



快速参考指南

# **Adaptive Server<sup>®</sup> Enterprise**

15.7

文档 ID: DC32973-01-1570-01

最后修订日期: 2011 年 9 月

版权所有 © 2011 by Sybase, Inc. 保留所有权利。

本出版物适用于 Sybase 软件 and 任何后续版本, 除非在新版本或技术声明中另有说明。此文档中的信息如有更改, 恕不另行通知。此处说明的软件按许可协议提供, 其使用和复制必须符合该协议的条款。

若要订购附加文档, 美国和加拿大的客户请拨打客户服务部门电话 (800) 685-8225 或发传真至 (617) 229-9845。

持有美国许可协议的其它国家/地区的客户可通过上述传真号码与客户服务部门联系。所有其他国际客户请与 Sybase 子公司或当地分销商联系。仅在定期安排的软件发布日期提供升级。未经 Sybase, Inc. 的事先书面许可, 不得以任何形式、任何手段 (电子的、机械的、手工的、光学的或其它手段) 复制、传播或翻译本出版物的任何部分。

Sybase 商标可在位于 <http://www.sybase.com/detail?id=1011207> 的“Sybase 商标页”(Sybase trademarks page) 处进行查看。Sybase 和列出的标记均是 Sybase, Inc. 的商标。® 表示已在美国注册。

SAP 和此处提及的其它 SAP 产品与服务及其各自的徽标是 SAP AG 在德国和世界各地其它几个国家/地区的商标或注册商标。

Java 和所有基于 Java 的标记都是 Sun Microsystems, Inc. 在美国和其它国家/地区的商标或注册商标。

Unicode 和 Unicode 徽标是 Unicode, Inc. 的注册商标。

IBM 和 Tivoli 是 International Business Machines Corporation 在美国和/或其它国家/地区的注册商标。

提到的所有其它公司和产品名均可能是与之相关的相应公司的商标。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

# 快速参考指南

主题	
<a href="#">数据类型</a>	1
<a href="#">标准和遵从性</a>	3
<a href="#">数据类型和加密列</a>	4
<a href="#">Adaptive Server 的全局变量</a>	5
<a href="#">保留字</a>	10
<a href="#">函数</a>	12
<a href="#">命令</a>	30
<a href="#">交互式 dbsql 命令</a>	69
<a href="#">系统过程</a>	72
<a href="#">目录存储过程</a>	100
<a href="#">扩展存储过程</a>	101
<a href="#">dbcc 存储过程</a>	103
<a href="#">系统表</a>	105
<a href="#">DBCC 表</a>	111
<a href="#">监控表</a>	112
<a href="#">sybpcidb 表</a>	123
<a href="#">实用程序</a>	125

## 数据类型

有关数据类型的详细信息，请参见《参考手册：构件块》。

数据类型种类	同义词	域	存储的字节数
<i>精确数值：整数</i>			
bigint		$2^{63}$ 到 $-2^{63} - 1$ （从 -9,223,372,036,854,775,808 到 +9,223,372,036,854,775,807）之间的整数，包括这两个值	8
int	integer	$2^{31} - 1$ (2,147,483,647) 到 $-2^{31}$ (-2,147,483,648)	4

数据类型种类	同义词	域	存储的字节数
smallint		$2^{15} - 1$ (32,767) 到 $-2^{15}$ (-32,768)	2
tinyint		0 到 255 (不允许使用负数)	1
unsigned bigint		0 到 18,446,744,073,709,551,615 之间的整数	8
unsigned int		0 到 4,294,967,295 之间的整数	4
unsigned smallint		0 到 65535 之间的整数	2
<i>精确数值: 小数</i>			
numeric (p, s)		$10^{38} - 1$ 到 $-10^{38}$	2 到 17
decimal (p, s)	dec	$10^{38} - 1$ 到 $-10^{38}$	2 到 17
<i>近似数值</i>			
float (precision)		与计算机有关	缺省精度 < 16 时为 4, 缺省精度 >= 16 时为 8
double precision		与计算机有关	8
real		与计算机有关	4
<i>货币</i>			
smallmoney		214,748.3647 到 -214,748.3648	4
money		922,337,203,685,477.5807 到 -922,337,203,685,477.5808	8
<i>日期/时间</i>			
smalldatetime		1900 年 1 月 1 日至 2079 年 6 月 6 日	4
datetime		1753 年 1 月 1 日至 9999 年 12 月 31 日	8
date		0001 年 1 月 1 日至 9999 年 12 月 31 日	4
time		12:00:00AM 到 11:59:59:999PM	4
bigdatetime		0001 年 1 月 1 日到 9999 年 12 月 31 日, 12:00.000000 AM 到 11:59:59.999999 PM	8
bigtime		12:00:00.000000 AM 到 11:59:59.999999 PM	8
<i>字符</i>			
char(n)	character	页大小	n
varchar(n)	character varying、char varying	页大小	实际条目长度
unichar	Unicode 字符	页大小	$n * @@unicharsize$ ( $@@unicharsize$ 等于 2)
univarchar	Unicode 字符 varying、char varying	页大小	实际字符数 * $@@unicharsize$

数据类型种类	同义词	域	存储的字节数
nchar(n)	national character、national char	页大小	$n * @@ncharsize$
nvarchar(n)	nchar varying、national char varying、national character varying	页大小	$@@ncharsize * \text{字符数}$
text		$2^{31} - 1$ (2,147,483,647) 字节或者更少	未初始化时为 0；初始化后是 2K 的倍数
unitext		1 - 1,073,741,823	未初始化时为 0；初始化后是 2K 的倍数
<i>二进制</i>			
binary(n)		页大小	$n$
varbinary(n)		页大小	实际条目长度
image		$2^{31} - 1$ (2,147,483,647) 字节或者更少	未初始化时为 0；初始化后是 2K 的倍数
<i>位</i>			
bit		0 或 1	1（1 个字节最多容纳 8 个 bit 列）

# 标准和遵从性

这些是 ANSI SQL 标准和 Transact-SQL 数据类型的符合级别。有关标准和符合级别的详细信息请参见《参考手册：构件块》。

**Transact-SQL — ANSI SQL 数据类型** char、varchar、smallint、int、bigint、decimal、numeric、float、real、date、time、double precision

**Transact-SQL 扩展 — 用户定义的数据类型** binary、varbinary、bit、nchar、datetime、smalldatetime、bigdatetime、bigtime、tinyint、unsigned smallint、unsigned int、unsigned bigint、money、smallmoney、text、unitext、image、nvarchar、unichar、univarchar、sysname、longsysname、timestamp

# 数据类型和加密列

下表列出了加密列支持的数据类型以及数据类型的加密列在磁盘上的长度。

数据类型	输入数据长度	加密列类型	最大加密数据长度（无 init_vector）	实际加密数据长度（无 init_vector）	最大加密数据长度（有 init_vector）	实际加密数据长度（有 init_vector）
date	4	varbinary	17	17	33	33
time	4	varbinary	17	17	33	33
smalldatetime	4	varbinary	17	17	33	33
bigdatetime	8	varbinary	17	17	33	33
bigtime	8	varbinary	17	17	33	33
datetime	8	varbinary	17	17	33	33
smallmoney	4	varbinary	17	17	33	33
money	8	varbinary	17	17	33	33
bit	8	varbinary	17	17	33	33
bigint	8	varbinary	17	17	33	33
unsigned bigint	8	varbinary	17	17	33	33
unichar(10)	2（1 个 unichar 字符）	varbinary	33	17	49	33
unichar(10)	20（10 个 unichar 字符）	varbinary	33	33	49	49
univarchar(20)	20（10 个 unichar 字符）	varbinary	49	33	65	49

---

## Adaptive Server 的全局变量

下面是 Adaptive Server 的全局变量及其简要说明。有关完整信息，请参见《参考手册：构件块》。

<code>@@active_instances</code>	返回集群中活动实例的数目。
<code>@@authmech</code>	指示用于鉴定用户的机制。
<code>@@bootcount</code>	返回安装已启动的次数。
<code>@@boottime</code>	返回 Adaptive Server 上次启动的日期和时间。
<code>@@bulkarraysize</code>	返回在使用批量复制接口传送之前要在本地服务器内存中放入缓冲区的行数。
<code>@@bulkbatchsize</code>	返回使用批量接口通过 <code>select into proxy_table</code> 传输到远程服务器的行数。
<code>@@char_convert</code>	如果字符集转换无效，则返回 0；如果字符集转换有效，则返回 1。
<code>@@cis_rpc_handling</code>	如果 cis rpc handling 关闭，则返回 0；如果 cis rpc handling 打开，则返回 1。
<code>@@cis_version</code>	返回组件集成服务的日期和版本。
<code>@@client_csexpansion</code>	返回从服务器字符集转换为客户端字符集时使用的扩展因子。
<code>@@client_csid</code>	如果客户端字符集从未初始化，则返回 -1；如果客户端字符集已初始化，则通过 <code>syscharsets</code> 返回连接的客户端字符集 ID。
<code>@@client_csname</code>	如果客户端字符集从未初始16 化，则返回 NULL；如果客户端字符集已初始化，则返回用于连接的字符集名称。
<code>@@clusterboottime</code>	返回首次启动集群的日期和时间，即使最初启动集群的实例已关闭
<code>@@clustercoordid</code>	返回当前集群事务协调器的实例 ID。
<code>@@clustermode</code>	返回字符串：“shared-disk cluster”。
<code>@@clustername</code>	返回集群的名称。
<code>@@cmpstate</code>	返回在高可用性环境下 Adaptive Server 的当前模式。
<code>@@connections</code>	返回用户尝试登录的次数。
<code>@@cpu_busy</code>	返回自 Adaptive Server 上次启动起，CPU 执行 Adaptive Server 工作所花费的时间（以时钟周期计）。
<code>@@cursor_rows</code>	该变量显示游标结果集中的总行数。
<code>@@curloid</code>	没有打开游标，没有行符合上次打开的游标的条件，或上次打开的游标已关闭或已释放。

<code>@@datefirst</code>	返回 <code>@@datefirst</code> 的当前值，指示每周指定的第一天，表示为 <code>tinyint</code> 数据类型。
<code>@@dbts</code>	返回当前数据库的时间戳。
<code>@@error</code>	返回系统最近一次生成的错误号。
<code>@@errorlog</code>	返回 Adaptive Server 错误日志所在目录相对于 <code>SSYBASE</code> 目录的完整路径。
<code>@@failedoverconn</code>	如果与主协同服务器的连接发生了故障切换而转到辅助协同服务器上执行，则返回一个大于 0 的值。
<code>@@fetch_status</code>	如果提取操作成功，则返回 0；如果提取操作失败，则返回 -1。 -2 留待将来使用。
<code>@@guestuserid</code>	返回 <code>guest</code> 用户的 ID。
<code>@@hacmpservername</code>	返回高可用性设置中的协同服务器的名称。
<code>@@haconnection</code>	如果连接启用了故障切换属性，则返回一个大于 0 的值。
<code>@@heapmemsize</code>	返回堆内存池的大小，单位为字节。
<code>@@identity</code>	返回最近一次生成的 <code>IDENTITY</code> 列值。
<code>@@idle</code>	返回 Adaptive Server 自上次启动起空闲的时间（以时钟周期计）。
<code>@@invaliduserid</code>	对无效的用户 ID 返回值 -1。
<code>@@instanceid</code>	返回从中执行它的实例的 ID。
<code>@@instancename</code>	返回从中执行它的实例的名称。
<code>@@invaliduserid</code>	对无效的用户 ID 返回值 -1。
<code>@@io_busy</code>	返回 Adaptive Server 执行输入和输出操作所花费的时间（以时钟周期计）。
<code>@@isolation</code>	返回当前 Transact-SQL 程序特定于会话的隔离级别值（0、1 或 3）。
<code>@@instanceid</code>	Job Scheduler 正在其上运行或启用后将在其上运行的实例的 ID。
<code>@@kernel_addr</code>	返回包含内核区域的第一个共享内存区域的起始地址。
<code>@@kernel_size</code>	返回属于第一个共享内存区域的一部分的内核区域的大小。
<code>@@kernelmode</code>	返回为 Adaptive Server 配置的模式（线程化或进程）。
<code>@@langid</code>	返回当前在整个服务器范围内使用的语言的语言 ID，该值在 <code>syslanguages.langid</code> 中指定。
<code>@@language</code>	返回当前使用的语言的名称，该名称在 <code>syslanguages.name</code> 中指定。



---

<code>@@lastkpgendate</code>	返回按 <code>sp_passwordpolicy</code> “keypair regeneration period” 策略选项所设置的那样生成最后一个密钥对的日期和时间。
<code>@@lastlogindate</code>	包括 <code>datetime</code> 数据类型，其值为当前会话建立之前的登录帐户的 <code>lastlogindate</code> 列。
<code>@@lock_timeout</code>	返回当前的 <code>lock_timeout</code> 设置，单位为毫秒。
<code>@@lwpid</code>	返回紧接着最近运行的轻量过程的对象 ID。
<code>@@max_connections</code>	返回在当前计算机环境下可同时与 Adaptive Server 连接的最大数目。
<code>@@max_precision</code>	返回由服务器设置的 <code>decimal</code> 和 <code>numeric</code> 数据类型所使用的精度级别。该值固定为常数 38。
<code>@@maxcharlen</code>	返回 Adaptive Server 的缺省字符集中一个字符的最大长度，单位为字节。
<code>@@maxgroupid</code>	返回最大的组用户 ID。最大值为 1048576。
<code>@@maxpagesize</code>	返回服务器的逻辑页大小。
<code>@@maxspid</code>	返回 <code>spid</code> 的最大有效值。
<code>@@maxsuid</code>	返回最大的服务器用户 ID。缺省值为 2147483647。
<code>@@maxuserid</code>	返回最大的用户 ID。最大值为 2147483647。
<code>@@mempool_addr</code>	返回全局内存池表的地址。
<code>@@min_poolsize</code>	返回命名高速缓存池的最小大小，单位为千字节。
<code>@@mingroupid</code>	返回最小的组用户 ID。最小值为 16384。
<code>@@minspid</code>	返回 1，即 <code>spid</code> 的最小值。
<code>@@minsuid</code>	返回最小的服务器用户 ID。最小值为 -32768。
<code>@@minuserid</code>	返回最小的用户 ID。最小值为 -32768。
<code>@@monitors_active</code>	减少 <code>sp_sysmon</code> 显示的消息数。
<code>@@ncharsize</code>	返回当前服务器的缺省字符集中一个字符的最大长度，单位为字节。
<code>@@nestlevel</code>	返回当前的嵌套级别。
<code>@@nextkpgendate</code>	返回按 <code>sp_passwordpolicy</code> “keypair regeneration period” 策略选项所设置的那样预定生成下一个密钥对的日期和时间。
<code>@@nodeid</code>	返回当前安装的 48 位节点标识符。
<code>@@optgoal</code>	返回查询优化当前的优化目标设置。
<code>@@options</code>	返回以十六进制表示的会话的 <code>set</code> 选项。
<code>@@optlevel</code>	返回当前的优化级别设置。

<code>@@opttimeoutlimit</code>	返回查询优化当前的优化超时限制设置。
<code>@@ospid</code>	（仅限线程化模式）返回服务器的操作系统 ID。
<code>@@pack_received</code>	返回 Adaptive Server 读取的输入包的数目。
<code>@@pack_sent</code>	返回 Adaptive Server 写入的输出包的数目。
<code>@@packet_errors</code>	返回 Adaptive Server 读写数据包时检测到的错误数。
<code>@@pagesize</code>	返回服务器的虚拟页的大小。
<code>@@parallel_degree</code>	返回当前的最大并行度设置。
<code>@@plwpid</code>	返回最近准备的轻量过程的对象 ID。
<code>@@probesuid</code>	对于 probe 用户 ID 返回值 2。
<code>@@procid</code>	返回当前执行过程的存储过程 ID。
<code>@@quorum_physname</code>	返回仲裁设备的物理路径。
<code>@@recovery_state</code>	根据这些返回值指示 Adaptive Server 是否正在进行恢复。
<code>@@remotestate</code>	（仅限高可用性）返回高可用性环境中主协同服务器的当前模式。
<code>@@repartition_degree</code>	返回当前的动态重新分区程度设置。
<code>@@resource_granularity</code>	返回查询优化的最大资源使用情况提示设置。
<code>@@rowcount</code>	返回上次查询所影响的行数。
<code>@@scan_parallel_degree</code>	返回用于非聚簇索引扫描的当前最大并行度设置。
<code>@@servername</code>	返回 Adaptive Server 的名称。
<code>@@setrowcount</code>	返回 set rowcount 的当前值。
<code>@@shmem_flags</code>	返回共享内存区域的属性。
<code>@@spid</code>	返回当前进程的服务器进程 ID。
<code>@@sqlstatus</code>	返回执行 fetch 语句时产生的状态信息（警告性例外）。
<code>@@ssl_ciphersuite</code>	如果 SSL 未用于当前连接，则返回 NULL；否则，它返回在当前连接的 SSL 握手期间您选择的密码成套程序的名称。
<code>@@stringsize</code>	返回从 toString() 方法返回的字符数据的数量。
<code>@@sys_tempdbid</code>	返回正在执行的实例的有效本地系统临时数据库的数据库 ID
<code>@@system_busy</code>	Adaptive Server 正在运行系统任务的时钟周期数。
<code>@@system_view</code>	返回特定于会话的系统视图设置 “instance” 或 “cluster”。

---

<code>@@tempdbid</code>	返回为会话指定的临时数据库的有效临时数据库 ID (dbid)。
<code>@@textcolid</code>	返回 <code>@@textptr</code> 所引用的列的列 ID。
<code>@@textdataptid</code>	返回一个文本分区的分区 ID，该文本分区包含被 <code>@@textptr</code> 引用的列。
<code>@@textdbid</code>	返回以下数据库的数据库 ID：该数据库中的对象具有被 <code>@@textptr</code> 引用的列。
<code>@@textobjid</code>	返回以下对象的对象 ID：该对象具有被 <code>@@textptr</code> 引用的列。
<code>@@textptnid</code>	返回一个数据分区的分区 ID，该数据分区包含被 <code>@@textptr</code> 引用的列。
<code>@@textptr</code>	返回某个进程上次插入或更新的 text、unitext 或 image 的文本指针（与 textptr 函数不同）。
<code>@@textptr_parameters</code>	如果 textptr_parameters 的当前状态为关闭，则返回 0；如果 textptr_parameters 的当前状态为打开，则返回 1。
<code>@@textsize</code>	返回 select 所返回的 text、unitext 或 image 数据的字节数限制。
<code>@@textts</code>	返回 <code>@@textptr</code> 所引用列的文本时间戳。
<code>@@thresh_hysteresis</code>	返回激活阈值所要求的可用空间的减小量。
<code>@@timeticks</code>	返回每个时钟周期中的微秒数。
<code>@@total_errors</code>	返回 Adaptive Server 读写数据时检测到的错误数。
<code>@@total_read</code>	返回 Adaptive Server 的磁盘读取操作的数目。
<code>@@total_write</code>	返回 Adaptive Server 的磁盘写入操作的数目。
<code>@@tranchained</code>	如果 Transact-SQL 程序的当前事务模式为非链式，则返回 0；如果为链式，则返回 1。
<code>@@trancount</code>	返回当前用户会话中事务的嵌套级别。
<code>@@transactional_rpc</code>	如果对远程服务器的 RPC 操作是事务型的，则返回 0；否则返回 1。
<code>@@transtate</code>	返回当前用户会话执行某个语句之后事务的当前状态。
<code>@@unicharsize</code>	返回 2，它是 unichar 类型的字符的大小。
<code>@@user_busy</code>	Adaptive Server 正在运行用户任务的时钟周期数。
<code>@@version</code>	返回 Adaptive Server 当前版本的日期、版本字符串等信息。
<code>@@version_as_integer</code>	返回 Adaptive Server 当前版本的最近升级版本的版本号，用整数表示。
<code>@@version_number</code>	返回 Adaptive Server 当前版本的完整版本号，用整数表示。

## 保留字

本节列出了各种保留字。有关详细信息，请参见《参考手册：构件块》。

### Transact-SQL 保留字

以下是 Adaptive Server 保留的关键字（SQL 命令语法的一部分）。

	保留字
<i>A</i>	add、all、alter、and、any、arith_overflow、as、asc、at、authorization、avg
<i>B</i>	begin、between、break、browse、bulk、by
<i>C</i>	cascade、case、char_convert、check、checkpoint、close、clustered、coalesce、commit、compute、confirm、connect、constraint、continue、controlrow、convert、count、count_big、create、current、cursor
<i>D</i>	database、dbcc、deallocate、declare、decrypt、default、delete、desc、deterministic、disk、distinct、drop、dummy、dump
<i>E</i>	else、encrypt、end、endtran、errlvl、errordata、errorexit、escape、except、exclusive、exec、execute、exists、exit、exp_row_size、external
<i>F</i>	fetch、fillfactor、for、foreign、from
<i>G</i>	goto、grant、group
<i>H</i>	having、holdlock
<i>I</i>	identity、identity_gap、identity_start、if、in、index、inout、insensitive、insert、install、intersect、into、is、isolation
<i>J</i>	jar、join
<i>K</i>	key、kill
<i>L</i>	level、like、lineno、load、lock
<i>M</i>	materialized、max、max_rows_per_page、min、mirror、mirrorexit、modify
<i>N</i>	national、new、noholdlock、nonclustered、not、null、nullif、numeric_truncation
<i>O</i>	of、off、offsets、on、once、online、only、open、option、or、order、out、output、over
<i>P</i>	partition、perm、permanent、plan、prepare、primary、print、privileges、proc、procedure、processexit、proxy_table、public
<i>Q</i>	quiesce
<i>R</i>	raiserror、read、readpast、readtext、reconfigure、references、remove、reorg、replace、replication、reservepagegap、return、returns、revoke、role、rollback、rowcount、rows、rule
<i>S</i>	save、schema、scroll、select、semi_sensitive、set、setuser、shared、shutdown、some、statistics、stringsize、stripe、sum、syb_identity、syb_restree、syb_terminate
<i>T</i>	table、temp、temporary、textsize、to、tracefile、tran、transaction、trigger、truncate、tsequal
<i>U</i>	union、unique、unpartition、update、use、user、user_option、using
<i>V</i>	values、varying、view

	保留字
<i>W</i>	waitfor、when、where、while、with、work、writetext
<i>X</i>	xmlextract、xmlparse、xmltest

## ANSI SQL 保留字

以下是未被 Adaptive Server 保留的 ANSI SQL 关键字。

	保留字
<i>A</i>	absolute、action、allocate、are、assertion
<i>B</i>	bit、bit_length、both
<i>C</i>	cascaded、case、cast、catalog、char、char_length、character、character_length、coalesce、collate、collation、column、connection、constraints、corresponding、cross、current_date、current_time、current_timestamp、current_user
<i>D</i>	date、day、dec、decimal、deferrable、deferred、describe、descriptor、diagnostics、disconnect、domain
<i>E</i>	end-exec、exception、extract
<i>F</i>	false、first、float、found、full
<i>G</i>	get、global、go
<i>H</i>	hour
<i>I</i>	immediate、indicator、initially、inner、input、insensitive、int、integer、interval
<i>J</i>	join
<i>L</i>	language、last、leading、left、local、lower
<i>M</i>	match、minute、module、month
<i>N</i>	names、natural、nchar、next、no、nullif、numeric
<i>O</i>	octet_length、outer、output、overlaps
<i>P</i>	pad、partial、position、preserve、prior
<i>R</i>	real、relative、restrict、right
<i>S</i>	scroll、second、section、semi_sensitive、session_user、size、smallint、space、sql、sqlcode、sqlerror、sqlstate、substring、system_user
<i>T</i>	then、time、timestamp、timezone_hour、timezone_minute、trailing、translate、translation、trim、true
<i>U</i>	unknown、upper、usage
<i>V</i>	value、varchar
<i>W</i>	when、whenever、write、year
<i>Z</i>	zone

可能的 ANSI SQL 保留字

采用 ISO/IEC 9075:1989 标准时，应避免使用以下词语，因为它们以后可能会成为 ANSI SQL 保留字。

	保留字
<i>A</i>	after、alias、async
<i>B</i>	before、boolean、breadth
<i>C</i>	call、completion、cycle
<i>D</i>	data、depth、dictionary
<i>E</i>	each、elseif、equals
<i>G</i>	general
<i>I</i>	ignore
<i>L</i>	leave、less、limit、loop
<i>M</i>	modify
<i>N</i>	new、none
<i>O</i>	object、oid、old、operation、operators、others
<i>P</i>	parameters、pendant、preorder、private、protected
<i>R</i>	recursive、ref、referencing、resignal、return、returns、routine、row
<i>S</i>	savepoint、search、sensitive、sequence、signal、similar、sqlexception、structure
<i>T</i>	test、there、type
<i>U</i>	under
<i>V</i>	variable、virtual、visible
<i>W</i>	wait、without

函数

以下是内置函数的简要说明和语法。有关完整信息，请参见 《参考手册：构件块》。

abs	返回表达式的绝对值。 <code>abs(numeric_expression)</code>
acos	返回具有指定余弦的角（以弧度表示）。 <code>acos(cosine)</code>
ascii	返回表达式中第一个字符的 ASCII 代码。 <code>ascii(char_expr   uchar_expr)</code>

---

asehostname	<p>返回 Adaptive Server 运行所在的物理或虚拟主机。</p> <p><code>asehostname</code></p>
asin	<p>返回指定正弦的角（以弧度表示）。</p> <p><code>asin(<i>sine</i>)</code></p>
atan	<p>返回指定正切的角（以弧度表示）。</p> <p><code>atan(<i>tangent</i>)</code></p>
atn2	<p>返回具有指定正弦和余弦的角（以弧度表示）。</p> <p><code>atn2(<i>sine</i>, <i>cosine</i>)</code></p>
avg	<p>计算所有（不同）值的数字平均值。</p> <p><code>avg([all   distinct] <i>expression</i>)</code></p>
audit_event_name	<p>返回对审计事件的描述。</p> <p><code>audit_event_name(<i>event_id</i>)</code></p>
authmech	<p>确定指定的已登录服务器进程 ID 使用哪种鉴定机制。</p> <p><code>authmech ([<i>spid</i>])</code></p>
biginttohex	<p>返回与指定的整数等值且与平台无关的 8 字节十六进制值。</p> <p><code>biginttohex (<i>integer_expression</i>)</code></p>
bintostr	<p>将十六进制数字序列转换为其等价的字母数字字符串或 varbinary 数据。</p> <p><code>select bintostr(<i>sequence of hexadecimal digits</i>)</code></p>
cache_usage	<p>以表所属高速缓存中所有对象的百分比形式返回高速缓存使用情况。</p> <p><code>cache_usage(<i>table_name</i>)</code></p>
case	<p>支持条件 SQL 表达式；可用于任何能够使用值表达式的情况。</p> <pre> case   when <i>search_condition</i> then <i>expression</i>   [when <i>search_condition</i> then <i>expression</i>]...   [else <i>expression</i>] end </pre> <p>case 和值语法：</p> <pre> case <i>expression</i>   when <i>expression</i> then <i>expression</i>   [when <i>expression</i> then <i>expression</i>]...   [else <i>expression</i>] end </pre>
cast	<p>将指定的值转换为其它数据类型。</p> <p><code>cast (<i>expression</i> as <i>datatype</i> [(<i>length</i>   <i>precision</i>[, <i>scale</i>)]])</code></p>

ceiling	返回大于或等于指定值的最小整数。 <code>ceiling(<i>value</i>)</code>
char	返回整数的等值字符。 <code>char(<i>integer_expr</i>)</code>
char_length	返回表达式中字符的数量。 <code>char_length(<i>char_expr</i>   <i>uchar_expr</i>)</code>
charindex	返回表示表达式起始位置的整数。 <code>charindex(<i>expression1</i>, <i>expression2</i> [, <i>start</i>])</code>
coalesce	支持条件 SQL 表达式；可用于任何可以使用值表达式的情况； <b>case</b> 表达式的替代形式。 <code>coalesce(<i>expression</i>, <i>expression</i> [, <i>expression</i>]...)</code>
col_length	返回已定义的列长度。 <code>col_length(<i>object_name</i>, <i>column_name</i>)</code>
col_name	返回已指定表 ID 和列 ID 的列的名称，且最长可为 255 个字节。 <code>col_name(<i>object_id</i>, <i>column_id</i> [, <i>database_id</i>])</code>
compare	您可以根据替代归类规则直接比较两个字符串。 <code>compare ({<i>char_expression1</i> <i>uchar_expression1</i>},           {<i>char_expression2</i> <i>uchar_expression2</i>}),           [<i>collation_name</i>   <i>collation_ID</i>])</code>
convert	将指定的值转换成另一数据类型或其它 <b>datetime</b> 显示格式。 <code>convert (<i>datatype</i> [(<i>length</i>)]   (<i>precision</i> [, <i>scale</i>]))           [<i>null</i>   <i>not null</i>], <i>expression</i> [, <i>style</i>])</code>
cos	返回指定角（以弧度表示）的余弦。 <code>cos(<i>angle</i>)</code>
cot	返回指定角（以弧度表示）的余切。 <code>cot(<i>angle</i>)</code>
count	以整数形式返回（不同）非空值的数量或选定的行数。 <code>count([<i>all</i>   <i>distinct</i>] <i>expression</i>)</code>
count_big	以 <b>bigint</b> 形式返回（不同）非空值的数量或选定的行数。 <code>count_big([<i>all</i>   <i>distinct</i>] <i>expression</i>)</code>
create_locator	为指定的 LOB 显式创建定位符，然后返回该定位符。 <code>create_locator (<i>datatype</i>, <i>lob_expression</i>)</code>



---

<code>current_bigdatetime</code>	<p>返回一个 <code>bigtime</code> 值，该值表示具有微秒级精度的当前时间。当前时间部分的精度受到系统时钟准确性的限制。</p> <p><code>current_bigdatetime()</code></p>
<code>current_bigtime</code>	<p>返回一个 <code>bigtime</code> 值，该值表示具有微秒级精度的当前时间。当前时间部分的精度受到系统时钟准确性的限制。</p> <p><code>current_bigtime()</code></p>
<code>current_date</code>	<p>返回当前日期。</p> <p><code>current_date()</code></p>
<code>current_time</code>	<p>返回当前时间。</p> <p><code>current_time()</code></p>
<code>curunreservedpgs</code>	<p>显示指定磁盘区段中的可用页数。</p> <p><code>curunreservedpgs (dbid, lstart, unreservedpgs)</code></p>
<code>data_pages</code>	<p>返回指定表、索引或特定分区所用的页数。其结果不包括用于内部结构的页。</p> <p><code>data_pages(dbid, object_id [, indid [, ptnid]])</code></p>
<code>datachange</code>	<p>用于测量自上次运行 <code>update statistics</code> 以来的数据分布更改量。具体而言，它测量给定对象、分区或列中发生的 <code>inserts</code>、<code>updates</code> 和 <code>deletes</code> 操作的次数，并帮助您确定调用 <code>update statistics</code> 是否有益于查询计划。</p> <p><code>datachange(object_name, partition_name, column_name)</code></p>
<code>datalength</code>	<p>返回指定列或字符串的实际长度（以字节表示）。</p> <p><code>datalength(expression)</code></p>
<code>dateadd</code>	<p>向指定日期或时间添加时间间隔。</p> <p><code>dateadd(date_part, integer, {date   time   bigtime   datetime,   bigdatetime})</code></p>
<code>datediff</code>	<p>计算两个指定的日期或时间之间的日期分量数。</p> <p><code>datediff(datepart, {date, date   time, time   bigtime, bigtime   datetime, datetime   bigdatetime, bigdatetime})</code></p>
<code>datename</code>	<p>以字符串形式返回指定 <code>date</code> 或 <code>time</code> 的指定 <code>datepart</code>。</p> <p><code>datename(datepart {date   time   bigtime   datetime   bigdatetime})</code></p>
<code>datepart</code>	<p>返回日期表达式的指定分量的整数值。</p> <p><code>datepart(date_part {date   time   datetime   bigtime   bigdatetime})</code></p>
<code>day</code>	<p>返回指定日期的 <code>datepart</code> 中表示日期的整数。</p> <p><code>day(date_expression)</code></p>

db_attr	返回指定数据库的 <code>durability</code> 、 <code>dml_logging</code> 和 <code>template</code> 设置，以及压缩级别。  <code>db_attr('database_name'   database_ID   NULL, 'attribute')</code>
db_id	显示指定数据库的 ID 号。  <code>db_id(database_name)</code>
db_instanceid	（仅限集群环境）返回指定的本地临时数据库的所有者实例 ID。如果指定的数据库是全局临时数据库或非临时数据库，则返回 NULL。  <code>db_instanceid(database_id   database_name)</code>
db_name	显示具有指定 ID 号的数据库的名称。  <code>db_name([database_id])</code>
db_recovery_status	（仅限集群环境）返回指定数据库的恢复状态。如果未包括 <code>database_ID</code> 或 <code>database_name</code> 的值，则返回当前数据库的恢复状态。  <code>db_recovery_status([database_ID   database_name])</code>
degrees	返回具有指定弧度的角的大小（以度表示）。  <code>degrees(numeric)</code>
derived_stat	返回指定对象和索引的派生统计信息。  <code>derived_stat("object_name"   object_id,               index_name   index_id,               ["partition_name"   partition_id,]               "statistic")</code>
difference	返回两个 <code>soundex</code> 值之间的差值。  <code>difference(expr1,expr2)</code>
dol_downgrade_check	返回指定数据库中含有宽于 8191 字节的可变长度列的仅数据锁定 (DOL) 表的数量。如果没有宽的可变长度列，则返回 0，您可以放心执行降级。  <code>dol_downgrade_check('database_name', target_version)</code>
exp	计算对常量求指定次幂所得的值。  <code>exp(approx_numeric)</code>
floor	返回小于或等于指定值的最大整数。  <code>floor(numeric)</code>
get_appcontext	返回指定环境中的属性值。 <code>get_appcontext</code> 由应用程序环境功能 (ACF) 提供。  <code>get_appcontext ("context_name", "attribute_name")</code>

---

getdate	返回当前系统日期和时间。 <code>getdate()</code>
getutcdate	返回协调通用时间 (UTC) 格式的日期和时间值。每次插入或选择行时都要计算 <code>getutcdate</code> 。 <code>getutcdate()</code>
has_role	返回关于是否已授予用户指定角色的信息。 <code>has_role ("role_name", option)</code>
hash	生成固定长度的散列值表达式。 <code>hash(expression , [algorithm])</code>
hashbytes	生成固定长度的散列值表达式。 <code>hashbytes(algorithm, expression[, expression...] [, using options])</code>
hextobigint	返回十六进制字符串的等值 <code>bigint</code> 值 <code>hextobigint(hexadecimal_string)</code>
hextoint	返回与十六进制字符串等值且与平台无关的整数。 <code>hextoint(hexadecimal_string)</code>
host_id	返回当前 Adaptive Server 客户端的客户端计算机操作系统进程 ID。 <code>host_id()</code>
host_name	显示客户端进程的当前宿主计算机名。 <code>host_name()</code>
instance_id	(仅限集群环境) 返回命名实例的 ID；如果没有为 <i>name</i> 提供值，则返回发出该函数的实例的 ID。 <code>instance_id([name])</code>
identity_burn_max	跟踪给定表的 <code>identity burn</code> 最大值。此函数只返回值；它不执行更新。 <code>identity_burn_max(table_name)</code>
index_col	显示指定表或视图中带索引的列的名称，最长可为 255 个字节。 <code>index_col(object_name, index_id, key_#, user_id)</code>
index_colorder	返回列的顺序。 <code>index_colorder(object_name, index_id, key_#[, user_id])</code>
index_name	当您提供了索引 ID、数据库 ID 和定义了其索引的对象时，返回索引名称。 <code>index_name(dbid, objid, indid)</code>
inttohex	返回与指定的整数等值且与平台无关的十六进制值。 <code>inttohex(integer_expression)</code>

isdate	确定输入表达式是否为有效的 <code>datetime</code> 值。 <code>isdate(character_expression)</code>
isnumeric	确定表达式是否为有效的 <code>numeric</code> 数据类型。 <code>isnumeric (character_expression)</code>
is_quiesced	指示数据库是否处于 <code>quiesce database</code> 模式。如果数据库已停顿，则 <code>is_quiesced</code> 返回 1，否则返回 0。 <code>is_quiesced(dbid)</code>
is_sec_service_on	确定是否启用了某个特定的安全服务。如果已启用该服务则返回 1；否则返回 0。 <code>is_sec_service_on(security_service_nm)</code>
is_singleusermode	如果 Adaptive Server 不在单用户模式下运行，则返回 0。如果 Adaptive Server 在单用户模式下运行，则返回 1。 <code>is_singleusermode()</code>
isnull	当 <code>expression1</code> 求值为 NULL 时，替代 <code>expression2</code> 中指定的值。 <code>isnull(expression1, expression2)</code>
isnumeric	确定表达式是否为有效的 <code>numeric</code> 数据类型。 <code>isnumeric (character_expression)</code>
instance_name	(仅限集群环境) 返回提供了其 ID 的 Adaptive Server 的名称；如果没有为 <code>id</code> 提供值，则返回发出该函数的 Adaptive Server 的名称。 <code>instance_name([id])</code>
lc_id	(仅限集群环境) 返回为其提供了名称的逻辑集群的 ID；如果没有提供名称，则返回当前逻辑集群的 ID。 <code>lc_id(logical_cluster_name)</code>
lc_name	(仅限集群环境) 返回提供了其 ID 的逻辑集群的名称；如果未提供 ID，则返回当前逻辑集群的名称。 <code>lc_name([logical_cluster_ID])</code>
lct_admin	管理最后机会阈值 (LCT)。它返回 LCT 的当前值，并中止已经到达其 LCT 的事务日志中的事务。 <code>lct_admin({{"lastchance"   "logfull"   "reserved_for_rollbacks"}, database_id, "reserve", {log_pages   0 }   "abort", process-id [, database-id]})</code>
left	返回字符串最左侧指定数目的字符。 <code>left(character_expression, integer_expression)</code>

---

len	<p>返回指定字符串表达式（不包括尾随空白）的字符数（而不是字节数）。</p> <p><code>len(string_expression)</code></p>
license_enabled	<p>如果启用了某功能的许可证，则返回 1；如果未启用许可证，则返回 0；如果指定的许可证名称无效，则返回 NULL。</p> <p><code>license_enabled("ase_server"   "ase_ha"   "ase_dtm"   "ase_java"   "ase_asm")</code></p>
list_appcontext	<p>列出当前会话中所有环境的全部属性。list_appcontext 由 ACF 提供。</p> <p><code>list_appcontext(["context_name"])</code></p>
locator_literal	<p>将二进制值标识为定位符文字。</p> <p><code>locator_literal(locator_type, literal_locator)</code></p>
locator_valid	<p>确定 LOB 定位符是否有效。</p> <p><code>locator_valid (locator_descriptor)</code></p>
lockscheme	<p>以字符串的形式返回指定对象的锁定方案。</p> <p><code>lockscheme(object_name)</code>  <code>lockscheme(object_id [, db_id])</code></p>
log	<p>计算指定数字的自然对数。</p> <p><code>log(approx_numeric)</code></p>
log10	<p>计算指定数字的以 10 为底数的对数。</p> <p><code>log10(approx_numeric)</code></p>
lower	<p>将大写字符转换为小写。</p> <p><code>lower(char_expr   uchar_expr)</code></p>
lprofile_id	<p>返回与指定名称关联的 ID，或者返回当前登录的 ID。</p> <p><code>lprofile_id(name),</code></p>
lprofile_name	<p>返回与指定 ID 关联的名称，或者返回当前登录的名称。</p> <p><code>lprofile_id(ID),</code></p>
ltrim	<p>删去指定表达式的前导空白。</p> <p><code>ltrim(char_expr   uchar_expr)</code></p>
max	<p>返回表达式的最大值。</p> <p><code>max(expression)</code></p>
min	<p>返回列中的最小值。</p> <p><code>min(expression)</code></p>

month	返回指定日期的 <code>datepart</code> 中表示月份的整数。 <code>month(date_expression)</code>
mut_excl_roles	返回有关两个角色之间互斥性的信息。 <code>mut_excl_roles (role1, role2 [membership   activation])</code>
newid	根据提供的参数生成两种不同格式的、人工可读的全局唯一 ID (GUID)。人工可读格式的 GUID 值长度为 32 个字节（无短划线）或 36 个字节（带短划线）。 <code>newid([optionflag])</code>
next_identity	检索下一个 insert 可用的下一个标识值。 <code>next_identity(table_name)</code>
nullif	允许为条件值编写 SQL 表达式。所有可以使用值表达式的地方都可以使用 <code>nullif</code> 表达式；可替代 <code>case</code> 表达式。 <code>nullif(expression, expression)</code>
object_attr	根据会话、表和数据库范围的设置来报告表的当前记录模式。 <code>object_attr(table_name, string)</code>
object_id	返回指定对象的对象 ID。 <code>object_id(object_name)</code>
object_name	返回具有指定对象 ID 的对象的名称；最长可为 255 个字节。 <code>object_name(object_id[, database_id])</code>
object_owner_id	返回对象的所有者 ID。 <code>object_owner_id(object_id[, database_id])</code>
pagesize	以字节为单位返回指定对象的页大小。 <code>pagesize(object_name[, ])</code> <code>pagesize(object_id[, db_id[, index_id]])</code>
partition_id	返回指定数据或索引分区名称的分区 ID。 <code>partition_id(table_name, partition_name[, index_name])</code>
partition_name	返回新分区的显式名称， <code>partition_name</code> 返回指定数据或索引分区 ID 的分区名称。 <code>partition_name(indid, ptnid[, dbid])</code>
partition_object_id	显示指定分区 ID 和数据库 ID 的对象 ID。 <code>partition_object_id(partition_id[, database_id] )</code>

---

patindex	<p>返回指定模式第一次出现时的起始位置。</p> <pre>patindex("%pattern%", char_expr   uchar_expr[, using {bytes   characters   chars}])</pre>
pi	<p>返回常量值 3.1415926535897936。</p> <pre>pi()</pre>
power	<p>返回求指定数字的给定次幂所得的值。</p> <pre>power(value, power)</pre>
proc_role	<p>返回有关是否已授予用户指定角色的信息。</p> <pre>proc_role("role_name")</pre>
pssinfo	<p>从 Adaptive Server 进程状态结构 (pss) 返回信息。</p> <pre>pssinfo(spid   0, 'pss_field')</pre>
radians	<p>将度转换为弧度。返回指定度数角的大小（以弧度表示）。</p> <pre>radians(numeric)</pre>
rand	<p>使用指定（可选）整数作为源值，返回 0 和 1 之间的随机浮点值。</p> <pre>rand([integer])</pre>
rand2	<p>返回 0 和 1 之间的随机值，该值由指定的种子值来生成，在 <b>select</b> 列表中使用时将每个返回的行计算该值。</p> <pre>rand2([integer])</pre>
replicate	<p>返回一个字符串，其中包含重复给定次数的指定表达式，或包含重复多次，直至其大小可占据 16KB 空间的指定表达式（取其次数少者）。</p> <pre>replicate(char_expr   uchar_expr, integer_expr)</pre>
reserve_identity	<p>reserve_identity 允许进程保留标识值块，以供该进程使用。</p> <pre>reserve_identity (table_name, number_of_values)</pre>
reserved_pages	<p>报告为数据库、对象或索引保留的页数。结果包括用于内部结构的页。</p> <pre>reserved_pages(dbid, object_id[, indid[, ptnid]])</pre>
return_lob	<p>取消引用定位符，并返回该定位符引用的 LOB。</p> <pre>return_lob (datatype, locator_descriptor)</pre>
reverse	<p>返回其字符逆序排列的指定字符串。</p> <pre>reverse(expression   uchar_expr)</pre>
right	<p>返回字符或二进制表达式中从右端起具有指定字符数的部分。返回值的数据类型与字符表达式的数据类型相同。</p> <pre>right(expression, integer_expr)</pre>

<code>rm_appcontext</code>	删除特定应用程序环境或所有应用程序环境。 <code>rm_appcontext</code> 由 ACF 提供。 <code>rm_appcontext("context_name", "attribute_name")</code>
<code>role_contain</code>	确定一个指定的角色是否包含在另一个指定的角色中。 <code>role_contain("role1", "role2")</code>
<code>role_id</code>	返回指定角色名的角色 ID。 <code>role_id("role_name")</code>
<code>role_name</code>	返回指定角色 ID 的角色名。 <code>role_name(role_id)</code>
<code>round</code>	返回指定数字舍入到指定的小数位数后所得的值。 <code>round(number, decimal_places)</code>
<code>row_count</code>	返回指定表中行数的估计值。 <code>row_count(dbid, object_id [,ptnid] [, "option"])</code>
<code>rtrim</code>	删去指定表达式的尾随空白。 <code>rtrim(char_expr   uchar_expr)</code>
<code>sdc_intempdbconfig</code>	(仅限集群环境) 如果系统当前处于临时数据库配置模式, 则返回 1; 否则返回 0。 <code>sdc_intempdbconfig()</code>
<code>set_appcontext</code>	为指定应用程序的属性定义的用户会话设置应用程序环境名称、属性名称和属性值。 <code>set_appcontext</code> 由 ACF 提供。 <code>set_appcontext("context_name", "attribute_name", "attribute_value")</code>
<code>setdata</code>	覆盖一个大对象 (LOB) 的部分或全部。 <code>setdata(locator_name, offset_value, new_value)</code>
<code>show_cached_plan_in_xml</code>	以 XML 格式显示语句高速缓存中的查询的执行查询计划。 <code>show_cached_plan_in_xml(statement_id, plan_id, [level_of_detail])</code>
<code>show_dynamic_params_in_xml</code>	以 XML 格式返回查询的文本。 <code>show_dynamic_params_in_xml(object_id)</code>
<code>show_role</code>	显示当前登录的当前处于活动状态的系统定义角色。 <code>show_role()</code>
<code>show_sec_services</code>	列出可供会话使用的安全服务。 <code>show_sec_services()</code>



---

sign	<p>返回指定值的符号：1（正）、0 或 -1（负）。</p> <p><code>sign(<i>numeric</i>)</code></p>
sin	<p>返回指定角（以弧度表示）的正弦。</p> <p><code>sin(<i>approx_numeric</i>)</code></p>
sortkey	<p>生成可用于根据归类行为对结果进行排序的值，该函数允许使用缺省设置（基于拉丁字符的字典排序顺序以及区分大小写或变音）以外的字符归类行为。</p> <p><code>sortkey(<i>char_expression</i>   <i>uchar_expression</i>)[, {<i>collation_name</i>   <i>collation_ID</i>}]</code></p>
soundex	<p>对于由有效单字节或双字节罗马字母的邻接序列所组成的字符串，它将返回四个字符的 <b>soundex</b> 代码。</p> <p><code>soundex(<i>char_expr</i>   <i>uchar_expr</i>)</code></p>
space	<p>返回由指定数量的单字节空格所组成的字符串。</p> <p><code>space(<i>integer_expr</i>)</code></p>
spid_instance_id	<p>（仅限集群环境）返回运行指定进程 ID (spid) 的实例 ID。</p> <p><code>spid_instance_id(<i>spid_value</i>)</code></p>
square	<p>计算表示为 <b>float</b> 的指定值的平方值。</p> <p><code>square(<i>numeric_expression</i>)</code></p>
sqrt	<p>计算指定数字的平方根。</p> <p><code>sqrt(<i>approx_numeric</i>)</code></p>
stddev	<p>计算包含数值表达式的样本的标准偏差，数据类型为 <b>double</b>。stddev 和 stdev 是 stddev_samp 的别名。</p>
stdev	<p>计算包含数值表达式的样本的标准偏差，数据类型为 <b>double</b>。stddev 和 stdev 是 stddev_samp 的别名。</p>
stddevp	<p>计算包含数值表达式的总体的标准偏差，数据类型为 <b>double</b>。stddevp 是 stddev_pop 的别名，并使用相同的语法。</p>
stddev_pop	<p>计算包含数值表达式的总体的标准偏差，数据类型为 <b>double</b>。</p> <p><code>stddev_pop ( [ all   distinct ] <i>expression</i> )</code></p>
stddev_samp	<p>计算包含数值表达式的样本的标准偏差，数据类型为 <b>double</b>。</p> <p><code>stddev_samp ( [ all   distinct ] <i>expression</i> )</code></p>
str	<p>返回指定数字的等值字符，并将具有字符或数值的输出填充到指定长度。</p> <p><code>str(<i>approx_numeric</i>[, <i>length</i> [, <i>decimal</i>]])</code></p>

str_replace	<p>将第一个字符串表达式 (<i>string_expression1</i>) 中出现的第二个字符串表达式 (<i>string_expression2</i>) 的所有实例替换为第三个表达式 (<i>string_expression3</i>)。</p> <p><code>str_replace("string_expression1", "string_expression2", "string_expression3")</code></p>
strtobin	<p>将字母数字字符序列转换为等值的十六进制数字。</p> <p><code>select strtobin("string of valid alphanumeric characters")</code></p>
stuff	<p>返回通过以下方法形成的字符串：从一个字符串中删除指定数量的字符，然后将这些字符替换为另一个字符串。</p> <p><code>stuff(char_expr1   uchar_expr1, start, length, char_expr2   uchar_expr2)</code></p>
substring	<p>返回通过从另一字符串中提取指定数量的字符所形成的字符串。</p> <p><code>substring(expression, start, length)</code></p>
sum	<p>返回值的总和。</p> <p><code>sum([all   distinct] expression)</code></p>
suser_id	<p>从 <code>syslogins</code> 表中返回服务器用户的 ID 号。</p> <p><code>suser_id([server_user_name])</code></p>
suser_name	<p>返回当前服务器用户的名称，或已指定服务器 ID 的用户的名称。</p> <p><code>suser_name([server_user_id])</code></p>
syb_quit	<p>终止连接。</p> <p><code>syb_quit()</code></p>
syb_sendmsg	<p>（仅限 UNIX）将消息发送到用户数据报协议 (UDP) 端口。</p> <p><code>syb_sendmsg ip_address, port_number, message</code></p>
sys_tempdbid	<p>（仅限集群环境）返回指定的实例的有效本地系统临时数据库的 ID。如果未指定 <i>instance_id</i>，则返回当前实例的有效本地系统临时数据库的 ID。</p> <p><code>sys_tempdbid(instance_id)</code></p>
tan	<p>计算指定角（以弧度表示）的正切。</p> <p><code>tan(angle)</code></p>
tempdb_id	<p>报告给定会话分配到的临时数据库。 <code>tempdb_id</code> 函数的输入是服务器进程 ID，其输出是将进程分配到的临时数据库。如果未提供服务器进程，则 <code>tempdb_id</code> 报告分配给当前进程的临时数据库的 <code>dbid</code>。</p> <p><code>tempdb_id()</code></p>
textptr	<p>返回一个指针，指向 <code>text</code>、<code>image</code> 或 <code>unitext</code> 列的第一页。</p> <p><code>textptr(column_name)</code></p>

---

textvalid	指向指定的 text、unitext、行内和行外 LOB 列的指针有效时返回 1；无效时返回 0。  textvalid("table_name.column_name", textpointer)
to_unichar	返回具有指定整数表达式值的 unichar 表达式。  to_unichar(integer_expr)
tran_dumpable_status	返回一个指明是否允许 dump transaction 的 true/false 值。  tran_dumpable_status("database_name")
tsequal	比较 timestamp 的值，以防止对某个在为浏览而被选中以来做过修改的进行进行更新。  tsequal(browsed_row_timestamp, stored_row_timestamp)
uhighsurr	如果 start 位置的 Unicode 值是代理对的上半部（应首先出现在对中），则返回 1。否则返回 0。该函数允许编写用于处理代理的显式代码。  uhighsurr(uchar_expr, start)
ulowsurr	如果 start 处的 Unicode 值是代理对的下半部（应出现在对中的第二部分），则返回 1。否则返回 0。该函数允许在 substr()、stuff() 和 right() 所执行的调整周围显式代码。  ulowsurr(uchar_expr, start)
upper	将指定的小写字符串转换为等值大写字符串。  upper(char_expr)
uscalar	返回表达式第一个 Unicode 字符的 Unicode 标量值。  uscalar(uchar_expr)
used_pages	报告表、索引或特定分区所用的页数。与 data_pages 不同，used_pages 还包括用于内部结构的页。此函数替换 Adaptive Server 15.0 之前的版本中使用的 used_pgs 函数。  used_pages(dbid, object_id[, indid[, ptnid]])
user	返回当前用户的名称。  user
user_id	返回数据库中指定用户或当前用户的 ID 号。  user_id([user_name])
user_name	返回指定用户或当前用户在数据库中的名称。  user_name([user_id])

<code>valid_name</code>	<p>如果指定字符串不是有效标识符，则返回 0；如果该字符串是有效标识符，则返回非 0 数字，且长度最大可以为 255 个字节。</p> <p><code>valid_name(character_expression[, maximum_length])</code></p>
<code>valid_user</code>	<p>如果指定的 ID 至少在一个数据库中是有效用户或别名，则返回 1。</p> <p><code>valid_user(server_user_id [, database_id])</code></p>
<code>var</code>	<p>计算包含数值表达式的样本的统计方差，数据类型为 <code>double</code>，并返回一组数值的方差。<code>var</code> 和 <code>variance</code> 是 <code>var_samp</code> 的别名。</p>
<code>var_pop</code>	<p>计算包含数值表达式的总体的统计方差，数据类型为 <code>double</code>。<code>varp</code> 是 <code>var_pop</code> 的别名，并使用相同的语法。</p> <p><code>var_pop ( [all   distinct] expression )</code></p>
<code>var_samp</code>	<p>计算包含数值表达式的样本的统计方差，数据类型为 <code>double</code>，并返回一组数值的方差。<code>var</code> 和 <code>variance</code> 是 <code>var_samp</code> 的别名，并使用相同的语法。</p> <p><code>var_samp ( [ all   distinct] expression )</code></p>
<code>variance</code>	<p>计算包含数值表达式的样本的统计方差，数据类型为 <code>double</code>，并返回一组数值的方差。<code>var</code> 和 <code>variance</code> 是 <code>var_samp</code> 的别名。</p>
<code>varp</code>	<p>计算包含数值表达式的总体的统计方差，数据类型为 <code>double</code>。<code>varp</code> 是 <code>var_pop</code> 的别名。</p>
<code>workload_metric</code>	<p>（仅限集群环境）查询指定实例的当前工作负荷指标，或更新指定实例的指标。</p> <p><code>workload_metric( instance_id   instance_name [, new_value ] )</code></p>
<code>xa_bqual</code>	<p>返回 ASCII XA 事务 ID 的 <code>bqual</code> 组成部分的二进制对应值。</p> <p><code>xa_bqual(xid, 0)</code></p>
<code>xa_gtrid</code>	<p>返回 ASCII XA 事务 ID 的 <code>gtrid</code> 组成部分的二进制对应值。</p> <p><code>xa_gtrid(xactname, int)</code></p>
<code>xact_connmigrate_check</code>	<p>（仅限集群环境）确定当前连接是否可以处理外部事务。</p> <p><code>xact_connmigrate_check("txn_name")</code></p>
<code>xact_owner_instance</code>	<p>（仅限集群环境）返回正在运行分布式事务的实例 ID。</p> <p><code>xact_owner_instance(XID)</code></p>
<code>year</code>	<p>返回指定日期的 <code>datepart</code> 中表示年份的整数。</p> <p><code>year(date_expression)</code></p>

---

## XML 函数

有关在数据库中支持 XML 的 Transact-SQL 函数的语法参数、示例和用法信息，请参见《XML 服务》。

### xmlextract

将 XML 查询表达式应用到 XML 文档并返回指定的结果。返回的信息中可以带有或不带有 XML 标记。

```
xmlextract_expression ::=  
    xmlextract (xml_query_expression,xml_data_expression  
        [optional_parameters])  
xml_query_expression ::= basic_string_expression  
xml_data_expression ::= general_string_expression  
optional_parameters ::=  
    options_parameter  
    | returns_type  
    | options_parameter returns_type  
options_parameter ::= [,] option option_string  
returns_type ::= [,] returns_datatype  
datatype ::= {string_type | computational_type | date_time_type }  
string_type ::= char (integer) | varchar (integer)  
                | unichar (integer) | univarchar (integer)  
                | text | unitext | image  
computational_type ::= integer_type | decimal_type | real_type  
                | date_time_type  
integer_type ::= [ unsigned ] {integer | int | tinyint | smallint | bigint}  
decimal_type ::= {decimal | dec | numeric } [ (integer [, integer ] ) ]  
real_type ::= real | float | double precision  
date_time_type ::= date | time | datetime  
option_string ::= [,] basic_string_expression
```

### xmlparse

分析作为参数传递的 XML 文档，并返回包含经过分析的文档的 image（缺省值）、binary 或 varbinary 值。

```
mlparse_call ::=  
    xmlparse (general_string_expression  
        [options_parameter] [returns_type])  
options_parameter ::= [,] option option_string  
option_string ::= basic_string_expression  
returns_type ::= [,] returns {image | binary | varbinary [(integer )]}
```

### xmlrepresentation

检查表达式的 image 参数，并返回一个整数，此整数指示该参数是否包含经过分析的 XML 数据或其它类型的 image 数据。

```
xmlrepresentation_call ::=  
    xmlrepresentation (parsed_xml_expression)
```

xmltable

从 XML 文档中提取数据，并以 SQL 表的形式返回该数据。

```

xmltable_expression ::= xmltable
    ( row_pattern passing xml_argument
      columns column_definitions
      options_parameter )
row_pattern ::= character_string_literal
xml_argument ::=
    xml_expression | column_reference | variable_reference
column_definitions ::=
    column_definition [ { , column_definition } ]
    column_definition ::=
        ordinality_column | regular_column

ordinality_column ::= column_name datatype for ordinality
regular_column ::=
    column_name datatype [ default literal ] [ null | not null ]
    [ path column_pattern ]
column_pattern ::= character_string_literal
options_parameter ::= [,] option option_string
options_string ::= basic_string_expression

```

(派生表语法) 从 SQL from 子句中返回 SQL 表。

```

from_clause ::= from table_reference [, table_reference]...
table_reference ::= table_view_name | ANSI_join | derived_table
table_view_name ::= 请参见 《参考手册：第二卷》的“命令”中的
select 命令。

```

ANSI\_join ::= 请参见 《参考手册：第二卷》的“命令”中的 select 命令。

```

derived_table ::=
    (subquery) as table_name [ (column_name [, column_name]...) ]
    xmltable_expression as table_name

```

xmltest

是对 XML 查询表达式求值的 SQL 谓词，它可以引用 XML 文档参数，并将返回布尔结果。xmltest 类似于 SQL like 谓词。

```

xmltest_predicate ::=
    xml_query_expression [not] xmltest xml_data
    [option option_string]
xml_data ::=
    xml_data_expression | (xml_data_expression)
xml_query_expression ::= basic_string_expression
xml_data_expression ::= general_string_expression
option_string ::= basic_string_expression

```

---

xmlvalidate

验证 XML 文档。

```
xmlvalidate_call ::=  
    xmlvalidate ( general_string_expression, [optional_parameters])  
optional_parameters ::= options_parameter  
    | returns_type  
    | options_parameter returns type  
options_parameter ::= [,] option option_string  
options_string ::= basic_string_expression  
returns_type ::= [,] returns string_type  
string_type ::= char (integer) | varchar (integer)  
    | unichar (integer) | univarchar (integer)  
    | text | unitext | image | java.lang.String
```

## 命令

以下是 Adaptive Server 命令的简要说明和语法。有关完整信息，请参见《参考手册：命令》。

### alter database

使用 `log off` 可减少分配给数据库日志以及存档数据库的修改页部分的空间量。

```
alter database database_name
  [on {default | database_device} [= size]
    [, database_device [= size]]...]
  [log on {default | database_device} [= size]
    [, database_device [= size]]...]
  set { [durability = { no_recovery | at_shutdown | full}]
    [[.] dml_logging = {full | minimal} ]
    [[.] template = { database_name | NULL}]
    [, compression = {none | row | page}]
    [, lob_compression = {compression_level | off}]
    [, inrow_lob_length = value [log off database_device
      [= size | [from logical_page_number] [to logical_page_number]]
    [, database_device
      [= size | [from logical_page_number] [to logical_page_number]]
    [with override]
    [for load]
    [for proxy_update]
```

### alter encryption key

（加密列）更改当前口令、添加和删除密钥副本、重新生成加密密钥。改变主密钥：

```
alter encryption key [dual] master
  with <char_string>
  { add encryption
    {with passwd <char_string>
      for user <user_name> [for recovery]
    | for automatic_startup
    }
  | modify encryption
    { with passwd <char_string>
      [for recovery]
    | for automatic_startup
    }
  | drop encryption
    { for user <user_name>
    | for recovery
    | for automatic_startup
    }
  | regenerate key
    [ with passwd <char_string>]
  | recovery encryption
    with passwd <char_string>
```



```
| modify owner <user_name>
}
```

改变 syb\_extpasswdkey 服务密钥:

```
alter encryption key syb_extpasswdkey
[ with { static key | master key}
  { regenerate key
    [ with { static key | master key }}
  | modify encryption
    [ with { static key | master key }} }
```

改变列加密密钥:

```
alter encryption key [[database.][owner].] keyname
{ [ as | not default ]
[dual] master
  [ with { static key | master key} ]
  regenerate key
  [ with { static key | master key [no] dual_control} ] | [with passwd
'password' | system_encr_passwd | login_passwd |
'base_key_password']
modify encryption
  [ with {passwd {'password' | system_encr_passwd |
    login_passwd } | master key }}
  [[no] dual_control]
  for automatic startup
add encryption [ with passwd 'password' | 'key_copy_password']
  for user user_name
  [for [login_association | recovery | automatic_startup]]
drop encryption for
  { user user_name | recovery
  [ for recovery ] | [ for automatic_startup ]}
  | [ with passwd 'password' ]
recover encryption with passwd 'password'
  | modify owner user_name
}
```

alter login

更改登录帐户的属性。

```
alter login login_name
{ [modify attribute_value_pair_list ]
| [add auto activated roles role_name [, role_name_list ]]
| [drop auto activated roles { ALL | role_name [, role_name_list ]}]
| [drop attribute_name_list ]
| [ with password caller_password
modify password [immediately] new_loginName_password ] }
```

alter login profile

更改登录配置文件的属性。

```
alter login profile login_profile_name
{ [as [ not ] default ]
| [modify attribute_value_pair_list ]
| [add auto activated roles role_name [, role_name_list ]]
```

	<pre>   [drop auto activated roles { ALL   <i>role_name</i> [, <i>role_name_list</i> ]}]   [drop <i>attribute_name_list</i> ] </pre>
alter...modify owner	<p>将数据库对象的所有权从一个所有者移交给另一个所有者。</p> <pre> alter { <i>object_type</i>   all } [<i>owner.</i>]{<i>object_name</i>   * } modify owner { <i>name_in_db</i>   loginname only <i>login_name</i> } [ preserve permissions ] </pre>
alter role	<p>定义角色之间的互斥关系；为角色添加、删除和更改口令；指定口令有效期、最小口令长度，以及指定的角色允许的最大登录尝试失败次数。 alter role 也用于锁定和解锁角色。</p> <pre> alter role <i>role1</i> {add   drop} exclusive {membership   activation} <i>role2</i>  alter role <i>role_name</i> [add passwd "<i>password</i>"   drop passwd] [lock   unlock]  alter role {<i>role_name</i>   "all overrides"} set {passwd expiration   min passwd length   max failed_logins} <i>option_value</i> </pre>
alter table	<p>向表添加新列；删除或修改现有的列；添加、更改或删除约束；更改现有表的属性；启用或禁用表上的触发器；更改表的压缩级别。</p> <p>支持添加、删除和修改计算列，并允许更改现有计算列的实现属性、可为空性或定义。</p> <p>用指定分区策略对表分区或重新分区，或对具有现有分区的表添加分区。</p> <pre> alter table [[<i>database.</i>][<i>owner.</i>]<i>table_name</i> {add <i>column_name datatype</i>} [default {<i>constant_expression</i>   user   null}] {identity   null   not null [not materialized]} [off row   in row] [[constraint <i>constraint_name</i>] {{unique   primary key} [clustered   nonclustered] [asc   desc] [with {fillfactor = <i>pct</i>, max_rows_per_page = <i>num_rows</i>, reservepagegap = <i>num_pages</i>} [on <i>segment_name</i>]   references [[<i>database.</i>]<i>owner.</i>]<i>ref_table</i> [(<i>ref_column</i>)] [match full]   check (<i>search_condition</i>)] [encrypt [with [<i>database.</i>]<i>owner.</i>] <i>keyname</i>] [decrypt_default {<i>constant_expression</i>   null}]] [compressed = <i>compression_level</i>   not compressed] [, <i>next_column</i>]...  add [constraint <i>constraint_name</i>] {unique   primary key} [clustered   nonclustered] (<i>column_name</i> [asc   desc][, <i>column_name</i> [asc   desc]...]) [with {fillfactor = <i>pct</i>, max_rows_per_page = <i>num_rows</i>, reservepagegap = <i>num_pages</i>}] [on <i>segment_name</i>]   foreign key (<i>column_name</i> [{, <i>column_name</i>}...]) references [[<i>database.</i>]<i>owner.</i>]<i>ref_table</i> [(<i>ref_column</i> [{, <i>ref_column</i>}...])] [match full] </pre>

---

```

| add lob-colname { text | image | unitext }
    [null] [ in row [ (length) ] ]
| check (search_condition)
| set dml_logging = {full | minimal | default} |
    [, compression = {none | page | row}}
    [lob_compression = off | compression_level]
| drop {column_name [, column_name]...
    | constraint constraint_name} | modify column_name
    [datatype [null | not null]]
    [[[encrypt [with keyname] [decrypt_default [value]] | decrypt]
    [[not] compressed]
    [compressed = compression_level | not compressed]
    [, next_column]...] | replace column_name
    default {constant_expression | user | null}
    | decrypt_default {constant_expression | null}
    | drop decrypt_default
lock {allpages | datarows | datapages} }
| with exp_row_size=num_bytes
    transfer table [on | off]} | partition number_of_partitions
| unpartition | partition_clause | add_partition_clause

```

用于分区的 alter table 语法:

```

partition_clause::=
    partition by range (column_name [, column_name]...)
        ([partition_name] values <= ({constant | MAX}
            [, {constant | MAX}] ...) [on segment_name]
            [compression_clause] [on segment_name]
            [, [partition_name] values <= ({constant | MAX}
                [, {constant | MAX}] ...) [on segment_name]...]

| partition by hash (column_name [, column_name]...)
    { (partition_name [on segment_name]
        [, partition_name [on segment_name]...]
        [compression_clause] [on segment_name]
        | number_of_partitions
        [on (segment_name [, segment_name] ...)]]

| partition by list (column_name)
    ([partition_name] values (constant [, constant] ...)
        [on segment_name]
        [compression_clause] [on segment_name]
        [, [partition_name] values (constant [, constant] ...)
            [on segment_name]])

| partition by roundrobin
    { (partition_name [on segment_name]
        [, partition_name [on segment_name]...]
        [compression_clause] [on segment_name]

```

```

| number_of_partitions
  [on (segment_name [, segment_name]...)]}

add_partition_clause::=
  add partition
    { ([partition_name] values <= ({constant | MAX}
      [, {constant | MAX}]...))
      [on segment_name]
      [compression_clause] [on segment_name]
      [, [partition_name] values <= ({constant | MAX}
        [, {constant | MAX}] ...))
        [on segment_name]...)}
  | modify partition {partition_name [, partition_name ...]}
  set compression [= {default | none | row | page}]

  | ([partition_name] values (constant[, constant] ...))
    [on segment_name]
    [, [partition_name] values (constant[, constant] ...))
      [on segment_name] ...)}

```

用于计算列的 alter table 语法:

```

alter table
  add column_name {compute | as}
    computed_column_expression...
    [materialized | not materialized]
  drop column_name
  modify column_name {null | not null |
    {materialized | not materialized} [null | not null] |
    {compute | as} computed_column_expression
    [materialized | not materialized]
    [null | not null]}

```

用于删除分区的 alter table 语法:

```

alter table table_name drop partition
  partition_name [, partition_name]...

```

alter thread pool

改变线程池。

```

alter thread pool pool_name with { pool name = "new_name"
  thread count = thread_count,
  [pool description = "description"]}
  [idle timeout = time_period]

```

begin...end

包含一系列 SQL 语句，以使控制流语言（例如 if...else）可以影响整个组的性能。

```

begin
  statement block
end

```

---

begin transaction	<p>标记用户定义的事务的起点。</p> <pre>begin tran[saction] [<i>transaction_name</i>]</pre>
break	<p>导致从 while 循环中退出。break 一般通过 if 测试激活。</p> <pre>while <i>logical_expression</i>     <i>statement</i> break     <i>statement</i> continue</pre>
checkpoint	<p>将所有脏页（自上次写入后已更新的页）写入数据库设备。</p> <pre>checkpoint [all   [<i>dbname</i>[, <i>dbname</i>, <i>dbname</i>, .....]]</pre>
close	<p>使游标失效。</p> <pre>close <i>cursor_name</i></pre>
commit	<p>标记用户定义的事务的终点。</p> <pre>commit [tran   transaction   work] [<i>transaction_name</i>]</pre>
compute clause	<p>生成在查询结果中显示为附加行的摘要值。</p> <pre><i>start_of_select_statement</i> compute <i>row_aggregate</i> (<i>column_name</i>)     [, <i>row_aggregate</i> (<i>column_name</i>)]... [by <i>column_name</i> [, <i>column_name</i>]]...</pre>
connect to...disconnect	<p>（仅限组件集成服务）连接到指定服务器和断开已连接的服务器。</p> <p>将此语法与 CIS 一起使用，可创建一个至另一服务器的直通连接。</p> <pre>connect to <i>server_name</i> disconnect     [from ASE]     [all]     [<i>connection_name</i>]</pre> <p>打开与 Adaptive Server 的新 JDBC 级连接，且不使用 CIS。可按任意顺序指定参数。如果不包括参数，Adaptive Server 会提示您提供连接参数。</p> <pre>connect     [to ASE <i>engine_name</i>]     [database <i>database_name</i>]     [as <i>connection_name</i>]     [user <i>user_id</i>]     [identified by password]]]</pre> <p>打开与 Adaptive Server 的新 JDBC 级连接。此语法不使用 CIS：</p> <pre>connect using <i>connect_string</i></pre>

continue

重新启动 while 循环。continue 一般通过 if 测试激活。

```
while boolean_expression
    statement
    break
    statement
continue
```

创建存档数据库

创建存档数据库。

```
create archive database db_name
[on db_device [= size]
  [, db_device [= size] ] ...]
with scratch_database = db_name
```

create database

创建新的数据库。

非集群环境：

```
create [inmemory] [temporary] database database_name
[use database_name as template]
[on {default | database_device} [= size]
  [, database_device [= size]]...]
[log on database_device [= size]
  [, database_device [= size]]...]
[with {dbid = number, default_location = "pathname", override}}
  | [{.]durability = { no_recovery
    | at_shutdown
    | full} ]
  [, compression = {none | row | page}}
  [, lob_compression = {compression_level | off}]
  [ [, inrow_lob_length = value ] }...]
[for {load | proxy_update}]
```

集群环境：

```
create [ [ global | system ] temporary ] database database_name
[ for instance instance_name ]
[on {default | database_device} [= size]
  [, database_device [= size]]...]
[log on database_device [= size]
  [, database_device [= size]]...]
[with {override | default_location = "pathname"}}
[for {load | proxy_update}]
```

create default

如果插入时没有显式地提供要插入列（或用户定义的数据类型的所有列）中的值，则指定一个值。

```
create default [owner.]default_name
as constant_expression
```

---

## create encryption key

创建加密密钥。所有与密钥和加密相关的信息都由 `create encryption key` 封装，从而允许您在加密过程中指定加密算法和密钥大小、密钥的缺省属性、用于加密密钥的可选的用户指定口令，以及是使用初始化矢量还是填充。

创建主密钥：

```
create encryption key [dual] master
    [for AES] with passwd char_literal
```

创建服务密钥：

```
create encryption key syb_extpasswdkey
    [ with { static key | master key } ]
create encryption key syb_syscommkey
    [ with { static key | master key } ]
```

创建列加密密钥：

```
create encryption key [[database.][owner].]keyname
    [as default]
    [for algorithm]
    [with [{passwd {char_literal | system_encr_passwd} | master key}]
    [key_length num_bits]
    [init_vector {null | random}]
    [pad {null | random}]
    [[no] dual_control]]
```

## create existing table

（仅限组件集成服务）创建代理表，然后检索和存储远程表中的元数据，并将数据放到该代理表中。允许将代理表映射到远程位置上的表、视图或过程上。

```
create existing table table_name (column_list)
    [on segment_name]
    [[external {table | procedure | file | connection_type} at pathname
    [column delimiter "string"]]
```

创建代理表的首选方法是使用 `create proxy_table` 命令，此命令无需定义列定义。

## create function

创建一个用户定义的函数，该函数是返回指定值的已保存 Transact-SQL 例程。

```
create function [ owner_name. ] function_name
    [ ( @parameter_name [as] parameter_datatype [ = default ]
    [ ,...n ] ) ]
    returns return_datatype
    [ with recompile ]
    as
    [begin]
    function_body
    return scalar_expression
    [end]
```

**create function (SQLJ)** 通过将 SQL 包装加入到 Java 静态方法中，创建用户定义的函数。可返回一个由该方法定义的值。

```
create function [owner.]sql_function_name
    ([ sql_parameter_name sql_datatype
      [(length)| (precision[, scale ])]
    [, sql_parameter_name sql_datatype
      [(length)| (precision[, scale ])]
    ...])
returns sql_datatype
    [(length)| (precision[, scale ])]
[modifies sql data]
[returns null on null input |
  called on null input]
[deterministic | not deterministic]
[exportable]
language java
parameter style java
external name 'java_method_name
    [(java_datatype[, java_datatype
    ...])]'
```

**create index** 创建表中一个或多个计算列或非计算列的索引。创建分区索引。

允许将计算列像普通列一样用作索引键，并创建基于函数的索引。基于函数的索引具有一个或多个用作索引键的表达式。现有的 **create index** 语法可创建计算列的索引，但基于函数的索引需要额外的语法。

```
create [unique] [clustered | nonclustered] index index_name
on [[database.]owner.]table_name
    (column_expression [asc | desc]
    [, column_expression [asc | desc]]...)
[with {fillfactor = pct,
      max_rows_per_page = num_rows,
      reservepagegap = num_pages,
      consumers = x, ignore_dup_key, sorted_data,
      [ignore_dup_row | allow_dup_row],
      statistics using num_steps values}]
[on segment_name]
[index_partition_clause]
```

创建索引分区：

```
index_partition_clause::=
[local index [partition_name [on segment_name]
[, partition_name [on segment_name]...]]]
```

创建基于函数的索引：

```
create [unique | nonclustered] index index_name
on [[database.]owner.] table_name
    (column_expression [asc | desc]
    [, column_expression [asc | desc]]...
```



---

create login	<p>创建登录帐户；指定口令、帐户的登录配置文件，以及为帐户分配的用户提供的参数。</p> <pre>create login <i>login_name</i> with [encrypted]     password <i>password</i>     [<i>attribute_value_pair_list</i>]</pre>
create login profile	<p>用指定的属性创建登录配置文件。</p> <pre>create login profile <i>login_profile_name</i>     [ as default ]     [ with { attributes from <i>login_name</i>   <i>attribute_value_pair_list</i> } ]</pre>
create plan	<p>创建抽象计划。</p> <pre>create plan <i>query plan</i>     [into <i>group_name</i>]     [and set @<i>new_id</i>]</pre>
create procedure	<p>创建存储过程或可以带一个或多个用户提供的参数的扩展存储过程 (ESP)。</p> <pre>create procedure [<i>owner</i>.]<i>procedure_name</i>[:<i>number</i>]     [[(<i>@parameter_name datatype [(length)   (precision [, scale])]</i>     [= default][output]     [, <i>@parameter_name datatype [(length)   (precision [, scale])]</i>     [= default][output]...)]]]     [with recompile]     as {<i>SQL_statements</i>   external name <i>dll_name</i>}</pre>
create procedure (SQLJ)	<p>通过将 SQL 包装加入到 Java 静态方法中，创建 SQLJ 存储过程。可以接受用户提供的参数并返回结果集和输出参数。</p> <pre>create procedure [<i>owner</i>.]<i>sql_procedure_name</i>     ([[<i>in</i>   <i>out</i>   <i>inout</i>] <i>sql_parameter_name</i>     <i>sql_datatype [(length)  </i>     (<i>precision</i>[, <i>scale</i>])]     [=default]     ...])     [, [<i>in</i>   <i>out</i>   <i>inout</i>] <i>sql_parameter_name</i>     <i>sql_datatype [(length)  </i>     (<i>precision</i>[, <i>scale</i>])]     [=default]     ...])     [modifies sql data]     [dynamic result sets <i>integer</i>]     [deterministic   not deterministic]     language java     parameter style java     external name '<i>java_method_name</i>     [([<i>java_datatype</i>[, <i>java_datatype</i>     ...]])]'</pre>

**create proxy\_table** (仅限组件集成服务) 创建代理表而不指定列的列表。 CIS 从它自远程表获得的元数据中派生出列列表。

```
create proxy_table table_name
[external [table | directory | file]]
at pathname
[column delimiter "<string>"]
```

**create role** 创建用户定义角色；指定在创建特定角色时所允许的口令有效期、最小口令长度和最大失败登录次数。也可以在创建角色时将一个口令与该角色关联。

```
create role role_name [with passwd "password"
[, {passwd expiration | min passwd length |
max failed_logins} option_value]]
```

**create rule** 为特定列或属于用户定义数据类型的任意列指定可接受值的域并创建访问规则。

```
create [[and | or] access]] rule
[owner.]rule_name
as condition_expression
```

**create schema** 为数据库用户创建一个表、视图和权限的新集合。

```
create schema authorization authorization_name
create_object_statement
[create_object_statement ...]
[permission_statement ...]
```

**create service** 使用指定的名称和参数将提供的 SQL 语句包装为一个存储过程。

```
create service service-name [secure security_options] [, userpath path]
[, alias alias-name]
type { xml | raw | soap }
[[(@parameter_name datatype [(length) | (precision [, scale ])]
[= default][output]
[, @parameter_name datatype [(length) | (precision [, scale ])]
[= default][output]]...[])]
as SQL_statements

security_options ::= (security_option_item [security_option_item])
```

**create table** 创建新表和 (可选) 完整性约束。

定义计算列。

定义表、行和分区的压缩级别。

定义加密列及其解密缺省值。

```
create table [[database.[owner].]table_name (column_name datatype
[default {constant_expression | user | null}]
[{identity | null | not null}]
[ in row [(length)] | off row ]
```

---

```

[[constraint constraint_name]
{{unique | primary key}
[clustered | nonclustered] [asc | desc]
[with {fillfactor = pct,
      max_rows_per_page = num_rows,
      reservepagegap = num_pages]
dml_logging = {full | minimal}}
[on segment_name]
| references [[database.]owner.]ref_table
  [(ref_column)]
  [match full]
  | check (search_condition)}}]
[encrypt [with [database.]owner.]key_name]
[decrypt_default constant_expression | null]]
[not compressed]
[compressed = {compression_level | not compressed}]
[[constraint [[database.]owner.]key_name]
{unique | primary key}
[clustered | nonclustered]
(column_name [asc | desc]
  [{, column_name [asc | desc]}...])
[with {fillfactor = pct
      max_rows_per_page = num_rows,
      reservepagegap = num_pages}]
[on segment_name]
| foreign key (column_name [{, column_name}...])
references [[database.]owner.]ref_table
  [(ref_column [{, ref_column}...])]
  [match full]
  | check (search_condition) ...}
[{, {next_column | next_constraint}...}]
[lock {datarows | datapages | allpages}]
[with {max_rows_per_page = num_rows,
      exp_row_size = num_bytes,
      reservepagegap = num_pages,
      identity_gap = value
      transfer table [on | off]
      dml_logging = {full | minimal}
      compression = {none | page | row}}]
lob_compression = off | compression_level
[on segment_name]
[partition_clause]
[ [ external table ] at pathname ]
[for load]
compression_clause::=
  with compression = {none | page | row}

```

将此语法用于分区：

```
partition_clause ::=
    partition by range (column_name[, column_name]...)
        ([partition_name] values <= ({constant | MAX}
            [, {constant | MAX}] ...)
            [compression_clause] [on segment_name]
            [, [partition_name] values <= ({constant | MAX}
            [, {constant | MAX}] ...)
            [compression_clause] [on segment_name]]...)

    | partition by hash (column_name[, column_name]...)
        { (partition_name
            [compression_clause] [on segment_name]
            [, partition_name
            [compression_clause] [on segment_name]]...)
        | number_of_partitions
            [on (segment_name[, segment_name] ...)]}

    | partition by list (column_name)
        ([partition_name] values (constant[, constant] ...)
            [compression_clause] [on segment_name]
            [, [partition_name] values (constant[, constant] ...)
            [compression_clause] [on segment_name]] ...)

    | partition by roundrobin
        { (partition_name [on segment_name]
            [, partition_name
            [compression_clause] [on segment_name]]...)
        | number_of_partitions
            [on (segment_name[, segment_name]...)]}
```

将此语法用于计算列

```
create table [[database.[owner].] table_name
    (column_name {compute | as}
        computed_column_expression
        [[materialized] [not compressed]] | [not materialized])
```

使用此语法创建虚拟散列表

```
create table [database.[owner].]table_name
...
    | {unique | primary key}
    using clustered
    (column_name [asc | desc] [{, column_name [asc | desc]}...] =
    (hash_factor [{, hash_factor}...])
    with max num_hash_values key
```

---

create thread pool

创建用户定义的线程池。

```
create thread pool pool_name with thread count = count
[, pool description = description ]
[idle timeout = time_period]
```

create trigger

创建触发器，它是一种经常用来强制实现完整性约束的存储过程。当用户试图对指定的表使用指定的数据修改语句时，就会自动执行触发器。

```
create trigger [owner.]trigger_name
on [owner.]table_name
for {insert , update} | instead of {insert, update, delete}}
[as
    [if update (column_name)
        [{and | or} update (column_name)...]
        SQL_statements
    [if update (column_name)
        [{and | or} update (column_name)...]
        SQL_statements]...]
```

create view

创建一个视图，视图是查看一个或多个表中的数据的一种替代方法。

```
create view [owner.]view_name
[(column_name[, column_name]...)]
as
select [distinct] select_statement
[with check option]
```

dbcc

数据库一致性检查程序 (dbcc) 检查数据库的逻辑和物理一致性，并提供统计、计划和修复功能。

有些 dbcc 命令只适用于共享磁盘集群。请参见针对各集群分别列出的 dbcc 语法。

```
dbcc addtempdb (dbid | database_name)
dbcc checkalloc [(database_name[, fix | nofix])]
dbcc checkcatalog [(database_name[, fix])
dbcc checkdb [(database_name[, skip_ncindex])]
dbcc checkindex ({table_name | table_id}, index_id
[, bottom_up[, partition_name | partition_id]])
dbcc checkstorage [(database_name)]
dbcc checktable (table_name | table_id
[, skip_ncindex | fix_spacebits | "check spacebits" |
bottom_up | NULL[, partition_name | partition_id])
dbcc checkverify (dbname[, tblname[, ignore_exclusions]])
dbcc complete_xact (xid, [{"commit", "1pc"} | "rollback"])
dbcc dbrepair (database_name, dropdb)
dbcc engine ({offline, [enginenum] | "online"})
```

```

dbcc fix_text ({table_name | table_id})
dbcc forget_xact (xid)
dbcc indexalloc (table_name | table_id, index_id
    [, optimized | fast | NULL [, fix | nofix | NULL
    [, partition_name | partition_id]])
dbcc monitor (increment, <group name>)
dbcc monitor (decrement, <group name>)
dbcc monitor (reset, <group name>)
dbcc pravailabletempdbs
dbcc rebuild_text (table_name | table_id | "all" [, column [, text_page
    [, data_partition_name | data_partition_id]])
dbcc reindex ({table_name | table_id})
dbcc serverlimits
dbcc stackused
dbcc tablealloc (table_name | table_id [, full | optimized | fast | NULL
    [, fix | nofix | NULL [, data_partition_name | data_partition_id]])
dbcc textalloc (table_name | table_id [, full | optimized | fast | NULL
    [, fix | nofix | NULL [, data_partition_name | data_partition_id]])
dbcc {traceon | traceoff} (flag [, flag ...])
dbcc tune ({ascinserts, {0 | 1} , table_name |
    cleanup, {0 | 1} |
    cpuaffinity, start_cpu {, on | off} |
    des_greedyalloc, dbid, object_name,
    " {on | off}" | deviochar vdevno, "batch_size" |
    doneinproc {0 | 1}})
dbcc upgrade_object [ ( dbid | dbname
    [, [database.owner].compiled_object_name' |
    'check' | 'default' | 'procedure' | 'rule' |
    'trigger' | 'view'
    [, 'force' ] ] )

```

仅适用于集群的 dbcc 语法:

```

dbcc nodetraceon(trace_flag_number)
dbcc nodetraceoff(trace_flag_number)
dbcc set_scope_in_cluster("cluster"|"instance"|"scope")
dbcc quorum

```

deallocate cursor

使游标不可访问并释放所有提交到该游标的内存资源。

```
deallocate [cursor] cursor_name
```

deallocate locator

删除内存中存储的大对象 (LOB) 并使其 LOB 定位符无效。

```
deallocate locator locator_descriptor
```

---

declare

为批处理或过程声明局部变量的名称和类型。变量声明：

```
declare @variable_name datatype  
[, @variable_name datatype]...
```

变量赋值：

```
select @variable = {expression | select_statement}  
[, @variable = {expression | select_statement} ...]  
[from table_list]  
[where search_conditions]  
[group by group_by_list]  
[having search_conditions]  
[order by order_by_list]  
[compute function_list [by by_list]]
```

declare cursor

通过将 select 语句与游标名关联来定义游标。可将 declare cursor 用于存档数据库。

```
declare cursor_name  
[semi_sensitive | insensitive] [scroll | no scroll] [release_locks_on_close]  
cursor for select_statement  
[for {read only | update [of column_name_list]]]
```

delete

删除表中的行。

```
delete  
[top unsigned_integer]  
[from] [[database.]owner.]{view_name|table_name}  
[where search_conditions]  
[plan "abstract plan"]  
  
delete [[database.]owner.]{table_name | view_name}  
[from] [[database.]owner.]{view_name [readpast]  
table_name  
[(index {index_name | table_name}  
[prefetch size][lru|mru])]}  
[readpast]  
[, [[database.]owner.]{view_name [readpast]  
table_name  
[(index {index_name | table_name}  
[prefetch size][lru|mru])]  
[readpast]} ...]  
[where search_conditions]  
[plan "abstract plan"]  
  
delete [from] [[database.]owner.]{table_name|view_name}  
where current of cursor_name
```

delete statistics

从 sysstatistics 系统表中删除统计数据。

```
delete [shared] statistics table_name  
[partition data_partition_name]  
[(column_name[, column_name] ...)]
```

disk init	<p>使物理设备或文件可供 Adaptive Server 使用。</p> <pre> disk init   name = "device_name",   physname = { 'physical_name'   'cache_name' }   skip_alloc={true   false},   [vdevno = virtual_device_number,]   size = number_of_blocks   [, type = 'inmemory' ]   [, vstart = virtual_address       , cntrltype = controller_number]   [, dsync = {true   false}]   [, directio = {true   false}]   [, instance = "instance_name"] </pre>
disk mirror	<p>创建在主设备发生故障时立即接替运行的软件镜像。</p> <pre> disk mirror   name = "device_name",   mirror = "physicalname"   [, writes = {serial   noserial}]   [clear = {TRUE   FALSE}] </pre>
disk refit	<p>通过 sysdevices 中的信息重建 master 数据库的 sysusages 和 sysdatabases 系统表。</p> <pre> disk refit </pre>
disk reinit	<p>重建 master 数据库的 sysdevices 系统表。将 disk reinit 作为过程的一部分使用，以恢复 master 数据库。</p> <pre> disk reinit   name = "device_name",   physname = "physicalname" ,   [vdevno = virtual_device_number ,]   size = number_of_blocks   [, vstart = virtual_address       , cntrltype = controller_number]   [, dsync = {true   false}]   [, directio = {true   false}]   [, instance = "instance_name"] </pre>
disk remirror	<p>在磁盘镜像因镜像设备发生故障而停止或者被 disk unmirror 命令临时禁用后重新启动它。</p> <pre> disk remirror   name = "device_name" </pre>
disk resize	<p>动态增加 Adaptive Server 使用的设备的大小。</p> <pre> disk resize   name = "device_name",   size = additional_space </pre>



---

disk unmirror	挂起用 <code>disk mirror</code> 命令启动的磁盘镜像，以允许硬件维护或硬件设备的更改。  <pre>disk unmirror   name = "device_name"   [, side = {"primary"   secondary}]   [, mode = {retain   remove}]</pre>
drop database	从 Adaptive Server 中删除一个或多个数据库，包括存档数据库。  <pre>drop database database_name [, database_name] ...</pre>
drop default	删除用户定义的缺省值。  <pre>drop default [owner.]default_name   [, [owner.]default_name] ...</pre>
drop encryption key	允许密钥所有者删除命名的加密密钥。  <pre>drop encryption key [(database.[owner].)keyname]</pre>
drop function	从当前数据库中删除一个或多个用户定义的函数。  <pre>drop function{ [ owner_name .] function_name } [ ,...n ]</pre>
drop function (SQLJ)	删除 SQLJ 函数。  <pre>drop func[tion] [owner.]function_name   [, [owner.]function_name] ...</pre>
drop index	从当前数据库的表中删除索引。  <pre>drop index table_name.index_name   [, table_name.index_name] ...</pre>
drop login	删除登录帐户或帐户列表。  <pre>drop login login_name [, login_name_list] [ with override ]</pre>
drop login profile	删除登录配置文件或登录配置文件列表。  <pre>drop login profile login_profile_name [, login_profile_name_list]   [ with override ]</pre>
drop procedure	删除过程。  <pre>drop proc[edure] [owner.]procedure_name   [, [owner.]procedure_name] ...</pre>
drop role	删除用户定义的角色。  <pre>drop role role_name [with override]</pre>
drop rule	删除用户定义的规则。  <pre>drop rule [owner.]rule_name[, [owner.]rule_name] ...</pre>

drop service	<p>drop service 命令用于从当前数据库中删除用户定义的 Web 服务。元数据和对应的存储过程都将被删除。</p> <pre>drop service <i>service-name</i></pre>
drop thread pool	<p>删除用户定义的池。</p> <pre>drop thread pool <i>pool_name</i></pre>
drop table	<p>从数据库中删除表定义及其所有的数据、索引、分区属性、触发器、加密属性和权限。</p> <pre>drop table [[<i>database.</i>]owner.]<i>table_name</i> [, [[<i>database.</i>]owner.]<i>table_name</i>] ...</pre>
drop trigger	<p>删除触发器。</p> <pre>drop trigger [owner.]<i>trigger_name</i> [, [owner.]<i>trigger_name</i>] ...</pre>
drop view	<p>从当前数据库中删除一个或多个视图。</p> <pre>drop view [owner.]<i>view_name</i> [, [owner.]<i>view_name</i>] ...</pre>
dump database	<p>以一种可用 load database 命令读取的格式制作整个数据库（包括事务日志）的备份副本。转储和装载通过 Backup Server 执行。</p> <pre>dump database <i>database_name</i> to [compress::[compression_level::]]<i>stripe_device</i> [at <i>backup_server_name</i>] [density = <i>density_value</i>, blocksize = <i>number_bytes</i>, capacity = <i>number_kilobytes</i>, dumpvolume = <i>volume_name</i>, file = <i>file_name</i>] [with shrink_log] with verify[= header   full] [stripe on [compress::[compression_level::]]<i>stripe_device</i> [at <i>backup_server_name</i>] [density = <i>density_value</i>, blocksize = <i>number_bytes</i>, capacity = <i>number_kilobytes</i>, dumpvolume = <i>volume_name</i>, file = <i>file_name</i>]] [[stripe on [compress::[compression_level::]]<i>stripe_device</i> [at <i>backup_server_name</i>] [density = <i>density_value</i>, blocksize = <i>number_bytes</i>, capacity = <i>number_kilobytes</i>, dumpvolume = <i>volume_name</i>, file = <i>file_name</i>]]....] [with{ density = <i>density_value</i>, blocksize = <i>number_bytes</i>,</pre>

---

```

capacity = number_kilobytes,
compression = compress_level
dumpvolume = volume_name,
file = file_name,
[dismount | nodismount],
[nounload | unload],
passwd = password,
retaindays = number_days,
[noinit | init],
notify = {client | operator_console}
}}

```

(Tivoli Storage Manager) 使用此语法可在 Tivoli Storage Manager 提供备份服务时复制数据库。

```

dump database database_name
to "syb_tsm::object_name"
[blocksizes = number_bytes]
[stripe on "syb_tsm::object_name"
[blocksizes = number_bytes]]...
[with{
blocksizes = number_bytes,
compression = compress_level,
passwd = password,
[noinit | init],
notify = {client | operator_console},
verify[ = header | full]
} ]

```

dump transaction

制作事务日志的副本并删除不活动的部分。进行例行日志转储：

```

dump tran[saction] database_name
to [compress::[compression_level::]]stripe_device
[at backup_server_name]
[density = density_value,
blocksizes = number_bytes,
capacity = number_kilobytes,
dumpvolume = volume_name,
file = file_name]
[stripe on [compress::[compression_level::]]stripe_device
[at backup_server_name]
[density = density_value,
blocksizes = number_bytes,
capacity = number_kilobytes,
dumpvolume = volume_name,
file = file_name]]
[[stripe on [compress::[compression_level::]]stripe_device
[at backup_server_name]
[density = density_value,
blocksizes = number_bytes,
capacity = number_kilobytes,
dumpvolume = volume_name,

```

```

        file = file_name]]...]
[with{
    density = density_value,
    blocksize = number_bytes,
    capacity = number_kilobytes,
    compression = compress_level,
    dumpvolume = volume_name,
    file = file_name,
    [dismount | nodismount],
    [nounload | unload],
    retaindays = number_days,
    [noinit | init],
    notify = {client | operator_console},
    standby_access}}]

```

截断日志而不生成备份副本：

```

dump tran[saction] database_name
with truncate_only

```

截断已达到容量上限的日志。仅在不得已的情况下才应使用这种方法，因为您将失去日志的内容：

```

dump tran[saction] database_name
with no_log

```

在某个数据库设备出现故障后备份日志：

```

dump tran[saction] database_name
to [compress::[compression_level::]]stripe_device
[at backup_server_name]
[density = density_value,
blocksize = number_bytes,
capacity = number_kilobytes,
dumpvolume = volume_name,
file = file_name]
[stripe on [compress::[compression_level::]]stripe_device
[at backup_server_name]
[density = density_value,
blocksize = number_bytes,
capacity = number_kilobytes,
dumpvolume = volume_name,
file = file_name]]
[[stripe on [compress::[compression_level::]]stripe_device
[at backup_server_name]
[density = density_value,
blocksize = number_bytes,
capacity = number_kilobytes,
dumpvolume = volume_name,
file = file_name]]...]
[with{
    density = density_value,
    blocksize = number_bytes,

```

---

```

capacity = number_kilobytes,
compression = compress_level
dumpvolume = volume_name,
file = file_name,
[dismount | nodismount],
[nounload | unload],
retaindays = number_days,
[noinit | init],
no_truncate,
notify = {client | operator_console}}]

```

Tivoli Storage Manager 提供备份服务时复制事务日志。

```

dump transaction database_name
to "syb_tsm::object_name"
[blocksize = number_bytes]
[stripe on "[syb_tsm::]object_name"
[blocksize = number_bytes]]...]
[with{
blocksize = number_bytes,
compression = compress_level,
passwd = password,
[noinit | init],
notify = {client | operator_console},
verify[ = header | full]
}]

```

execute

运行过程或动态执行 Transact-SQL 命令。

```

[exec[ute]] [@return_status =]
[[[server.]database.]owner.]procedure_name[;number]
[[@parameter_name =] value |
[@parameter_name =] @variable [output]
[, [@parameter_name =] value |
[@parameter_name =] @variable [output]...]]
[with recompile]

```

或

```

exec[ute] ("string" | char_variable
[+ "string" | char_variable]...)

```

fetch

从游标结果集中返回一行或一组行。

```

fetch [next | prior | first | last | absolute
fetch_offset | relative fetch_offset]
[from] cursor_name
[into fetch_target_list]

```

goto label

分岔转到用户定义的标签。

```

label:
goto label

```

grant

为单个用户、用户组和角色指派权限。为用户、系统或用户定义角色分配角色。

授予访问数据库对象的权限：

```
grant {all [privileges] | permission_list}
    on {table_name [correlation_name]
        [(column_list)]
        | view_name [(column_list)]
        | stored_procedure_name}
        | function_name
        | keyname}
    [where search_condition]
    [as pred_name]
    to {public | name_list | role_list}
    [with grant option]
```

授予使用内置函数的权限：

```
grant select
    on [builtin] builtin
    to {name_list | role_list}
```

授予执行特定命令的权限：

```
grant {all [privileges] | command_list}
    to {public | name_list | role_list}
```

授予对某些 dbcc 命令的访问权限：

```
grant dbcc {dbcc_command [on {all | database}]
    [, dbcc_command [on {all | database}] , ...]}
    to {user_list | role_list}
```

授予创建加密密钥的权限：

```
grant create encryption key to {user_list | role_list | group_list}
```

授予对表或表中某些列的列表的 decrypt 权限：

```
grant decrypt on [ owner.]tablename[(columnname [{columnname}])]
    to {user | group | role}
```

授予特定系统表的缺省权限：

```
grant default permissions on system tables
```

为用户或角色授予某个角色：

```
grant {role role_granted [, role_granted ...]}
    to grantee [, grantee...]
```

将服务器用户标识切换成任何其它的服务器登录，且根据目标登录角色来限制其使用：

```
grant set {proxy | tracing} to role_list
    [restrict role role_list | all | system]
```

---

group by 和 having  
子句

在 select 语句中使用，用于将表划分为组，并仅返回与 having 子句中的条件匹配的组。group by 通常与集合一起使用，以指定如何对 select 查询的未集合的列进行分组。having 子句应用于这些组。

*Start of select statement*

[group by [all] *aggregate\_free\_expression*  
[, *aggregate\_free\_expression*]...]

[having *search\_conditions*]

*End of select statement*

if...else

施加 SQL 语句的执行条件。

if *logical\_expression* [plan "*abstract plan*"]  
*statements*

[else  
[if *logical\_expression*] [plan "*abstract plan*"]  
*statement*]

insert

向表或视图中添加新行。

insert [into] [*database*.*owner*.]]{*table\_name*|*view\_name*}  
[(*column\_list*)]  
{values (*expression* [, *expression*]...)  
|*select\_statement* [plan "*abstract plan*"]}

kill

注销进程。

kill *spid* with statusonly

load database

装载使用 dump database 创建的用户数据库的备份副本，包括其事务日志，并实现通过数据库转储装载的存档数据库。

进行例行数据库装载：

load database *database\_name*  
from [compression=]*stripe\_device*  
[at *backup\_server\_name*]  
[density = *density\_value*,  
blocksize = *number\_bytes*,  
dumpvolume = *volume\_name*,  
file = *file\_name*]  
with verify only [= header | full]  
[stripe on [compression=]*stripe\_device*  
[at *backup\_server\_name*]  
[density = *density\_value*,  
blocksize = *number\_bytes*,  
dumpvolume = *volume\_name*,  
file = *file\_name*]  
[[stripe on [compression=]*stripe\_device*  
[at *backup\_server\_name*]  
[density = *density\_value*,

```

        blocksize = number_bytes,
        dumpvolume = volume_name,
        file = file_name]]...]
[with{
    density = density_value,
    blocksize = number_bytes,
    compression,
    dumpvolume = volume_name,
    file = file_name,
    [dismount | nodismount],
    [nounload | unload],
    passwd = password,
    notify = {client | operator_console},
    [override]]}]

```

返回标头或文件信息但不装载备份:

```

load database database_name
from [compress::stripe_device
    [at backup_server_name]
    [density = density_value,
    blocksize = number_bytes,
    dumpvolume = volume_name,
    file = file_name]
[stripe on [compress::stripe_device
    [at backup_server_name]
    [density = density_value,
    blocksize = number_bytes,
    dumpvolume = volume_name,
    file = file_name]
[[stripe on [compress::stripe_device
    [at backup_server_name]
    [density = density_value,
    blocksize = number_bytes,
    dumpvolume = volume_name,
    file = file_name]]...]]
[with{
    density = density_value,
    blocksize = number_bytes,
    compression,
    dumpvolume = volume_name,
    file = file_name,
    [dismount | nodismount],
    [nounload | unload],
    passwd = password,
    listonly [= full],
    headeronly,
    notify = {client | operator_console}
}]]

```



---

实现存档数据库：

```
load database database_name
  from dump_device
  [ [stripe on stripe_device] ...]
  [with [norecovery],[passwd=password]]
```

如站点已获得 Tivoli Storage Manager 许可则装载数据库的副本：

```
load database database_name
  from syb_tsm::[[-S source_sever_name][-D source_database_name]
  ::]object_name [blocksize = number_bytes]
  [stripe on syb_tsm::[[-S source_sever_name]
  [-D source_database_name]::]object_name
  [blocksize = number_bytes]]
  [[stripe on syb_tsm::[[-S source_sever_name]
  [-D source_database_name]::]object_name
  [blocksize = number_bytes]]...]
  [with{
    blocksize = number_bytes,
    passwd = password,
    listonly [= full],
    headeronly,
    notify = {client | operator_console},
    [[verifyonly | verify] [= header | full]]
  }]
```

load transaction

装载使用 dump transaction 创建的事务日志的备份副本。

进行例行日志装载：

```
load tran[saction] database_name
  from [compress::]stripe_device
  [at backup_server_name]
  [density = density_value,
  blocksize = number_bytes,
  dumpvolume = volume_name,
  file = file_name]
  [stripe on [compress::]stripe_device
  [at backup_server_name]
  [density = density_value,
  blocksize = number_bytes,
  dumpvolume = volume_name,
  file = file_name]
  [[stripe on [compress::]stripe_device
  [at backup_server_name]
  [density = density_value,
  blocksize = number_bytes,
  dumpvolume = volume_name,
  file = file_name]]...]
  [with{
    density = density_value,
```

```

        blocksize = number_bytes,
        compression,
        dumpvolume = volume_name,
        file = file_name,
        [dismount | nodismount],
        [nounload | unload],
        notify = {client | operator_console}
    ]}]

```

返回标头或文件信息但不装载备份日志：

```

load tran[saction] database_name
  from [compress::stripe_device
        [at backup_server_name]
        [density = density_value,
          blocksize = number_bytes,
          dumpvolume = volume_name,
          file = file_name]
        [stripe on [compress::stripe_device
                    [at backup_server_name]
                    [density = density_value,
                      blocksize = number_bytes,
                      dumpvolume = volume_name,
                      file = file_name]
                    [[stripe on [compress::stripe_device
                                [at backup_server_name]
                                [density = density_value,
                                  blocksize = number_bytes,
                                  dumpvolume = volume_name,
                                  file = file_name]]...]]
        [with{
          density = density_value,
          blocksize = number_bytes,
          compression,
          dumpvolume = volume_name,
          file = file_name,
          [dismount | nodismount],
          [nounload | unload],
          listonly [= full],
          headeronly,
          notify = {client | operator_console}
          until_time = datetime}]

```

将事务日志装载到存档数据库中：

```

load tran[saction] database_name
  from dump_device
  [[stripe on stripe_device] ...]

```

---

（仅限 Tivoli Storage Manager）您的站点许可使用 Tivoli Storage Manager 时装载事务日志的副本。

```
load transaction database_name
  from syb_tsm::[[-S source_sever_name][[-D source_database_name]
    ::]object_name [blocksize = number_bytes]
  [stripe on syb_tsm::[[-S source_sever_name]
    [-D source_database_name::]object_name
    [blocksize = number_bytes]]
  [[stripe on syb_tsm::[[-S source_sever_name]
    [-D source_database_name::]object_name
    [blocksize = number_bytes]]...]
  [with{
    blocksize = number_bytes,
    passwd = password,
    listonly [= full],
    headeronly,
    notify = {client | operator_console},
    until_time = datetime
  }]
```

lock table

在事务中显式锁定表。

```
lock table table_name in {share | exclusive} mode
  [wait [numsecs] | nowait]
```

merge

将源表中的行传送到目标表中：

- 插入位于源中且在目标中没有匹配键列的行。
- 用源行中的值更新其键列已经在目标中存在的行。

```
merge by {insert | update}
  into [(database.owner.identifier [as table_alias]
  using [(database.owner.identifier [as table_alias]
    | (select_query) as alias_name [column_list]
  on merge_search_condition
  [ when matched [and search_conditions ]
    then {update set {col_name = expression} | delete} ]
  [ when not matched [and search_conditions ]
    then insert [(column_list)] values (value_list)
```

mount

将数据库附加到目标或辅助 Adaptive Server。

```
mount database all | database_mapping[, database_mapping, ...]
  from "manifest_file"
  [using device_mapping [, device_mapping...]
  [with listonly]

database_mapping:
  origdbname as newdbname
  | newdbname = origdbname
```

	<pre>   origdbname   newdbname  device_mapping     logical_device_name as new_physical_name   new_physical_name = logical_device_name   original_physical_name   new_physical_name </pre>
online database	<p>在常规装载序列完成后，将数据库标记为可供公共用户使用；如果需要，将装载的数据库升级到当前版本的 Adaptive Server；在装载了使用 <code>for standby_access</code> 选项转储的事务日志后使数据库联机。还可以使用 <code>online database</code> 使存档数据库联机。</p> <pre> online database <i>database_name</i> [for standby_access] </pre>
open	<p>打开要进行处理的光标。</p> <pre> open <i>cursor_name</i> </pre>
order by clause	<p>按排序顺序返回指定列中的查询结果。</p> <pre> [Start of select statement]  [order by     {[<i>table_name</i>.  <i>view_name</i>.]       <i>column_name</i>   <i>select_list_number</i>   <i>expression</i>}     [asc   desc]     [, {[<i>table_name</i>.  <i>view_name</i>.]       <i>column_name</i>   <i>select_list_number</i>   <i>expression</i>}     [asc   desc]}...]  [End of select statement] </pre>
prepare transaction	<p>由 DB-Library 在两阶段提交应用程序中使用，以查看服务器是否已准备好提交事务。</p> <pre> prepare tran[saction] </pre>
print	<p>在用户屏幕上输出用户定义的消息。</p> <pre> print     {<i>format_string</i>   @<i>local_variable</i>       @@<i>global_variable</i>}     [, <i>arg_list</i>] </pre>
quiesce database	<p>挂起和恢复对指定的一系列数据库进行更新。</p> <pre> quiesce database <i>tag_name</i> hold <i>database_list</i> [for external dump]     [to <i>manifest_file</i> [with override]] </pre> <p>或：</p> <pre> quiesce database <i>tag_name</i> release </pre>

---

raiserror	<p>在用户屏幕上输出用户定义的错误消息并设置系统标记来记录发生了错误这一情况。</p> <pre>raiserror <i>error_number</i>           [{<i>format_string</i>   @<i>local_variable</i>}] [, <i>arg_list</i>]           [with errordata <i>restricted_select_list</i>]</pre>
readtext	<p>从指定偏移位置开始，读取指定字节数或字符数的 <i>text</i>、<i>unitext</i> 和 <i>image</i> 值。</p> <pre>readtext [(<i>database</i>.)<i>owner</i>.]<i>table_name.column_name</i>           <i>text_pointer</i> <i>offset</i> <i>size</i>           [holdlock   noholdlock] [readpast]           [using {bytes   chars   characters}]           [at isolation{             [read uncommitted   0]               [read committed   1]               [repeatable read   2]               [serializable   3]]]</pre>
reconfigure	<p><b>reconfigure</b> 命令目前不起作用，包含它是为了使现有脚本不进行任何修改就可以运行。</p> <pre>reconfigure</pre>
remove java	<p>当数据库中安装 Java 类后从数据库中删除一个或多个 Java-SQL 类、软件包或 JAR。</p> <pre>remove java class <i>class_name</i> [, <i>class_name</i>]...     package <i>package_name</i> [, <i>package_name</i>]...     jar <i>jar_name</i> [, <i>jar_name</i>]...[retain classes]</pre>
reorg	<p>根据使用的选项，回收页上未使用的空间、删除行转移或将表中所有行重新写入到新页中。</p> <pre>reorg compact <i>table_name</i> [partition <i>partition_name</i>]           [with {resume, time = <i>no_of_minutes</i>, compress}] reorg forwarded_rows <i>table_name</i> [partition <i>partition_name</i>]           [with {resume, time = <i>no_of_minutes</i>, compress}] reorg rebuild <i>table_name</i> [<i>index_name</i> [partition <i>index_partition_name</i>]] reorg reclaim_space <i>table_name</i> [<i>index_name</i>] [partition <i>partition_name</i>]           [with {resume, time = <i>no_of_minutes</i>, compress}]</pre>
return	<p>无条件退出批处理或过程，并提供可选返回状态。<b>return</b> 之后的语句不会执行。</p> <pre>return [<i>integer_expression</i>] [plan "<i>abstract_plan</i>"]</pre>

revoke

撤销访问数据库对象的权限：

```
revoke [grant option for]
{all [privileges] | permission_list}
on {table_name [(column_list)]
    | view_name [(column_list)]
    | stored_procedure_name}
    | keyname}
from {public | name_list | role_list}
[cascade]
```

撤销选择内置函数的权限：

```
revoke select
on [builtin] builtin
to {name_list | role_list}
```

撤销创建数据库对象的权限、执行 set proxy 或执行 set session authorization：

```
revoke {all [privileges] | command_list}
from {public | name_list | role_list}
```

撤销运行 set proxy 或 set tracing 的权限：

```
[revoke set {proxy | tracing}
from {public | name_list | role_list}
```

撤销用户或其它角色的角色：

```
revoke role {role_name [, role_list ...]} from
{grantee [, grantee ...]}
```

撤销对某些 dbcc 命令的访问权限：

```
revoke dbcc {dbcc_command [on {all | database}]
[, dbcc_command [on {all | database}], ...]}
from {user_list | role_list}
```

撤销其他用户、组和角色创建加密密钥的权限。

```
revoke create encryption key from user | role | group
```

撤销行过滤谓词。

```
revoke {all [privileges]
    | [all] permission_list}
on table_name (column_list)
[with { pred_name | {all | no} predicates}]
from {public | name_list | role_list}
```

撤销对表或表中某些列的列表的 decrypt 权限：

```
revoke decrypt on [owner.] tablename[(columnname [{columnname}]])
from user | group | role
```

撤销 public 的缺省权限：

```
revoke default permissions on system tables
```

---

rollback	<p>将用户定义的事务回退到事务内的指定保存点或事务的起始点。</p> <pre>rollback [tran   transaction   work]         [transaction_name   savepoint_name]</pre>
rollback trigger	<p>回退触发器中所做的工作，包括导致触发器引发的数据修改，并发出可选的 <code>raiserror</code> 语句。</p> <pre>rollback trigger         [with raiserror_statement]</pre>
save transaction	<p>在事务内设置一个保存点。</p> <pre>save transaction savepoint_name</pre>
select	<p>从数据库对象中检索行。</p> <pre>select ::=     select [all   distinct]     [top unsigned_integer]     select_list     [into_clause]     [from_clause]     [where_clause]     [group_by_clause]     [having_clause]     [order_by_clause]     [compute_clause]     [read_only_clause]     [isolation_clause]     [browse_clause]     [plan_clause]     [for_xml_clause]      select_list ::=  into_clause ::=     into [(database.) owner.] table_name         [(colname encrypt [with (database.[owner].)keyname] [,             colname encrypt_clause ...])]           [compressed = compression_level   not compressed]         [in row [(length)]   off row ]         [{[external table at]             'server_name.(database).[owner].object_name'               external directory at 'pathname'               external file at 'pathname' [column delimiter 'string']}]         [on segment_name]         dml_logging = (full   minimal)         [partition_clause]         [lock {datarows   datapages   allpages}]         [with [, into_option[, into_option] ...]]]</pre> <p>有关 <code>select_list</code> 的详细信息，请参见《参考手册：命令》中的“参数”部分。</p>

```

| into existing table table_name

partition_clause ::=
    partition by range (column_name[, column_name]...)
        ([partition_name] values <= ({constant | MAX}
            [, {constant | MAX}] ...)[on segment_name]
            [compression_clause] [on segment_name]
            [, [partition_name] values <= ({constant | MAX}
            [, {constant | MAX}] ...)[on segment_name]...]
            [compression_clause] [on segment_name]

    | partition by hash (column_name[, column_name]...)
        { (partition_name [on segment_name]
            [compression_clause] [on segment_name]
            [, partition_name [on segment_name]...]
            [compression_clause] [on segment_name]
            | number_of_partitions
            [on (segment_name[, segment_name] ...)])

    | partition by list (column_name)
        ([partition_name] values (constant[, constant] ...)
            [compression_clause] [on segment_name]
            [, [partition_name] values (constant[, constant] ...)
            [compression_clause] [on segment_name]

    | partition by roundrobin
        { (partition_name [on segment_name]
            [, partition_name [on segment_name]...]
            [compression_clause] [on segment_name]
            | number_of_partitions
            [on (segment_name [, segment_name]...)])

into_option ::=
    | max_rows_per_page = num_rows
    | exp_row_size = num_bytes
    | reservepagegap = num_pages
    | identity_gap = gap
    | compression = {none | page | row}
    | lob_compression = off | compression_level]

from_clause ::=
    from table_reference [, table_reference]...

table_reference ::=
    table_view_name | ANSI_join

table_view_name ::=
    [[database.]owner.] [{table_name | view_name}
    [as] [correlation_name]
    [(index {index_name | table_name})]
    [parallel [degree_of_parallelism]]
    [prefetch size][lru | mru]}

```



---

```

[holdlock | noholdlock]
[readpast]
[shared]

ANSI_join ::=
    table_reference join_type join table_reference
        join_conditions
    join_type ::= inner | left [outer] | right [outer]
    join_conditions ::= on search_conditions

compression_clause ::=
    with compression = {none | page | row}

where_clause ::=
    where search_conditions
    for update [of column_list]

group_by_clause ::=
    group by [all] aggregate_free_expression
        [, aggregate_free_expression]...

having_clause ::=
    having search_conditions

order_by_clause ::=
    order by sort_clause [, sort_clause]...

sort_clause ::=
    {[[[database.]owner.]{table_name.|view_name.}]column_name
    | select_list_number
    | expression }
    [asc | desc]

compute_clause ::=
    compute row_aggregate (column_name)
        [, row_aggregate (column_name)]...
    [by column_name [, column_name]...]

read_only_clause ::=
    for {read only | update [of column_name_list]}

isolation_clause ::=
    at isolation
        {read uncommitted | 0}
        | {read committed | 1}
        | {repeatable read | 2}
        | {serializable | 3}

browse_clause ::=
    for browse

plan_clause ::=
    plan "abstract plan"

```

set

在用户的工作会话期间设置 Adaptive Server 查询处理选项；在触发器或存储过程内设置某些选项。

```

set advanced_aggregation on/off
set @variable = expression [, @variable = expression...]
set ansinull {on | off}
set ansi_permissions {on | off}
set arithabort [arith_overflow | numeric_truncation] {on | off}
set arithignore [arith_overflow] {on | off}
set bulk array size number
set bulk batch size number
set builtin_date_strings number
set {chained, close on endtran, nocount, noexec, parseonly,
    self_recursion, showplan, sort_resources} {on | off}
set char_convert {off | on [with {error | no_error}] |
    charset [with {error | no_error}]}
set cis_rpc_handling {on | off}
set [clientname client_name | clienthostname host_name
    | clientapplname application_name]
set compression {on | off | default}
set cursor rows number for cursor_name
set {datefirst number, dateformat format, language language}
set delayed_commit {on | off | default}
set deferred_name_resolution { on | off }
set dml_logging {minimal | default}
set encryption passwd 'password_phrase'
    for {key | column} {keyname | column_name}
set export_options [on | off]
set fipsflagger {on | off}
set flushmessage {on | off}
set fmtonly {on | off}
set forceplan {on | off}
set identity_insert [database.owner.]table_name {on | off}
set identity_update table_name {on | off}
set index_union on | off
set literal_autoparam on | off

```

---

```
set lock {wait [numsecs] | nowait}
set metrics_capture on | off
set offsets {select, from, order, compute, table,
             procedure, statement, param, execute} {on | off}
set option show
set opttimeoutlimit
set parallel_degree number
set plan {dump | load} [group_name] {on | off}
set plan exists check {on | off}
set plan for show
set plan optgoal {allrows_mix | allrows_dss}
set plan opttimeoutlimit number
set plan replace {on | off}
set prefetch [on|off]
set print_minlogged_mode_override
set proc_output_params {on | off}
set proc_return_status {on | off}
set process_limit_action {abort | quiet | warning}
set proxy login_name
set quoted_identifier {on | off}
set repartition_degree number
set repthreshold number
set resource_granularity number
set role {"sa_role" | "sso_role" | "oper_role" |
         role_name [with passwd "password"]} {on | off}
set {rowcount number, textsize number}
set scan_parallel_degree number
set send_locator {on | off }
set session authorization login_name
set switch [serverwide] {on | off} trace_flag [, [trace_flag,] [with option [,
option]]
set show_exec_info ["on" | "off"]
set show_sqltext {on | off}
set show_transformed_sql, {on|off}
set statement_cache on | off
```

	<pre> set statistics {io, subquerycache, time, plancost} {on   off} set statistics simulate {on   off} set strict_dtm_enforcement {on   off} set string_truncation {on   off} set system_view {instance   cluster   clear} set textsize {number} set tracefile [filename] [off] [for spid] set transaction isolation level{     [read uncommitted   0]       [read committed   1]       [repeatable read   2]       [serializable   3]} set transactional_rpc {on   off} </pre>
setuser	<p>允许数据库所有者充当其他用户。</p> <pre> setuser ["user_name"] </pre>
shutdown	<p>关闭发出 shutdown 命令的 Adaptive Server、其本地 Backup Server 或远程 Backup Server。</p> <pre> shutdown [srvname] [with {wait [= "hh:mm:ss"]   nowait}] </pre> <p>用于集群的语法：</p> <pre> shutdown {cluster   [instance_name]} [with {wait   nowait}] </pre>
transfer table	<p>启动增量表传输。</p> <pre> transfer table [[db.]owner.]table [to   from] destination_file     [ for { ase   bcp   iq   csv } ]     [ with {column_separator=string}, {column_order=option},     {encryption=option}, {row_separator=string},     {resend=id}, {progress=sss}, {tracking_id=nnn}     {sync = true   false}}, {fixed_length = true   false}     , null_byte = true   false}] </pre>
truncate lob	<p>将 LOB 截断成指定长度。</p> <pre> truncate lob locator_descriptor [ ( result_length)] </pre>
truncate table	<p>从表或分区中删除所有行。</p> <pre> truncate table [[database.]owner.]table_name     [partition partition_name] </pre>

---

union 运算符

返回由两个或多个查询的结果组成的一个结果集。除非指定 **all** 关键字，否则重复行会从结果集中删除。

```
select [top unsigned_integer] select_list
    [into clause] [from clause] [where clause]
    [group by clause] [having clause]
    [union [all]
select [top unsigned_integer] select_list
    [from clause] [where clause]
    [group by clause] [having clause]]...
[order by clause]
[compute clause]
```

unmount

关闭数据库并从 Adaptive Server 中删除它。

```
unmount database dbname_list to manifest_file
```

update

通过添加数据或者修改现有数据，更改现有行中的数据。

```
update [top unsigned_integer]
[[database.]owner.]{table_name | view_name}
set [[[database.]owner.]{table_name.|view_name.}
column_name1 =
{expression1 | NULL | (select_statement)} |
variable_name1 =
{expression1 | NULL | (select_statement)}
[, column_name2 =
{expression2 | NULL | (select_statement)}]...|
[, variable_name2 =
{expression2 | NULL | (select_statement)}]...

[from [[database.]owner.]{view_name [readpast]]
table_name
    [(index {index_name | table_name}
        [prefetch size][lru|mru])]]
    [readpast]
[, [[database.]owner.]{view_name [readpast]]
table_name
    [(index {index_name | table_name}
        [prefetch size][lru|mru])]]]
    [readpast] ...]
[where search_conditions]
[plan "abstract plan"]

update [[database.]owner.]{table_name | view_name}
set [[[database.]owner.]{table_name.|view_name.}
column_name1 =
    {expression1 | NULL | (select_statement)} |
variable_name1 =
    {expression1 | NULL | (select_statement)}
[, column_name2 =
    {expression2 | NULL | (select_statement)}]...|
```

	<pre> [, variable_name2 =   {expression2   NULL   (select_statement)}]... where current of cursor_name </pre>
update all statistics	更新给定表的所有统计信息。可以对单个数据分区运行 update all statistics。 <pre> update all statistics table_name [partition data_partition_name] </pre>
update index statistics	更新索引中所有列的统计信息。 <pre> update index statistics   table_name [[partition data_partition_name]     [index_name [partition index_partition_name]]]   [using step values]   [with consumers = consumers] [, sampling=N percent] </pre>
update statistics	为索引、表或分区中的所有列更新指定索引中键值的分布信息，并重置全局非聚簇索引的数据更改计数器。 <pre> update statistics table_name   [[partition data_partition_name] [(column_list)]     index_name [partition index_partition_name]]   [using step values]   [with consumers = consumers][, sampling=N percent] </pre>
update table statistics	update table statistics 更新存储在 systabstats 表中的统计信息，如行计数、集群比等。update table statistics 不会影响存储在 sysstatistics 中的列统计信息。 <pre> update table statistics table_name   [partition data_partition_name]   [index_name [partition index_partition_name]] </pre>
use	指定要使用的数据库。 <pre> use database_name </pre>
waitfor	指定执行语句块、存储过程或事务的特定时间、时间间隔或事件。 <pre> waitfor {delay time   time time   errorexit   processexit   mirrorexit} </pre>
where 子句	在 select、insert、update 或 delete 语句中设置搜索条件。 <pre> where [not] expression comparison_operator expression where {[not] expression comparison_operator expression}   {...} where [not] expression [not] like "match_string"   [escape "escape_character"] where [not] expression is [not] null where [not] expression [not] between expression and expression where [not] expression [not] in ({value_list   subquery}) where [not] exists (subquery) where [not] expression comparison_operator {any   all} (subquery) </pre>

---

	<code>where [not] <i>column_name</i> <i>join_operator</i> <i>column_name</i></code>
	<code>where [not] <i>logical_expression</i></code>
	<code>where [not] <i>expression</i> {and   or} [not] <i>expression</i></code>
	<code>where <i>column_name</i> is [not] null</code>
while	设置重复执行语句或语句块的条件。只要指定条件为真，语句就会重复执行。
	<code>while <i>logical_expression</i> [plan "<i>abstract plan</i>"] <i>statement</i></code>
writetext	允许对现有 <code>text</code> 、 <code>unitext</code> 或 <code>image</code> 列进行需要日志操作最少的交互式更新。
	<code>writetext [[<i>database</i>.]<i>owner</i>.]<i>table_name.column_name</i></code> <code><i>text_pointer</i> [readpast] [with log] <i>data</i></code>

## 交互式 dbsql 命令

以下是 Adaptive Server 的交互式 `dbsql` 命令的语法和简要说明。有关详细信息，请参见《参考手册：命令》。

clear	清除 Interactive SQL 窗格。
	<code>clear</code>
configure	打开 Interactive SQL 的“选项”对话框。
	<code>configure</code>
connect	建立与数据库的连接。
	<code>connect</code> <code>[to <i>engine_name</i>]</code> <code>[database <i>database_name</i>]</code> <code>[as <i>connection_name</i>]</code> <code>[user] <i>user_id</i> identified by password</code> <code><i>engine_name</i>, <i>database_name</i>, <i>connection_name</i>, <i>user_id</i>,</code> <code><i>password</i> : {<i>identifier</i>   <i>string</i>   <i>hostvar</i>}</code> <code>connect using <i>connect_string</i> : {<i>identifier</i>   <i>string</i>   <i>hostvar</i>}</code>
disconnect	删除与数据库的当前连接。
	<code>disconnect [{<i>identifier</i>   <i>string</i>   <i>hostvar</i>}   current   all]</code>
exit	退出 Interactive SQL。
	<code>{exit   quit   bye} [{<i>number</i>   <i>connection_variable</i>}</code>

input	<p>将数据从外部文件或键盘导入数据库表中。</p> <pre>input into [ owner.]table_name [ from filename   prompt] [ format { ascii   dbase   dbaseII   dbaseIII   excel   fixed   foxpro   lotus }}] [ escape character character] [ escapes { on   off } [ by order   by name ] [ delimited by string ] [ column widths (integer , . . . ) ] [ nostrip ] [ ( column_name, . . . ) ] [ encoding {identifier   string}]</pre>
output	<p>将数据从外部文件或键盘导入数据库表中。</p> <pre>output to filename [ append ] [ verbose ] [ format {ascii   dbase   dbaseII  dbaseIII   excel   fixed   foxpro   lotus   sql   xml}}] [ escape character character ] [ escapes { on   off} [ delimited by string ] [ quote string [ all ] ] [ column widths (integer , . . . ) ] [ hexadecimal { on   off   asis } ] [ encoding {string   identifier}]</pre>
parameters	<p>指定 Interactive SQL 命令文件的参数。</p> <pre>parameters parameter1, parameter2, . . .</pre>
read	<p>从文件中读取 Interactive SQL 语句。</p> <pre>read [ encoding {identifier   string}] file_name [ parameters ]</pre>
set connection	<p>将当前数据库连接更改到另一台服务器。</p> <pre>set connection {identifier   string   hostvar}</pre>
set option	<p>更改 Interactive SQL 选项的值。</p> <pre>set [ temporary] option [{identifier   string   hostvar}.   public.] {identifier   string   hostvar  builtin_date_strings} = [option_value]  set permanent set</pre>
start logging	<p>开始将执行的 SQL 语句记录到日志文件中。</p> <pre>start logging file_name</pre>



---

stop logging	停止记录当前会话中已执行的 SQL 语句。 stop logging
system	从 Interactive SQL 内启动可执行文件。 system '[path] file_name'

## 系统过程

以下是 Adaptive Server 系统存储过程的语法和简要说明。有关完整信息，请参见《参考手册：过程》。

<code>sp_activeroles</code>	显示所有活动角色。  <code>sp_activeroles [expand_down]</code>
<code>sp_add_qpgroup</code>	添加抽象计划组。  <code>sp_add_qpgroup new_name</code>
<code>sp_add_resource_limit</code>	创建 Adaptive Server 登录名和/或应用程序执行查询、查询批处理或事务时可以使用的服务器资源的数目限制。  <code>sp_add_resource_limit name, appname, rangename, limittype, limitvalue [, enforced [, action [, scope ]]]</code>
<code>sp_add_time_range</code>	将指定的时间范围添加到 Adaptive Server。  <code>sp_add_time_range name, startday, endday, starttime, endtime</code>
<code>sp_addalias</code>	允许 Adaptive Server 用户在数据库中以另一个用户的名义操作。  <code>sp_addalias loginame, name_in_db</code>
<code>sp_addauditrecord</code>	允许用户在审计追踪中输入用户定义的审计记录（注释）。  <code>sp_addauditrecord [text [, db_name [, obj_name [, owner_name [, dbid [, objid]]]]]]</code>
<code>sp_addaudittable</code>	安装审计功能后，再添加一个系统审计表。  <code>sp_addaudittable devname</code>
<code>sp_addengine</code>	<code>sp_addengine</code> 不在线程化模式中运行。  <code>sp_addengine engine_number, engine_group [, instance_id]</code>
<code>sp_addexeclass</code>	创建或更新可以绑定到客户端应用程序、登录和存储过程的用户定义执行类。  <code>sp_addexeclass classname, priority, timeslice, engine_group [, instance_id]</code>
<code>sp_addextendedproc</code>	在 master 数据库中创建扩展存储过程 (ESP)。  <code>sp_addextendedproc esp_name, dll_name</code>
<code>sp_addexternlogin</code>	（仅适用于组件集成服务）创建替代的登录帐号和口令，以便在通过组件集成服务与远程服务器通信时使用。  <code>sp_addexternlogin server, loginame, externname [, externpasswd] [rolename]</code>

---

sp_addgroup	向数据库添加组。在授予和撤消特权时可使用组作为集合名称。 <b>sp_addgroup</b> <i>grpname</i>
sp_addlanguage	定义替代语言的月份和星期日的名称及其日期格式。 <b>sp_addlanguage</b> <i>language, alias, months, shortmons, days, datefmt, datefirst</i>
sp_addmessage	将用户定义的消息添加到 sysusermessages 以供存储过程 print 和 raiserror 调用以及 sp_bindmsg 使用。 <b>sp_addmessage</b> <i>message_num, message_text</i> [, <i>language</i> [, <i>with_log</i> [, <i>replace</i> ]]]
sp_addobjectdef	(仅限组件集成服务) 指定本地表和外部存储位置之间的映射。 <b>sp_addobjectdef</b> <i>tablename, objectdef</i> [, " <i>objecttype</i> "]
sp_addremotelogin	通过将条目添加到 master.dbo.sysremotelogins 中对新远程服务器用户进行授权。 <b>sp_addremotelogin</b> <i>remoteserver</i> [, <i>loginame</i> [, <i>remotename</i> ]]
sp_addsegment	定义数据库中数据库设备上的段。 <b>sp_addsegment</b> <i>segname, dbname, devname</i>
sp_addserver	定义远程服务器或定义本地服务器的名称; 指定使用 host 和 port 参数时用于远程过程调用 (RPC) 的服务器。 <b>sp_addserver</b> <i>lname</i> [, <i>class</i> [, <i>pname</i> ]] 仅用于组件集成服务 (CIS): <b>sp_addserver</b> ' <i>logical_server_name</i> ', ASEnterprise, ' <i>host:port:filter</i> '
sp_addthreshold	创建一个阈值以监控数据库段上的空间。当段上的可用空间低于指定水平时, Adaptive Server 执行关联的存储过程。 <b>sp_addthreshold</b> <i>dbname, segname, free_space, proc_name</i>
sp_addtype	创建用户定义的数据类型。 <b>sp_addtype</b> <i>typename,</i> <i>phystype</i> [( <i>length</i> )   ( <i>precision</i> [, <i>scale</i> ])] [, " <i>identity</i> "   <i>nulltype</i> ]
sp_addumpdevice	将转储设备添加到 Adaptive Server。 <b>sp_addumpdevice</b> {" <i>tape</i> "   " <i>disk</i> "}, <i>logicalname,</i> <i>physicalname</i> [, <i>tapesize</i> ]
sp_adduser	将新用户添加到当前数据库。 <b>sp_adduser</b> <i>loginame</i> [, <i>name_in_db</i> [, <i>grpname</i> ]]

sp_altermessage	启用或禁用在 Adaptive Server 错误日志中记录系统定义的消息或用户定义的消息。  sp_altermessage <i>message_id</i> , <i>parameter</i> , <i>parameter_value</i>
sp_audit	允许系统安全员配置审计选项。  sp_audit <i>option</i> , <i>login_name</i> , <i>object_name</i> [, <i>setting</i> ]  或:  sp_audit 'restart'
sp_autoconnect	(仅限组件集成服务) 为特定用户定义到远程服务器的直通连接, 这样就可以使指定的用户在登录时自动进入直通模式。  sp_autoconnect <i>server</i> , {true   false} [, <i>loginame</i> ]
sp_autoformat	一种用于生成可读结果集数据的实用程序存储过程, sp_autoformat 将重新设置长度可变的字符数据的宽度格式, 以便只显示非空字符。输出中将截断尾随空白。  sp_autoformat <i>fulltablename</i> [, <i>selectlist</i> , <i>whereclause</i> , <i>orderby</i> ]
sp_bindcache	将数据库、表、索引、text 对象或 image 对象绑定到数据高速缓存。  sp_bindcache <i>cachename</i> , <i>dbname</i> [, [ <i>ownername</i> .] <i>tablename</i> [, <i>indexname</i>   "text only"]]
sp_bindefault	将用户定义的缺省值绑定到列或用户定义的数据类型。  sp_bindefault <i>defname</i> , <i>objname</i> [, <i>futureonly</i> ]
sp_bindexclass	将执行类与客户端应用程序、登录、存储过程或缺省执行类关联起来。  sp_bindexclass " <i>object_name</i> ", " <i>object_type</i> ", " <i>scope</i> ", " <i>classname</i> "
sp_bindmsg	将用户消息绑定到参照完整性约束或检查约束。  sp_bindmsg <i>constrname</i> , <i>msgid</i>
sp_bindrule	将规则绑定到列或用户定义的数据类型。  sp_bindrule <i>rulename</i> , <i>objname</i> [, <i>futureonly</i> ]
sp_cacheconfig	创建、配置、重新配置和删除数据高速缓存, 并提供有关信息。  sp_cacheconfig [ <i>cachename</i> [, " <i>cache_size</i> [P   K   M   G]" [, <i>logonly</i>   <i>mixed</i>   <i>inmemory_storage</i> ][, <i>strict</i>   <i>relaxed</i> ]] [, " <i>cache_partition</i> =[1   2   4   8   16   32   64]" [, <i>instance instance_name</i> ]

---

<code>sp_cachestrategy</code>	<p>对表、索引、text 对象或 image 对象启用或禁用预取（大 I/O）和 MRU 高速缓存替换策略。</p> <pre>sp_cachestrategy <i>dbname</i>, [<i>ownername</i>.]<i>tablename</i> [, <i>indexname</i>   "text only"   "table only" [, {<i>prefetch</i>   <i>mr</i>u}, {"on"   "off"}]]</pre>
<code>sp_changedbowner</code>	<p>更改用户数据库的所有者。</p> <pre>sp_changedbowner <i>loginame</i> [, true]</pre>
<code>sp_changegroup</code>	<p>更改用户的组。</p> <pre>sp_changegroup <i>grpname</i>, <i>username</i></pre>
<code>sp_checknames</code>	<p>检查当前数据库中是否存在包含 7 位 ASCII 字符集以外字符的名称。</p> <pre>sp_checknames [help   silent]</pre>
<code>sp_checkreswords</code>	<p>检测并显示属于 Transact-SQL 保留字的标识符。检查服务器名、设备名、数据库名、段名、用户定义的数据类型、对象名、列名、用户名、登录名和远程登录名。</p> <pre>sp_checkreswords [<i>user_name_param</i>]</pre>
<code>sp_checksourc</code>	<p>检查已编译对象的源文本以及计算列的源文本是否存在。编译对象可以是谓词的用户定义名称，或者如果谓词没有外部名称，则是其内部名称。</p> <pre>sp_checksourc [<i>objname</i> [, <i>tablename</i> [, <i>username</i>]]]</pre>
<code>sp_chgattribute</code>	<p>为以后分配表或索引的空间而更改 <code>max_rows_per_page</code>、<code>fillfactor</code>、<code>reservepagegap</code> 或 <code>exp_row_size</code> 值；设置表的 <code>concurrency_opt_threshold</code>。提供优化索引锁定的用户界面。</p> <pre>sp_chgattribute <i>objname</i>, {"max_rows_per_page"   "fillfactor"   "reservepagegap"   "exp_row_size"   "concurrency_opt_threshold"   "optimistic_index_lock"   "identity_burn_max"   "plldegree"} , <i>value</i>, <i>optvalue</i>  sp_chgattribute <i>objname</i>, {"identity_gap", <i>set_number</i>   "dealloc_first_txtpg", <i>value</i>}</pre>
<code>sp_cleanpwdchecks</code>	<p><code>sp_cleanpwdchecks</code> 是一个自定义存储过程，允许您定义何时及如何删除存储在用户定义表中的、与登录名和口令相关的属性。</p> <pre>sp_cleanpwdchecks, <i>login_name</i></pre>
<code>sp_clearpsex</code>	<p>清除由 <code>sp_setpsex</code> 设置的 Adaptive Server 会话的执行属性。</p> <pre>sp_clearpsex <i>spid</i>, <i>exeattr</i></pre>

**sp\_clearstats** 为所有服务器用户或指定的用户开始新的会计周期。执行 **sp\_reportstats** 可以输出上一个周期的统计信息。

```
sp_clearstats [loginame]
```

**sp\_client\_addr** 显示每个具有附加客户端应用程序（包括 **spid** 和客户端主机名）的 Adaptive Server 任务的 IP（Internet 协议）地址。

```
sp_client addr [spid]
```

**sp\_clusterlockusage** 在集群环境中——报告集群中可用、已用和保留的锁。

```
sp_clusterlockusage
```

**sp\_cluster** （仅限集群环境）执行大量与集群有关的过程。

将连接迁移到另一个逻辑集群或实例：

```
sp_cluster connection, migrate, lc_name, instance_name, "spid_list"
```

确定之前迁移到新实例的连接迁移是否处于挂起状态，如果是则终止迁移：

```
sp_cluster connection, ['migrate_status' | 'migrate_cancel' ][, 'spid_list']
```

修改未完成的操作，例如取消操作或更改操作的计时：

```
sp_cluster logical, "action", lc_name, {
    cancel, action_handle |
    modify_time, action_handle, wait_option[, timeout] |
    release, action_handle }
```

将资源或一个或多个路由添加到逻辑集群中：

```
sp_cluster logical, "add", lc_name, {
    route, route_type, key_list |
    instance, instance_list |
    failover, instance_list }
```

将路由从一个逻辑集群移到另一个逻辑集群：

```
sp_cluster logical, "alter", lc_name, route, route_type, key_list
```

新建逻辑集群：

```
sp_cluster logical, "create", lc_name
```

停止一个或多个实例上的逻辑集群或停止整个逻辑集群，并将这些实例或该集群置于非活动状态：

```
sp_cluster logical, "deactivate", lc_name, {
    "cluster" |
    "instance", instance_list }
[, wait_option[, timeout[, @handle output ]]]
```

---

删除一个逻辑集群或从该逻辑集群中删除一个或多个资源：

```
sp_cluster logical, "drop", lc_name,  
    {cluster | instance, instance_list |  
    failover, instance_list |  
    route, route_type, key_list }
```

逆向执行手动故障切换，恢复原始基本实例：

```
sp_cluster logical, "failback", lc_name, {  
    cluster[, wait_option[, timeout[, @handle output ]]] |  
    instance, from_instance_list, to_instance_list[, wait_option[,  
    timeout[, @handle output ]]] }
```

启动从基本实例到故障切换实例的手动故障切换。

```
sp_cluster logical, "failover", lc_name, {cluster  
    [, to_instance_list[, wait_option[, timeout[, @handle output ]]]  
    | instance, from_instance_list, to_instance_list[, wait_option[,  
    timeout[, @handle output ]]] }
```

手动收集一组连接并将其迁移到其它逻辑集群：

```
sp_cluster logical, 'gather', lc_name
```

显示 sp\_cluster logical 的完整语法：

```
sp_cluster logical, "help"
```

停止一个或多个实例上的逻辑集群或停止整个逻辑集群：

```
sp_cluster logical, "offline", lc_name,  
    {cluster | instance, instance_list }  
    [, wait_option[, timeout[, @handle output ]]]
```

启动一个或多个实例上的缺省逻辑集群：

```
sp_cluster logical, "online", { lc_name[, instance_list] }
```

设置逻辑集群规则：开放式逻辑集群、故障切换模式、系统视图、启动模式和负载配置文件：

```
sp_cluster logical, "set", lc_name, { open  
    | failover, failover_mode  
    | system_view, view_mode  
    | startup, { automatic | manual }  
    | load_profile, profile_name }  
    login_distribution, { affinity | "round-robin" }
```

显示有关逻辑集群的信息：

```
sp_cluster logical, "show"  
    [, lc_name[, {action[, state] | route[, type[, key]]}]]
```

允许设置和管理逻辑集群的负荷配置文件：

```
sp_cluster profile, [ "show" [, profile_name ]
| "create", profile_name
| "drop", profile_name
| "set", profile_name [, weight [, wt_metric [, wt_value ]
| threshold [, thr_metric [, thr_value ] ] ] ]
```

允许设置和管理逻辑集群的负荷配置文件：

```
sp_cluster profile, [ "show" [, profile_name ] | "create", profile_name |
"drop", profile_name | "set", profile_name [, weight [, wt_metric [,
wt_value ] | threshold [, thr_metric [, thr_value ] ] ] ]
```

sp\_cmp\_all\_qplans

比较两个抽象计划组中的所有抽象计划。

```
sp_cmp_all_qplans group1, group2 [, mode]
```

sp\_cmp\_qplans

比较两个抽象计划。

```
sp_cmp_qplans id1, id2
```

sp\_commonkey

定义两个表或视图之间的公用键（经常被连接的列）。

```
sp_commonkey tabaname, tabbname, col1a, col1b
[, col2a, col2b, ..., col8a, col8b]
```

sp\_companion

执行集群操作，例如在高可用性系统中将 Adaptive Server 配置为辅助协同服务器，将协同服务器从一种故障切换模式转换为另一种故障切换模式。sp\_companion 在辅助协同服务器上运行。

```
sp_companion
{server_name
{, configure
[, {with_proxydb | NULL}]
[, srvlogin]
[, server_password]
[, cluster_login]
[, cluspassword]}
| drop
| suspend
| resume
| prepare_failback
| do_advisory}
{, all
| help
| group_attribute_name
| base_attribute_name}
```

sp\_compatmode

检验是否可以使用完全兼容模式。

```
sp_compatmode
```



sp_configure	<p>按组显示配置参数、其当前值、非缺省值设置、最近设置的值以及此设置所使用的内存量。只显示其显示级别不高于用户的显示级别的参数。</p> <pre> sp_configure [configname [, configvalue]   group_name   non_unique_parameter_fragment] 'drop instance' [, instance_name] [display_nondefault_settings]  sp_configure "configuration file", 0, {"write"   "read"   "verify"   "restore"} "file_name" </pre>
sp_copy_all_qplans	<p>将一个抽象计划组中的所有计划复制到另一个组中。</p> <pre> sp_copy_all_qplans src_group, dest_group </pre>
sp_copy_qplan	<p>将一个抽象计划复制到抽象计划组。</p> <pre> sp_copy_qplan src_id, dest_group </pre>
sp_countmetadata	<p>显示 Adaptive Server 中索引、对象或数据库的数目。</p> <pre> sp_countmetadata "configname" [, dbname] </pre>
sp_cursorinfo	<p>报告有关特定游标或为会话激活的所有执行游标的信息。</p> <pre> sp_cursorinfo [{cursor_level   null}] [, cursor_name] </pre>
sp_dbextend	<p>使您能够：</p> <ul style="list-style-type: none"> <li>在数据库/段对和设备上安装自动数据库扩展过程。</li> <li>为单个段和设备定义特定于站点的策略。</li> <li>模拟数据库扩展机制的执行，在进行大容量卷装载前研究这一操作。</li> </ul> <p>所有参数都是字符串参数：</p> <pre> sp_dbextend 'help'[, command]  sp_dbextend [ ['set', ['threshold', dbname, segmentname, freespace   'database', dbname, segmentname { [ [ , growby ] [, maxsize ] } ]   'device', devicename { [ [ , growby ] [, maxsize ] } ] } ]   'clear', 'threshold', dbname, segmentname  sp_dbextend 'clear', 'database' [, dbname [, segmentname ] ] sp_dbextend 'clear', 'device' [, devicename ]  sp_dbextend 'modify', 'database', dbname, segmentname, { 'growby'   'maxsize' }, newvalue  sp_dbextend 'modify', 'device', devicename, { 'growby'   'maxsize' }, newvalue  sp_dbextend { 'list'   'listfull' } [, 'database' [, dbname [, segmentname [, order_by_clause ] ] ] ]  sp_dbextend { 'list'   'listfull' } [, 'device' [, devicename [, order_by_clause ]] ] </pre>

	<pre> sp_dbextend { 'list'   'listfull' }, [ 'threshold' [ , @dbname     [ , @segmentname ] ] ] sp_dbextend 'check', 'database' [ , dbname [ , segmentname ] ] sp_dbextend { 'simulate'   'execute' }, dbname, segmentname [ , iterations ] sp_dbextend 'trace', { 'on'   'off' } sp_dbextend 'reload [defaults]' sp_dbextend { 'enable'   'disable' }, 'database' [ , dbname [ , segmentname     ] ] sp_dbextend 'who' [ , 'spid'   'block'   'all' ] </pre>
sp_dboption	<p>显示或更改数据库选项，并启用异步日志服务功能。</p> <pre> sp_dboption [dbname, optname, optvalue [ , dockptf]] </pre>
sp_dbrecovery_order	<p>指定用户数据库恢复的顺序，并列出数据库或所有数据库的用户定义的恢复顺序。</p> <pre> sp_dbrecovery_order [database_name [ , rec_order [ , force [ , relax   strict     ]]]] </pre>
sp_dbremap	<p>强制 Adaptive Server 识别 alter database 所进行的更改。只有在 Adaptive Server 消息要求这样做时才运行此过程。</p> <pre> sp_dbremap dbname </pre>
sp_defaultloc	<p>（仅限组件集成服务）为本地数据库中的对象定义缺省存储位置。</p> <pre> sp_defaultloc dbname, defaultloc, defaulttype </pre>
sp_deletesmobj	<p>从 IBM Tivoli Storage Manager (TSM) 中删除指定的备份对象。</p> <pre> sp_deletesmob "syb_tsm", "server_name", "database_name",     "object_type", "dump_type", "until_time", "bs_name" </pre>
sp_depends	<p>显示有关数据库对象依赖性的信息，这些对象是数据库中依赖于指定的表或视图的视图、触发器、用户定义的函数和过程，以及数据库中指定的视图、触发器或过程所依赖的表和视图。</p> <pre> sp_depends objname[, column_name] </pre>
sp_deviceattr	<p>（仅限 UNIX 平台）更改现有数据库设备文件的设备参数设置。</p> <pre> sp_deviceattr logicalname, optname, optvalue </pre>
sp_diskdefault	<p>在用户未指定数据库设备或通过 create database 或 alter database 命令指定 default 的情况下，指定数据库设备是否可用于数据库存储。</p> <pre> sp_diskdefault logicalname, {defaulton   defaultoff} </pre>
sp_displayaudit	<p>显示审计选项的状态。</p> <pre> sp_displayaudit ["procedure"   "object"   "login"   "database"   "global"       "default_object"   "default_procedure" [ , "name"]] </pre>

---

sp_displaylevel	<p>设置或显示哪些 Adaptive Server 配置参数显示在 sp_configure 输出中。</p> <p>sp_displaylevel [<i>loginame</i> [, <i>level</i>]]</p>
sp_displaylogin	<p>显示有关登录帐户的信息。通过使用通配符表达式 (%), 还可以获取有关匹配登录名的信息。同时还会显示存储在磁盘上的登录口令的加密版本。</p> <p>sp_displaylogin [<i>'user_id'</i>   [<i>'loginame'</i>   <i>wildcard</i>]]</p>
sp_displayroles	<p>显示授予另一个角色的所有角色、登录名或登录配置文件、角色的表格式的整个层次树, 以及其它为指定角色配置的与登录安全性有关的参数, 包括锁定角色的日期、锁定原因以及锁定该角色的登录服务器用户 ID (suid)。对于受口令保护的角色, 还显示角色口令加密版本。</p> <p>sp_displayroles [<i>grantee_name</i> [, <i>mode</i>]]</p>
sp_downgrade	<p>(仅限 master 数据库) 验证是否已准备好降级至早期的 15.0.x 版。同时降级 Adaptive Server 15.0.2 已修改的系统目录更改。</p> <p>sp_downgrade @cmd = {'prepare'   'downgrade'   'help'}, @toversion = 'n'[, @verbose = 0   1][, @override = 0   1]</p>
sp_dropalias	<p>删除通过 sp_addalias 建立的别名用户名称标识。</p> <p>sp_dropalias <i>loginame</i> [, <i>force</i>]</p>
sp_drop_all_qplans	<p>删除抽象计划组中的所有抽象计划。</p> <p>sp_drop_all_qplans <i>name</i></p>
sp_drop_qpgroup	<p>删除抽象计划组。</p> <p>sp_drop_qpgroup <i>group</i></p>
sp_drop_qplan	<p>删除抽象计划。</p> <p>sp_drop_qplan <i>id</i></p>
sp_drop_resource_limit	<p>从 Adaptive Server 中删除一个或多个资源限制。</p> <p>sp_drop_resource_limit { <i>name</i>, <i>appname</i> } [, <i>rangename</i>, <i>limittype</i>, <i>enforced</i>, <i>action</i>, <i>scope</i>]</p>
sp_drop_time_range	<p>从 Adaptive Server 中删除用户定义的时间范围。</p> <p>sp_drop_time_range <i>name</i></p>
sp_dropdevice	<p>删除 Adaptive Server 数据库设备或转储设备。</p> <p>sp_dropdevice <i>logicalname</i></p>
sp_dropengine	<p>sp_dropengine 不在线程化模式中运行。</p> <p>sp_dropengine <i>engine_number</i> [, <i>engine_group</i>] [, <i>instance_id</i>]</p>
sp_dropexeclass	<p>删除用户定义的执行类。</p> <p>sp_dropexeclass <i>classname</i></p>

<code>sp_dropextendedproc</code>	删除扩展存储过程 (ESP)。 <code>sp_dropextendedproc <i>esp_name</i></code>
<code>sp_dropexternlogin</code>	(仅限组件集成服务) 删除以前通过 <code>sp_addexternlogin</code> 定义的远程登录的定义。 <code>sp_dropexternlogin <i>server</i> [, <i>loginame</i> [, <i>rolename</i> ]]</code>
<code>sp_dropglockpromote</code>	从表或数据库中删除锁升级值。 <code>sp_dropglockpromote {"database"   "table"}, <i>objname</i></code>
<code>sp_dropgroup</code>	从数据库中删除组。 <code>sp_dropgroup <i>grpname</i></code>
<code>sp_dropkey</code>	从 <code>syskeys</code> 表中删除使用 <code>sp_primarykey</code> 、 <code>sp_foreignkey</code> 或 <code>sp_commonkey</code> 定义的键。 <code>sp_dropkey <i>keytype</i>, <i>tablename</i> [, <i>deptabname</i>]</code>
<code>sp_droplanguage</code>	从服务器中删除替代语言并从 <code>master.dbo.syslanguages</code> 中删除相关的行。 <code>sp_droplanguage <i>language</i> [, <i>dropmessages</i>]</code>
<code>sp_dropmessage</code>	从 <code>sysusermessages</code> 中删除用户定义的消息。 <code>sp_dropmessage <i>message_num</i> [, <i>language</i>]</code>
<code>sp_dropobjectdef</code>	(仅限组件集成服务) 删除为本地对象提供的外部存储映射。 <code>sp_dropobjectdef <i>tablename</i></code>
<code>sp_dropremotelogin</code>	删除远程用户登录名。 <code>sp_dropremotelogin <i>remoteserver</i> [, <i>loginame</i> [, <i>remotename</i> ]]</code>
<code>sp_droprowlockpromote</code>	从数据库或表中删除行锁升级阈值。 <code>sp_droprowlockpromote {"database"   "table"}, <i>objname</i></code>
<code>sp_dropsegment</code>	从数据库中删除段或从特定的数据库设备取消段的映射。 <code>sp_dropsegment <i>segname</i>, <i>dbname</i> [, <i>device</i>]</code>
<code>sp_dropserver</code>	从已知服务器列表中删除服务器, 或在同一操作中删除远程登录名和外部登录名。 <code>sp_dropserver <i>server</i> [, <i>droplogins</i>]</code>
<code>sp_dropthreshold</code>	从段中删除可用空间阈值。 <code>sp_dropthreshold <i>dbname</i>, <i>segname</i>, <i>free_space</i></code>
<code>sp_droptype</code>	删除用户定义的数据类型。 <code>sp_droptype <i>typename</i></code>

---

sp_dropuser	<p>从当前数据库中删除用户。</p> <pre>sp_dropuser <i>name_in_db</i></pre>
sp_dumpoptimize	<p>指定在转储数据库操作期间 Backup Server 转储的数据量。</p> <pre>sp_dumpoptimize [ 'archive_space = {maximum   minimum   default }' ] sp_dumpoptimize [ 'reserved_threshold = {nnn   default }' ] sp_dumpoptimize [ 'allocation_threshold = {nnn   default }' ]</pre>
sp_encryption	<p>报告加密信息。</p> <pre>sp_encryption help   helpkey sp_encryption help   helpkey [ , <i>key_name</i>   wildcard ] [ , all_dbs   key_copy   display_cols ] sp_encryption help   'helpkey', { 'master'   'dual master' } [ , 'display_keys'   'all_dbs' ] sp_encryption 'help' [ , 'servicekeyname' [ , 'display_objs' ] ] sp_encryption 'helpextpasswd' sp_encryption helpcol [ , <i>table_name</i>   <i>column_name</i> ] sp_encryption helpuser [ , <i>user_name</i>   wildcard ] [ , key_copy ] sp_encryption 'mkey_startup_file' [ , { 'new_path'   'default_location'   'null' } ] [ , { sync_with_mem   sync_with_qrm } ] ] sp_encryption 'downgrade_kek_size' [ , 'true'   'false' ] sp_encryption system_encr_passwd, 'newpasswd' [ , 'oldpasswd' ]</pre>
sp_engine	<p>使您可以让引擎联机或脱机。在线程化模式下，使用 alter thread pool 可让引擎联机。</p> <pre>sp_engine { "online"   [offline   can_offline] [ , <i>engine_id</i> ]   ["shutdown", <i>engine_id</i>] }</pre>
sp_estspace	<p>估计表及其索引要求的空间量以及创建索引需要的时间。</p> <pre>sp_estspace <i>table_name</i>, <i>no_of_rows</i>, <i>fill_factor</i>, <i>cols_to_max</i>, <i>textbin_len</i>, <i>iosec</i>, <i>page_size</i></pre>
sp_export_qpgroup	<p>将指定用户和抽象计划组的所有计划导出到用户表。</p> <pre>sp_export_qpgroup <i>usr</i>, <i>group</i>, <i>tab</i></pre>
sp_extendsegment	<p>将段的范围扩展到其它数据库设备。</p> <pre>sp_extendsegment <i>segname</i>, <i>dbname</i>, <i>devname</i></pre>
sp_extengine	<p>启动和停止 EJB Server。显示 EJB Server 的状态信息。</p> <pre>sp_extengine 'ejb_server', '{ start   stop   status }'</pre>

<code>sp_extrapwdchecks</code>	<p>一个自定义存储过程，它可以包含用于口令复杂程度检查的用户定义的逻辑。您可以根据安全性需要对 <code>sp_extrapwdchecks</code> 进行配置。</p> <p><code>sp_extrapwdchecks</code> 安装在 master 数据库中。</p> <p><code>sp_extrapwdchecks caller_password, new_password, login_name</code></p>
<code>sp_familylock</code>	<p>报告由并行执行某个语句的一系列进程（协调进程及其工作进程）持有的所有锁的相关信息。</p> <p><code>sp_familylock [fpid1 [, fpid2]]</code></p>
<code>sp_find_qplan</code>	<p>在给定的查询文本或计划文本的模式的情况下查找抽象计划。</p> <p><code>sp_find_qplan pattern [, group]</code></p>
<code>sp_fixindex</code>	<p><code>sp_fixindex</code> 可在系统表损坏时修复其上的一组索引（而非单个索引）。如果目标表具有位置索引或聚簇索引，<code>sp_fixindex</code> 将重建数据层（它在处理系统表的位置索引或聚簇索引时回收数据层中未使用的空间）。</p> <p><code>sp_fixindex database_name, table_name [, index_id   null] [, index_name   null] [, force_option]</code></p>
<code>sp_flushstats</code>	<p>将内存中的统计信息刷新到 <code>systabstats</code> 和 <code>sysstatistics</code> 系统表中。</p> <p><code>sp_flushstats [objname]</code></p>
<code>sp_forceonline_db</code>	<p>提供对数据库中所有页的访问，这些页以前被恢复过程标记为可疑。</p> <p><code>sp_forceonline_db dbname, {sa_on   sa_off   all_users}</code></p>
<code>sp_forceonline_object</code>	<p>提供对以前被恢复过程标记为可疑的索引的访问。</p> <p><code>sp_forceonline_object dbname, objname, indid, {sa_on   sa_off   all_users} [, no_print]</code></p>
<code>sp_forceonline_page</code>	<p>提供对以前被恢复过程标记为可疑的页的访问。</p> <p><code>sp_forceonline_page dbname, pgid, {sa_on   sa_off   all_users}</code></p>
<code>sp_foreignkey</code>	<p>在当前数据库中定义表或视图的外键。</p> <p><code>sp_foreignkey tabname, pktabname, col1 [, col2] ... [, col8]</code></p>
<code>sp_freedll</code>	<p>卸载以前装入 XP Server 内存的动态链接库 (DLL)，以支持扩展存储过程 (ESP) 的执行。</p> <p><code>sp_freedll dll_name</code></p>
<code>sp_getmessage</code>	<p>从 <code>print</code> 和 <code>raiserror</code> 语句的 <code>sysmessages</code> 和 <code>sysusermessages</code> 检索存储的消息字符串。</p> <p><code>sp_getmessage message_num, result output [, language]</code></p>

---

sp_grantlogin	<p>（仅限 Windows）当集成安全模式或混合模式（具有命名管道）处于活动状态时，将 Adaptive Server 角色或 default 权限指派给 Windows 用户和组。</p> <pre>sp_grantlogin {login_name   group_name}                ["role_list"   default]</pre>
sp_ha_admin	<p>对高可用性系统中的配置了 Sybase 故障切换功能的 Adaptive Server 执行管理任务。sp_ha_admin 是使用 <i>installhavss</i> 脚本（UNIX 平台）或 <i>insthasv</i> 脚本（Windows NT）安装的。</p> <pre>sp_ha_adminin [cleansessions   help]</pre>
sp_help	<p>报告与数据库对象（在 sysobjects 中列出的任何对象）、系统数据类型或用户定义的数据类型、用户定义的函数以及计算列和基于函数的索引有关的信息。列显示 optimistic_index_lock。</p> <pre>sp_help [objname]</pre>
sp_help_resource_limit	<p>报告资源限制的有关信息。</p> <pre>sp_help_resource_limit [name [, appname [, limittime                         [, limitday [, scope [, action[, verbose]]]]]]]</pre>
sp_help_qpgroup	<p>报告有关抽象计划组的信息。</p> <pre>sp_help_qpgroup [ group [, mode ]]</pre>
sp_help_qplan	<p>报告有关抽象计划的信息。</p> <pre>sp_help_qplan id [, mode ]</pre>
sp_helpapptrace	<p>确定 Adaptive Server 正在跟踪的会话。sp_helpapptrace 会返回 Adaptive Server 正在跟踪的所有会话的服务器进程 ID (spid)、对其进行跟踪的会话的 spid 以及跟踪文件的名称。</p> <pre>sp_helpapptrace</pre>
sp_helppartition	<p>列出表或索引的有关分区的信息。</p> <pre>sp_helppartition [ tablename [, { null   indexname   'all' }[, partitionname ] ]]</pre>
sp_helpcache	<p>显示与绑定到数据高速缓存的对象或指定高速缓存大小所需的开销量有关的信息。</p> <pre>sp_helpcache {cache_name   "cache_size[P   K   M   G]" ,               "instance_instance_name"}</pre>
sp_helpcomputedcolumn	<p>报告有关指定表中的计算列的信息。</p> <pre>sp_helpcomputedcolumn {tablename}</pre>
sp_helpconfig	<p>报告有关配置参数的帮助信息。</p> <pre>sp_helpconfig "configname"[, "size"]</pre>

sp_helpconstraint	报告有关在指定表中使用的完整性约束的信息。 sp_helpconstraint [objname][, detail]
sp_helpdb	报告有关特定数据库或所有数据库的信息。 sp_helpdb [dbname [, order]]
sp_helpdevice	报告有关特定设备或有关所有 Adaptive Server 数据库设备和转储设备的信息。 sp_helpdevice [devname]
sp_helpextendedproc	显示当前数据库中的扩展存储过程 (ESP) 以及与其相关联的 DLL 文件。 sp_helpextendedproc [esp_name]
sp_helpexternlogin	(仅限组件集成服务) 报告有关外部登录名的信息。 sp_helpexternlogin [server[, loginame[, rolename]]]
sp_helpgroup	报告有关当前数据库中的特定组或所有组的信息。 sp_helpgroup [grpname]
sp_helpindex	报告有关表中创建的索引的信息。报告有关计算列索引和基于函数的索引的信息。 sp_helpindex objname
sp_helpjava	显示有关在数据库中安装的 Java 类和关联 JAR 的信息。 sp_helpjava ["class"[, java_class_name[, "detail"   "depends"]]   "jar", jar_name[, "depends"]]
sp_helpjoins	列出两个表或视图中适于连接的列。 sp_helpjoins lefttab, righttab
sp_helpkey	报告有关特定表或视图的主键、外键或公用键的信息，或有关当前数据库中的所有键的信息。 sp_helpkey [tablename]
sp_helplanguage	报告有关特定替代语言或所有语言的信息。 sp_helplanguage [language]
sp_helplog	报告包含事务日志第一页的设备的名称。 sp_helplog
sp_helpobjectdef	(仅限组件集成服务) 报告远程对象定义的所有者、对象和类型信息。 sp_helpobjectdef [objname]
sp_helpremotelogin	报告有关特定远程服务器的登录或有关所有远程服务器的登录的信息。 sp_helpremotelogin [remoteserver[, remotename]]



---

sp_helpprotect	报告数据库对象、用户、组或角色的权限。 <code>sp_helpprotect [name[, username[, "grant" [, "none"   "granted"   "enabled"   role_name[, permission_name]]]]]</code>
sp_helpsegment	报告有关当前数据库中的特定段或所有段的信息。 <code>sp_helpsegment [segname]</code>
sp_helpserver	报告有关特定远程服务器或所有远程服务器的信息。 <code>sp_helpserver [server]</code>
sp_helpsort	显示 Adaptive Server 的缺省排序顺序和字符集。 <code>sp_helpsort</code>
sp_helptext	显示编译对象的源文本，以及用户定义的函数、计算列或基于函数的索引定义的源文本。 <code>sp_helptext objname[.grouping_num][, numlines[, printopts]]</code>
sp_helpthread	显示当前线程池配置。 <code>sp_helpthread [pool_name]</code>
sp_helpthreshold	报告与当前数据库中所有阈值或特定段中所有阈值关联的段、可用空间值、状态和存储过程。 <code>sp_helpthreshold [segname]</code>
sp_helpuser	报告有关特定用户、组或别名的信息或者有关当前数据库中所有用户的信息。另外还标识用户拥有的对象和用户定义数据类型。 <code>sp_helpuser [name_in_db [, display_object]]</code>
sp_hidetext	隐藏指定编译对象的源文本，以及计算列和基于函数的索引键的文本。 <code>sp_hidetext [objname[, tabname[, username]]]</code>
sp_import_qpgroup	将抽象计划从用户表导入到抽象计划组中。 <code>sp_import_qpgroup tab, usr, group</code>
sp_indsuspect	检查用户表中排序顺序更改后的恢复期间标记为可疑的索引。 <code>sp_indsuspect [tab_name]</code>
sp_jreconfig	管理 Java PCA/JVM。启用或禁用参数和指令、更改配置值和报告配置值。 <code>sp_jreconfig{   add array_arg, new_string     array_clear array_arg     array_enable array_arg     array_disable array_arg     delete array_arg, string_value     disable { directive   argument   array_arg, string_value }  </code>

```

enable { directive | argument | array_arg, string_value } |
list { list_type [, formatted ] | units | units, units_type[, formatted ] } |
reload_config |
report { directive[, formatted ] | directive, args[, formatted ]
|argument[, formatted ] } |
update { argument, old_value, new_value } }

```

sp\_ldapadmin

创建或列出 LDAP URL 搜索字符串、检验 LDAP URL 搜索字符串或登录名，或者指定访问帐户和可调优的 LDAPUA 相关参数。

```
sp_ldapadmin command [, option1 [, option2]]
```

有效的 *command* [, *option1* [, *option2*]] 选项为：

```

'set_primary_url', 'url'
'set_secondary_url', 'url'
'set_dn_lookup_url', 'url'
'set_secondary_dn_lookup_url', 'url'
'set_access_acct', 'distinguished_name', 'password'
'set_secondary_access_acct', 'distinguished_name', 'password'
'set_failback_interval', time_in_minutes
'suspend', {'primary' | 'secondary'}
'activate', {'primary' | 'secondary'}
'list'
'list_urls'
'list_access_acct'
'check_url', 'url'
'reinit_descriptors'
'check_login', 'name'
'set_timeout', timeout_in_milli_seconds
'set_log_interval', log_interval_in_minutes
'set_num_retries', num_retries
'set_max_ldapua_native_threads', max_ldapua_native_threads
'set_max_ldapua_desc', max_ldapua_desc
'set_abandon_ldapua_when_full', {true|false}
'starttls_on_primary', {true|false}
'starttls_on_secondary', {true|false}
'help'

```

sp\_listener

按服务器动态启动和停止任何给定端口上的 Adaptive Server 监听器。

对于线程化模式，语法是：

```
sp_listener "command", "server_name | network", remaining
```

或：

```
sp_listener "command", "[protocol:]machine:port:CN=common_name"
```

对于进程模式，语法是：

```
sp_listener "command", "server_name | network", engine | remaining
```

---

	或:
	<code>sp_listener "command", "[protocol:]machine:port:"CN=common_name", engine</code>
<code>sp_listsuspect_db</code>	列出由于恢复时检测到损坏而当前具有脱机页的所有数据库。
	<code>sp_listsuspect_db</code>
<code>sp_listsuspect_object</code>	列出数据库中由于在恢复期间检测到损坏情况而导致当前脱机的所有索引。
	<code>sp_listsuspect_object [dbname]</code>
<code>sp_listsuspect_page</code>	列出数据库中由于恢复时检测到损坏而当前处于脱机状态的所有页。
	<code>sp_listsuspect_page [dbname]</code>
<code>sp_lmconfig</code>	在 Adaptive Server 上配置与许可证管理相关的信息。
	<code>sp_lmconfig</code> <code>[</code> <code>    ['edition' [, <i>edition_type</i> ]]</code> <code>    ['license type' [, <i>license_type_name</i> ]]</code> <code>    ['smtp host' [, <i>smtp_host_name</i> ]]</code> <code>    ['smtp port' [, <i>smtp_port_number</i> ]]</code> <code>    ['email sender' [, <i>sender_email_address</i> ]]</code> <code>    ['email recipients' [, <i>email_recipients</i> ]]</code> <code>    ['email severity' [, <i>email_severity</i> ]]</code> <code>]</code>
<code>sp_lock</code>	报告有关当前持有锁的进程的对象名和 ID。
	<code>sp_lock [spid1[, spid2]]   [@verbose = int]</code>
<code>sp_locklogin</code>	锁定 Adaptive Server 帐户，防止用户登录；或显示所有锁定帐户的列表。
	<code>sp_locklogin login   NULL   wildcard_string , "lock"   "unlock",</code> <code>    [except_login_name   except_role_name]</code> <code>    [, number_of_inactive_days]</code>
	或:
	<code>sp_locklogin</code>
<code>sp_logdevice</code>	将日志和数据在同一设备上的数据库的事务日志移到单独的数据库设备。
	<code>sp_logdevice dbname, devname</code>
<code>sp_loginconfig</code>	(仅限 Windows NT) 显示一个或所有集成安全性参数的值。
	<code>sp_loginconfig ["parameter_name"]</code>
<code>sp_logininfo</code>	(仅限 Windows NT) 显示通过 <code>sp_grantlogin</code> 授予 Windows NT 用户和组的所有角色。
	<code>sp_logininfo ["login_name"   "group_name"]</code>

sp_logiosize	<p>为当前数据库的事务日志执行 I/O 时，将 Adaptive Server 使用的日志 I/O 大小更改为不同的内存池。</p> <pre>sp_logiosize ["default"   "size"   "all"]</pre>
sp_logintrigger	<p>设置并显示全局登录触发器。全局登录触发器与个人登录脚本具有相同的特点。对于每个试图登录的用户，包括系统管理员和系统安全员，全局登录触发器都会在任何个人登录脚本之前执行。</p> <pre>sp_logintrigger 'global login trigger name'</pre>
sp_maplogin	<p>将外部用户映射到 Adaptive Server 登录名。</p> <pre>sp_maplogin (authentication_mech   null), (client_username   null),              (action   login_name   null)</pre>
sp_merge_dup_inline_default	<p>删除现有重复行内缺省对象，将唯一行内缺省对象转换为可共享行内缺省对象。</p> <pre>sp_merge_dup_inline_default [report_only = {yes   no}                              [, show_progress = {yes   no}]]</pre>
sp_metrics	<p>备份、删除和刷新 QP 指标（总是在作为各自数据库中的第 1 组的缺省运行组中捕获）及其查询的统计信息。</p> <pre>sp_metrics ['backup' backup_group_ID   'drop', 'gid' [, 'id' ]               'flush'   'help', 'command']</pre>
sp_modify_resource_limit	<p>通过指定新的限制值和/或在超出该限制时要执行的操作，更改资源限制。</p> <pre>sp_modify_resource_limit {name, appname}                         rangename, limittype, limitvalue, enforced, action, scope</pre>
sp_modify_time_range	<p>更改与指定时间范围关联的开始日期、开始时间、结束日期和/或结束时间。</p> <pre>sp_modify_time_range name, startday, endday, starttime, endtime</pre>
sp_modifystats	<p>允许系统管理员或拥有权限的任何用户执行该过程和更新目标表的统计信息，以修改 sysstatistics 中的列的密度值。</p> <pre>sp_modifystats [database].[owner].table_name,                 {"column_group"   "all"},                 MODIFY_DENSITY,                 {range   total},                 {absolute   factor},                 "value"</pre> <p>或：</p> <pre>sp_modifystats [database].[owner].table_name, column_name,                 REMOVE_SKEW_FROM_DENSITY</pre>

---

sp_modifythreshold	<p>通过使阈值与其它阈值过程、可用空间级别或段名关联，对该阈值进行修改。</p> <pre>sp_modifythreshold <i>dbname, segname, free_space</i> [, <i>new_proc_name</i>][, <i>new_free_space</i>][, <i>new_segname</i>]</pre>
sp_monitor	<p>显示有关 Adaptive Server 的统计信息。</p> <pre>sp_monitor [[<i>connection</i>   <i>statement</i>], [<i>cpu</i>   <i>diskio</i>   <i>elapsed time</i>]] [<i>event</i>, [<i>spid</i>]] [<i>procedure</i>, [<i>dbname</i>, [<i>procname</i>], <i>summary</i>   <i>detail</i>]]] [<i>enable</i>] [<i>disable</i>] [<i>help</i>], [<i>deadlock</i>][<i>procstack</i>]</pre>
sp_monitorconfig	<p>显示与索引、对象、数据库和内核资源内存池的元数据描述符有关的高速缓存使用情况统计信息。sp_monitorconfig 还报告用于参照完整性查询的辅助扫描描述符的有关统计信息以及事务描述符和 DTX 参与者的使用情况统计信息。</p> <pre>sp_monitorconfig "<i>configname</i>"[, "<i>result_tbl_name</i>"][, "full"]</pre>
sp_monitor_server	<p>提供服务器范围的监控信息。</p> <pre>sp_monitor_server [<i>server_name</i>]</pre>
sp_object_stats	<p>显示表和索引的锁争用、锁等待时间和死锁统计信息。</p> <pre>sp_object_stats <i>interval</i>[, <i>top_n</i>[, <i>dbname</i>, <i>objname</i>[, <i>rpt_option</i>]]]</pre>
sp_opt_querystats	<p>返回选定查询的性能分析。</p> <pre>sp_opt_querystats "<i>query_text</i>"   <i>help</i> [, "<i>diagnostic_options</i>"   <i>null</i> [, <i>database_name</i>] [, <i>user_name</i>]]</pre>
sp_options	<p>显示选项值。</p> <pre>sp_options [ [<i>show</i>   <i>help</i> [, <i>option_name</i>   <i>category_name</i>   <i>null</i> [, <i>dflt</i>   <i>non_dflt</i>   <i>null</i> [, <i>spid</i>]]]]]</pre>
sp_passthru	<p>(仅限组件集成服务) 使用户可以将 SQL 命令缓冲区传递到远程服务器。</p> <pre>sp_passthru <i>server, command, errcode, errmsg, rowcount</i> [, <i>arg1, arg2, ... argn</i>]</pre>
sp_passwordpolicy	<p>一个接口，具有 sso_role 的用户可以使用该接口配置登录名和口令策略选项。</p> <p>指定、删除并列出新的口令复杂程度选项：</p> <pre>sp_passwordpolicy {"set"   "clear"   "list"}, <i>policy_option, option_value</i> <p>检验口令复杂程度选项：</p> <pre>sp_passwordpolicy 'validate password options'</pre> </pre>

为网络登录口令加密生成非对称密钥对：

```
sp_passwordpolicy "regenerate keypair"
```

使口令到期：

```
sp_passwordpolicy "expire role passwords", "[rolename | wildcard]"
sp_passwordpolicy "expire login passwords", "[login_name | wildcard]"
sp_passwordpolicy "expire stale role passwords", "datetime"
sp_passwordpolicy "expire stale login passwords", "datetime"
```

显示所有命令、选项及其值的简要说明：

```
sp_passwordpolicy "help"
```

sp\_pciconfig

管理 Java PCI Bridge。启用或禁用参数和指令、更改配置值和报告配置值。请勿使用 sp\_pciconfig 更改参数或指令，除非 Sybase 技术支持部门指示这样做。

```
sp_pciconfig{
    disable { directive | argument } |
    enable { directive | argument } |
    list { list_type [, formatted ] | units | units, units_type[, formatted ] } |
    report { directive[, formatted ] |
        directive, args[, formatted ] |
        argument[, formatted ] } |
    update { number_arg, old_value new_value } }
```

sp\_placeobject

将表或索引的将来空间分配放置于特定段上。

```
sp_placeobject segname, objname
```

sp\_plan\_dbccdb

为新的 dbccdb 和 dbccalt 数据库推荐合适的大小、为 dbccdb 和 dbccalt 列出合适的设备，并为目标数据库建议高速缓存大小和合适的工作进程数。

```
sp_plan_dbccdb [dbname]
```

sp\_poolconfig

创建、删除数据高速缓存中的内存池并调整其大小，同时提供有关内存池的信息。

在现有高速缓存中创建内存池或者更改内存池的大小：

```
sp_poolconfig cache_name[, "mem_size [P | K | M | G]", "config_poolK"
    [, "affected_pool K", instance instance_name]
```

更改池的清洗大小：

```
sp_poolconfig cache_name, "affected_poolK", "wash=size[P|K|M|G]"
```

更改池的异步预取百分比：

```
sp_poolconfig cache_name, "affected_poolK",
    "local async prefetch limit=percent "
```

---

sp_post_xpload	在跨字节顺序类型不同的平台执行完 load database 后，检查并重建索引。 sp_post_xpload
sp_primarykey	定义表或视图的主键。 sp_primarykey <i>tablename</i> , <i>col1</i> [, <i>col2</i> , <i>col3</i> , ..., <i>col8</i> ]
sp_processmail	(仅限 Windows NT) 使用 xp_findnextmsg、xp_readmail、xp_sendmail 和 xp_deletemail 系统扩展存储过程 (ESP)，阅读、处理、发送和删除 Adaptive Server 收件箱中的邮件。 sp_processmail [ <i>subject</i> ] [, <i>originator</i> ] [, <i>dbuser</i> ] [, <i>dbname</i> ] [, <i>filetype</i> ] [, <i>separator</i> ]]]]
sp_procxmode	显示或更改与存储过程关联的执行模式。 sp_procxmode [ <i>procname</i> ] [, <i>tranmode</i> ]]
sp_querysmobj	(仅限 Tivoli Storage Manager) 在 Tivoli Storage Manager (TSM) 中查询一系列 Adaptive Server 备份对象。仅在您的站点已获得 TSM 许可时才支持 sp_querysmobj。 sp_querysmobj "syb_tsm", "output_file", "server_name" {, "database_name", "object_name", "dump_type", "until_time", "bs_name"}
sp_recompile	导致使用指定表的各个存储过程和触发器在下次运行时重新编译。 sp_recompile <i>objname</i>
sp_refit_admin	(对于集群环境) 提供接口以执行各种与 disk refit 相关的操作，例如显示 disk refit 进程的当前状态、重置 disk refit 进程的状态、跳过实例的 disk refit 进程等。 sp_refit_admin ['help']   'status'   ['reset'   'skiperfit' [, <i>instance_name</i> ]] [[ 'removedevice', <i>device_name</i> ]
sp_remap	重新映射来自版本 4.8 之后、版本 10.0 之前的数据库的存储过程、触发器、规则、缺省值或视图，使其与版本 10.0 和更高版本兼容。对于升级过程未能重新映射的原有对象，请使用 sp_remap。 sp_remap <i>objname</i>
sp_remoteoption	显示或更改远程登录选项。 sp_remoteoption [ <i>remoteserver</i> ] [, <i>loginame</i> ] [, <i>remotename</i> ] [, <i>optname</i> ] [, <i>optvalue</i> ]]]]
sp_remotesql	(仅限组件集成服务) 建立与远程服务器的连接，将查询缓冲区从客户端传递到远程服务器，然后再将结果传递回客户端。 sp_remotesql <i>server</i> , <i>query</i> [, <i>query2</i> , ..., <i>query254</i> ]

sp_rename	更改当前数据库中用户创建的对象或用户定义的数据类型的名称。 <code>sp_rename objname, newname [,"index"   "column"]</code>
sp_rename_qpgroup	重命名抽象计划组。 <code>sp_rename_qpgroup old_name, new_name</code>
sp_renamedb	更改用户数据库的名称。 <code>sp_renamedb dbname, newname</code>
sp_reportstats	报告有关系统使用情况的统计信息。 <code>sp_reportstats [loginame]</code>
sp_revokelogin	(仅限 Windows NT) 当集成安全模式或混合模式 (具有命名管道) 处于活动状态时, 撤消 Windows NT 用户和组的 Adaptive Server 角色和缺省权限。 <code>sp_revokelogin {login_name   group_name}</code>
sp_role	授予 Adaptive Server 登录帐号角色, 或撤销其角色。 <code>sp_role {"grant"   "revoke"}, rolename, loginame</code>
sp_securityprofile	列出与登录配置文件关联的属性或绑定。 <code>sp_securityprofile 'attributes', 'login profile', {wildcard   login_profile_name   'default'} sp_securityprofile 'bindings', 'login profile' [, {wildcard   login_profile_name   'default'} [, 'login' ,{wildcard   login_name}]] sp_securityprofile 'help'</code>
sp_sendmsg	将消息发送到 “用户数据报协议” (UDP) 端口。 <code>sp_sendmsg ip_address, port_number, message</code>
sp_serveroption	显示或更改远程服务器选项。 <code>sp_serveroption [server, optname, optvalue]</code>
sp_set_qplan	更改现有计划的抽象计划文本, 而不更改相关查询。 <code>sp_set_qplan id, plan</code>
sp_setlangalias	指派或更改替代语言的别名。 <code>sp_setlangalias language, alias</code>
sp_setpglockpromote	设置或更改数据库、表或 Adaptive Server 的锁升级阈值。 <code>sp_setpglockpromote {"database"   "table"}, objname, new_lwm, new_hwm, new_pct sp_setpglockpromote server, NULL, new_lwm, new_hwm, new_pct</code>



---

<code>sp_setpsex</code>	<p>在会话处于活动状态时，为会话设置自定义执行属性。</p> <p><code>sp_setpsex <i>spid</i>, <i>exeattr</i>, <i>value</i></code></p>
<code>sp_setrowlockpromote</code>	<p>为某个数据行锁定表、数据库中的所有数据行锁定表或服务器上的所有数据行锁定表设置或更改行锁升级阈值。</p> <p><code>sp_setrowlockpromote "server", NULL, <i>new_lwm</i>, <i>new_hwm</i>, <i>new_pct</i></code>  <code>sp_setrowlockpromote {"database"   "table"}, <i>objname</i>, <i>new_lwm</i>,  <i>new_hwm</i>, <i>new_pct</i></code></p>
<code>sp_setsuspect_granularity</code>	<p>显示或设置用户数据库的恢复故障隔离模式，该模式控制在检测到数据损坏时将如何进行恢复。</p> <p><code>sp_setsuspect_granularity [<i>dbname</i>  [, "database"   "page" [, "read_only"]]]</code></p>
<code>sp_setsuspect_thresh old</code>	<p>显示或设置 Adaptive Server 在数据库中的可疑页数达到多少（允许的最大可疑页数）之后就会将整个数据库标记为可疑。</p> <p><code>sp_setsuspect_threshold [<i>dbname</i> [, <i>threshold</i>]]</code></p>
<code>sp_setup_table_transfer</code>	<p>在包含标记为增量传输的表的每个数据库中各运行一次，以在相应数据库中创建 <code>spt_TableTransfer</code> 表。</p> <p><code>sp_setup_table_transfer</code></p>
<code>sp_show_options</code>	<p>输出在当前会话中设置的所有服务器选项。</p> <p><code>sp_show_options</code></p>
<code>sp_showcontrolinfo</code>	<p>显示有关线程池指派、绑定的客户端应用程序、登录名和存储过程的信息。</p> <p><code>sp_showcontrolinfo [<i>object_type</i>, <i>object_name</i>, <i>spid</i>]</code></p>
<code>sp_showexeclass</code>	<p>显示与指定的执行类关联的执行类属性，以及与其关联的线程池名称。</p> <p><code>sp_showexeclass [<i>execlassname</i>]</code></p>
<code>sp_showoptstats</code>	<p><code>sp_showoptstats</code> 在功能上与 XML 文档中的 <code>optdiag</code> 独立实用程序类似，但采用系统过程格式，它提取和显示系统表（如 <code>systabstats</code> 和 <code>sysstatistics</code>）中的各种数据对象的统计信息和直方图。</p> <p><code>sp_showoptstats [[<i>database_name</i>.<i>owner</i>].<i>table_name</i>],  [<i>column_name</i>], [h]</code></p>
<code>sp_showplan</code>	<p>针对任何用户连接，显示当前 SQL 语句或同一批处理中前一语句的 <code>showplan</code> 输出。</p> <p><code>sp_showplan <i>spid</i>, <i>batch_id</i> output,  <i>context_id</i> output,  <i>stmt_num</i> output</code></p>

若要显示当前 SQL 语句的 showplan 输出而不指定 batch\_id、context\_id 或 stmt\_num，请使用以下语法：

```
sp_showplan spid, null, null, null
```

sp\_showpsex

显示 Adaptive Server 上运行的所有客户端会话的执行类、当前优先级和线程池关联。

```
sp_showpsex [spid]
```

sp\_spaceusage

报告表、索引或事务日志的空间使用情况并估计数据库中表和索引的碎片数。估计值是使用数据和索引行的平均行长度以及表中的行数计算的。可将空间使用情况和碎片数据存档，以用于以后的报告和趋势分析。sp\_spaceusage 支持多种操作，包括 help、display、archive 和 report，以指示当前的 Adaptive Server 空间使用情况。

“help” 操作语法：

```
sp_spaceusage 'help'[, 'all']
sp_spaceusage 'help' [, {'display' | 'display summary'
                          | 'report' | 'report summary' | 'archive'}
                        [, {'table' | 'index' | 'tranlog'}]]
```

“display” 操作语法：

```
sp_spaceusage 'display summary [using unit= {KB | MB | GB | PAGES}
]',
    {'table' | 'index'}, name
    [,where_clause [,order_by[,command] ] ]
sp_spaceusage 'display [using unit= {KB | MB | GB | PAGES} ]',
    {'table' | 'index'}, name
    [,select_list
    [,where_clause [,order_by[,command] ] ] ]
sp_spaceusage 'display [using unit={KB | MB | GB | PAGES} ]',
    'tranlog' [, name[,select_list[,where_clause [,order_by]]]]
```

“archive” 操作语法：

```
sp_spaceusage 'archive [ using_clause ]',
    {'table' | 'index'}, name[,where_clause[,command] ]
sp_spaceusage 'archive [ using_clause ]',
    'tranlog' [,name[,where_clause] ]
```

“report” 操作语法：

```
sp_spaceusage 'report summary [ using_clause ]',
    {'table' | 'index'}, name
    [,where_clause [,order_by[,from_date [,to_date]]]]
sp_spaceusage 'report [ using_clause ]',
    {'table' | 'index'}, name
    [,select_list[,where_clause [,order_by[,from_date [,to_date]]]]]
```

	<pre> sp_spaceusage 'report [ using_clause ]',     'tranlog' [, name     [,select_list[,where_clause [,order_by     [,from_date [,to_date]]]]]] using_clause = USING using_item [, using_item ...] using_item = { unit={ KB   MB   GB   PAGES }       dbname=database_name   prefix=string } </pre>
sp_spaceused	<p>显示指定的表或当前数据库中的所有表使用的行数、数据页数、索引大小和空间的估计值。</p> <pre> sp_spaceused [objname [,1] ] </pre>
sp_ssladmin	<p>添加、删除或显示 Adaptive Server 的服务器认证列表。</p> <pre> sp_ssladmin {[addcert, certificate_path [, password   NULL]] [dropcert, certificate_path] [lsert] [help]} [lsiphers] [setciphers, {"FIPS"   "Strong"   "Weak"   "All"   quoted_list_of_ciphersuites}] </pre>
sp_syntax	<p>显示 Transact-SQL 语句、系统过程、实用程序和 Adaptive Server 的其它例程的语法，具体取决于服务器上有哪些产品和相应的 sp_syntax 脚本。</p> <pre> sp_syntax word [, mod][, language] </pre>
sp_sysmon	<p>显示性能信息。</p> <pre> sp_sysmon begin_sample sp_sysmon { end_sample   interval }[, section[, applmon] ] [, 'cache wizard' [, top_N[, filter] ] ] </pre>
sp_tab_suspectptn	<p>列出具有可疑分区的表。基于字符的分区键的域分区表会在排序顺序更改后变为可疑，散列分区表会在跨平台转储装载后变为可疑。</p> <pre> sp_tab_suspectptn [table_name] </pre>
sp_tempdb	<p>sp_tempdb 允许用户：</p> <ul style="list-style-type: none"> <li>创建和管理临时数据库组。</li> <li>将用户或应用程序绑定到 default 或其它临时数据库组，或绑定到特定的本地临时数据库。</li> <li>管理到本地临时数据库和临时数据库组的绑定。</li> </ul> <pre> sp_tempdb [ [ { "create"   "drop" } , "groupname" ]   [ { "add"   "remove" } , "tempdbname", "groupname" ]   [ { "bind", "objtype", "objname", "bindtype", "bindobj" [, "scope", "hardness" ] }   { "unbind", "objtype", "objname" [, "scope" ] "instance_name" } ]   </pre>

	<pre>[ "unbindall_db", "tempdbname" ]   [ show [, "all"   "gr"   "db"   "login"   "app" [, "name" ] ]   [ who, "dbname" ] [ help ]]</pre>
sp_tempdb_markdrop	<p>(在集群环境中) 将本地系统临时数据库置于删除状态。</p> <pre>sp_tempdb_markdrop database_name [, {'mark'   'unmark'}]</pre>
sp_thresholdaction	<p>在日志段中的可用页数低于最后机会阈值时自动执行，除非该阈值与其它过程关联。Sybase 不提供此过程。</p> <p>达到阈值时，Adaptive Server 会按照位置将以下参数传递到阈值过程：</p> <pre>sp_thresholdaction @dbname,                    @segment_name,                    @space_left,                    @status</pre>
sp_tran_dumpable_status	<p>如果无法对数据库执行事务转储，sp_tran_dumpable_status 将显示无法转储的原因。</p> <pre>sp_tran_dumpable_status [database_name]</pre>
sp_transactions	<p>报告与活动事务有关的信息。</p> <pre>sp_transactions ["xid", xid_value]   ["state", {"heuristic_commit"   "heuristic_abort"   "prepared"   "indoubt"} [, "xactname"]]   ["gtrid", gtrid_value]</pre>
sp_unbindcache	<p>解除数据库、表、索引、text 对象或 image 对象与数据高速缓存的绑定。</p> <pre>sp_unbindcache dbname [, {owner}tablename [, indexname   "text only"]]</pre>
sp_unbindcache_all	<p>解除绑定到某个高速缓存的所有对象与该高速缓存的绑定。</p> <pre>sp_unbindcache_all cache_name</pre>
sp_unbindefault	<p>解除创建的缺省值与列或用户定义数据类型的绑定。</p> <pre>sp_unbindefault objname [, futureonly]</pre>
sp_unbindexclass	<p>在指定作用域内，删除以前与客户端应用程序、登录名、存储过程或缺省执行类关联的执行类属性。</p> <pre>sp_unbindexclass object_name, object_type, scope</pre>
sp_unbindmsg	<p>解除用户定义的消息与约束的绑定。</p> <pre>sp_unbindmsg constrname</pre>
sp_unbindrule	<p>解除规则与列或用户定义数据类型的绑定。</p> <pre>sp_unbindrule objname [, futureonly [, "accessrule"   "all"]]</pre>

---

**sp\_version** 返回上次运行的安装脚本（*installmaster*、*installdbccdb* 等）的版本信息以及关于它是否成功运行的信息。

`sp_version [script_file, [all]]`

**sp\_volchanged** 通知 Backup Server 操作员在转储或装载期间执行了所请求的卷处理操作。

`sp_volchanged session_id, devname, action  
[, fname [, vname]]`

**sp\_webservices** 创建和管理 Adaptive Server Web 服务引擎使用的代理表。

创建代理表：

`sp_webservices 'add', 'wsdl_uri' [, sds_name]  
[, 'method_name=proxy_table  
[,method_name=proxy_table]* ' ]`

显示 **sp\_webservices** 的用法信息：

`sp_webservices help [, 'option']`

列出映射到 WSDL 文件的代理表：

`sp_webservices 'list' [, 'wsdl_uri'] [, sds_name]`

修改超时设置：

`sp_webservices 'modify', 'wsdl_uri', 'timeout=time'`

删除映射到 WSDL 文件的代理表：

`sp_webservices 'remove', 'wsdl_uri' [, sds_name]`

### 用户定义的 Web 服务的选项

为用户定义的 Web 服务创建数据库别名：

`sp_webservices 'addalias' alias_name , database_name`

配置用户定义的 Web 服务：

`sp_webservices 'deploy', ['all' | 'service_name']`

在用户定义的 Web 服务中删除数据库别名：

`sp_webservices 'dropalias' alias_name`

列出用户定义的 Web 服务中映射到 WSDL 文件的代理表：

`sp_webservices 'listudws' [, 'service_name']`

列出用户定义的 Web 服务的别名或数据库别名。

`sp_webservices 'listalias'`

取消配置用户定义的 Web 服务：

`sp_webservices 'undeploy', ['all' | 'service_name']`

**sp\_who** 报告有关当前所有的 Adaptive Server 用户和进程的信息，或有关特定用户或进程的信息。包括 `thread_pool` 列，用以说明 Adaptive Server 用来执行任务的线程池。

`sp_who [loginame | "spid"]`

## 目录存储过程

以下是 Adaptive Server 目录存储过程的语法和简要说明。有关完整信息，请参见《参考手册：过程》。

**sp\_column\_privileges** 返回表或视图中的一列或多列的权限信息。

`sp_column_privileges table_name [, table_owner  
[, table_qualifier [, column_name]]]`

**sp\_columns** 返回可以存储在一列或多列中的数据类型的相关信息。

`sp_columns table_name [, table_owner]  
[, table_qualifier [, column_name]]`

**sp\_databases** 返回 Adaptive Server 中的数据库的列表。

`sp_databases`

**sp\_datatype\_info** 返回有关特定 ODBC 数据类型或所有 ODBC 数据类型的信息。

`sp_datatype_info [data_type]`

**sp\_fkeys** 返回使用 `create table` 或 `alter table` 命令在当前数据库中创建的外键约束的相关信息。

`sp_fkeys pktable_name [, pktable_owner]  
[, pktable_qualifier [, fktable_name]  
[, fktable_owner [, fktable_qualifier]]]`

**sp\_pkeys** 返回使用 `create table` 或 `alter table` 命令为单个表创建的主键约束的相关信息。

`sp_pkeys table_name [, table_owner]  
[, table_qualifier]`

**sp\_server\_info** 返回 Adaptive Server 属性名和当前值的列表。

`sp_server_info [attribute_id]`

**sp\_special\_columns** 返回唯一地标识表或视图中的行的一组最佳列；还可以返回一组 `timestamp` 列，这些列值是在事务更新行中的任意值后自动生成的。

`sp_special_columns table_name [, table_owner]  
[, table_qualifier [, col_type]]`

---

sp_sproc_columns	返回有关存储过程的输入参数和返回参数的信息。 <pre>sp_sproc_columns <i>procedure_name</i> [, <i>procedure_owner</i>]                 [, <i>procedure_qualifier</i>] [, <i>column_name</i>]</pre>
sp_statistics	返回单个表的索引列表。 <pre>sp_statistics <i>table_name</i> [, <i>table_owner</i>]                 [, <i>table_qualifier</i>] [, <i>index_name</i>] [, <i>is_unique</i>]</pre>
sp_stored_procedures	返回有关一个或多个存储过程的信息。 <pre>sp_stored_procedures [<i>sp_name</i> [, <i>sp_owner</i> [, <i>sp_qualifier</i>]]]</pre>
sp_table_privileges	返回表或视图中的所有列的特权信息。 <pre>sp_table_privileges <i>table_name</i> [, <i>table_owner</i>] [, <i>table_qualifier</i>]</pre>
sp_tables	返回可以出现在 from 子句中的对象的列表。 <pre>sp_tables [<i>table_name</i>] [, <i>table_owner</i>] [, <i>table_qualifier</i>] [, <i>table_type</i>]</pre>

## 扩展存储过程

以下是 Adaptive Server 扩展存储过程的语法和简要说明。有关完整信息，请参见《参考手册：过程》。

xp_cmdshell	在运行 Adaptive Server 的主机系统上执行本机操作系统命令。 <pre>xp_cmdshell <i>command</i> [, <i>no_output</i>] [<i>return_status</i>   <i>no_wait</i>]</pre>
xp_deletemail	(仅限 Windows NT) 从 Adaptive Server 收件箱中删除邮件。 <pre>xp_deletemail [<i>msg_id</i>]</pre>
xp_enumgroups	(仅限 Windows NT) 显示指定的 Windows NT 域的组。 <pre>xp_enumgroups [<i>domain_name</i>]</pre>
xp_findnextmsg	(仅限 Windows NT) 检索 Adaptive Server 收件箱中的下一个邮件标识符。 <pre>xp_findnextmsg @<i>msg_id</i> = @<i>msg_id</i> output[, <i>type</i>]                 [, <i>unread_only</i> = {true   false}]</pre>
xp_logevent	(仅限 Windows NT) 在 Adaptive Server 中向 Windows NT 事件日志记录用户定义的事件。 <pre>xp_logevent <i>error_number</i>, <i>message</i> [, <i>type</i>]</pre>

xp\_readmail

(仅限 Windows NT) 查阅 Adaptive Server 收件箱中的邮件。

```
xp_readmail [msg_id]
[, recipients output]
[, sender output]
[, date_received output]
[, subject output]
[, cc output]
[, message output]
[, attachments output]
[, suppress_attach = {true | false}]
[, peek = {true | false}]
[, unread = {true | false}]
[, msg_length output]
[, bytes_to_skip [output]]
[, type [output]]
```

xp\_sendmail

(仅限 Windows NT) 向指定的收件人发送邮件。邮件可以是文本，也可以是 Transact-SQL 查询的结果。

```
xp_sendmail recipient [, recipient] . . .
[, subject]
[, cc_recipient] . . .
[, bcc_recipient] . . .
[, {query | message}]
[, attachname]
[, attach_result = {true | false}]
[, echo_error = {true | false}]
[, include_file [, include_file] . . .]
[, no_column_header = {true | false}]
[, no_output = {true | false}]
[, width]
[, separator]
[, dbuser]
[, dbname]
[, type]
[, include_query = {true | false}]
```

xp\_startmail

(仅限 Windows NT) 启动 Adaptive Server 邮件会话。

```
xp_startmail [mail_user] [, mail_password]
```

xp\_stopmail

(仅限 Windows NT) 停止 Adaptive Server 邮件会话。

```
xp_stopmail
```



## dbcc 存储过程

以下是 Adaptive Server dbcc 存储过程的语法和简要说明。有关完整信息，请参见《参考手册：过程》。

<code>sp_dbcc_alterws</code>	将指定工作空间的大小改为指定值，并初始化该工作空间。
------------------------------	----------------------------

```
sp_dbcc_alterws dbname, wsname, "wssize[K|M]"
```

<code>sp_dbcc_configreport</code>	生成报告，描述指定数据库的 <code>dbcc checkstorage</code> 操作使用的配置信息。
-----------------------------------	---

sp\_dbcc\_configreport [*dbname*]

sp_dbcc_createws	在指定段与数据库上创建指定类型与大小的工作空间。
------------------	--------------------------

```
sp_dbcc_createws dbname, segname, [wsname], wstype, "wssize[K|M]"
```

sp_dbcc_deletedb	从 dbccdb 中删除与指定目标数据库相关的所有信息。
------------------	------------------------------

sp\_dbcc\_deletedb [*dbname* | *dbid*]

**sp\_dbcc\_deletehistory** 删除在指定日期和时间之前对目标数据库执行的 **dbcc checkstorage** 操作的结果。**sp\_dbcc\_deletehistory** 不释放与已删除的历史数据相关的任何空间，因为工作空间是预先分配的并具有固定大小。

```
sp_dbcc_deletehistory [cutoffdate [, dbname | dbid]]
```

sp\_dbcc\_differentialre  
port

```
sp_dbcc_differentialreport [dbname [, objectname]],
    [db_op] [, "date1" [, "date2"]]
```

sp_dbcc_evaluatedb	重新计算目标数据库的配置信息，并与当前配置信息进行比较。
--------------------	------------------------------

sp\_dbcc\_evaluatedb [dbname]

sp_dbcc_exclusions	允许用户创建和管理供 checkverify 和 sp_dbcc_faultreport 使用的持久排除列表。
--------------------	---

sp\_dbcc\_exclusions *dbname*, *op*, *type*, *exclusion\_list*

**sp\_dbcc\_faultreport** 生成一个报告，该报告中包含在指定日期针对目标数据库中的指定对象执行的 **dbcc checkstorage** 操作的故障统计信息。该报告按顺序列出表和索引。

```
sp_dbcc_faultreport [report_type [, dbname [, objectname  
[, date [, hard_only [, exclusion_mode[, exclusion_faults  
[, exclusion_tables [, exclusion_combo  
[, display_recommendations [, opid [, fault_type in]]]]]]]]]]]
```

sp\_dbcc\_fullreport 在指定的 *date* 或在此之前为 *database..object\_name* 运行  
sp\_dbcc\_summaryreport、sp\_dbcc\_configreport、sp\_dbcc\_statisticsreport 和  
sp\_dbcc\_faultreport short。

```
sp_dbcc_fullreport [dbname [, objectname [, date]]]
```

sp_dbcc_help_fault	提供指定故障类型和推荐修复的说明。 <code>sp_dbcc_help_fault [fault_type]</code>
sp_dbcc_patch_finishtime	简化中止的 checkverify 和 checkstorage 操作的报告。 <code>sp_dbcc_patch_finishtime dbname, opid [,optype [,seq [,finishtime]]]</code>
sp_dbcc_recommendations	分析与指定的操作 ID 或日期相对应的 checkstorage 操作报告的故障，并为目标数据库中的指定对象生成推荐的更正操作列表。 <code>sp_dbcc_recommendations dbname [, "date" [, opid [, "objectname"]]]</code>
sp_dbcc_runcheck	在指定数据库上运行 dbcc checkstorage，然后运行 sp_dbcc_summaryreport 或指定的报告。 <code>sp_dbcc_runcheck dbname [, user_proc]</code>
sp_dbcc_statisticsreport	为目标数据库中的指定对象生成分配统计信息报告。 <code>sp_dbcc_statisticsreport [dbname [, objectname [, date]]]</code>
sp_dbcc_summaryreport	生成指定数据库的摘要报告。 <code>sp_dbcc_summaryreport [dbname [, date [, op_name [, display_recommendations]]]]</code>
sp_dbcc_updateconfig	使用目标数据库的配置信息更新 dbccdb 中的 dbcc_config 表。 <code>sp_dbcc_updateconfig dbname, type, "str1" [, "str2"]</code>

---

## 系统表

以下是有关 Adaptive Server 系统表的简要说明。有关完整信息，请参见《参考手册：表》，或者，有关表、列以及表间关系的视觉表示形式，请参见《系统表框图》。

syblicenseslog	（仅限 master 数据库）对在 Adaptive Server 中使用的最大许可证数目的每次更新（每 24 小时进行一次）在 syblicenseslog 中都有对应的一行。列：status、logtime、maxlicenses。
sysalternates	（所有数据库）每个映射到（或者别名为）当前数据库用户的 Adaptive Server 用户在 sysalternates 中都有对应的一行。列：suid、altsuid
sysaltusages	（空数据库）将存档数据库中的页码映射到数据库转储及其分条或修改页面区域中的实际页。列：dbid、altsuid、lstart、start、size、vstart、vdevno、segmap
sysattributes	（所有数据库）定义对象的属性。列：class、attribute、object_type、object_cinfo、object_cinfo2、object、object_info1、object_info2、object_info3、int_value、char_value、text_value、image_value、comments
sysauditoptions	（sybsecurity 数据库）每个服务器范围的审计选项在 sysauditoptions 中都有对应的一行，用以指示该选项的当前设置。列：num、val、minval、maxval、name、sval、comment
sysaudits_01 – sysaudits_08	（sybsecurity 数据库）包含审计追踪。列：event、eventmod、spid、eventtime、sequence、suid、dbid、objid、xactid、loginname、dbname、objname、objowner、extrainfo、nodeid、instanceid
syscacheinfo	（master 数据库）提供有关数据高速缓存和池的配置的信息。列：cache_name、cache_status、cache_type、config_size、run_size、config_replacement、run_replacement、config_partitions、run_partitions、overhead、cacheid、instanceid、scope
syscachepoolinfo	为每个数据高速缓存池提供一行内容，用以包括数据高速缓存的信息。列：cache_name、cache_status、cache_type、cache_config_size、cache_run_size、cache_config_replacement、cache_run_replacement、cache_config_partitions、cache_run_partitions、cache_overhead、pool_io_size、pool_config_size、pool_run_size、pool_apf_percent、pool_wash_size、cacheid、instanceid、scope
syscharsets	（仅限 master 数据库）所定义的由 Adaptive Server 使用的每个字符集和排序顺序在 syscharsets 中都有对应的一行。列：type、id、csid、status、name、description、definition、sortfile

syscolumns	（所有数据库）每个表和视图中的每一列以及过程中的每个参数在 syscolumns 中都有对应的一行。列：id、number、colid、status、type、length、offset、usertype、cdefault、domain、name、printfmt、prec、scale、remote_type、remote_name、xstatus、xtype、xdbid、accessrule、status2、status3、computedcol、encrtype、lobcomp_lvl、encrlen、encrykeyid、encrykeydb、encrdate、inrowlen
syscomments	（所有数据库）每个视图、规则、缺省值、触发器、表约束、分区、过程、计算列、基于函数的索引键和其它形式的编译对象在 syscomments 中都有对应的条目。列：id、number、colid、texttype、language、text、colid2、status、partitionid
sysconfigures	（仅限 master 数据库）每个可由用户设置的配置参数在 sysconfigures 中都有对应的一行。列：config、value、comment、status、name、parent、value2、value3、value4、instanceid
sysconstraints	（所有数据库）每当用户使用 create table 或 alter table 声明新的检查约束或参照约束时，Adaptive Server 都将在 sysconstraints 表中插入一行。在用户执行 alter table 删除约束之前，该行将一直保留。列：colid、constrid、tableid、error、status、spare2
syscoordinations	（sysystemdb 数据库）包含有关参与分布式事务的远程 Adaptive Server（远程参与者）及其协调状态的信息。列：participant、starttime、coordtype、owner、protocol、state、bootcount、dbid、logvers、spare、status、xactkey、gtrid、partdata、srvname、nodeid、instanceid
syscurconfigs	（仅限 master 数据库）包含与每个配置参数对应的条目（与 sysconfigures 相同），但这些参数的值是当前值，而不是缺省值。另外，它还包含四行，用于说明配置结构。列：config、value、comment、status、value2、defvalue、minimum_value、maximum_value、memory_used、display_level、datatype、message_num、apf_percent、nodeid、instanceid、type
sysdatabases	（仅限 master 数据库）Adaptive Server 中的每个数据库在 sysdatabases 中都有对应的一行。在安装 Adaptive Server 后，master 数据库、model 数据库、sysystemprocs 和 tempdb 数据库在 sysdatabases 中都有相应的条目。如果已安装了审计功能，sybsecurity 数据库在其中也有相应的条目。列：name、dbid、suid、status、version、logptr、crdate、dumptrdate、status2、audflags、deftabaud、defvwaud、defpfraud、def_remote_type、def_remote_loc、status3、status4、audflags2、instanceid、durability
sysdepends	（所有数据库）每个由过程、视图或者触发器引用的过程、视图或表在 sysdepends 中都有对应的一行。列：id、number、depid、depnumber、status、selall、resultobj、readobj、columns

---

sysdevices	（仅限 master 数据库）每个磁带转储设备、磁盘转储设备、用于数据库的磁盘和用于数据库的磁盘分区在 sysdevices 中都有对应的一行。列：low、high、status、cntrltype、name、phyname、mirrorname、vdevno、crdate、resizedate、status2、instanceid、uuid
sysencryptkeys	数据库中创建的每个密钥（包括缺省密钥）在数据库特定的系统目录 sysencryptkeys 中都有相应的条目。列：id、ekalgorithm、type、status、eklen、value、uid、eksalt、ekpairid、pwwdate、expdate、ekpwdwarn
sysengines	（仅限 master 数据库）每个当前处于联机状态的 Adaptive Server 引擎在 sysengines 中都有对应的一行。列：engine、osprocid、osprocname、status、affinitied、cur_kpid、last_kpid、idle_1、idle_2、idle_3、idle_4、starttime、nodeid、instanceid
sysgams	（所有数据库）存储数据库的全局分配映射。
sysindexes	（所有数据库）每个聚簇索引、每个非聚簇索引、每个没有聚簇索引的表和每个包含 text 或 image 列的表在 sysindexes 中都有对应的一行。列：name、id、indid、doampg、ioampg、oampgtrips、status3、status2、ipgtrips、first、root、distribution、usagecnt、segment、status、maxrowsperpage、minlen、maxlen、maxirow、keycnt、keys1、keys2、soid、csid、base_partition、fill_factor、res_page_gap、exp_rowsize、keys3、identitygap、crdate、partitiontype、conditionid
sysinstances	报告实例状态的虚设表。集群配置中定义每个实例在 sysinstances 中都有对应的一行。列：id、name、state、hostname、starttime、connections_active、engines_online
sysjars	（所有数据库）数据库中保留的每个 Java 存档文件在 sysjars 中都有对应的一行。列：jid、jstatus、jname、jbinary
syskeys	（所有数据库）每个主键、外键或者公用键在 syskeys 中都有对应的一行。列：id、type、depid、keycnt、size、key1 ... key8、depkey1 ... depkey8、spare1
syslanguages	（仅限 master 数据库）Adaptive Server 已知的每种语言在 syslanguages 中都有对应的一行。列：langid、dateformat、datefirst、upgrade、name、alias、months、shortmonths、days
syslisteners	（仅限 master 数据库）每种可用来与当前 Adaptive Server 连接的网络协议在 syslisteners 中都有对应的一行。列：net_type、address_info、spare、nodeid、instanceid
syslocks	（仅限 master 数据库）包含有关活动锁的信息，它是在用户进行查询时动态创建的。列：id、dbid、page、type、spid、class、fid、context、row、loid、partitionid、nodeid、instanceid

sysloginroles	（仅限 master 数据库）拥有系统角色的服务器登录的每个实例在 sysloginroles 中都有对应的一行。列：suid、srid、status
syslogins	（仅限 master 数据库）每个有效的 Adaptive Server 用户帐户在 syslogins 中都有对应的一行。列：suid、status、accddate、totcpu、totio、spacelimit、timelimit、resultlimit、dbname、name、password、language、pwdate、audflags、fullname、srvname、logincount、procid、lastlogindate、crdate、locksuid、lockreason、lockdate、crsuid、lpid
syslogs	（所有数据库）包含事务日志。Adaptive Server 使用它来进行恢复。它对用户来说是无用的。列：xactid、op
syslogshold	（仅限 master 数据库）包含有关每个数据库的最早活动事务（如果有）以及事务日志的 Replication Server 截断点（如果有）的信息，但它不是常规表。它是在用户进行查询时动态创建的。列：dbid、reserved、spid、page、xactid、masterxactid、starttime、name、xloid
sysmessages	（仅限 master 数据库）Adaptive Server 可返回的每个系统错误或者警告在 sysmessages 中都有对应的一行。列：error、severity、dlevel、description、langid、sqlstate
sysmonitors	（仅限 master 数据库）每个监控计数器在 sysmonitors 中都有对应的一行。列：field_name、group_name、field_id、value、description、nodeid、instanceid
sysobjects	（所有数据库）每个表、视图、存储过程、扩展存储过程、日志、规则、缺省值、触发器、检查约束、参照约束、计算列、基于函数的索引键和（仅在 tempdb 中）临时对象以及其它形式的编译对象在 sysobjects 中都有对应的一行。对象 type 为 N 时，每个分区条件 ID 在该表中也都有相应的一行。列：name、id、uid、type、userstat、sysstat、indexdel、schemacnt、sysstat2、systat3、crdate、expdate、deltrig、instrig、updtrig、seltrig、ckfirst、cache、audflags、objspare、versionts、loginame、identburnmax、spacestate、erlchgts、lobcomp_lvl
sysoptions	（所有数据库）sp_options 查询的一个虚设表。列：spid、name、category、currentsetting、defaultsetting、scope、number
syspartitionkeys	（所有数据库）表的散列、范围和列表分区的每个分区键在 syspartitionkeys 中都有对应的一行。所有列都为非空。列：indid、id、colid、position
syspartitions	（所有数据库）每个数据分区和每个索引分区在 syspartitions 中都有对应的一行。列：name、indid、id、partitionid、segment、status、dataampage、indoampage、firstpage、rootpage、data_partitionid、crdate、cdataptname、lobcomp_lvl

---

syspoolinfo	（仅限 master 数据库）提供有关数据高速缓存和池的信息。列： cache_name、io_size、config_size、run_size、apf_percent、wash_size、 cacheid、instanceid、scope
sysprocedures	（所有数据库）每个视图、缺省值、规则、触发器、过程、声明缺省 值、分区条件、检查约束、计算列、基于函数的索引键和其它形式的编 译对象在 sysprocedures 中都有对应的条目。列：type、qp_setting、id、 sequence、status、number、version
sysprocesses	（仅限 master 数据库）包含有关 Adaptive Server 进程的信息，但它不是 常规表。列：spid、kpid、enginenum、status、suid、hostname、 program_name、hostprocess、cmd、cpu、physical_io、memusage、 blocked、dbid、uid、gid、tran_name、time_blocked、network_pktsz、 fid、execlass、priority、affinity、id、stmtnum、linenum、origsuid、 block_xloid、clientname、clienthostname、clientappliance、sys_id、 ses_id、loggedindatetime、ipaddr、nodeid、instanceid、pad、lcid
sysprotects	（所有数据库）包含有关用户、组和角色已经被授予或者被撤消的权限 的信息。列：id、uid、action、protecttype、columns、grantor、pred_id、 protstatus
sysquerymetrics	（所有数据库）从持久数据提供各个查询的集合的历史查询处理指标。 列：uid、gid、hashkey、id、sequence、exec_min、exec_max、 exec_avg、elap_min、elap_max、elap_avg、lio_min、lio_max、lio_avg、 pio_min、pio_max、pio_avg、cnt、abort_cnt、qtext
sysqueryplans	（所有数据库）每个抽象查询计划在 sysqueryplans 中都有对应的两行或 者更多行。它使用数据行锁。列：uid、dbid、qpddate、spcrid、 hashkey2、key1、key2、key3、gid、hashkey、id、type、sequence、 status、text
sysreferences	（所有数据库）每个在表或者列上声明的参照完整性约束在 sysreferences 中都有对应的一行。列：indexid、constrid、tableid、 reftabid、keycnt、status、frgnbid、pmrydbid、spare2、fokey1 ... fokey16、refkey1 ... refkey16、frgnbname、pmrydbname
sysremotelogins	（仅限 master 数据库）每个可以在 Adaptive Server 上执行远程过程调用 的远程用户在 sysremotelogins 中都有对应的一行。列：remoteserverid、 remoteusername、suid、status
sysresourcelimits	（仅限 master 数据库）每个由 Adaptive Server 定义的资源限制在 sysresourcelimits 中都有对应的一行。列：name、appname、rangeid、 limitid、enforced、action、limitvalue、scope、spare
sysroles	（所有数据库）将服务器角色 ID 映射到本地角色 ID。列：id、lrid、 type、status

syssecmechs	（仅限 master 数据库）包含有关 Adaptive Server 可用的每个安全性机制所支持的安全性服务的信息。列：sec_mech_name、available_service
syssegments	（所有数据库）每个段（命名的磁盘区段集）在 syssegments 中都有对应的一行。列：segment、name、status
sysservers	（仅限 master 数据库）Adaptive Server 在其上可以执行远程过程调用的每个远程 Adaptive Server、Backup Server 或者 Open Server 在 sysservers 中都有对应的一行。列：srvid、srvstatus、srvstatus2、srvstat2、srvname、srvnetname、srvclass、srvsecmech、srvcost、srvprincipal
syssessions	（仅限 master 数据库）每个连接到具有故障切换属性的 Adaptive Server 的客户端在 syssessions 中都有对应的一行。列：sys_id、ses_id、state、spare、status、dbid、name、nodeid、instanceid、ses_data
syslices	（所有数据库）分区表的每个段（页链）在 syslices 中都有对应的一行。syslices 仅在 Adaptive Server 升级过程中使用。列：state、id、partitionid、firstpage、controlpage、spare
sysrvroles	（仅限 master 数据库）每个系统角色或者用户定义的角色在 sysrvroles 中都有对应的一行。列：srid、name、password、pwdate、status、logincount、locksuid、lockreason、lockdate
sysstatistics	（所有数据库）用户表上的每个建立了索引的列和每个分区在 sysstatistics 中都有对应的一行或者多行。未索引的列在其中也可能有对应的行。列：statid、id、sequence、moddate、formatid、usedcount、colidarray、c0...c79、indid、ststatus、partitionid、spare2、spare3
systabstats	（所有数据库）每个聚簇索引、每个非聚簇索引、每个无聚簇索引的表以及每个分区在 systabstats 中都有对应的一行。列：indid、id、activestatid、indexheight、leafcnt、pagecnt、rowcnt、forwrowcnt、delrowcnt、dpagecrcnt、ipagecrcnt、drowcrcnt、oamapgcnt、extent0pgcnt、datarowsize、leafrowsize、status、pllijoindegree、spare2、rslastoam、rslastpage、frlastoam、frlastpage、conopt_thld、plldegree、emptypgcnt、spare4、partitionid、spare5、statmoddate、unusedpgcnt、oamapgcnt
systhresholds	（所有数据库）为数据库定义的每个阈值在 systhresholds 中都有对应的一行。列：segment、free_space、status、proc_name、suid、currauth
systimeranges	（仅限 master 数据库）存储指定的时间范围，Adaptive Server 将使用该时间范围来控制应用资源限制的时间。列：name、id、startday、endday、starttime、endtime



---

systransactions	（仅限 master 数据库）包含有关 Adaptive Server 事务的信息，但它不是常规表。列：xactkey、starttime、failover、type、coordinator、state、connection、status、status2、spid、masterdbid、loid、namelen、xactname、srvname、nodeid、instanceid
systypes	（所有数据库）每个系统提供的数据类型和用户定义的数据类型在 systypes 中都有对应的一行。域（由规则定义）和缺省值将在其存在时提供。列：uid、usertype、variable、allownulls、type、length、tdefault、domain、name、printfmt、prec、scale、ident、hierarchy、xtypeid、xdbid、accessrule
sysusages	（仅限 master 数据库）分配给数据库的每个磁盘分配区段在 sysusages 中都有对应的一行。每个数据库都包含指定数目的数据库（逻辑）页数。列：dbid、segmap、lstart、size、vstart、pad、unreservedpgs、crdate、vdevno
sysusermessages	（所有数据库）Adaptive Server 可返回的每个用户定义消息在 sysusermessages 中都有对应的一行。列：error、uid、description、langid、dlevel
sysusers	（所有数据库）数据库中允许的每个用户以及每个组或者角色在 sysusers 中都有对应的一行。列：suid、uid、gid、name、environ
sysxtypes	（所有数据库）每个扩展的 Java-SQL 数据类型在 sysxtypes 中都有对应的一行。列：xtid、xtstatus、xtmetatype、xtcontainer、xtname、xtsource、xtbinaryinrow、xtbinaryoffrow

## DBCC 表

以下是有关 Adaptive Server dbcc 表的简要说明。有关完整信息，请参见《参考手册：表》。

dbcc_config	说明当前正在执行的或者上次完成的 dbcc checkstorage 操作。列：dbid、type_code、value、stringvalue
dbcc_counters	存储 dbcc checkstorage 所执行的分析的结果。列：dbid、id、indid、partitionid、devid、opid、type_code、value
dbcc_exclusions	存储应排除在通过 checkverify 所执行的处理或通过 sp_dbcc_faultreport 所执行的故障报告之外的故障、表或这两者的组合。列：dbid、type、fault_type、table_name

dbcc_fault_params	为在 dbcc_faults 表中输入的故障提供附加的说明性信息。列: dbid、opid、faultid、type_code、intvalue、realvalue、binaryvalue、stringvalue、datevalue
dbcc_faults	提供 dbcc checkstorage 检测到的每个故障的说明。列: dbid、id、indid、partitionid、devid、opid、faultid、type_code、status
dbcc_operation_log	记录对 dbcc checkstorage 操作的使用情况。列: dbid、opid、optype、suid、start、finish、seq、id、maxseq
dbcc_operation_results	为记录在 dbcc_operation_log 表中的操作提供附加的说明性信息。列: dbid、opid、optype、type_code、intvalue、realvalue、binaryvalue、stringvalue、datevalue、seq
dbcc_types	提供 dbcc checkstorage 使用的数据类型的定义。

## 监控表

以下是有关 Adaptive Server 监控表的简要说明。有关完整信息, 请参见《参考手册: 表》, 或者, 有关表、列以及表间关系的视觉表示形式, 请参见《系统表框图》张贴画。

monCachedObject	存储当前在数据高速缓存中具有页的所有表、分区和索引的统计信息。列: CacheID、InstanceID、DBID、IndexID、PartitionID、CachedKB、CacheName、ObjectID、DBName、OwnerUserID、OwnerName、ObjectName、PartitionName、ObjectType、TotalSizeKB、ProcessesAccessing
monCachePool	存储为所有数据高速缓存分配的所有池的统计信息。列: CacheID、InstanceID、IOBufferSize、AllocatedKB、PhysicalReads、Stalls、PagesTouched、PagesRead、BuffersToMRU、BuffersToLRU、CacheName、LogicalReads、PhysicalWrites、APFReads、APFPercentage、WashSize
monCachedProcedures	存储当前存储在过程高速缓存中的所有存储过程、触发器和编译计划的统计信息。列: ObjectID、InstanceID、OwnerUID、DBID、PlanID、MemUsageKB、CompileDate、ObjectName、ObjectType、OwnerName、DBName、RequestCnt、TempdbRemapCnt、AvgTempdbRemapTime、ExecutionCount、CPUTime、ExecutionTime、PhysicalReads、LogicalReads、PhysicalWrites、PagesWritten

---

monCachedStatement	存储有关语句高速缓存的详细监控信息，包括有关语句上次执行期间所用资源、语句执行频率、特定计划的有效设置、语句的并发使用数等信息。列：SSQLID、HashKey、UserID、SUserID、DBID、DBName、CachedDate、LastUsedDate、CurrentUsageCount、StatementSize、MaxUsageCount、SessionSettings、ParallelDegree、QuotedIdentifier、TransactionIsolationLevel、TransactionMode、SAAuthorization、SystemCatalogUpdates、ExecutionMetrics、MetricsCount、MaxElapsedTime、MinElapsedTime、AvgElapsedTime、MaxLIO、MinLIO、AvgLIO、MaxPIO、MinPIO、AvgPIO、NumRecompilesPlanFlushes、NumRecompilesSchemaChanges、MaxPlanSize、MinPlanSize、LastRecompiledDate、UseCount、HasAutoParams、OptimizationGoal、OptimizerLevel
monCIPC	（特定于 Cluster Edition）提供从当前实例或所有实例看到的有关集群内部所有消息传送的摘要数字。列：InstanceID、ReceiveCount、TransmitCount、Multicast、Synchronous、ReceiveSoftError、ReceiveHardError、TransmitsSoftError、TransmitHardError、Retransmits、Switches、FailedSwitches、RegularBuffersInUse、FreeRegularBuffers、MaxRegularBuffersInUse、LargeBuffersInUse、FreeLargeBuffers、MaxLargeBuffersInUse
monCIPCEndpoints	（特定于 Cluster Edition）提供详细摘要，为集群实例中的每个子系统提供通信量数据。列：InstanceID、ReceiveCount、TransmitCount、ReceiveBytes、TransmitBytes、ReceiveQ、MaxReceiveQ、DoneQ、MaxDoneQ、MaxRecvQTime、AvgRecvQTime、EndPoint
monCIPCLinks	（特定于 Cluster Edition）监控集群中实例间链接的状态。列：InstanceID、LocalInterface、RemoteInterface、PassiveState、PassiveStateAge、ActiveState、ActiveStateAge
monCIPCMesh	（特定于 Cluster Edition）按实例提供从当前实例到集群中所有其它实例的连接网络的摘要数字。列：InstanceID、FarInstanceID、Received、Dropped、Transmitted、Resent、Retry、ControlRx、ControlTx、SendQ、MaxSendQ、SentQ、MaxSentQ、MaxSendQTime、AvgSendQTime、Mesh、MinRTT、MaxRTT、AverageRTT
monCLMObjectActivity	（特定于 Cluster Edition）收集集群锁信息。列：InstanceID、DBID、Object_PartitionID、LockRequests、LocalMaster、Waited、Granted、RWConflictWaited、AvgRWConflictWaitTime、MaxRWConflictWaitTime、WWConflictWaited、AvgWWConflictWaitTime、MaxWWConflictWaitTime、ClusterMsgWaits、AvgClusterMsgWaitTime、MaxClusterMsgWaitTime、DowngradeReqRecv、DowngradeReqRecvWithNoBlocker、ClusterDeadlock、Locktype

monClusterCacheManager	(特定于 Cluster Edition) 存储有关每个实例上运行的集群高速缓存管理器守护程序的诊断信息。monClusterCacheManager 按实例报告集群范围的信息。列: InstanceID、RequestsQueued、RequestsRequeued、RequestsServiced、DiskWrites、SleepCount、DaemonName、TransfersInitiated、Downgrades、Releases、AvgServiceTime、MaxQSize
monCMSFailover	(特定于 Cluster Edition) 跟踪集群成员资格服务 (CMS) 检测到失败的时间、获取新的集群视图、重新同步心跳、发布失败事件以及完成失败事件。列: InstanceID、FailedInstanceID、FailDetectTime、InitViewTime、FinalViewTime、ResynchHBTime、NotifyFailTime、EventdoneTime
monDataCache	存储有关 Adaptive Server 数据高速缓存的统计信息。列: CacheID、InstanceID、RelaxedReplacement、BufferPools、CacheSearches、PhysicalReads、LogicalReads、PhysicalWrites、Stalls、CachePartitions、CacheName、Status、Type、CacheSize、ReplacementStrategy、APFReads、Overhead
monDBRecovery	(特定于 Cluster Edition) 包含集群中所有实例的行, 并包含有助于恢复的每个数据库的行。列: DBID、InstanceID、MaxOpenXacts、MaxPFTSEntries、Buckets、LogBTotPages、LogBTotAPFWaited、LogBTotIO、AnlTotRec、AnlPhase1Recs、AnlPhase1RedoRecs、AnlPhase2Recs、AnlPhase2RedoRecs、AnlTotPages、AnlTotAPFWaited、AnlTotIO、RedoOps、RedoOpsNotRedonePFTS、RedoOpsRedonePFTS、RedoOpsRedoneTS、RedoOpsNotRedoneTS、RedoLogTotPages、RedoLogTotAPFWaited、RedoLogTotIO、RedoRecTotPage、RedoRecTotAPFWaited、RedoRecTotIO、UndoRecsUndone、UndoLogTotPages、UndoLogTotAPFWaited、UndoLogTotIO、UndoRecTotPages、UndoRecTotAPFWaited、UndoRedTotIO、DBName、FailedInstanceID、Command、RecType、LobBStartTime、LogBEndTime、AnlStartTime、AnlEndTime、RedoStartTime、RedoEndTime、UndoStartTime、UndoEndTime
monDBRecoveryLRTypes	(特定于 Cluster Edition) 跟踪恢复期间查看的日志记录。恢复进程至少查看了一条日志记录的每个日志记录类型在该表中都有对应的一行。列: DBID、InstanceID、NumRecs、LogRecType

---

monDeadLock	提供有关死锁的信息。使用 <code>deadlock pipe max messages</code> 可以调整返回消息的最大数。列：DeadLockID、VictimKPID、InstanceID、ResolveTime、ObjectDBID、PageNumber、RowNumber、HeldFamilyID、HeldSPID、HeldKPID、HeldProcDBID、HeldProcedureID、HeldBatchID、HeldContextID、HeldLineNumber、WaitFamilyID、WaitSPID、WaitKPID、WaitTime、ObjectName、HeldUserName、HeldAppName、HeldTranName、HeldLockType、HeldCommand、WaitUserName、WaitLockType、HeldSourceCodeID、WaitSourceCodeID、HeldClientAppName、HeldClientName、HeldClientHostName、HeldHostName、HeldNumLocks、HeldProcDBName、HeldProcedureName、HeldStmtNumber、ObjectDBName、ObjectID、WaitAppName、WaitBatchID、WaitClientAppName、WaitClientHostName、WaitClientName、WaitCommand、WaitContextID、WaitHostName、WaitLineNumber、WaitProcDBID、WaitProcDBName、WaitProcedureID、WaitProcedureName、WaitStmtNumber、WaitTranName
monDeviceIO	返回与数据库设备上的活动有关的统计信息。列：InstanceID、Reads、APFReads、Writes、DevSemaphoreRequests、DevSemaphoreWaits、IOTime、LogicalName、PhysicalName
monDeviceSpaceUsage	提供有关在其上分配数据库设备的文件系统的信息。空间信息仅可用于文件系统设备。列：InstanceID、VDevNo、LogicalName、PhysicalName、DeviceSizeMB、FileSystemName、FileSystemSizeMB、FileSystemFreeMB
monEngine	提供有关 Adaptive Server 引擎的统计信息。列：EngineNumber、ThreadID、InstanceID、CurrentKPID、PreviousKPID、CPUTime、SystemCPUTime、UserCPUTime、IOCPUTime、IdleCPUTime、Yields、Connections、DiskIOChecks、DiskIOPolled、DiskIOCompleted、MaxOutstandingIOs、ProcessesAffinitied、ContextSwitches、HkgcMaxQSize、HkgcPendingItems、HkgcHWMItems、HkgcOverflows、Status、Starttime、StopTime、AffinitiedToCPU、OSPID
monErrorLog	从 Adaptive Server 错误日志返回最新的错误消息。列：SPID、InstanceID、KPID、FamilyID、EngineNumber、ErrorNumber、Severity、State、Time、ErrorMessage
monFailoverRecovery	（特定于 Cluster Edition）包含集群锁管理程序 (CLM)、数据库恢复和集群成员资格服务 (CMS) 模块的集合的故障切换恢复诊断信息。列：InstanceID、ModuleName、FailedInstanceID、StartTime、EndTime
monIOController	提供有关 I/O 控制器的信息。列：InstanceID、ControllerID、KTID、EngineNumber、BlockingPolls、NonBlockingPolls、EventPolls、NonBlockingEventPolls、FullPolls、Events、EventHWM、Pending、Completed、Reads、Writes、Deferred、Type

monIOQueue	提供设备 I/O 统计信息，显示为每个设备上常规数据库和临时数据库的数据和日志 I/O。列：InstanceID、IOs、IOTime、LogicalName、IOType
monLicense	提供 Adaptive Server 当前检出的所有许可证的列表。列：InstanceID、Quantity、Name、Edition、Type、Version、Status、LicenseExpiry、GraceExpiry、LicenseID、Filter、Attributes
monLocks	返回已授予的锁和挂起的锁请求的列表。列：SPID、InstanceID、KPID、DBID、ParentSPID、LockID、Context、DBName、ObjectID、LockState、LockType、LockLevel、WaitTime、PageNumber、RowNumber、BlockedBy、BlockedState、SourceCodeID
monLockTimeout	提供有关锁超时的信息。列：InstanceID、LockWaitPeriod、LockTimeoutLevel、ObjectDBID、ObjectDBName、ObjectID、ObjectName、PageNumber、RowNumber、ExpiredAtTime、HeldSPID、HeldKPID、HeldUserName、HeldAppName、HeldHostName、HeldClientName、HeldClientAppName、HeldClientHostName、HeldTranName、HeldCommand、HeldFamilyID、HeldProcDBID、HeldProcDBName、HeldProcDBName、HeldProcedureName、HeldBatchID、HeldContextID、HeldLineNumber、HeldStmtNumber、HeldLockType、HeldNumLocks、HeldNumTimeoutsCausedByTran、HeldNumTimeoutsCausedByLock、HeldSourceCodeID、WaitSPID、WaitKPID、WaitUserName、WaitAppName、WaitHostName、WaitClientName、WaitClientAppName、WaitClientHostName、WaitTranName、WaitCommand、WaitFamilyID、WaitProcDBID、WaitProcDBName、WaitProcedureName、WaitBatchID、WaitContextID、WaitLineNumber、WaitStmtNumber、WaitLockType、WaitNumTimeoutsCausedByTran、WaitSourceCodeID、HeldProcedureID、WaitProcedureID
monLogicalCluster	（特定于 Cluster Edition）显示有关系统上当前配置的逻辑集群的信息。列：LCID、Attributes、ActiveConnections、BaseInstances、ActiveBaseInstances、FailoverInstances、ActiveFailoverInstances、Name、State、DownRoutingMode、FailoverMode、StartupMode、SystemView、Roles、LoadProfile、ActionnRelease、Gather
monLogicalClusterAction	（特定于 Cluster Edition）显示从启动到操作释放这段时间对本地集群执行的所有管理操作。列：Handle、State、LCID、LogicalClusterName、Action、FromInstances、ToInstances、InstancesWaiting、WaitType、StartTime、Deadline、CompleteTime、ConnectionsRemaining、NonMigConnections、NonHACConnections

---

monLogicalClusterInstance	（特定于 Cluster Edition）显示有关实例与逻辑集群之间的多对多关系的信息。列：LCID、LogicalClusterName、InstanceID、InstanceName、Type、FailoverGroup、State、ActiveConnections、NonMigConnections、NonHACConnections、LoadScore
monLogicalClusterRoute	（特定于 Cluster Edition）显示有关已配置的路由（应用程序、登录和别名绑定）的信息。列：LCID、LogicalClusterName、RouteType、RouteKey
monNetworkIO	返回 Adaptive Server 与客户端连接之间所有通信的网络 I/O 统计信息。列：InstanceID、PacketsSent、PacketsReceived、BytesSent、BytesReceived
monOpenDatabases	提供与当前位于服务器元数据高速缓存中的数据库有关的状态信息和统计信息。列：DBID、InstanceID、BackupInProgress、LastBackupFailed、TransactionLogFull、AppendLogRequests、AppendLogWaits、DBName、BackupStartTime、SuspendedProcesses、QuiesceTag、LastCheckpointTime、LastTranLogDumpTime
monOpenObjectActivity	提供有关所有打开的表和索引的统计信息。列：DBID、ObjectID、IndexID、InstanceID、DBName、ObjectName、LogicalReads、PhysicalReads、APFReads、PagesRead、PhysicalWrites、PagesWritten、RowsInserted、RowsDeleted、RowsUpdated、Operations、LockRequests、LockWaits、OptSelectCount、LastOptSelectDate、UsedCount、LastUsedDate、HkgcRequests、HkgcPending、HkgcOverflows、PhysicalLocks、PhysicalLocksRetained、PhysicalLocksRetainWaited、PhysicalLocksDeadlocks、PhysicalLocksWaited、PhysicalLocksPageTransfer、TransferReqWaited、AvgPhysicalLocksWaitTime、AvgTransferReqWaitTime、TotalServiceRequests、PhysicalLocksDowngraded、PagesTransferred、ClusterPageWrites、AvgServiceTime、AvgTimeWaitedOnLocalUsers、AvgTransferSendWaitTime、AvgIOServiceTime、AvgDowngradeServiceTime、SharedLockWaitTime、ExclusiveLockWaitTime、UpdateLockWaitTime、ObjectCacheDate

monOpenPartitionActivity	提供有关服务器上每个打开的分区的使用信息。列：DBID、ObjectID、IndexID、PartitionID、InstanceID、DBName、ObjectName、PartitionName、LogicalReads、PhysicalReads、APFReads、PagesRead、PhysicalWrites、PagesWritten、RowsInserted、RowsDeleted、RowsUpdated、OptSelectCount、LastOptSelectDate、UsedCount、LastUsedDate、HkgcRequests、HkgcPending、HkgcOverflows、PhysicalLocks、PhysicalLocksRetained、PhysicalLocksRetainWaited、PhysicalLocksDeadlocks、PhysicalLocksWaited、PhysicalLocksPageTransfer、TransferReqWaited、AvgPhysicalLockWaitTime、AvgTransferReqWaitTime、TotalServiceRequests、PhysicalLocksDowngraded、PagesTransferred、ClusterPageWrites、AvgServiceTime、AvgTimeWaitedOnLocalUsers、AvgTransferSendWaitTime、AvgIOServiceTime、AvgDowngradeServiceTime、ObjectCacheDate
monPCIBridge	包含有关 Java PCI Bridge 的信息。列：InstanceID、Status、ConfiguredSlots、ActiveSlots、ConfiguredPCIMemoryKB、UsedPCIMemoryKB
monPCIEngine	显示有关 PCI Bridge 及其插件的引擎信息。列：InstanceID、Engine、Status、PLBStatus、NumberOfActiveThreads、PLBRequests、PLBWakeUpRequests
monPCISlots	包含有关绑定到 PCI Bridge 中每个槽的插件的信息。列：InstanceID、Slot、Status、Modulename、engine
monPCM	(特定于 Cluster Edition) 跟踪集群中的对等协调模块 (PCM) 客户端活动，且每个 PCM 客户端在该表中都有对应的一行。列：InstanceID、Sent、Fragments_sent、Fragments_received、Received、Reply、Unicast、Multicast、Sync、Async、MinBytes、AvgBytes、MaxBytes、MinDialog、AvgDialog、MaxDialog、Dialog、MinTimeSyncApi、AvgTimeSyncApi、MaxTimeSyncApi、MinTimeAsyncApi、AvgTimeAsyncApi、MaxTimeAsyncApi、MinTimeCIPCMalloc、AvgTimeCIPCMalloc、MaxTimeCIPCMalloc、MinTimeCIPCAllocCB、AvgTimeCIPCAllocCB、MaxTimeCIPCAllocCB、MinTimeCIPCUnicastmsg、AvgTimeCIPCUnicastmsg、MaxTimeCIPCUnicastmsg、MinTimeCIPCMulticastmsg、AvgTimeCIPCMulticastmsg、MaxTimeCIPCMulticastmsg、MinTimeClientRecvCB、AvgTimeClientRecvCB、MaxTimeClientRecvCB、ModuleName
monProcedureCache	返回与 Adaptive Server 过程高速缓存有关的统计信息。列：Requests、Loads、Writes、Stalls、InstanceID



---

monProcedureCache MemoryUsage	每个过程高速缓存分配器在该表中都有一行。列: InstanceID、AllocatorID、ModuleID、Active、HWM、ChunkHWM、AllocatorName、NumReuseCaused
monProcedureCache ModuleUsage	每个从过程高速缓存中分配内存的模块在该表中都有一行。列: InstanceID、ModuleID、Active、HWM、NumPagesReused、ModuleName
monProcess	提供有关当前正在执行的或等待的进程的详细统计信息。列: SPID、InstanceID、KPID、ServerUserID、BatchID、ContextID、LineNumber、SecondsConnected、DBID、EngineNumber、Priority、FamilyID、Login、Application、Command、NumChildren、SecondsWaiting、WaitEventID、BlockingSPID、BlockingXLOID、DBName、EngineGroupName、ExecutionClass、MasterTransactionID、HostName、ClientName、ClientHostName、ClientAppName
monProcessActivity	提供有关进程活动的详细统计信息。列: SPID、InstanceID、KPID、ServerUserID、CPUTime、WaitTime、PhysicalReads、LogicalReads、PagesRead、PhysicalWrites、PagesWritten、MemUsageKB、LocksHeld、TableAccesses、IndexAccesses、WorkTables、TempDbObjects、ULCBytesWritten、ULCFlushes、ULCFlushFull、ULCMaxUsage、ULCCurrentUsage、Transactions、Commits、Rollbacks、HostName、Application、ClientName、ClientHostName、ClientAppName
monProcessLookup	提供有关服务器上每个进程的标识信息。列: SPID、InstanceID、KPID、Login、Application、ClientHost、ClientIP、ClientOSPID、ClientName、ClientHostName、ClientAppName
monProcessMigration	(特定于 Cluster Edition) 显示有关当前迁移的连接的信息。列: SPID、KPID、LogicalCluster、Instance、MigrationLogicalCluster、MigrationInstance、Command
monProcessNetIO	提供每个进程的网络 I/O 活动信息。列: SPID、InstanceID、KPID、NetworkPacketSize、PacketSent、PacketsReceived、BytesSent、BytesRecieved、NetworkEngineNumber
monProcessObject	提供有关进程当前访问的对象的统计信息。列: SPID、InstanceID、KPID、DBID、ObjectID、PartitionID、IndexID、OwnerUserID、LogicalReads、PhysicalReads、PhysicalAPFReads、DBName、ObjectName、PartitionName、ObjectType、PartitionSize
monProcessProcedures	返回进程正在执行的所有过程的列表。列: SPID、InstanceID、KPID、DBID、OwnerUID、ObjectID、PlanID、MemUsageKB、CompileDate、ContextID、LineNumber、DBName、OwnerName、ObjectName、ObjectType、ExecutionCount、CPUTime、ExecutionTime、PhysicalReads、LogicalReads、PhysicalWrites、PagesWritten

monProcessSQLText	提供进程当前正在执行的 SQL 文本。列：SPID、InstanceID、KPID、ServerUserID、BatchID、LineNumber、SequenceInLine、SQLText
monProcessStatement	提供有关当前正在执行的语句的信息。列：SPID、InstanceID、KPID、DBID、ProcedureID、PlanID、BatchID、ContextID、LineNumber、CPUTime、WaitTime、MemUsageKB、PhysicalReads、LogicalReads、PagesModified、PacketsSent、PacketsReceived、NetworkPacketSize、PlansAltered、RowsAffected、DBName、StartTime
monProcessWaits	提供服务器上的当前进程正在等待的所有等待事件的列表。列：SPID、InstanceID、KPID、ServerUserID、WaitEventID、Waits、WaitTime
monProcessWorkerThread	提供有关当前配置的每个工作进程的活动的统计信息。列：SPID、InstanceID、KPID、ThreadsActive、MaxParallelDegree、MaxScanParallelDegree、ParallelQueries、PlansAltered、FamilyID
monRepLogActivity	从 Replication Agent 所更新的监控计数器中收集信息。列：DBID、SPID、InstanceID、LogRecordsScanned、LogRecordsProcessed、NumberOfScans、TotalTimeForLogScans、LongestTimeForLogScans、AvgTimeForLogScans、Updates、Inserts、Deletes、StoredProcedures、SQLStatements、DDL、Writetext、LobColumns、CLR、Checkpoints、BeginTransaction、CommitTransaction、AbortedTransaction、PreparedTransaction、DelayedCommit、MaintenanceUserTransaction、NumberOfLogExtentions、TotalTimeOfLogExtentions、LongestTimeOfLogExtentions、AvgTimeOfLogExtentions、MaxHashSchemaSize、NumberOfSchemasReused、NumberOfSchemaFwdLookup、TotalTimeOfSchemaFwdLookup、LongestTimeOfSchemaFwdLookup、AvgTimeOfSchemaFwdLookup、NumberOfSchemaBckwLookup、TotalTimeOfSchemaBckwLookup、LongestTimeOfSchemaBckwLookup、AvgTimeOfSchemaBckwLookup、NumberOfMempoolAllocates、NumberOfMempoolFrees、MempoolCurrentSize、MempoolHighUsage、DBName
monRepScanners	提供有关 Rep Agent 扫描程序任务在哪里花费时间的信息。列：DBID、SPID、InstanceID、EngineBinding、LogRecordsScanned、LogrecordsProcessed、NumberOfTruncPointRequested、NumberOfTruncPointMoved、DBName、Status、SleepStatus、StartMarker、EndMarker、CurrentMarker、OldestTransaction
monRepScannersTotalTime	提供有关 Rep Agent 扫描程序任务在哪里花费时间的信息。列：DBID、SPID、InstanceID、LogRecProcessed、BytesPacked、TotalTime、MRPBootstrapTime、ScanTime、ProcessTime、SchemaLookupsTime、PackTime、QueueingTime、HashBindingSize、HashBindingEntries、HashBindingCollisions、YieldsOnFullQueue、WaitsOnSenderThread、WaitTimeOnSenderThread、LongestWaitOnSenderThread

---

monRepSenders	提供有关 Rep Agent 发送者任务的处理信息。列：DBID、SPID、InstanceID、EngineBinding、MessageQueueSize、MessagesInQueue、NumberOfScannerYields、NumberOfScannerSleeps、NumberOfBytesSent、LastRepServerError、NumberOfRetries、SleepsOnEmptyQueue、NumberOfQueueFlushes、SleepTimeOnEmptyQueue、LongestSleepTimeOnEmptyQueue、MaxQueueSize、DBName、Dataserver、ReplicationServer、Username、Status、SleepStatus
monSQLRepActivity	提供有关所有打开对象的成功复制的 SQL 语句的统计信息。列：DBID、ObjectID、InstanceID、DBName、ObjectName、UpdateStmts、InsertSelectStmts、DeleteStmts、SelectIntoStmts、RowsThreshold
monSQLRepMisses	提供有关所有打开对象的未成功复制的 SQL 语句的统计信息。列：DBID、ObjectID、InstanceID、DBName、ObjectName、Threshold、QueryLimitation、Configuration
monState	提供有关 Adaptive Server 的总体状态的信息。列：InstanceID、LockWaitThreshold、LockWaits、DaysRunning、CheckPoints、NumDeadlocks、Diagnostic Dumps、Connections、MaxRecovery、Transactions、StartDate、CountersCleared
monStatementCache	提供有关语句高速缓存的统计信息。列：InstanceID、TotalSizeKB、UsedSizeKB、NumStatements、NumSearches、HitCount、NumInserts、NumRemovals、NumRecompilesSchemaChanges、NumRecompilesPlanFlushes
monSysLoad	（特定于 Cluster Edition）按引擎提供趋势统计信息。列：InstanceID、EngineNumber、SteadyState、Avg_1min、Avg_5min、Avg_15min、Max_1min、Max_5min、Max_15min、Max_1min_Time、Max_5min_Time、Max_15min_Time、Statistic、Sample、Peak、Peak_time、StatisticID
monSysPlanText	提供当前执行的查询的查询计划的历史记录。列：PlanID、InstanceID、SPID、KPID、BatchID、ContextID、SequenceNumber、DBID、ProcedureID、DBName、PlanText
monSysSQLText	提供最近执行的 SQL 文本或当前执行的 SQL 文本。列：SPID、InstanceID、KPID、ServerUserID、BatchID、SequenceInBatch、SQLText
monSysStatement	提供有关服务器上最近执行的语句的历史记录。列：SPID、InstanceID、KPID、DBID、ProcedureID、PlanID、BatchID、ContextID、LineNumber、CpuTime、WaitTime、MemUsageKB、PhysicalReads、LogicalReads、PagesModified、PacketsSent、PacketsReceived、NetworkPacketSize、PlansAltered、RowsAffected、ErrorStatus、HashKey、SqlId、ProcNestLevel、StatementNumber、DBName、StartTime、EndTime
monSysWaits	针对进程等待的事件提供一个服务器范围的视图。列：InstanceID、WaitEventID、WaitTime、Waits

monSysWorkerThread	返回与工作线程配置和执行有关的服务器范围的统计信息。列： InstanceID、ThreadsActive、TotalWorkerThreads、HighWater、 ParallelQueries、PlansAltered、WorkerMemory、TotalWorkerMemory、 WorkerMemoryHWM、MaxParallelDegree、MaxScanParallelDegree
monTableColumns	说明每个监控表的所有列。列：TableID、ColumnID、TypeID、 Precision、Scale、Length、Indicators、TableName、ColumnName、 TypeName、Description、Label、Language
monTableCompression	包含表的压缩历史记录。列：InstanceID、DBID、TableID、PartitionID、 TableName、CompRowInserted、CompRowUpdated、CompRowForward、 CompRowScan、RowPageDecompressed、RowDecompressed、 ColDecompressed、RowCompNoneed、PageCompNoneed、 PagesCompressed、BytesSavedPageLevel
monTableParameters	提供监控表中用于优化监控表查询性能的所有列的说明。列：TableID、 ParameterID、TypeID、Precision、Scale、Length、TableName、 ParameterName、TypeName、Description
monTables	提供对所有监控表的说明。列：TableID、Columns、Parameters、 Indicators、Size、TableName、Description、Language
monTableTransfer	MonTableTransfer 提供 Adaptive Server 活动内存中表的历史传输信息。 列：InstanceID、DBID、TableID、TableName、SequenceID、 TrackingID、PercentDone、BeginTime、EndTime、EndCode、 TransferFloor、TransferCeiling、RowsSent、BytesSent、Format
monTask	特定于处于线程化模式中的 Adaptive Server，每个任务都有相应的一 行。列：InstanceID、KTID、ThreadPoolID、ThreadID、KPID、SPID、 Name、ThreadPoolName
monTempdbActivity	（特定于 Cluster Edition）提供有关所有打开的本地临时数据库的统计信 息，包括在 tempdb 配置模式下启动实例时的全局系统 tempdb。列： DBID InstanceID、DBName、AppendLogRequest、AppendLogWaits、 LogicalReads、PhysicalReads、APFReads、PagesRead、PhysicalWrites、 PagesWritten、LockRequests、LockWaits、CatLockRequests、 CatLockWaits、AssignedCnt、SharableTabCnt
monThread	特定于处于线程化模式中的 Adaptive Server：每个线程都有相应的一 行。列：InstanceID、ThreadID、KTID、OSThreadID、AltOSThreadID、 ThreadPoolID、State、ThreadAffinity、ThreadPoolName、TaskRuns、 TotalTicks、IdleTicks、SleepTicks、BusyTicks、UserTime、SystemTime、 MinorFaults、MajorFaults、VoluntaryCtxtSwitches、 NonVoluntaryCtxtSwitches

---

monThreadPool	特定于处于线程化模式中的 Adaptive Server: 每个线程池都有相应的一行。列: ThreadPoolID、Size、TargetSize、Tasks、ThreadPoolName、ThreadPoolDescription、Type、IdleTimeout
monWaitClassInfo	为所有等待类提供文本说明 (例如, 等待磁盘读取完成)。列: WaitClassID、Description
monWaitEventInfo	为强制进程在 Adaptive Server 内等待的每种可能情况提供文本说明。列: WaitEventID、WaitClassID、Description、Language
monWorkload	(特定于 Cluster Edition) 根据负荷配置文件显示每个实例上每个逻辑集群的工作负荷分值。列: LCID、InstanceID、LoadProfileID、LoadScore、ConnectionsScore、CpuScore、RunQueueScore、IoLoadScore、EngineScore、UserScore、LogicalClusterName、InstanceName、LoadProfileName
monWorkloadPreview	(特定于 Cluster Edition) 提供有关负荷配置文件在不启用时对工作负荷分值的影响的估计值。列: InstanceID、LoadProfileID、LoadScore、ConnectionScore、CpuScore、RunQueueScore、IoLoadScore、EngineScore、UserScore、InstanceName、LoadProfileName
monWorkloadProfile	(特定于 Cluster Edition) 显示当前配置的工作负荷配置文件。列: ProfileID、ConnectionsWeight、CpuWeight、RunQueueWeight、IoLoadWeight、EngineWeight、UserWeight、LoginThreshold、DynamicThreshold、Hysteresis、Name、Type
monWorkloadRaw	(特定于 Cluster Edition) 提供每个实例的原始工作负荷统计信息。您不需要具有 mon_role 角色即可查询此监控表。列: InstanceID、ConnectionsRaw、CpuRaw、RunQueueRaw、IoLoadRaw、EngineRaw、UserRaw、InstanceName
monWorkQueue	提供有关工作队列的信息。列: InstanceID、CurrentLength、MaxLength、TotalRequests、QueuedRequests、WaitTime、Name

## sybpcidb 表

sybpcidb 数据库存储 Java PCI 桥和 PCA/JVM 插件的配置信息。本节按字母顺序列出 sybpcidb 表。

pca_jre_arguments	存储有关配置 PCA/JVM 插件所用参数的信息。列: jre_args_directive_index、jre_args_name、jre_args_units、 jre_args_number_value、jre_args_string_value、jre_args_description、 jre_args_enabled、jre_args_status
-------------------	--

pca_jre_directives	存储有关配置 PCA/JVM 所用指令的信息。列：jre_directives_index、jre_directives_name、jre_directives_description、jre_directives_enabled、jre_directives_status
pci_arguments	存储定义配置 PCI Bridge 所用每个参数的信息。列： pci_args_directive_index、pci_args_name、pci_args_units、 pci_args_number_value、pci_args_string_value、pci_args_description、 pci_args_enabled、pci_args_status
pci_directives	存储配置 PCI Bridge 的指令。列：pci_directives_index、 pci_directives_name、pci_directives_description、pci_directives_enabled、 pci_directives_status
pci_slotinfo	包含介绍每个槽的信息，包括槽的指令和参数的表名。列： slot_number、slot_name、slot_pca_directives_table_name、 slot_pca_arguments_table_name、slot_status
pci_slot_syscalls	包含 PCI 桥使用的运行时分配模式的运行时系统调用配置信息。列： syscall_slot_number、syscall_system_call、syscall_dispatch_name、 syscall_enabled、syscall_status

---

## 实用程序

以下是 Adaptive Server 实用程序的语法和简要说明。有关完整信息，请参见《实用程序指南》。

### backupserver

Backup Server 程序的可执行形式。

```
backupserver
[-C server_connections]
[-S b_servename]
[-I interfaces_file]
[-e error_log_file]
[-M sybmultbuf_binary]
[-N network_connections]
[-T trace_value]
[-L Sybase_language_name]
[-J Sybase_character_set_name]
[-c tape_config_file]
[-D n]
[-A pathname]
[-P active_service_threads]
[-V level_number]
[-p n]
[-m max_shared_memory]
```

- 或：backupserver -v

### bcp

按照用户指定的格式将数据库表复制到操作系统文件中或从操作系统文件中复制数据库表。

```
bcp [(database_name.)owner].table_name [: [partition_id | slice_number]]
[partition partition_name] {in | out} datafile
[-f formatfile]
[-e errfile]
[-d discardfileprefix]
[-F firstrow]
[-L lastrow]
[-b batchsize]
[-m maxerrors]
[-n]
[-c]
[-t field_terminator]
[-r row_terminator]
[-U username]
[-P password]
[-I interfaces_file]
[-S server]
[-a display_charset]
[-z language]
[-A packet_size]
[-J client_charset]
```

```

[-T text_or_image_size]
[-E]
[-g id_start_value]
[-N]
[-W]
[-X]
[-M LabelName LabelValue]
[-labeled]
[-K keytab_file]
[-R remote_server_principal]
[-C]
[-V [security_options]]
[-Z security_mechanism]
[-Q]
[-Y]
[-y sybase_directory]
[-x trusted.txt_file]
[--maxconn maximum_connections]
[--show-fi]
[--hide-vcc]
[--colpasswd [[database_name.owner].table_name.]column_name
[password]]]
[--keypasswd [[database_name.owner].]key_name [password]]]
[--initstring 'Transact-SQL_command']

```

或

bcp -v

certauth

将服务器认证请求转换为一个 CA（证书发放机构）签署的认证。

```

certauth
[-r]
[-C caCert_file]
[-Q request_filename]
[-K caKey_filename]
[-N serial_number]
[-O SignedCert_filename]
[-P caPassword]
[-s start_time]
[-T valid_time]

```

或：certauth -v

certpk12

将 PKCS #12 文件导出或导入到认证文件和私有密钥中。

```

certpk12
{-O Pkcs12_file | -I Pkcs12_file}
[-C Cert_file]
[-K Key_file]
[-P key_password]
[-E Pkcs12_password]

```

- 或：certpk12 -v



---

certreq

创建一个服务器认证请求和相应的私有密钥。

```
certreq
[-F input_file]
[-R request_filename]
[-K PK_filename]
[-P password]
```

- 或: certreq -v

charset

*仅限 UNIX 平台* — 在 Adaptive Server 中装载字符集和排序顺序文件。位于 \$SYBASE/\$SYBASE\_ASE/bin 中。

```
charset
[-Ppassword]
[-Sserver]
[-linterface]
sort_order
[ charset ]
```

- 或: charset -v

cobpre

COBOL 的预编译程序。

cpre

预编译 C。

dataserver

*仅限 UNIX 平台* — Adaptive Server 程序的可执行形式。

```
dataserver [-f] [-g] [-G] [-h] [-H] [-m] [-q] [-v] [-X]
[-a path_to_CAPs_directive_file]
[-b master_device_size [k | K | m | M | g | G | t | T]]
[-c config_file_for_server]
[-d device_name]
[-e path_to_error_log]
[-i interfaces_file_directory]
[-K keytab_file]
[-L config_file_name_for_connectivity]
[-M shared_memory_repository_directory]
[-N licinstant]
[-n sa_login_name]
[-p sa_login_name]
[-r mirror_disk_name]
[-s server_name]
[-T trace_flag]
[-u sa/sso_name]
[-w master | model database]
[-y [password] ]
[-z page_size [ k | K ]]
```

用于 Cluster Edition 的语法:

```

dataserver
-u, --admin-name=sa/sso_name
    --buildquorum=[force]
-a, --caps-file=filename
-F, --cluster-input=filename
    --cluster-takeover
-L, --conn-config-file=[filename]
    --create-cluster-id [=quorum]
-D, --default-db-size=size_spec
-e, --error-log=[filename]
-G, --event-log-server=logserv_name
-f, --forcebuild
-H, --ha-server
-h, --help=[{0|1|2|3}[,display_width]]
    --instance=instance_name
-y, --key-password=[key_password]
-K, --keytab-file=filename
-N, --license-prop-file=filename
-z, --logical-page-size=page_size
-Z, --master-db-size=size_spec
-d, --master-dev=master_device_name
-b, --master-dev-size=[size_spec]
    --master_key_password [=password]
-r, --master-mirror=filename
-m, --masterrecover
-g, --no-event-logging
-Q, --quorum-dev=quorum_dev
-q, --recover-quiet
-w, --rewrite-db=database_name
-p, --sa-name={SSO_login_account | sso_role | sa_role}
-k, --server-principal=s_principal
-M, --shared-mem-dir=directory_name
-X, --sybmon
-T, --trace=trace_flag
-v, --version

```

或

```
dataserver -v
```

ddlgen

这是一个基于 Java 的工具，可生成 Adaptive Server 中的服务器级和数据库级对象的定义。

```

ddlgen
-Ulogin
-Ppassword
-S[[ssl:]server | host_name : port_number]
[-I interfaces_file]
[-Tobject_type]
[-Nobject_name]

```

---

```

[-Ddbname]
[-Xextended_object_type]
[-Ooutput_file]
[-Error_file]
[-Lprogress_log_file]
[-Jclient_charset]
[-LC -N logical_cluster_name
-F[ % | SGM | GRP | USR | R | D | UDD | U | V |
    P | XP | I | RI | KC | TR | PC ]

```

或

```
ddlgen -v
```

defncopy

将指定视图、规则、缺省值、触发器或过程的定义从数据库复制到操作系统文件中或从操作系统文件复制到数据库中。

```

defncopy
[-X]
[-a display_charset]
[-I interfaces_file]
[-J [client_charset]]
[-K keytab_file]
[-P password]
[-R remote_server_principal]
[-S [server_name]]
[-U username]
[-V security_options]
[-Z security_mechanism]
[-z language]
{ in file_name database_name |
  out file_name database_name [owner.]object_name
  [[owner.]object_name...]}

```

- 或: defncopy -v

dscp

仅限 UNIX 平台 — 允许通过 UNIX 平台中的命令行来查看和编辑 interfaces 文件中的服务器条目。位于 \$SYBASE/\$SYBASE\_OCS/bin 中。

```
dscp [-p]
```

- 或: dscp -v

dsedit

在 UNIX 平台上 — dsedit 实用程序允许在 UNIX 平台中使用基于 X11/Motif 的 GUI 来查看和编辑 interfaces 文件中的服务器条目。在 Windows 上 — dsedit.exe 实用程序创建和修改 interfaces 文件中的网络连接信息。

```
dsedit
```

- 或: dsedit -v

## extractjava

将保留的 JAR 及其包含的类从 Adaptive Server 复制到客户端文件中。

extractjava (Windows 中的 extrjava)

```
-j jar_name
-f file_name
[-S server_name]
[-U user_name]
[-P password]
[-D database_name]
[-I interfaces_file]
[-a display_charset]
[-J client_charset]
[-z language]
[-t timeout]
[-v]
```

- 或: extractjava -v

## installjava

将 JAR 从客户端文件安装到 Adaptive Server 中。

installjava

```
-f file_name
[ -new | -update ]
[-j jar_name ]
[-S server_name ]
[-U user_name ]
[-P password ]
[-D database_name ]
[-I interfaces_file ]
[-a display_charset ]
[-J client_charset ]
[-z language ]
[-t timeout ]
[-v]
```

- 或: installjava -v

## isql

Adaptive Server 的交互式 SQL 语法分析程序。

```
isql [-b] [-e] [-F] [-p] [-n] [-v] [-W] [-X] [-Y] [-Q]
[-a display_charset]
[-A packet_size]
[-c cmdend]
[-D database]
[-E editor]
[-h header]
[-H hostname]
[-i inputfile]
[-I interfaces_file]
[-J client_charset]
[-K keytab_file]
[-l login_timeout]
[-m errorlevel]
```

---

```
[-o outputfile]
[-P password]
[-R remote_server_principal]
[-s colseparator]
[-S server_name]
[-t timeout]
-U username
[-V [security_options]]
[-w columnwidth]
[-z locale_name]
[-Z security_mechanism]
[--conceal]
```

langinstall

在 Adaptive Server 上安装新语言。

```
langinstall
[-S server]
[-U user]
[-P password]
[-R release_number]
[-I path]
语言
character_set
```

- 或: langinstall -v

optdiag

显示优化程序统计信息或将更新的统计信息装载到系统表中。

```
optdiag [binary] [simulate] statistics
{ -i input_file | database[.owner[.table[column]] ] } [-o output_file] }
[-U user_name]
[-P password]
[-T trace_value]
[-I interfaces_file]
[-S server]
[-v]
[-h]
[-s]
[-z language]
[-J client_character_set]
[-a display_charset]
```

preupgrade

对安装或数据库执行测试以确定其是否为升级做好准备，并报告发现的问题。

```
preupgrade [-v] [-h] [-N]
[-p [skip_sybprocs]
[-D database_name]
[-I interfaces_file]
[-P password]
[-S server_name]
[-U user_name]
[-X option[,option]...]
```

pwdcrypt	在 <i>libtcl.cfg</i> 文件中创建和输出一个加密的 LDAP 口令。
	pwdcrypt
qptune	使用户可以修复丢失的统计信息、确定最佳的查询计划、优化目标或其它配置设置，以及在查询级或服务器级应用这些设置。
	qptune [-U <i>username</i> ] [-P <i>password</i> ] [-S <i>hostname:port/database</i> ] [-A <i>action</i> ] [-M <i>mode</i> ] [-T <i>appTime</i> ] [-i <i>inputFile</i> ] [-o <i>outputFile</i> ] [-f <i>fileList(.)</i> ] [-c <i>configFile</i> ] [-l <i>limit</i> ] [-e <i>evalField</i> ] [-d <diff%(.diff_abs)>] [-m <i>missingCount</i> ] [-n <i>login</i> ] [-J <i>charset</i> >] [-N (noexec)] [-g (applyOptgoal)] [-v (verbose)] [-s (sort)] [-h (help)]
qrmutil	仅限 <i>Cluster Edition</i> — 允许备份、恢复和重新配置仲裁设备。qrmutil 位于 <i>\$SYBASE/\$SYBASE_ASE/bin</i> 中。
	--additional-run-parameters= <i>parameter_list</i> --ase-config-extract= <i>file_name</i> --ase-config-info --ase-config-store= <i>file_name</i> --ase-config-version= <i>version_number</i> --buildquorum=[force]--cluster-take-over --config-file= <i>file_name</i> --diag={all   boot   toc   nodes   locks   config   cms} --display={boot   nodes   heartbeat   master   cluster   instance   config   state} --drop-cluster=[force] --drop-instance= <i>instance_name</i> --errorlog= <i>file_name</i> --extract-config= <i>file_name</i> -h, --help -F, --cluster-input= <i>file_name</i> --fence-capable= <i>device_path</i> --installation= <i>installation_mode</i> -s, --instance= <i>instance_name</i>

---

```

--instance-node=node_name
--interfaces-dir=path_to_interfaces_file
--max-instances=number_of_instances
--master-dev=master_device
--membership-mode=membership_mode
--primary-address=interconnect_address
--primary-port=port_number
--primary-protocol=protocol
-Q, --quorum-dev=quorum_device
--register-node=node_name
--secondary-address=interconnect_address
--secondary-port=port_number
--secondary-protocol=protocol
--traceflags=traceflag_list
--unregister-node=node_name
--verify-node=node_name
-v, --version]

```

**showserver**      仅限 *UNIX* 平台 — 显示当前在本地计算机上运行的 Adaptive Server 和 Backup Server （仅适用于 *UNIX* 平台）。

```
showserver
```

**sqldbgr**      sqldbgr 是一个调试存储过程和触发器的命令行实用程序。

```

sqldbgr
  -U username
  -P password
  -S host:port

```

**sqlloc**      仅限 *UNIX* 平台 — 使用基于 X11/Motif 的 GUI 来安装和修改 Adaptive Server 的语言、字符集和排序顺序的缺省设置。

```

sqlloc
  [-S server]
  [-U user]
  [-P password]
  [-s sybase dir]
  [-I interfaces file]
  [-r resource file]

```

- 或：sqlloc -v

**sqllocres**      仅限 *UNIX* 平台 — 使用资源文件安装和修改 Adaptive Server 的语言、字符集和排序顺序的缺省设置。

```

sqllocres
  [-S server]
  [-U user]
  [-P password]
  [-s sybase dir]
  [-I interfaces file]
  [-r resource file]

```

- 或: sqllocres -v

sqlsrvr

仅限 Windows 平台 — Adaptive Server 程序的可执行形式。

```
sqlserver [-f] [-g] [-G] [-h] [-H] [-m] [-P] [-q] [-v] [-X]
[-a path_to_CAPs_directive_file]
[-b master_device_size] [k | K | m | M | g | G | t | T]
[-c config_file_for_server]
[-d device_name]
[-e path_to_error_log]
[-i interfaces_file_directory]
[-K keytab_file]
[-L config_file_name_for_connectivity]
[--master_key_password [=password]
[-M shared_memory_repository_directory]
[-p sa_login_name]
[-r mirror_disk_name]
[-s server_name]
[-T trace_flag]
[-u sa/sso_name]
[-w master | model database]
[-y [password] ]
[-z page_size [ k | K ]]
```

sqlupgrade

仅限 UNIX 平台 — 使用基于 X11/Motif 的 GUI 将当前安装的 Adaptive Server 版本升级为最新版本。

```
sqlupgrade
[-s sybase_dir]
[-r resource_file]
```

- 或: sqlupgrade -v

sqlupgraderes

仅限 UNIX 平台 — 使用资源文件将当前安装的 Adaptive Server 版本升级为最新版本。

```
sqlupgraderes
[-s sybase_dir]
[-r resource_file]
```

- 或: sqlupgraderes -v

srvbuild

仅限 UNIX 平台 — 以缺省的或用户指定的关键配置属性值来创建新的 Adaptive Server、Backup Server、Monitor Server 或 XP Server。

```
srvbuild
[-s sybase_dir]
[-l interfaces_file]
[-r resource_file]
```

- 或: srvbuild -v



---

srvbuildres

仅限 *UNIX* 平台 — 使用资源文件，以缺省的或用户指定的关键配置属性值来创建新的 Adaptive Server、Backup Server、Monitor Server 或 XP Server。

```
srvbuildres
  [-s sybase_dir]
  [-l interfaces_file]
  [-r resource_file]
```

- 或：srvbuildres -v

startserver

仅限 *UNIX* 平台 — 启动 Adaptive Server 或 Backup Server。

```
startserver [[-f runserverfile] [-m]] ...
```

sybdiag

是一个基于 Java 的工具，用于收集综合的 Adaptive Server 配置和环境数据。Sybase 技术支持部门使用此信息来诊断服务器问题，从而加快客户案例处理。

```
sybdiag -U username [-P password] -S [server_name | host:port]
  [-l interfaces_file]
  [-L log_file]
  [-N num_threads]
  [-O output_directory]
  [-R resource_file]
  [-T feature_list]
  [-h]
  [-m message_level]
  [-v]
```

sybcluster

仅限 *集群环境* — 管理 Sybase 共享磁盘集群。sybcluster 允许创建、启动、停止和管理集群或集群中的任何实例。

```
sybcluster
  [-C cluster_name ]
  [-d discovery_list ]
  [-F agent_connection ]
  [-h ]
  [-l instance_name ]
  [-i input_file_path ]
  [-L ]
  [-m message_level ]
  [-P [ password ]]
  [-U user_name ] (the default value is "uafadmin")
  [-v ]
```

**sybmigrate** 将 Adaptive Server 从一个页大小转换为另一个页大小，并用于在各平台之间迁移。

```
sybmigrate [-v] [-h] [-f]
            [-D 1 | 2 | 3 | 4]
            [-l interfaces_file]
            [-r input_resource_file]
            [-m setup | migrate | validate | report]
            [-rn status | space_est | repl | diff | password]
            [-l log_file]
            [-t output_template_resource_file]
            [-J client_charset]
            [-z language]
            [-T trace_flags]
            [-Tase trace_flags]
            [-f]
```

**sybtssmpasswd** 在 Tivoli Storage Manager (TSM) 客户端计算机上记录或更改用户口令，以及创建 TSM 加密口令文件 *TSM.PWD*。该文件位于 TSM 配置文件中 PASSWORDDIR 配置参数所指定的目录中。

```
sybtssmpasswd
```

**xpserver** 手工启动 XP Server。

```
xpserver -S XP_Server
xpserver
-SXP_Server
  [-linterfaces_file]
  [-ppriority]
  [-sstack_size]
  [-u]
  [-v]
  [-x]
```