("databasename",REPAIR\_REBUILD) WITH TABLOCK (修复数据库)

DBCC CHECKTABLE("tablename",repair\_rebuild) with tablock (修复表)

DBCC DBREINDEX ('t\_icitem' , ' ') 修复此表所有的索引。

ALTER DATABASE [jm] SET MULTI\_USER (改为多用户模式)

REPAIR\_ALLOW\_DATA\_LOSS：执行由REPAIR\_REBUILD 完成的所有修复，包括对行和页进行分配和取消分配以改正分配错误、结构行或页的错误，以及删除已损坏的文本对象。这些修复可能会导致一些数据丢失。修复操作可以在用户事务下完成以允许用户回滚所做的更改。如果回滚修复，则数据库仍会含有错误，应该从备份进行恢复。如果由于所提供修复等级的缘故遗漏某个错误的修复，则将遗漏任何取决于该修复的修复。修复完成后，备份数据库。

REPAIR\_FAST 进行小的、不耗时的修复操作，如修复非聚集索引中的附加键。这些修复可以很快完成，并且不会有丢失数据的危险。

REPAIR\_REBUILD 执行由REPAIR\_FAST 完成的所有修复，包括需要较长时间的修复（如重建索引），执行这些修复时不会有丢失数据的危险。

dbcc shrinkdatabase (jm) 压缩数据库

SQLSERVER DBCC命令大全

DBCC DROPCLEANBUFFERS:从缓冲池中删除所有缓存，清除缓冲区

在进行测试时，使用这个命令可以从SQLSERVER的数据缓存data cache(buffer)清除所有的测试数据，以保证测试的公正性。

需要注意的是这个命令只移走干净的缓存，不移走脏缓存。由于这个原因，在执行这个命令前，应该先执行CheckPoint，将所有脏的缓存写入磁盘，

这样在运行DBCC RROPCLEANBUFFERS 时，可以保证所有的数据缓存被清理，而不是其中的一部分。

DBCC CacheStats:显示存在于当前buffer Cache中的对象的信息，例如：hit rates，编译的对象和执行计划

DBCC ErrorLog :如果很少重启mssqlserver服务，那么服务器的日志(不是数据库事务日志)会增长得很快，而且打开和查看日志的速度也会很慢

使用这个命令，可以截断当前的服务器日志，主要是生成一个新的日志。可以考虑设置一个调度任务，每周执行这个命令自动截断服务器日志。

使用存储过程sp\_cycle\_errorlog也可以达到同样的目的

一、DBCC 帮助类命令

DBCC HELP('?') ：查询所有的DBCC命令

DBCC HELP('命令') ：查询指定的DBCC命令的语法说明

DBCC USEROPTIONS ：返回当前连接的活动(设置)的SET选项

二、DBCC 检查验证类命令

DBCC CHECKALLOC('数据库名称') ：检查指定数据库的磁盘空间分配结构的一致性

DBCC CHECKCATALOG ('数据库名称') ：检查指定数据库的系统表内和系统表间的一致性

DBCC CHECKCONSTRAINTS ('tablename') ：检查指定表上的指定约束或所有约束的完整性

DBCC CHECKDB ：检查数据库中的所有对象的分配和结构完整性

DBCC CHECKFILEGROUP ：检查指定文件组中所有表在当前数据库中的分配和结构完整性

DBCC CHECKTABLE ：检查指定表或索引视图的数据、索引及test、ntest和image页的完整性

DBCC CHECKIDENT ：如果存在大量数据删除，考虑在删除后，使用 dbcc checkident 重置一下自增值

http://social.msdn.microsoft.com/Forums/zh-CN/sqlserverzhchs/thread/8fa3e3a8-2ff2-4a68-be3e-92e76c380ef9/

检查指定的当前标识值

DBCC SQLPERF(UMSSTATS)：最关键的一个参考数据num runnable：表明当前有多少个线程再等待运行，如果大于等于2,考虑CPU达到瓶颈

Scheduler ID:当前机器有多少个逻辑CPU就有多少个Scheduler ID，具体怎麽看可以看一下我的这篇文章

SQLSERVER独特的任务调度算法"SQLOS"

三、DBCC 维护类命令

DBCC CLEANTABLE ('db\_name','table\_name') ：回收Alter table drop column语句删除可变长度列或text

DBCC DBREINDEX ：重建指定数据库的一个或多个索引 跟ALTER INDEX REBUILD差不多

DBCC INDEXDEFRAG：对表或视图上的索引和非聚集索引进行碎片整理

DBCC PINTABLE (db\_id,object\_id) ：将表数据驻留在内存中

查看哪些表驻留在内存的方法是：0：没有驻留 ，1：驻留

1 USE [GPOSDB]

2 GO

3 SELECT OBJECTPROPERTY(OBJECT\_ID('dbo.SystemPara'), 'tableispinned')

DBCC UNPINTABLE (db\_id,object\_id) ：撤消驻留在内存中的表

DBCC SHRINKDATABASE(db\_id,int) ：收缩指定数据库的数据文件和日志文件大小

DBCC SHRINKFILE(file\_name,int)：收缩相关数据库的指定数据文件和日志文件大小

四、DBCC 性能调节命令

DBCC dllname(FREE) ：在内存中卸载指定的扩展过程动态链接库（dll)

sp\_helpextendedproc 查看加载的扩展PROC

DBCC DROPCLEANBUFFERS ：从缓冲池中删除所有缓冲区

DBCC FREEPROCCACHE ：从执行计划缓冲区删除所有缓存的执行计划

DBCC INPUTBUFFER ：显示从客户机发送到服务器的最后一个语句

DBCC OPENTRAN (db\_name) ：查询某个数据库执行时间最久的事务，由哪个程序拥有

DBCC SHOW\_STATISTICS ：显示指定表上的指定目标的当前统计信息分布

DBCC SHOWCONTIG ：显示指定表的数据和索引的碎片信息

DBCC SQLPERF (logspace) ：查看各个DB的日志情况

(iostats) 查看IO情况

(threads) 查看线程消耗情况

返回多种有用的统计信息

DBCC CACHESTATS ：显示SQL Server 2000内存的统计信息

DBCC CURSORSTATS ：显示SQL Server 2000游标的统计信息

DBCC MEMORYSTATS ：显示SQL Server 2000内存是如何细分的

DBCC SQLMGRSTATS ：显示缓冲区中先读和预读准备的SQL语句

五、DBCC 未公开的命令

DBCC ERRLOG ：初始化SQL Server 2000的错误日志文件

DBCC FLUSHPROCINDB (db\_id) ：清除SQL Server 2005服务器内存中的某个数据库的存储过程缓存内容

DBCC BUFFER (db\_name,object\_name,int(缓冲区个数)) ：显示缓冲区的头部信息和页面信息

DBCC DBINFO (db\_name) ：显示数据库的结构信息

DBCC DBTABLE ：显示管理数据的表(数据字典)信息

DBCC IND (db\_name,table\_name,index\_id) ：查看某个索引使用的页面信息

DBCC REBUILDLOG ：重建SQL Server 2000事务日志文件

DBCC LOG (db\_name,3) (-1~4) ：查看某个数据库的事物日志信息 显示格式可以为：-1，0，1，2，3，4 每个数字代表不同的格式

DBCC PAGE ：查看某个数据库数据页面信息

DBCC PROCBUF ：显示过程缓冲池中的缓冲区头和存储过程头

DBCC PRTIPAGE ：查看某个索引页面的每行指向的页面号

DBCC PSS (user,spid,1) ：显示当前连接到SQL Server 2000服务器的进程信息

DBCC RESOURCE ：显示服务器当前使用的资源情况

DBCC TAB (db\_id,object\_id) ：显示数据页面的结构

六、DBCC跟踪标记

跟踪标记用于临时设置服务器的特定特征或关闭特定行为，常用于诊断性能问题或调试存储过程或复杂的系统

DBCC TRACEON (3604) ：打开跟踪标记

DBCC TRACEOFF ：关闭跟踪标记

DBCC TRACESTATS ：查看跟踪标记状态

七、官方使用DBCC的建议

1、在系统使用率较低时运行 CHECKDB。

2、请确保未同时执行其它磁盘 I/O 操作，例如磁盘备份。

3、将 tempdb 放到单独的磁盘系统或快速磁盘子系统中。

4、允许 tempdb 在驱动器上有足够的扩展空间。 使用带有 ESTIMATE ONLY 的 DBCC 估计 tempdb 将需要多少空间。

5、避免运行占用大量 CPU 的查询或批处理作业。

6、在 DBCC 命令运行时，减少活动事务。

7、使用 NO\_INFOMSGS 选项减少一些信息的输出。

8、考虑使用带有 PHYSICAL\_ONLY 选项的 DBCC CHECKDB 来检查页和记录的物理结构。

PHYSICAL\_ONLY 选项：只检查物理错误，不检查逻辑错误

物理错误比逻辑出更严重，因为物理错误一般SQLSERVER都不能修复的，而逻辑错误大部分SQLSERVER都可以修复