

SYBASE®

参考手册：表

**Adaptive Server® Enterprise**

15.5

文档 ID: DC37416-01-1550-01

最后修订日期: 2009 年 11 月

版权所有 © 2010 Sybase, Inc. 保留所有权利。

本出版物适用于 Sybase 软件和任何后续版本, 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他国家 / 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, 本书的任何部分不得以任何形式、任何手段 (电子的、机械的、手动、光学的或其它手段) 进行复制、传播或翻译。

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

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.

# 目录

关于本手册 .....	ix	
<b>第 1 章</b>	<b>系统表 .....</b>	<b>1</b>
	系统表的位置 .....	1
	master 中的系统表 .....	1
	sybsecurity 中的系统表 .....	2
	sybssystemdb 中的系统表 .....	3
	所有数据库中的系统表 .....	3
	关于 sybdiagdb 数据库 .....	4
	关于 syblicenseslog 表 .....	4
	使用 Cluster Edition 中的系统表 .....	4
	timestamp 列 .....	4
	已更改的标识值 .....	5
	控制虚设表实现 .....	5
	使用系统表的规则 .....	6
	系统表的权限 .....	6
	用于系统表的锁定方案 .....	7
	保留列 .....	8
	更新系统表 .....	8
	系统表上的触发器 .....	8
	syblicenseslog .....	9
	sysalternates .....	10
	sysaltusages .....	11
	sysattributes .....	13
	sysauditoptions .....	15
	sysaudits_01 至 sysaudits_08 .....	16
	syscharsets .....	18
	syscolumns .....	19
	syscomments .....	21
	sysconfigures .....	23
	sysconstraints .....	25
	syscoordinations .....	26
	syscurconfigs .....	27
	sysdatabases .....	29

sysdepends	32
sysdevices	33
sysencryptkeys	35
sysengines	36
sysgams	37
sysindexes	38
sysinstances	41
sysjars	42
syskeys	43
syslanguages	44
syslisteners	45
syslocks	46
sysloginroles	48
syslogins	49
syslogs	51
syslogshold	52
sysmessages	53
sysmonitors	54
sysobjects	55
sysoptions	59
syspartitionkeys	60
syspartitions	61
sysprocedures	63
sysprocesses	64
sysprotects	67
sysquerymetrics	68
sysqueryplans	70
sysreferences	71
sysremotelogins	72
sysresourcelimits	73
sysroles	74
syssecmechs	75
syssegments	76
sysservers	77
syssessions	79
syslices	80
sysssrvroles	81
sysstatistics	82
sysstabstats	83
systhresholds	85
systemranges	87
systransactions	88
systypes	90
sysusages	93

	sysusermessages .....	94
	sysusers .....	95
	sysxtypes .....	96
<b>第 2 章</b>	<b>dbccdb 表 .....</b>	<b>97</b>
	dbccdb 工作空间 .....	97
	dbccdb 日志 .....	98
	dbcc_config .....	99
	dbcc_counters .....	100
	dbcc_exclusions .....	101
	dbcc_fault_params .....	102
	dbcc_faults .....	103
	dbcc_operation_log .....	104
	dbcc_operation_results .....	105
	dbcc_types .....	106
<b>第 3 章</b>	<b>监控表 .....</b>	<b>113</b>
	monCachedObject .....	114
	monCachePool .....	115
	monCachedProcedures .....	116
	monCachedStatement .....	117
	monCIPC .....	119
	monCIPCEndpoints .....	120
	monCIPCLinks .....	121
	monCIPCMesh .....	122
	monCLMObjectActivity .....	123
	monClusterCacheManager .....	125
	monCMSFailover .....	126
	monDataCache .....	127
	monDBRecovery .....	128
	monDBRecoveryLRTypes .....	130
	monDeadLock .....	131
	monDeviceIO .....	133
	monEngine .....	134
	monErrorLog .....	136
	monFailoverRecovery .....	136
	monIOQueue .....	137
	monLicense .....	138
	monLocks .....	139
	monLogicalCluster .....	141
	monLogicalClusterAction .....	143
	monLogicalClusterInstance .....	144
	monLogicalClusterRoute .....	145

monNetworkIO .....	146
monOpenDatabases .....	147
monOpenObjectActivity .....	148
monOpenPartitionActivity .....	151
monPCIBridge .....	154
monPCIEngine .....	154
monPCISlots .....	155
monPCM .....	156
monProcedureCache .....	158
monProcedureCacheMemoryUsage .....	159
monProcedureCacheModuleUsage .....	160
monProcess .....	161
monProcessActivity .....	162
monProcessLookup .....	163
monProcessMigration .....	163
monProcessNetIO .....	164
monProcessObject .....	165
monProcessProcedures .....	166
monProcessSQLText .....	167
monProcessStatement .....	168
monProcessWaits .....	169
monProcessWorkerThread .....	170
monState .....	171
monStatementCache .....	172
monSysLoad .....	173
monSysPlanText .....	174
monSysSQLText .....	175
monSysStatement .....	176
monSysWaits .....	178
monSysWorkerThread .....	179
monTableColumns .....	180
monTableParameters .....	181
monTables .....	182
monTableTransfer .....	183
monTempdbActivity .....	184
monWaitClassInfo .....	185
monWaitEventInfo .....	186
monWorkload .....	187
monWorkloadPreview .....	188
monWorkloadProfile .....	189
monWorkloadRaw .....	190

---

<b>第 4 章</b>	<b>sybpcidb 表</b> .....	<b>191</b>
	pca_jre_arguments .....	192
	pca_jre_directives .....	193
	pci_arguments .....	194
	pci_directives .....	195
	pci_slotinfo .....	196
	pci_slot_syscalls .....	197
<b>索引</b> .....		<b>199</b>





# 关于本手册

《Adaptive Server 参考手册》包含有关 Sybase<sup>®</sup> Adaptive Server<sup>®</sup> Enterprise 和 Transact-SQL<sup>®</sup> 语言的四本指南：

- 《构件块》介绍了 Transact-SQL 的各个部件：数据类型、内置函数、全局变量、表达式、标识符、保留字和 SQLSTATE 错误。要想成功使用 Transact-SQL，您需要首先了解这些构件块的功能，以及它们对 Transact-SQL 语句结果的影响。
- 《命令》提供了有关用于创建语句的各种 Transact-SQL 命令的参考信息。
- 《过程》提供了有关系统过程、目录存储过程、扩展存储过程和 dbcc 存储过程的参考信息。所有过程都是使用 Transact-SQL 语句创建的。
- 《表》提供了有关系统表的参考信息。系统表中存储了有关服务器、数据库和用户的信息，以及服务器的其它详细信息。它还提供有关 dbccdb 和 dbccalt 数据库中的表的信息。

## 读者

《Adaptive Server 参考手册》是适合于各种层次的 Transact-SQL 用户的参考工具。

## 如何使用本手册

- [第 1 章“系统表”](#) 包含了有关 master 数据库、auditing 数据库和任何用户数据库（如 pubs2）中所有系统表的信息。本章还介绍了 syblicenseslog，它不是一个系统数据库，但包含有关许可证的信息。
- [第 2 章“dbccdb 表”](#) 包含了有关 dbccdb 和 dbccalt 数据库中的表的信息。
- [第 3 章“监控表”](#) 包含有关监控表的信息。

## 相关文档

Adaptive Server<sup>®</sup> Enterprise 文档集包括：

- 针对所用平台的发行公告 — 包含未能及时写入手册的最新信息。  
最新版本的发行公告可能已可获得。要了解本产品 CD 发行以后增加的重要产品或文档信息，请使用 Sybase<sup>®</sup> Product Manuals 网站。
- 针对所用平台的安装指南 — 介绍所有 Adaptive Server 产品及相关 Sybase 产品的安装、升级和一些配置过程。

- 
- **New Feature Summary**（《新增功能摘要》）— 介绍 Adaptive Server 中的新增功能，为支持这些功能所做的系统更改，以及可能会影响现有应用程序的更改。
  - **Active Messaging Users Guide**（《Active Messaging 用户指南》）— 介绍如何使用 Active Messaging 功能捕获 Adaptive Server Enterprise 数据库中的事务（数据更改），并将它们作为事件实时发送给外部应用程序。
  - 《组件集成服务用户指南》— 说明如何使用组件集成服务功能来连接远程 Sybase 和非 Sybase 数据库。
  - 针对所用平台的《配置指南》— 提供执行特定配置任务的操作说明。
  - 《词汇表》— 定义 Adaptive Server 文档中使用的技术术语。
  - **Historical Server Users Guide**（《Historical Server 用户指南》）— 介绍如何使用 Historical Server 从 Adaptive Server 获取性能信息。
  - 《Adaptive Server Enterprise 中的 Java》— 介绍如何在 Adaptive Server 数据库中安装 Java 类，以及如何将它们用作数据类型、函数及存储过程。
  - 《Job Scheduler 用户指南》— 提供有关如何使用命令行或图形用户界面 (GUI) 在本地或远程 Adaptive Server 上安装、配置、创建和调度作业的操作说明。
  - 《迁移技术指南》— 介绍迁移到 Adaptive Server 不同版本的策略和工具。
  - **Monitor Client Library Programmers Guide**（《Monitor Client Library 程序员指南》）— 介绍如何编写访问 Adaptive Server 性能数据的 Monitor Client Library 应用程序。
  - **Monitor Server Users Guide**（《Monitor Server 用户指南》）— 介绍如何使用 Monitor Server 从 Adaptive Server 获取性能统计信息。
  - **Monitoring Tables Diagram**（《监控表框图》）— 以张贴画的形式阐明监控表及其实体关系。大图只在印刷品中提供；采用 PDF 格式时提供缩略图。

- Performance and Tuning Series (《性能和调优系列》) — 一个丛书系列, 介绍如何调整 Adaptive Server 以获得最优性能:
  - Basics (《基础知识》) — 通晓和研究 Adaptive Server 性能问题所需具备的基础知识。
  - Improving Performance with Statistical Analysis (《利用统计分析改进性能》) — 介绍了 Adaptive Server 如何存储和显示统计信息, 及如何使用 `set statistics` 命令分析服务器统计信息。
  - Locking and Concurrency Control (《锁定和并发控制》) — 介绍如何使用锁定方案改进性能, 及如何选择索引以最大限度地降低并发操作次数。
  - Monitoring Adaptive Server with `sp_sysmon` (《使用 `sp_sysmon` 监控 Adaptive Server》) — 讨论如何使用 `sp_sysmon` 监控性能。
  - Monitoring Tables (《监控表》) — 介绍如何从 Adaptive Server 的监控表中查询统计信息和诊断信息。
  - Physical Database Tuning (《物理数据库调优》) — 介绍如何管理物理数据放置、为数据分配的空间及临时数据库。
  - Query Processing and Abstract Plans (《查询处理和抽象计划》) — 介绍优化程序如何处理查询, 以及如何使用抽象计划更改某些优化程序计划。
- 《快速参考指南》 — 这是一本袖珍手册, 完整地列出了各种命令、函数、系统过程、扩展系统过程、数据类型和实用程序的名称和语法 (该手册在用 PDF 格式阅读时采用正常大小)。
- 《参考手册》 — 一个丛书系列, 包含详细的 Transact-SQL<sup>®</sup> 信息:
  - 《构件块》 — 讨论数据类型、函数、全局变量、表达式、标识符、通配符和保留字。
  - 《命令》 — 提供了命令的文档资料。
  - 《过程》 — 介绍系统过程、目录存储过程、系统扩展存储过程及 `dbcc` 存储过程。
  - 《表》 — 讨论系统表、监控表及 `dbcc` 表。

- 
- 《系统管理指南》 —
    - 《卷 1》 — 介绍系统管理的基本知识，包括配置参数说明、资源问题、字符集、排序顺序及诊断系统问题的操作说明。《卷 1》第二部分深入讨论了安全性管理。
    - 《卷 2》 — 包括管理物理资源、镜像设备、配置内存和数据高速缓存、管理多处理器服务器和用户数据库、装入和卸下数据库、创建和使用段、使用 `reorg` 命令及检查数据库一致性的操作说明和指导。《卷 2》后半部分介绍如何备份和恢复系统数据库和用户数据库。
  - System Tables Diagram (《系统表框图》) — 以张贴画的形式阐明系统表及其实体关系。大图只在印刷品中提供；采用 PDF 格式时提供缩略图。
  - 《Transact-SQL 用户指南》 — 提供有关 Transact-SQL 这一 Sybase 关系数据库语言增强版的文档资料。本指南可作为数据库管理系统入门用户的教科书，还包括 `pubs2` 和 `pubs3` 示例数据库的详细说明。
  - Troubleshooting: Error Messages Advanced Resolutions (《故障排除：错误消息高级解析》) — 包括可能会遇到的问题的故障排除步骤。此处讨论的问题是 Sybase 技术支持部门人员最常听到的问题。
  - 《加密列用户指南》 — 介绍了如何利用 Adaptive Server 配置和使用加密列。
  - In-Memory Database Users Guide (《内存数据库用户指南》) — 介绍了如何配置和使用内存数据库。
  - Using Adaptive Server Distributed Transaction Management Features (《使用 Adaptive Server 分布式事务管理功能》) — 介绍如何在分布式事务处理环境中配置、使用 Adaptive Server DTM 功能以及排除其故障。
  - 《将 Backup Server 与 IBM® Tivoli® Storage Manager 配合使用》 — 说明如何设置和使用 IBM Tivoli Storage Manager 以创建 Adaptive Server 备份。
  - 《在高可用性系统中使用 Sybase 故障切换》 — 提供有关使用 Sybase 的故障切换功能将 Adaptive Server 配置为高可用性系统中的协同服务器的操作说明。
  - Unified Agent and Agent Management Console (《Unified Agent 和 Agent Management Console》) — 介绍 Unified Agent，它用于提供管理、监控和控制分布式 Sybase 资源的运行时服务。

- 《实用程序指南》— 提供有关在操作系统级别执行的 Adaptive Server 实用程序（如 isql 和 bcp）的文档资料。
- Web Services Users Guide（《Web 服务用户指南》）— 介绍如何配置、使用 Adaptive Server Web 服务及排除其故障。
- 《适用于 CICS、Encina 和 TUXEDO 的 XA 接口集成指南》— 提供有关将 Sybase DTM XA 接口与 X/Open XA 事务管理器配合使用的说明。
- 《Adaptive Server Enterprise 中的 XML 服务》— 介绍了 Sybase 本机 XML 处理器、基于 Sybase Java 的 XML 支持及数据库中的 XML，并提供了 XML 服务中可用的查询和映射函数的文档资料。

#### 其它信息来源

使用 Sybase Getting Started CD、SyBooks™ CD 和 Sybase Product Manuals 网站可以了解有关产品的更多信息：

- Getting Started CD 包含 PDF 格式的发行公告和安装指南，还可能包含 SyBooks CD 中未收纳的其它文档或更新信息。它随软件一起提供。若要阅读或打印 Getting Started CD 上的文档，需要使用 Adobe Acrobat Reader，该软件可以使用 CD 上提供的链接从 Adobe Web 站点免费下载。
- SyBooks CD 含有产品手册，它随软件一起提供。基于 Eclipse 的 SyBooks 浏览器使您能够以易于使用的、基于 HTML 的格式阅读手册。

有些文档可能是以 PDF 格式提供的，您可以通过 SyBooks CD 上的 PDF 目录访问这些文档。若要阅读或打印 PDF 文件，您需要使用 Adobe Acrobat Reader。

有关安装和启动 SyBooks 的说明，请参见 Getting Started CD 上的《SyBooks 安装指南》或 SyBooks CD 上的 *README.txt* 文件。

- Sybase Product Manuals Web 站点是 SyBooks CD 的联机版本，您可以使用一种标准 Web 浏览器来访问它。除了产品手册之外，还可以找到有关 EBFs/Maintenance（EBF/ 维护）、Technical Documents（技术文档）、Case Management（案例管理）、Solved Cases（解决的案例）、Newsgroups（新闻组）和 Sybase Developer Network（Sybase 开发人员网络）的链接。

要访问 Sybase 产品手册网站，请转到位于 <http://www.sybase.com/support/manuals/> 的“产品手册” (Product Manuals)。

---

## Web 上的 Sybase 认证

Sybase 网站上的技术文档在不断地更新。

### ❖ 查找有关产品认证的最新信息

- 1 将 Web 浏览器的地址指向位于 <http://www.sybase.com/support/techdocs/> 的“技术文档”(Technical Documents)。
- 2 单击“认证报告”(Certification Report)。
- 3 在“认证报告”(Certification Report) 过滤器中选择相应的产品、平台和时间范围，然后单击“查找”(Go)。
- 4 单击“认证报告”(Certification Report) 标题显示此报告。

### ❖ 查找有关组件认证的最新信息

- 1 将 Web 浏览器的地址指向位于 <http://certification.sybase.com/> 的“可用性和认证报告”(Availability and Certification Reports)。
- 2 在“按基本产品搜索”(Search by Base Product) 下选择产品系列和产品，或在“按平台搜索”(Search by Platform) 下选择平台和产品。
- 3 选择“搜索”(Search) 以显示所选项目的可用性和认证报告。

### ❖ 创建 Sybase Web 站点（包括支持页）的个人化视图

建立 MySybase 配置文件。MySybase 是一项免费服务，它允许您创建 Sybase Web 页的个人化视图。

- 1 将 Web 浏览器的地址指向位于 <http://www.sybase.com/support/techdocs/> 的“技术文档”(Technical Documents)。
- 2 单击“MySybase”并创建 MySybase 配置文件。

## Sybase EBF 和软件维护

### ❖ 查找有关 EBF 和软件维护的最新信息

- 1 将 Web 浏览器的地址指向位于 <http://www.sybase.com/support> 的“Sybase 支持页”(Sybase Support Page)。
- 2 选择“EBF/ 维护”(EBFs/Maintenance)。如果出现提示信息，请输入您的 MySybase 用户名和口令。
- 3 选择一个产品。

- 4 指定时间范围并单击“查找”(Go)。即会显示 EBF/ 维护版本的列表。

锁形图标表示因为您没有注册为“技术支持联系人”(Technical Support Contact)，因此您没有某些 EBF/ 维护版本的下载授权。如果您尚未注册，但拥有 Sybase 代表提供的或通过支持合同获得的有效信息，请单击“Edit Roles”(编辑角色)将“Technical Support Contact”(技术支持联系人)角色添加到 MySybase 配置文件中。

- 5 单击信息图标可显示 EBF/ 维护报告，单击产品说明可下载软件。

## 约定

以下各部分将说明在本手册中使用的约定。

SQL 是一种形式自由的语言。没有规定每一行中的单词数量或者必须折行的地方。然而，为便于阅读，本手册中所有示例和大多数语法语句都经过了格式设置，以便语句的每个子句都在一个新行上开始。有多个成分的子句会扩展到其它行，这些行会有缩进。复杂命令使用已修改的 Backus Naur Form (BNF) 表示法进行了格式处理。

表 1 说明本手册中出现的语法语句的约定：

**表 1：本手册的字体和语法定义**

元素	示例
命令名、过程名、实用程序名和其它关键字用 sans serif 字体显示。	<code>select</code> <code>sp_configure</code>
数据库名和数据库类型用 sans serif 字体显示。	<code>master</code> 数据库
书名采用正常字体并加书名号；文件名、变量和路径名用斜体显示。	《系统管理指南》 <code>sql.ini</code> 文件 <code>column_name</code> \$SYBASE/ASE 目录
变量（即代表您要填充的值的词语）作为查询或语句的一部分出现时用斜体的 Courier 字体显示。	<code>select column_name</code> <code>from table_name</code> <code>where search_conditions</code>
键入小括号作为命令的一部分。	<code>compute row_aggregate (column_name)</code>
双冒号加等号表示语法是用 BNF 表示法编写的。请勿输入此符号。表示“被定义为”。	<code>::=</code>
大括号表示至少选择括号中的一个选项。不要输入大括号。	<code>{cash, check, credit}</code>
中括号表示可以选择括号中的一个或多个选项，也可不选。不要输入中括号。	<code>[cash   check   credit]</code>
逗号表示可以选择任意多个所显示的选项。可用逗号作为命令的一部分来分隔选项。	<code>cash, check, credit</code>
竖线 ( ) 表示只可选择所显示的选项中的一个。	<code>cash   check   credit</code>

---

**元素****示例**

---

省略号 (...) 表示可以将最后一个单元 *重复* 任意多次。

```
buy thing = price [cash | check | credit]
[, thing = price [cash | check | credit]]...
```

您必须至少购买一件产品，并给出其价格。可以选择一种付款方式：方括号中的选项之一。还可选择购买其它物品：可根据需要购买任意数量的物品。对于要买的每种产品，给出其名称、价格和付款方式（可选）。

---

- 语法语句（显示命令的语法和所有选项）显示如下：

```
sp_dropdevice [device_name]
```

对于具有多个选项的命令：

```
select column_name
from table_name
where search_conditions
```

在语法语句中，关键字（命令）采用常规字体，而标识符为小写。斜体表示用户提供的內容。

- 说明 Transact-SQL 命令用法的示例显示如下：

```
select * from publishers
```

- 计算机输出的示例如下所示：

pub_id	pub_name	city	state
0736	New Age Books	Boston	MA
0877	Binnet & Hardley	Washington	DC
1389	Algodata Infosystems	Berkeley	CA

(3 rows affected)

本手册中的大多数示例都用小写显示。不过，输入 Transact-SQL 关键字时可以忽略大小写。例如，**SELECT**、**Select** 和 **select** 是相同的。

Adaptive Server 是否区分数据库对象（如表名）的大小写，取决于安装在 Adaptive Server 上的排序顺序。通过重新配置 Adaptive Server 的排序顺序，可改变单字节字符集的区别大小写设置。有关详细信息，请参见《系统管理指南》。



**辅助功能特性**

本档具有为提供辅助功能而进行了专门设计的 HTML 版本。可以利用适应性技术（如屏幕阅读器）浏览 HTML 文档，也可以用屏幕放大器进行查看。

Adaptive Server HTML 文档已经过测试，符合美国政府“第 508 节辅助功能”的要求。符合“第 508 节”的文档一般也符合非美国的辅助功能原则，如 World Wide Web 协会 (W3C) 针对网站的原则。

---

**注释** 您可能需要对辅助功能工具进行配置以实现最优化。某些屏幕阅读器按照大小写来辨别文本，例如将“ALL UPPERCASE TEXT”看作首字母缩写，而将“MixedCase Text”看作单词。对工具进行配置，规定语约定，您可能会感觉更方便。有关工具的信息，请查阅文档。

---

有关 Sybase 如何支持辅助功能的信息，请参见位于 <http://www.sybase.com/accessibility> 的“Sybase 辅助功能” (Sybase Accessibility)。“Sybase 辅助功能” (Sybase Accessibility) 站点包括指向“第 508 节”和 W3C 标准相关信息的链接。

**如果需要帮助**

对于购买了支持合同的客户安装的每一个 Sybase 产品，都会有一位或多位指定人员获得与 Sybase 技术支持部门联系的授权。如果使用手册或联机帮助不能解决问题，可让指定人员与 Sybase 技术支持部门联系或与所在区域的 Sybase 子公司联系。



# 系统表

系统表是 Sybase 提供的表。Adaptive Server version 15.0 中的大多数系统表现在都是行锁定表。那些不是行锁定表的系统表，将在单独的系统表说明中加以注明。

主题	页码
<a href="#">系统表的位置</a>	1
<a href="#">使用系统表的规则</a>	6

## 系统表的位置

系统表可能位于：

- master 数据库、
- sybsecurity 数据库、
- sybssystemdb 数据库或
- 所有数据库

master 数据库中的大部分表都是系统表。其中一些表也会出现在用户数据库中。当发出 `create database` 命令之后，将自动创建它们。

## master 中的系统表

以下系统表仅出现在 master 数据库中：

系统表	内容
<a href="#">syscharsets</a>	每个字符集或排序顺序在其中都有一行。
<a href="#">sysconfigures</a>	每个可由用户设置的配置参数在其中都有一行。
<a href="#">syscurconfigs</a>	有关 Adaptive Server 当前使用的配置参数的信息。
<a href="#">sysdatabases</a>	Adaptive Server 上的每个数据库在其中都有一行。
<a href="#">sysdevices</a>	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其中都有一行。

系统表	内容
<a href="#">sysengines</a>	当前处于联机状态的每个 Adaptive Server 引擎在其中都有相应的一行。
<a href="#">syslanguages</a>	服务器能识别的每种语言（美式英语除外）在其中都有相应的一行。
<a href="#">syslisteners</a>	当前 Adaptive Server 使用的每种网络连接类型在其中都有相应的一行。
<a href="#">syslocks</a>	有关活动锁的信息。
<a href="#">sysloginroles</a>	每个具有系统角色的服务器登录名在其中都有相应的一行。
<a href="#">syslogins</a>	每个有效的 Adaptive Server 用户帐户在其中都有相应的一行。
<a href="#">syslogshold</a>	有关每个数据库最早的活动事务和 Replication Server <sup>®</sup> 截断点的信息。
<a href="#">sysmessages</a>	每个系统错误或警告在其中都有相应的一行。
<a href="#">sysmonitors</a>	每个监控计数器在其中都有相应的一行。
<a href="#">sysprocesses</a>	有关服务器进程的信息。
<a href="#">sysremotelogins</a>	每个远程用户在其中都有相应的一行。
<a href="#">sysresourcelimits</a>	每个资源限制在其中都有相应的一行。
<a href="#">syssecmechs</a>	可供 Adaptive Server 使用的每个安全机制的可用安全服务的有关信息。
<a href="#">syssservers</a>	每个远程 Adaptive Server 在其中都有相应的一行。
<a href="#">sysssessions</a>	仅用于已在高可用性系统中将 Adaptive Server 配置为用于 Sybase 故障切换的情况。每个连接到具有故障切换属性的 Adaptive Server 的客户端，在 sysssessions 中都有相应的一行。
<a href="#">sysssrvroles</a>	每个全服务器范围的角色在其中都有相应的一行。
<a href="#">systimeranges</a>	每个已指定的时间范围在其中都有相应的一行。
<a href="#">systransactions</a>	每个事务在其中都有相应的一行。
<a href="#">sysusages</a>	分配给数据库的每个磁盘区段在其中都有相应的一行。

## sybsecurity 中的系统表

以下系统表仅出现在 sybsecurity 数据库中：

系统表	内容
<a href="#">sysauditoptions</a>	每个全局审计选项在其中都有相应的一行。
<a href="#">sysaudits_01</a> 至 <a href="#">sysaudits_08</a>	审计追踪。每个审计记录在审计表中都有相应的一行。

所有与审计相关的系统表都是所有页锁定表。

## sysystemdb 中的系统表

以下系统表仅出现在 sysystemdb 数据库中：

系统表	内容
<a href="#">syscoordinations</a>	分布式事务的每个远程参与者在其中都有一行。

## 所有数据库中的系统表

以下系统表出现在所有数据库中：

系统表	内容
<a href="#">sysalternates</a>	映射到数据库用户的每个 Adaptive Server 用户在其中都有一行。
<a href="#">sysattributes</a>	每个对象属性定义在其中都有一行。
<a href="#">syscolumns</a>	表或视图中的每一列以及过程中的每个参数在其中都有一行。
<a href="#">syscomments</a>	每个视图、规则、缺省值、触发器和过程在其中都有一行或多行（提供 SQL 定义语句）。
<a href="#">sysconstraints</a>	与表或列相关联的每个参照约束和检查约束在其中都有一行。
<a href="#">sysdepends</a>	过程、视图或触发器引用的每个过程、视图或表在其中都有一行。
<a href="#">sysgams</a>	整个数据库的分配位图。
<a href="#">sysindexes</a>	每个聚簇或非聚簇索引、每个无索引的表以及每个包含 text 或 image 数据的表在其中都有一行。
<a href="#">sysjars</a>	数据库中保留的每个 Java 存档 (JAR) 文件在其中都有一行。
<a href="#">syskeys</a>	每个主键、外键或公用键在其中都有一行；由用户设置（而不是由 Adaptive Server 维护）。
<a href="#">syslogs</a>	事务日志。
<a href="#">sysobjects</a>	每个表、视图、过程、规则、触发器缺省值、日志和临时对象（仅在 tempdb 中）在其中都有一行。
<a href="#">syspartitionkeys</a>	每个分区键都有一行。
<a href="#">syspartitions</a>	分区表的每个分区或索引在其中都有一行。
<a href="#">sysprocedures</a>	每个视图、规则、缺省值、触发器和过程在其中都有一行（提供内部定义）。
<a href="#">sysprotects</a>	用户权限信息。
<a href="#">sysquerymetrics</a>	将集合的历史查询信息收集在一个持久目录中。sysquerymetrics 是一个视图，而不是一个表。
<a href="#">sysqueryplans</a>	抽象查询计划和 SQL 文本。
<a href="#">sysreferences</a>	在表或列上声明的每个参照完整性约束在其中都有一行。
<a href="#">sysroles</a>	将全服务器范围的角色映射到本地数据库组。
<a href="#">syssegments</a>	每个段（命名的磁盘区段集）在其中都有一行。
<a href="#">syslices</a>	已过时，仅用于升级期间。它在 Adaptive Server 15.0 版本之前的名称是 syspartitions。

系统表	内容
<a href="#">sysstatistics</a>	用户表上的每个已索引列在其中都有相应的一行或多行。未索引的列在其中也可能有相应的行。
<a href="#">systabstats</a>	每个表在其中都有相应的一行，而每个非聚簇索引在其中也有相应的一行。
<a href="#">systhresholds</a>	为数据库定义的每个阈值在其中都有相应的一行。
<a href="#">systypes</a>	每个系统提供的数据类型和用户定义的数据类型在其中都有相应的一行。
<a href="#">sysusermessages</a>	每个用户定义的消息在其中都有相应的一行。
<a href="#">sysusers</a>	数据库中允许的每个用户在其中都有相应的一行。
<a href="#">sysxtypes</a>	每个扩展 Java-SQL 数据类型在其中都有相应的一行。它使用行级锁定。

## 关于 *sybdiagdb* 数据库

Sybase 技术支持部门可能会在您的系统上创建 *sybdiagdb* 数据库以进行调试。该数据库保存技术支持代表所使用的诊断配置数据。

## 关于 *syblicenseslog* 表

[第 9 页的 \*syblicenseslog\*](#) 中提供了对 *syblicenseslog* 表的说明。从技术角度上说，它不是系统表，但您可能需要查阅它以获得与关闭 Adaptive Server 有关的许可证信息。

## 使用 Cluster Edition 中的系统表

本节介绍对 Cluster Edition 的系统表的一般更改。在表标题下列出了对特定表的更改。

### *timestamp* 列

在 Adaptive Server 中，如果表包括 *timestamp* 列，则当更改行时，该列的值会被更新。客户端应用程序可以利用此功能来使用称为“优化锁定”的访问方法检测到行的更改。*timestamp* 列中的值在数据库中是唯一的。然而，在 Cluster Edition 中，不能保证 *timestamp* 列值在数据库的各表之间按渐增顺序排序，但对于特定表，可保证它们按渐增顺序排序。

## 已更改的标识值

Cluster Edition 中的标识列的行为方式与 Adaptive Server 的非聚簇版本的标识列不同。虽然 Cluster Edition 可确保标识值是唯一的，但由于性能原因，标识值可能不会单调递增。

在非聚簇 Adaptive Server 中，一组标识值被保存到内存中，以便在 inserts 从内存中访问下一个值时减少磁盘 I/O。在 Cluster Edition 中，相同大小的 set 被保存到内存中，但该 set 会在各集群实例之间进行分配。在标识 set 大小为 250000 的两实例集群中，第一个实例插入值 {1、2、3 等}，第二个实例插入值 {125000、125001、125002 等}。

next-identity 函数将报告在其中执行 next-identity 的实例中的表的下一个标识值。例如，对于实例 1，next-identity 将返回 4，对于实例 2，则返回 125003。

identity-burn-max 的行为与非聚簇 Adaptive Server 的 identity-burn-max 行为保持相同，因为保存大小和保存行为在 Cluster Edition 中没有更改。

## 控制虚设表实现

某些存储过程（如 sp\_who 和 sp\_lock）从虚设表（如 sysprocesses 和 syslocks）中读取。由于它们的行未存储在磁盘上，虚设表会显示共享磁盘集群的共享数据性质的一种例外情况，并应用特殊功能。

可以通过使用 set system\_view 命令控制虚设表查询是从集群的本地实例还是从集群的所有实例返回行。set system\_view 是会话级命令。此外，set system\_view 还控制监控表实现。

有关在逻辑集群级别设置缺省系统视图的信息，请参见《集群用户指南》。

缺省情况下，Adaptive Server 只从本地实例检索行。

- 要指定虚设表查询为所有实例实现行，请使用 cluster 选项。例如：

```
set system_view cluster
```
- 要指定虚设表查询为本地实例实现行，请使用 instance 选项。例如：

```
set system_view instance
```

要检索当前 system\_view 设置，请选择 @@system\_view 全局变量。

Adaptive Server 支持以下虚设表的全集群范围实现：

- sysprocesses
- syslocks
- sysengines
- syslisteners
- sysmonitors
- syssechmechs
- syscurconfigs

---

**注释** 无论 `system_view` 设置如何，`sysinstances` 始终设置用于全集群范围实现。

---

## 使用系统表的规则

本节说明系统表的规则、限制和使用信息。

---

**注释** 缺省情况下，列定义为 NOT NULL。可空列使用 “null” 关键字来描述，并在本手册中的表的列说明中列出。

---

## 系统表的权限

系统表的使用权限，如同其它任何表的权限一样，可以由数据库所有者控制。缺省情况下，安装 Adaptive Server 时，`installmodel` 脚本会授予 “public”（所有用户）对大多数系统表和对表中的大多数字段的 `select` 访问权限。相反，Adaptive Server 在建立新的数据库时分配系统表的缺省权限。但是，对于某些系统表，例如 `sysssrvroles`，并不授予访问权限；对于某些系统表中的某些字段，也不授予访问权限。例如，在缺省情况下，所有用户可以选择 `sysobjects` 中除 `audflags` 以外的所有其它列。有关详细信息，请参见《系统管理指南，卷 1》。

`sp_helpprotect system_table_name`



例如，要检查 master 中 sysssrvroles 的权限，请执行：

```
use master
go
sp_helprotect sysssrvroles
go
```

## 用于系统表的锁定方案

在 Adaptive Server 的所有页锁定方案中，锁在数据页和索引页上获取。有关锁定方案的详细信息，请参见《性能和调优指南：锁定》。

除以下系统表使用所有页锁定之外，其它所有系统表都使用数据行锁定：

- sysusermessages
- syslices
- sysmessages

此外，以下系统表为“虚设”目录，即非面向行的目录，它们看起来好像使用所有页锁定：

- syslogs
- sysgams
- sysprocesses
- syslocks
- syscurconfigs
- syssecmechs
- sysmonitors
- sysengines
- systestlog
- syslisteners
- syslogshold

## 保留列

列说明中的“保留”一词是指，Adaptive Server 当前不使用该列。

## 更新系统表

不允许对系统表进行直接更新，甚至数据库所有者也不能这样做。不过，Adaptive Server 提供了系统过程作为替代方法，应使用这些过程来执行通常所需的任何系统表更新和添加操作。

如果不能使用系统过程对系统表进行必要的修改，则可以允许对系统表进行直接更新。要允许直接更新，系统安全员必须使用 `sp_configure` 重新设置称为 `allow updates to system tables` 的配置参数。有关详细信息，请参见《系统管理指南》。

## 系统表上的触发器

不能在系统表上创建触发器。如果尝试在系统表上创建触发器，Adaptive Server 将返回错误消息并取消该触发器。

## syblicenseslog

### 仅限 master 数据库

#### 说明

每 24 小时对 Adaptive Server 中使用的许可证最大数目进行的每次更新在 syblicenseslog 中都有相应的一行。syblicenseslog 每 24 小时更新一次。如果任何时候关闭 Adaptive Server，则在完成关闭之前，许可证使用管理器记录当前在 syblicenseslog 中使用的许可证数。启动 Adaptive Server 后，重新开始 24 小时周期。

**注释** syblicenseslog 不是系统表。它的类型为“U”，其对象 ID 大于 255。

#### 列

syblicenseslogs 的列为：

名称	数据类型	说明
status	smallint	使用的许可证的最大数量的状态；可以为以下值之一： <ul style="list-style-type: none"> <li>• 0 = 没有超过许可证数量</li> <li>• 1 = 超过许可证数量</li> <li>• -1 = 管家无法监控许可证数量</li> </ul>
logtime	datetime	写入日志的日期和时间
maxlicenses	int	在 24 小时内使用的许可证的最大数量

## sysalternates

### 所有数据库

#### 说明

映射到或别名为当前数据库用户的每个 Adaptive Server 用户在 `sysalternates` 中都有相应的一行。当用户尝试访问数据库时，Adaptive Server 将在 `sysusers` 中查找有效的 `uid` 条目。如果未找到，则在 `sysalternates.suid` 中进行查找。如果找到了用户的 `suid`，则将该用户视为其 `suid` 在 `sysalternates.altsuid` 中列出的数据库用户。

#### 列

`sysalternates` 的列为：

名称	数据类型	说明
<code>suid</code>	<code>int</code>	被映射用户的服务器用户 ID
<code>altsuid</code>	<code>int</code>	其他用户映射到的用户的服务器用户 ID

#### 索引

- `suid` 上的唯一聚簇索引。

## sysaltusages

### 空数据库

#### 说明

`sysaltusages` 系统表将存档数据库中的页码映射到数据库转储及其设备或修改页面区域中的实际页。但是，与传统数据库中的 `sysusages` 表不同，`sysaltusages` 表不会映射数据库中的每个逻辑页。`sysaltusages` 映射以下页：

- 已存储在数据库转储中的页
- 已修改并因此重新定位到修改页面区域中的页

请参见《系统管理指南，卷 2》中的第 14 章“存档数据库访问”。

#### 列

`sysaltusages` 的列为：

名称	数据类型	说明
<code>dbid</code>	<code>int</code>	存档数据库的数据库 ID
<code>altsuid</code>	<code>int</code>	物理连续页块所驻留的存档数据库段的位置 在 <code>location</code> 列中，值 5 和 6 表示该位置在数据库转储、事务日志转储或其设备中；值 7 和 8 表示该位置在已修改页区域中。值 4 用于填充物理上不可用的页的间隙。
<code>lstart</code>		物理连续页块起始位置的逻辑页码。
<code>start</code>		
<code>size</code>		物理连续页块的逻辑页的数量。
<code>vstart</code>		<code>vdevno</code> 指定的设备上连续页块起始位置的偏移。
<code>vdevno</code>		连续页块所驻留的设备的编号。
<code>segmap</code>		将此页块分配到的段的映射。

**注释** 由于 `sysaltusages` 是行锁定目录，因此您可能需要定期使用 `reorg` 回收在逻辑上已删除的空间。

空数据库存储新的 `sysaltusages` 表。空数据库用于为 `sysaltusages` 表所在的位置提供灵活性。

空数据库可以是任何数据库（也有一些数据库例外，如 **master** 和 **temporary** 数据库。） Sybase 建议您提供一个只用作空数据库的专用数据库，因为：

- **sysaltusages** 的大小可能会根据它支持的存档数据库的数量而变化。您不能减小数据库的大小，但是若数据库太大，可以将其删除，并在需要时，重新创建一个较小的数据库。
- 它允许您打开 "trunc log on checkpoint" 选项，以便自动截断数据库日志。

除了承载 **sysaltusages** 表之外，此数据库与任何其它数据库类似。您可以通过设定阈值和其它空间管理机制来管理数据库内的空间。

**空数据库** 必须输入以下内容来指定用作空数据库的数据库：

```
sp_dboption <db name>, "scratch database", "true"
```

每个存档数据库一次只能指定给一个空数据库，然而多个存档数据库可以使用同一个空数据库。如果您有大量的存档数据库，您可能想要定义多个空数据库。

# sysattributes

## 所有数据库

### 说明

系统属性定义了数据库、表、索引、用户、登录和过程等对象的属性。对象的每一属性定义（由各种系统过程配置）在 `sysattributes` 中都有相应的一行。`master.sysattributes` 将 Adaptive Server 的有效属性值和类的完整集合作为一个整体进行了定义。它还存储全服务器范围的对象（如数据库和登录）的属性定义。

只能使用系统过程来访问 `sysattributes`。修改 `sysattributes` 所需的权限取决于您使用的系统过程。

### 列

`sysattributes` 的列为：

名称	数据类型	说明
<code>class</code>	<code>smallint</code>	属性类 ID。它说明属性的类别。在 <code>master.sysattributes</code> 中，特殊类 0 指定 Adaptive Server 的所有有效属性类。
<code>attribute</code>	<code>smallint</code>	属性 ID <code>attribute</code> 在值为 1 ( <code>DECRYPT-DEFAULT_ID</code> ) 的加密列中为类型为 EC 并且属于类 25 的对象指定缺省解密。
<code>object_type</code>	<code>char(2)</code>	由一个或两个字母组成的字符 ID，用于定义与属性相关联的对象的类型。
<code>object_cinfo</code>	<code>varchar(255)</code> <code>null</code>	对象的字符串标识符（如应用程序的名称）。并非所有属性都使用该字段。
<code>object_cinfo2</code>	<code>varchar(255)</code> <code>null</code>	SDC 环境中对象的字符串标识符（如应用程序的名称）。并非所有属性都使用该字段。
<code>object</code>	<code>int null</code>	对象标识符。它可以是对象 ID、用户 ID、解密缺省 ID 或数据库 ID，具体取决于对象的类型。如果对象是表的一部分（如索引），则该列包含关联表的对象 ID。
<code>object_info1</code> , <code>object_info2</code> , <code>object_info3</code>	<code>int null</code>	定义标识对象所需的其它信息。并非所有属性都使用该字段。该字段的内容取决于所定义的属性。 <ul style="list-style-type: none"> <li><code>object_info_1</code> — 包括其加密列定义解密缺省值的表的表 ID。</li> <li><code>object_info2</code> — 指定包括解密缺省值的加密列的 <code>colid</code>。</li> </ul>
<code>int_value</code>	<code>int null</code>	属性的整数值（如用户的显示级别）。
<code>char_value</code>	<code>varchar(768)</code> <code>null</code>	属性的字符值（如高速缓存名）。
<code>text_value</code>	<code>text null</code>	属性的文本值。
<code>image_value</code>	<code>image null</code>	属性的图像值。
<code>comments</code>	<code>varchar(255)</code> <code>null</code>	有关属性定义的注释或其它信息。

表 1-1 列出了 `object_type` 中最常用的相关值。这些值提供了 `sysattributes` 的其它信息，并不作为独立的值使用。因此，请仅与类 ID 一起使用这些值。

**表 1-1: `sysattributes` 的 `object_type` 列的有效值**

值	说明
D	数据库
I	索引
L	登录
P	过程
T	表
U	用户
AP	应用程序
DC	转储条件
EL	外部登录 (OMNI)
OD	对象定义 (OMNI)
TC	事务协调 (ASTC)
TG	临时数据库组 (多个临时数据库)
TP	文本页 (OMNI)
QP	查询计划 (抽象计划)
UR	用户角色
GR	组角色
LG	登录 (用于 MTDB 绑定)
EG	引擎组
PS	口令安全性

## 索引

- `class`、`attribute`、`object_type`、`object`、`object_info1`、`object_info2`、`object_info3`、`object_cinfo` 上的唯一聚簇索引。
- `object_type`、`object`、`object_info1`、`object_info2`、`object_info3`、`object_cinfo` 上的非聚簇索引。



## sysauditoptions

### sybsecurity 数据库

**说明** 每个全服务器范围的审计选项在 `sysauditoptions` 中都有相应的一行，并且 `sysauditoptions` 会指示该选项的当前设置。其它类型的审计选项设置存储在其它表中。例如，数据库特定的选项设置存储在 `sysdatabases` 中；而对象特定的选项设置存储在 `sysobjects` 中。每个选项的缺省值是 0 或“off”。只有系统安全员能够访问 `sysauditoptions`。

**列** `sysauditoptions` 的列为：

名称	数据类型	说明
num	smallint	全服务器范围选项的数量。
val	smallint	当前值；可以为以下值之一： <ul style="list-style-type: none"> <li>• 0 = off</li> <li>• 1 = pass</li> <li>• 2 = fail</li> <li>• 3 = on</li> </ul>
minval	smallint	该选项的最小有效值。
maxval	smallint	该选项的最大有效值。
name	varchar(30)	选项的名称。
sval	varchar(30)	当前值的等值字符串：例如，“on”、“off”、“nonfatal”。
comment	varchar(255)	选项的说明。

## sysaudits\_01 至 sysaudits\_08

### sybsecurity 数据库

#### 说明

这些系统表包含审计追踪。每次只有一个表处于活动状态。活动表是由 `current audit table` 配置参数值决定的。安装的系统最多可以有八个审计表。例如，如果安装的系统有三个审计表，则将表命名为 `sysaudits_01`、`sysaudits_02` 和 `sysaudits_03`。每个审计记录在审计表中都有相应的一行。

#### 列

sysaudits\_01 — sysaudits\_08 的列为：

名称	数据类型	说明
event	smallint	审计的事件的类型。
eventmod	smallint	有关事件的详细信息。可能的值有： <ul style="list-style-type: none"> <li>• 0 = 此事件无修饰符。</li> <li>• 1 = 事件通过了权限检查。</li> <li>• 2 = 事件未通过权限检查。</li> </ul>
spid	smallint	导致写入审计记录的进程的服务器进程 ID。
		对于 Cluster Edition, 为 int
eventtime	datetime	审计事件的日期和时间。
sequence	smallint	单个事件中记录的序列号；某些事件需要多个审计记录。
suid	smallint	执行审计事件的用户的服务器登录 ID。
dbid	int null	发生审计事件的数据库的 ID 或对象 / 存储过程 / 触发器（取决于事件类型）所在数据库的 ID。
objid	int null	访问的对象或存储过程 / 触发器的 ID。
xactid	binary(6) null	包含审计事件的事务的 ID。对于多数据库事务，这是来自发起该事务的数据库的事务 ID。
loginname	varchar(30) null	与 <code>suid</code> 相对应的登录名。
dbname	varchar(30) null	与 <code>dbid</code> 相对应的数据库名。
objname	varchar(255) null	与 <code>objid</code> 相对应的对象名。
objowner	varchar(30) null	<code>objid</code> 的所有者名。
extrainfo	varchar(255) null	有关审计事件的其它信息。该字段包含一系列用分号隔开的项目。请参见表 1-2。
nodeid	tinyint null	留作将来使用（不能用于集群环境）
instanceid	tinyint	实例的 ID（只能用于集群环境）

**注释** 由于 Cluster Edition 的数据类型的这一更改，Sybase 强烈建议您在升级之前存档并截断审计表。这会降低由于 `sybsecurity` 数据库中空间不足而导致升级失败的可能性。

`extrainfo` 列中包含一系列用分号分隔的项目，如表 1-2 中所示：

**表 1-2: `extrainfo` 列中的项目**

项目	内容
角色	列出活动的角色。这些角色用空格隔开。
关键字或选项	用于事件的关键字或命令选项的名称。例如，对于 <code>alter table</code> 命令，可以使用 <code>add column</code> 或 <code>drop constraint</code> 选项。如果使用多个关键字或选项，则将其用逗号隔开。
以前值	如果事件导致更新值，则它表示更新之前的值。
当前值	如果事件导致更新值，则它表示新值。
其它信息	为事件记录的其它与安全性相关的信息。
代理信息	初始登录名（如果在 <code>set proxy</code> 有效时发生该事件）。
主要信息	来自基础安全机制的主体名称（如果用户的登录名是安全缺省登录名，并且用户通过统一登录登录到 Adaptive Server 上）。如果没有使用安全缺省登录名，则该字段的值为 <code>NULL</code> 。

对于更改审计配置参数的安全性相关事件，其 `extrainfo` 列可能如下所示：

```
sso_role;suspend auditing when full;1;0;;;
```

该示例表示系统安全员已将配置参数 `suspend auditing when full` 从 1（挂起所有涉及审计事件的进程）更改为 0（截断下一个审计表并将其作为当前的审计表）。

## syscharsets

### 仅限 master 数据库

**说明** 所定义的由 Adaptive Server 使用的每个字符集和排序顺序在 `syscharsets` 中都有相应的一行。在 `master..sysconfigures` 中将其中一个排序顺序标记为缺省排序顺序，它是实际使用的唯一排序顺序。

**列** `syscharsets` 的列为：

名称	数据类型	说明
<code>type</code>	<code>smallint</code>	该行表示的实体类型。编号 1001 到 1999 表示字符集。编号 2000 到 2999 表示排序顺序。
<code>id</code>	<code>tinyint</code>	字符集或排序顺序的 ID。排序顺序是用排序顺序 ID 和字符集 ID ( <code>csid</code> ) 组合定义的。字符集是用 <code>id</code> 定义的，此 ID 必须是唯一的。Sybase 保留从 0 到 200 的 ID 号。
<code>csid</code>	<code>tinyint</code>	如果该行表示字符集，则不使用此字段。如果该行表示排序顺序，则此字段是创建排序顺序时所基于的字符集的 ID。该表中必须存在具有该 ID 的字符集行。
<code>status</code>	<code>smallint</code>	内部系统状态信息位。
<code>name</code>	<code>varchar(30)</code>	字符集或排序顺序的唯一名称。只能使用 7 位 ASCII 字母 A–Z 或 a–z、数字 0–9 和下划线 ( <code>_</code> )，并且必须以字母开头。
<code>description</code>	<code>varchar(255)</code>	字符集或排序顺序功能的可选说明。
<code>definition</code>	<code>image</code>	字符集或排序顺序的内部定义。此字段中的数据结构取决于 <code>type</code> 。
<code>sortfile</code>	<code>varchar(30) null</code>	排序顺序文件的名称。

**索引**

- `id`、`csid` 上的唯一聚簇索引
- `name` 上的唯一非聚簇索引

# syscolumns

## 所有数据库

**说明** 每个表和视图中的每一列以及每个过程中的每一参数在 `syscolumns` 中都有相应的一行。

每个与表相关联的计算列和基于函数的索引键在其中都有相应的一行。

**列** `syscolumns` 的列为：

名称	数据类型	说明
<code>id</code>	<code>int</code>	该列所属表的 ID 或与该参数相关联的过程的 ID。
<code>number</code>	<code>smallint</code>	对过程进行分组时的子过程数（0 表示非过程条目）。
<code>colid</code>	<code>smallint</code>	列 ID。
<code>status</code>	<code>tinyint</code>	<ul style="list-style-type: none"> <li>位 0-2（值 1、2 和 4）— 表示位的位置（如果列使用 <code>bit</code> 数据类型）。如果列使用 <code>text/image</code> 数据类型，位 0 和 1 表示如下所示的复制状态： <ul style="list-style-type: none"> <li>01 = 始终复制</li> <li>10 = 仅在更改后复制</li> <li>00 = 不复制</li> </ul> </li> <li>位 3（值 8）— 表示 NULL 值在该列中是否合法。</li> <li>位 4（值 16）— 表示该列是否存在多项检查约束。</li> <li>位 5 和 6 — 供内部使用。</li> <li>位 7（值 128）— 表示标识列。</li> </ul>
<code>type</code>	<code>tinyint</code>	物理存储类型；从 <code>systypes</code> 中复制。
<code>length</code>	<code>int</code>	数据的物理长度；从 <code>systypes</code> 中复制或由用户提供。
<code>offset</code>	<code>smallint</code>	该列所在行中的偏移；如果为负值，则为可变长度列。
<code>usertype</code>	<code>smallint</code>	用户类型 ID；从 <code>systypes</code> 中复制。
<code>cdefault</code>	<code>int</code>	生成该列缺省值的过程的 ID。
<code>domain</code>	<code>int</code>	该列第一项规则或检查约束的约束 ID。
<code>name</code>	<code>varchar(255)</code> <code>not null</code>	列名
<code>printfmt</code>	<code>varchar(255)</code> <code>null</code>	保留
<code>prec</code>	<code>tinyint null</code>	有效位数（如果列使用 <code>numeric</code> 数据类型）。
<code>scale</code>	<code>tinyint null</code>	小数点右侧的位数（如果列使用 <code>numeric</code> 数据类型）。
<code>remote_type</code>	<code>int null</code>	将本地名映射到远程名。组件集成服务的访问方法需要使用它，以使该软件能够将参数中的本机列数据类型信息传递到 <code>access_server</code> 类服务器。
<code>remote_name</code>	<code>varchar(255)</code> <code>null</code>	将本地名映射到远程名。组件集成服务的访问方法需要使用它，以使用正确的远程表列名来构造查询。

名称	数据类型	说明
xstatus	int null	具有扩展数据类型的列的状态。其值为： <ul style="list-style-type: none"> <li>• 0 = 在行内</li> <li>• 1 = 在行外</li> </ul> 对于没有扩展数据类型的列，xstatus 为 NULL。
xtype	int null	类的 ID。 如果表中的列或过程中的参数将 Java 类作为其数据类型，则使用该列。在使用时，字段不为 NULL，type 的值为 0x39。有关详细信息，请参见《Adaptive Server Enterprise 中的 Java》。
xdbid	int null	类的数据库 ID。对于系统类，该值为 -1。否则，该值为当前数据库 ID。 如果表中的列或过程中的参数将 Java 类作为其数据类型，则使用该列。字段不为 NULL，type 的值为 0x39。有关详细信息，请参见《Adaptive Server Enterprise 中的 Java》。
accessrule	int null	sysprocedures 中访问规则的对象 ID。有关详细信息，请参见《系统管理指南》的第 11 章“管理用户权限”中的“行级访问控制”。
status2	int null	它表示 SQLJ 存储过程的参数模式以及 SQLJ 函数的返回类型。 提供三个内部位，用于支持计算列： <ul style="list-style-type: none"> <li>• 0x00000010，值 16 — 该列为计算列。</li> <li>• 0x00000020，值 32 — 该列为已实现的计算列。</li> <li>• 0x00000040，值 64 — 该列为视图中的计算列。</li> <li>• 0x00001000，值 4096 — 该加密列具有解密缺省值。</li> </ul> syscolumns 中的 status2 字段使用以下编码来表示列的加密属性： <ul style="list-style-type: none"> <li>• 0x80，值 128 — 该列已加密。</li> <li>• 0x100，值 256 — 已使用初始化矢量加密该列。</li> <li>• 0x200，值 512 — 已使用随机填充加密该列。</li> <li>• 0x400，值 1024 — 代理表已加密。</li> <li>• 0x1000，值 4096 — 该加密列具有解密缺省值。</li> </ul>
status3	int	0x0001，值 1 — 表示基于函数的索引键的隐藏计算列。
computedcol	int	存储计算列定义的对象 ID。
encrtype	int null	采用加密形式的数据的类型。
enclen	int null	加密数据的长度。
encrykeyid	int null	密钥的对象 ID。
encrykeydb	varchar(30) null	创建加密密钥的数据库的名称；如果它与加密列位于同一数据库中，则为 NULL
encrdate	datetime null	加密密钥的创建日期；从 sysobjects.crdate 中复制。

索引

- id、number、colid 上的唯一聚簇索引

## syscomments

### 所有数据库

#### 说明

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。`text` 列包含初始定义语句。如果 `text` 列的长度大于 255 个字节，条目将跨行显示。每个对象最多可占据 65,025 行。

它还存储计算列、基于函数的索引或分区定义的有关文本。例如，为域分区存储 “`values <= value_list`”。

`create service` 命令在 `syscomments` 中存储文本，因为它使用 `create procedure` 基础结构。

#### 列

`syscomments` 的列为：

名称	数据类型	说明
<code>id</code>	<code>int</code>	该文本适用的对象 ID。
<code>number</code>	<code>smallint</code>	对过程进行分组时的子过程数（0 表示非过程条目）。
<code>colid</code>	<code>smallint</code>	此过程的注释的列计数器的低位部分。值介于 0 到 32767 之间。如果在这么多行中无法容纳所有的过程文本，则该计数器与 <code>colid2</code> 一起发生作用。
<code>texttype</code>	<code>smallint</code>	指示注释类型。其值包括： <ul style="list-style-type: none"> <li>0 — 系统提供的注释，用于视图、规则、缺省值、触发器和过程</li> <li>1 — 用户提供的注释，用于添加对对象或列进行说明的条目</li> </ul>
<code>language</code>	<code>smallint</code>	保留。
<code>text</code>	<code>varchar(255)</code> <code>null</code>	SQL 定义语句的实际文本。
<code>colid2</code>	<code>smallint</code>	此过程的注释的列计数器的高位部分。值介于 0 到 32767 之间。只有对于注释文本超过 32,768 行的过程，值才为 0。
<code>status</code>	<code>smallint null</code>	表示对象状态的位： <ul style="list-style-type: none"> <li>0x1 — <code>SYSCOM_TEXT_HIDDEN</code> 表示文本是隐藏的</li> <li>0x2 — 保留为内部使用</li> <li>0x4 — <code>SYSCOM_QUOTED_ID_ON</code> 表示创建对象时启用带引号的标识符</li> </ul>
<code>partitionid</code>	<code>int null</code>	分区 ID。否则为空。

**注释** 不要将定义语句从 `syscomments` 的 `text` 列中删除。Adaptive Server 升级进程需要使用这些语句。要加密定义语句，请执行系统过程 `sp_hidetext`。要查看是否已删除在 11.5 版或更高版本中创建的语句，请执行 `sp_checksource`。如果删除了该语句，则必须重新创建原先创建该语句的对象，或者重新安装原先创建该对象的应用程序，这样就会重新创建该语句。

要防止对数据库对象的文本进行未经授权的访问，可施加限制，仅限对象所有者和系统管理员拥有 `syscomments` 表中 `text` 列的 `select` 权限。这种限制既适用于通过存储过程进行的访问，也适用于通过 `select` 语句进行的直接访问。在已评估的配置中运行 `Adaptive Server` 时，必须施加这种限制。要施加这种限制，系统安全员必须用系统过程 `sp_configure` 重新设置称为 `allow select on syscomments.text column` 的参数。有关已评估配置的信息，请参见《系统管理指南》。

### 索引

- `id`、`number`、`colid2`、`colid`、`texttype` 上的唯一聚簇索引



## sysconfigures

### 仅限 master 数据库

说明 每个可由用户设置的配置参数在 `sysconfigures` 中都有相应的一行。

列 `sysconfigures` 的列为：

名称	数据类型	说明
<code>config</code>	<code>smallint</code>	配置参数编号。
<code>value</code>	<code>int</code>	用户可修改的 <code>integer</code> 数据类型的参数值。对于数据类型为 <code>character</code> 的参数，其值为 0。
<code>comment</code>	<code>varchar(255)</code>	配置参数的名称。
<code>status</code>	<code>int</code>	表示配置参数类型的值。有关详细信息，请参见表 1-3。
<code>name</code>	<code>varchar(255)</code> <code>null</code>	配置参数的名称（其值与 <code>comment</code> 相同）。
<code>parent</code>	<code>smallint null</code>	父项的配置参数编号；如果有多个父项，则将其它父项编号存储在 <code>sysattributes</code> 中。
<code>value2</code>	<code>varchar(255)</code> <code>null</code>	用户可修改的 <code>character</code> 数据类型的参数值。对于具有 <code>integer</code> 数据类型的参数，其值为 <code>NULL</code> 。 <code>value2</code> 还用于存储缓冲池的池大小。
<code>value3</code>	<code>int null</code>	存储缓冲池的清洗大小。
<code>value4</code>	<code>int null</code>	存储缓冲池的异步预取百分比；如果未指定值或使用缺省值，则为 1。
<code>instanceid</code>	<code>tinyint</code>	实例的 ID。只能用于集群环境。

表 1-3 提供了有关 `status` 列的信息。

**表 1-3: Status 列说明**

Status 类型	十进制	十六进制	说明
<code>CFG_NO_OPTIONS</code>	0	0x0	参数没有选项。
<code>CFG_SYSTEM_OPTION</code>	1	0x01	参数为系统选项。
<code>CFG_SYSTEM_GROUP</code>	2	0x02	参数为系统组。
<code>CFG_STATIC</code>	4	0x04	参数为静态参数。
<code>CFG_DYNAMIC</code>	8	0x08	参数为动态参数。
<code>CFG_CALCULATED</code>	16	0x10	对参数进行计算。
<code>CFG_READONLY</code>	32	0x20	为只读参数。
<code>CFG_MEMORY_USED</code>	64	0x40	参数将占用内存。
<code>CFG_CONFIG_FILE</code>	128	0x80	为外部可见参数。
<code>CFG_SYSTEM_TAB</code>	256	0x100	参数仅在系统表中为外部可见。
<code>CFG_EXTRAS_OPTION</code>	512	0x200	参数适用于 <code>CFG_EXTRAS</code> ，而不适用于 <code>DS_CONFIG</code> 。
<code>CFG_CFGBLK</code>	1024	0x400	参数存储在配置块中。
<code>CFG_CACHE_GROUP</code>	2048	0x800	参数为高速缓存组。

Status 类型	十进制	十六进制	说明
CFG_CACHE_OPTION	4096	0x1000	参数为高速缓存选项。
CFG_BUFFER_POOL_GROUP	8192	0x2000	参数为缓冲池组。
CFG_BUFFER_POOL_OPTION	16384	0x4000	参数为缓冲池选项。
CFG_INTERNAL	32768	0x8000	参数仅供内部使用。
CFG_FNOF_LPAGESIZE	65536	0x10000	参数条目取决于逻辑页大小。

索引

- name、parent、config 上的唯一聚簇索引
- config 上的非聚簇索引
- parent、config 上的非聚簇索引

## sysconstraints

### 所有数据库

#### 说明

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 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。通过执行 `drop table` 删除表后，将从 `sysconstraints` 表中删除所有与该表关联的行。

该表对于每个与特定表相关联的检查约束、参照约束、计算列和基于函数的索引键也都有对应的一行。

#### 列

`sysconstraints` 的列为：

名称	数据类型	说明
<code>colid</code>	<code>smallint</code>	表中的列编号
<code>constrid</code>	<code>int</code>	约束的对象 ID
<code>tableid</code>	<code>int</code>	声明约束的表的 ID
<code>error</code>	<code>int</code>	约束特定的错误消息
<code>status</code>	<code>int</code>	约束的类型如下： <ul style="list-style-type: none"> <li>• 0x0040 = 参照约束</li> <li>• 0x0080 = 检查约束</li> <li>• 0x0100 = 计算列对象约束</li> </ul>
<code>spare2</code>	<code>int</code>	未使用

#### 索引

- `tableid`、`colid` 上的唯一聚簇索引
- `constrid` 上的非聚簇索引

## syscoordinations

### 仅限 subsystemdb 数据库

说明 syscoordinations 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 syscoordinations 的列为：

名称	数据类型	说明
participant	smallint	参与者 ID
starttime	datetime	事务的启动日期
coordtype	tinyint	systransactions 表定义中表示协调方法或协议的值
owner	tinyint	行所有者（供内部使用）
protocol	smallint	留作内部使用
state	int	表示远程参与者当前状态的值（请参见表 1-4）
bootcount	int	留作内部使用
dbid	smallint	事务开始时的数据库 ID。
logvers	tinyint	留作内部使用
spare	tinyint	留作内部使用
status	int	留作内部使用
xactkey	binary(14)	唯一的 Adaptive Server 事务关键字
gtrid	varchar(255) null	由 Adaptive Server 协调的分布式事务的全局事务 ID（留作内部使用）
partdata	varbinary(255) null	留作内部使用
srvname	varchar(30) null	本地服务器的名称（对于远程服务器为 null）
nodeid	tinyint null	不能用于非集群环境 — 留作将来使用
instanceid	tinyint	只适用于集群环境 — 实例的 ID

表 1-4 列出了 state 列的值：

**表 1-4: syscoordinations state 值**

state 值	参与者状态
1	Begun
4	Prepared
7	Committed
9	In Abort Tran

索引

- xactkey、participant、owner 上的唯一聚簇索引

## syscurconfigs

### 仅限 master 数据库

**说明** syscurconfigs 是在执行查询时动态创建的。每个配置参数在该表中都有对应的条目（与在 sysconfigures 中一样），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** syscurconfigs 的列为：

名称	数据类型	说明
config	smallint	配置参数编号。
value	int	数据类型为 integer 的参数的当前运行值。对于数据类型为 character 的参数，其值为 0。
comment	varchar(255)	关于配置参数的注释信息。留作内部使用。
status	int	表示配置参数类型的值。有关详细信息，请参见表 1-5。
value2	varchar(255) null	数据类型为 character 的参数的当前运行值。对于数据类型为 integer 的参数，其值为 NULL。
defvalue	varchar(255) null	配置参数的缺省值。
minimum_value	int null	配置参数的最小值。
maximum_value	int null	配置参数的最大值。
memory_used	int null	每个配置参数所用内存量的整数值。负值表示内存被共享使用。
display_level	int null	配置参数的显示级别。值为 1、5 和 10。
datatype	int null	配置参数的数据类型。
message_num	int null	该配置参数的 sp_helpconfig 消息的消息号。
apf_percent	int null	缓冲池异步预取百分比的当前运行值。仅对表示缓冲池的行有效。
nodeid	tinyint null	留作将来使用（在集群环境中不可用）
instanceid	tinyint	实例的 ID（只能用于集群环境） <ul style="list-style-type: none"> <li>• 不适用 — 参数没有单位</li> <li>• 数字 — 项目数</li> <li>• 时钟周期 — 时钟周期数</li> <li>• 微秒</li> <li>• 毫秒</li> <li>• 秒</li> <li>• 分钟</li> <li>• 小时</li> <li>• 天</li> <li>• 字节</li> <li>• 千字节</li> <li>• 兆字节</li> <li>• 内存页 (2K)</li> <li>• 虚拟页 (2K)</li> <li>• 逻辑页</li> <li>• 百分比</li> <li>• 比率</li> <li>• 开关 — 布尔值</li> <li>• ID — ID 号</li> <li>• 名称</li> <li>• 行</li> </ul>

名称	数据类型	说明
type	varchar(10) null	<p>指定在配置参数的结构定义中将配置参数声明为动态参数还是静态参数。其值包括：</p> <ul style="list-style-type: none"> <li>• Dynamic — 立即生效。</li> <li>• Static — 在重新启动 Adaptive Server 后生效。</li> </ul>

表 1-5 提供了有关 status 列的信息。

**表 1-5: Status 列说明**

Status 类型	十进制	十六进制	说明
CFG_NO_OPTIONS	0	0x0	参数没有选项。
CFG_SYSTEM_OPTION	1	0x01	参数为系统选项。
CFG_SYSTEM_GROUP	2	0x02	参数为系统组。
CFG_STATIC	4	0x04	参数为静态参数。
CFG_DYNAMIC	8	0x08	参数为动态参数。
CFG_CALCULATED	16	0x10	对参数进行计算。
CFG_READONLY	32	0x20	为只读参数。
CFG_MEMORY_USED	64	0x40	参数将占用内存。
CFG_CONFIG_FILE	128	0x80	为外部可见参数。
CFG_SYSTEM_TAB	256	0x100	参数在系统表中仅为外部可见。
CFG_EXTRAS_OPTION	512	0x200	参数适用于 CFG_EXTRAS，而不适用于 DS_CONFIG。
CFG_CFGBLK	1024	0x400	参数存储在配置块中。
CFG_CACHE_GROUP	2048	0x800	参数为高速缓存组。
CFG_CACHE_OPTION	4096	0x1000	参数为高速缓存选项。
CFG_BUFFER_POOL_GROUP	8192	0x2000	参数为缓冲池组。
CFG_BUFFER_POOL_OPTION	16384	0x4000	参数为缓冲池选项。
CFG_INTERNAL	32768	0x8000	参数仅供内部使用。
CFG_FNOF_LPAGE_SIZE	65536	0x10000	参数条目取决于逻辑页大小。

## sysdatabases

### 仅限 master 数据库

说明

Adaptive Server 中的每个数据库在 `sysdatabases` 中都有相应的一行。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

`sysdatabases` 的列为：

名称	数据类型	说明
<code>name</code>	<code>sysname</code>	数据库的名称
<code>dbid</code>	<code>smallint</code>	数据库 ID
<code>suid</code>	<code>int</code>	数据库所有者的服务器用户 ID
<code>status</code>	<code>smallint</code>	控制位；表 1-6 中列出了用户可以用 <code>sp_dboption</code> 设置的控制位
<code>version</code>	<code>smallint</code>	未使用
<code>logptr</code>	<code>int</code>	指向事务日志的指针
<code>crdate</code>	<code>datetime</code>	创建日期
<code>dumptrdate</code>	<code>datetime</code>	上次执行 <code>dump transaction</code> 时的日期
<code>status2</code>	<code>smallint null</code>	附加控制位（请参见第 30 页的表 1-7）
<code>audflags</code>	<code>int null</code>	数据库的审计设置
<code>deftabaud</code>	<code>int null</code>	为表定义缺省审计设置的位屏蔽
<code>defvwaud</code>	<code>int null</code>	为视图定义缺省审计设置的位屏蔽
<code>defpraud</code>	<code>int null</code>	为存储过程定义缺省审计设置的位屏蔽
<code>def_remote_type</code>	<code>smallint null</code>	在没有通过存储过程 <code>sp_addobjectdef</code> 提供存储位置的情况下，指定要用于远程表的缺省对象类型
<code>def_remote_loc</code>	<code>varchar(349) null</code>	在没有通过存储过程 <code>sp_addobjectdef</code> 提供存储位置的情况下，指定要用于远程表的缺省存储位置
<code>status3</code>	<code>int null</code>	附加控制位
<code>status4</code>	<code>int null</code>	附加控制位
<code>audflags2</code>	<code>varbinary(16) null</code>	留作将来使用
<code>instanceid</code>	<code>tinyint</code>	实例的 ID（只适用于 Cluster Edition）
<code>durability</code>	<code>int</code>	数据库的持久性级别。其值包括： <ul style="list-style-type: none"> <li>1 – full</li> <li>5 – at_shutdown</li> <li>6 – no_recovery</li> </ul>

表 1-6 列出了 status 列的位表示形式。

**表 1-6: sysdatabases 表中的 status 控制位**

十进制	十六进制	状态
1	0x01	在此数据库上启动了升级
2	0x02	升级已成功
4	0x04	<ul style="list-style-type: none"> <li>select into/bulkcopy</li> <li>可由用户设置</li> </ul>
8	0x08	<ul style="list-style-type: none"> <li>trunc log on chkpt</li> <li>可由用户设置</li> </ul>
16	0x10	<ul style="list-style-type: none"> <li>no chkpt on recovery</li> <li>可由用户设置</li> </ul>
32	0x20	用 for load 选项创建数据库（否则在装载数据库时崩溃），用于指示不进行恢复
64	0x04	为要恢复的所有数据库启动恢复
256	0x100	<ul style="list-style-type: none"> <li>可疑数据库</li> <li>未恢复</li> <li>无法打开或使用</li> <li>只能用 dbcc dbrepair 删除</li> </ul>
512	0x200	<ul style="list-style-type: none"> <li>ddl in tran</li> <li>可由用户设置</li> </ul>
1024	0x400	<ul style="list-style-type: none"> <li>read only</li> <li>可由用户设置</li> </ul>
2048	0x800	<ul style="list-style-type: none"> <li>dbo use only</li> <li>可由用户设置</li> </ul>
4096	0x1000	<ul style="list-style-type: none"> <li>single user</li> <li>可由用户设置</li> </ul>
8192	0x2000	<ul style="list-style-type: none"> <li>allow nulls by default</li> <li>可由用户设置</li> </ul>

表 1-7 列出了 status2 列的位表示形式。

**表 1-7: sysdatabases 表中的 status2 控制位**

十进制	十六进制	状态
1	0x0001	abort tran on log full ; 可以由用户设置
2	0x0002	no free space acctg ; 可以由用户设置
4	0x0004	auto identity ; 可以由用户设置
8	0x0008	identity in nonunique index ; 可以由用户设置
16	0x0010	数据库处于脱机状态
32	0x0020	在恢复结束之前，数据库一直处于脱机状态



十进制	十六进制	状态
64	0x0040	表具有自动标识功能，并且在 identity 列上具有唯一约束
128	0x0080	数据库中有可疑页
256	0x0100	将表结构写入磁盘。如果在恢复完成后出现此位，则说明为打开的数据库配置的此服务器参数值可能太低。请使用 <code>sp_configure</code> 增加此参数。
512	0x0200	数据库正在进行升级
1024	0x0400	将数据库联机以用于备用访问
2048	0x0800	在用户设置后，可防止通过别名机制跨数据库访问
-32768	0xFFFF8000	数据库中的某些日志不在仅供日志使用的设备上

表 1-8 列出了 status3 列的位表示形式。

**表 1-8: sysdatabases 表中的 status3 控制位**

十进制	十六进制	状态
0	0x0000	常规或标准的数据库或 create 语句中不带代理更新的数据库。
1	0x0001	已指定 proxy_update 选项，并且数据库是用户创建的代理数据库
2	0x0002	数据库是根据高可用性要求创建的代理数据库。
4	0x0004	数据库具有一个根据高可用性要求创建的代理数据库。
8	0x0008	禁止访问数据库，因为正在关闭数据库。
16	0x0010	数据库是已故障切换的数据库。
32	0x0020	数据库是类型为 master 的已装入数据库。
64	0x0040	数据库是已装入数据库。
128	0x0080	使用 quiesce database 命令禁止数据库写入。
256	0x0100	用户创建的 tempdb。
512	0x0200	在故障切换状态下，禁止对服务器中的数据库进行外部访问。
1024	0x0400	用于启用或禁用异步日志记录服务线程的、由用户提供的选项。用户可通过在特定数据库上将 sp_dboption enable async logging service 选项设置为 true 来启用此功能。
4096	0x1000	数据库已成功关闭。
8192	0x2000	drop database 正在进行中。

- 索引
- name 上的唯一聚簇索引
  - dbid 上的非聚簇索引

## sysdepends

### 所有数据库

**说明** 过程、视图或触发器引用的每个过程、视图或表在 `sysdepends` 中都有相应的一行。

**列** `sysdepends` 的列为：

名称	数据类型	说明
<code>id</code>	<code>int</code>	对象 ID。
<code>number</code>	<code>smallint</code>	过程编号。
<code>depid</code>	<code>int</code>	相关对象 ID。
<code>depnumber</code>	<code>smallint</code>	相关过程编号。
<code>status</code>	<code>smallint</code>	内部状态信息。
<code>selall</code>	<code>bit</code>	如果将对象用于 <code>select *</code> 语句，则为 <code>on</code> 。
<code>resultobj</code>	<code>bit</code>	如果正在更新对象，则为 <code>on</code> 。
<code>readobj</code>	<code>bit</code>	如果正在读取对象，则为 <code>on</code> 。
<code>Columns</code>	<code>varbinary</code>	存储一个列 ID 位图，这些列 ID 表示的列在存储过程主体中引用。该位图提供了已编译对象的列级相关性跟踪信息，并由 <code>sp_depends</code> 进行解码，以报告存储过程、触发器和视图的列级相关性。

**索引**

- `id`、`number`、`depid`、`depnumber` 上的唯一聚簇索引

## sysdevices

### 仅限 master 数据库

说明

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` 中都有相应的一行。Adaptive Server 分发介质的 `sysdevices` 中包含四个条目：一个用于主设备（用于数据库），一个用于磁盘转储设备，两个用于磁带转储设备。

**注释** 对于 Adaptive Server 15.0 版本，设备标识号存储在 `vdevno` 列中，而不再作为 `high` 或 `low` 列的一部分进行存储。因此，您可能需要对那些根据先前模式来确定设备标识号的脚本和存储过程进行修改。

列 `sysdevices` 的列为：

名称	数据类型	说明
<code>low</code>	<code>int</code>	不用于转储设备—虚拟页的块偏移量（以 2K 字节为单位）
<code>high</code>	<code>int</code>	最后一个虚拟页的块偏移量（以 2K 字节为单位）
<code>status</code>	<code>smallint</code>	表示设备类型、缺省值和镜像状态的位图（请参见表 1-9）
<code>cntrlrtype</code>	<code>smallint</code>	控制器类型： <ul style="list-style-type: none"> <li>0 = 数据库设备</li> <li>2 = 磁盘转储设备或流式磁带</li> <li>3–8 = 磁带转储设备</li> </ul>
<code>name</code>	<code>sysname</code>	转储设备、数据库设备或内存存储高速缓存的逻辑名称
<code>phyname</code>	<code>varchar(127)</code>	物理设备或内存存储高速缓存的名称
<code>mirrorname</code>	<code>varchar(127) null</code>	镜像设备的名称
<code>vdevno</code>	<code>int</code>	设备标识号
<code>crdate</code>	<code>datetime null</code>	设备的添加日期
<code>resizedate</code>	<code>datetime null</code>	针对此设备最近一次运行 <code>disk resize</code> 的日期
<code>status2</code>	<code>int</code>	此设备的附加状态位（请参见表 1-10）
<code>instanceid</code>	<code>tinyint</code>	实例的 ID（只能用于集群环境）
<code>uuid</code>	<code>varbinary(16)</code>	留作将来使用（只能用于集群环境）

`status` 列的位表示形式具有累积性，如下所示。例如，“3”表示同时是缺省磁盘的物理磁盘。

**表 1-9: status 列的位表示形式。**

十进制	十六进制	状态
1	0x01	缺省磁盘
2	0x02	物理磁盘
4	0x04	未使用— 逻辑磁盘
8	0x08	跳过标题
16	0x10	转储设备
32	0x20	串行写
64	0x40	设备已镜像
128	0x80	读取已镜像
256	0x100	仅辅助镜像端
512	0x200	已启用镜像
1024	0x400	主设备已镜像
2048	0x800	内部使用— 已禁用镜像
4096	0x1000	内部使用— 必须将主设备解除镜像
8192	0x2000	内部使用— 必须将辅助设备解除镜像
16384	0x4000	UNIX 文件设备使用 <code>dsync</code> 设置（直接写入物理介质）

表 1-10 显示了 `status2` 列的位表示形式。

**表 1-10: status2 列的位表示形式。**

十进制	十六进制	状态
1	0x01	对此设备启用直接 I/O

索引

- `name` 上的唯一聚簇索引

## sysencryptkeys

### 所有数据库

说明 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 `sysencryptkeys` 的列为：

字段	类型	说明
<code>id</code>	<code>int</code>	加密密钥 ID。
<code>ekalgorithm</code>	<code>int</code>	加密算法。
<code>type</code>	<code>smallint</code>	标识密钥类型。其值为： <ul style="list-style-type: none"> <li>• <code>0x1</code>（十进制 1）— 对称密钥</li> <li>• <code>0x4</code>（十进制 4）— 缺省密钥</li> <li>• <code>0x10</code>（十进制 16）— 密钥副本</li> <li>• <code>0x40</code>（十进制 64）— 恢复密钥副本</li> </ul>
<code>status</code>	<code>int</code>	内部状态信息。位表示形式为： <ul style="list-style-type: none"> <li>• <code>0x1</code>（十进制 1）— 密钥使用初始化矢量</li> <li>• <code>0x2</code>（十进制 2）— 密钥使用随机填充</li> <li>• <code>0x4</code>（十进制 4）— 密钥被加密以用于保护丢失的口令</li> <li>• <code>0x8</code>（十进制 8）— 用于登录访问的加密密钥副本</li> <li>• <code>0x10</code>（十进制 16）— 使用登录口令加密的密钥副本</li> <li>• <code>0x20</code>（十进制 32）— 使用系统加密口令加密的密钥副本</li> <li>• <code>0x100</code>（十进制 256）— 使用用户口令加密的密钥</li> </ul>
<code>eklen</code>	<code>smallint</code>	用户指定的密钥长度。
<code>value</code>	<code>varbinary(1282)</code>	加密的密钥值。包含对称密钥加密。为加密密钥，Adaptive Server 将 AES 和来自系统加密口令、用户指定口令或登录口令的 128 位密钥一起使用。
<code>uid</code>	<code>int null</code>	密钥副本被授予人的用户 ID。
<code>eksalt</code>	<code>varbinary(20)</code>	用于验证加密密钥解密的随机值。
<code>ekpairid</code>	<code>int null</code>	尚未使用。
<code>pwdate</code>	<code>datetime null</code>	上次更改口令的日期。
<code>expdate</code>	<code>int null</code>	尚未使用。
<code>ekpwdwarn</code>	<code>int null</code>	尚未使用。

`sysencryptkeys` 的状态位。

**表 1-11: `sysencryptkeys` 状态位**

十进制	十六进制	状态
	<code>0x00000004</code>	<code>EK_KEYRECOVERY()</code> — 为保护丢失的口令而加密的密钥
	<code>0x00000008</code>	<code>EK_LOGINACCESS()</code> — 为登录访问而加密的密钥
	<code>0x00000010</code>	<code>EK_LOGINPASS ()</code> — 用登录口令加密的密钥
	<code>0x00000100</code>	<code>EK_USERPWD()</code> — 用用户加密口令加密的密钥

## sysengines

### 仅限 master 数据库

说明 当前处于联机状态的每个 Adaptive Server 引擎在 sysengines 中都有相应的一行。

列 sysengines 的列为：

名称	数据类型	说明
engine	smallint	引擎号
osprocid	int	<ul style="list-style-type: none"> <li>• <i>UNIX 平台</i> — 操作系统进程 ID（可以为 NULL）</li> <li>• <i>Windows</i> — 线程句柄。</li> </ul>
osprocname	char(32)	操作系统进程名称（可以为 NULL）
status	char(12)	以下状态之一：online、in offline、in create、in destroy、debug、bad status
affinitied	int	与该引擎相关的 Adaptive Server 进程数
cur_kpid	int	当前在该引擎上运行的进程（如果有）的内核进程 ID
last_kpid	int	以前在该引擎上运行的进程的内核进程 ID
idle_1	tinyint	保留
idle_2	tinyint	保留
idle_3	tinyint	保留
idle_4	tinyint	保留
starttime	datetime	引擎开始联机的日期和时间
nodeid	tinyint null	留作将来使用（不能用于集群环境）
instanceid	tinyint	实例的 ID（只能用于集群环境）

## sysgams

### 所有数据库

#### 说明

**sysgams** 存储数据库的全局分配映射 (GAM)。GAM 存储数据库所有分配单元的位图，每一位对应一个分配单元。不能从 **sysgams** 中进行选择或者查看。

# sysindexes

## 所有数据库

**说明** 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` 中都有相应的一行。每个基于函数的索引或在计算列上创建的索引在该表中也都有相应的一行。

**列** `sysindexes` 的列为：

名称	数据类型	说明
<code>name</code>	<code>varchar(255)</code> <code>null</code>	索引或表名。
<code>id</code>	<code>int</code>	索引的 ID 或索引所属的表的 ID。
<code>indid</code>	<code>smallint</code>	有效值包括： <ul style="list-style-type: none"> <li>• 0 = 如果是表。</li> <li>• 1 = 如果是所有页锁定表上的聚簇索引。</li> <li>• &gt;1 = 如果是 DOL 锁定表上的非聚簇索引或聚簇索引。</li> <li>• 255 = 如果是 <code>text</code>、<code>image</code>、文本链或 Java 行外结构（大对象，即 LOB 结构）。</li> </ul>
<code>doampg</code>	<code>int</code>	旧函数
<code>ioampg</code>	<code>int</code>	旧函数
<code>oampgtrips</code>	<code>int</code>	在刷新之前，OAM 页在高速缓存中循环而未被重新使用的次数
<code>status3</code>	<code>smallint</code>	内部系统状态信息。
<code>status2</code>	<code>smallint</code>	内部系统状态信息（请参见表 1-13）
<code>ipgtrips</code>	<code>int</code>	在刷新之前，索引页在高速缓存中循环而未被重新使用的次数
<code>first</code>	<code>int</code>	旧函数
<code>root</code>	<code>int</code>	旧函数
<code>distribution</code>	<code>int</code>	未使用。以前用于存储索引的分布页的页码。
<code>usagecnt</code>	<code>smallint</code>	保留
<code>segment</code>	<code>smallint</code>	对象所在的段号
<code>status</code>	<code>smallint</code>	内部系统状态信息（请参见表 1-12）
<code>maxrowsperpage</code>	<code>smallint</code>	每页的最大行数
<code>minlen</code>	<code>smallint</code>	最小行大小
<code>maxlen</code>	<code>smallint</code>	最大行大小
<code>maxirow</code>	<code>smallint</code>	非叶索引行的最大大小
<code>keycnt</code>	<code>smallint</code>	对于所有页锁定表中的聚簇索引，则为键的个数；对所有其它索引，则为键的个数加 1
<code>keys1</code>	<code>varbinary(255)</code> <code>null</code>	如果条目是索引，则为键列的说明



名称	数据类型	说明
keys2	varbinary(255) null	如果条目是索引，则为键列的说明
soid	tinyint	创建索引时使用的排序顺序 ID；如果键中没有字符数据，则为 0
csid	tinyint	创建索引时使用的字符集 ID；如果键中没有字符数据，则为 0
base_partition	int null	旧函数
fill_factor	smallint null	使用 <code>sp_chgattribute</code> 设置的表的填充因子值
res_page_gap	smallint null	表中 <code>reservepagegap</code> 的值
exp_rowsize	smallint null	期望的数据行大小
keys3	varbinary(255) null	如果条目是索引，则为键列的说明
identitygap	int null	表的标识间隔
crdate	datetime null	创建日期
partitiontype	smallint null	其值包括： <ul style="list-style-type: none"> <li>• 1 = 范围</li> <li>• 2 = 散列</li> <li>• 3 或 NULL = [ 缺省值 ] 循环</li> <li>• 4 = 列表</li> </ul>
conditionid	int null	分区条件的 ID。如果 <code>partitiontype</code> 为循环或散列，则为 Null

表 1-12 列出了 `status` 列的位表示形式。

**表 1-12: `sysindexes` 表 `status` 列中的 `status` 位**

十进制	十六进制	状态
1	0x1	如果试图插入重复键，则将中止当前命令或触发器。
2	0x2	唯一索引。
4	0x4	如果试图插入重复行，则将中止当前命令或触发器；对于 DOL 锁定表，则始终为 0。
16	0x10	表是具有聚簇索引的所有页锁定表。
64	0x40	如果是所有页锁定表，则索引允许重复行；如果是 DOL 锁定表，则始终为 0。
128	0x80	内部使用的已排序对象切换。可以用 <code>create clustered index</code> 、 <code>reorg rebuild</code> 或 <code>alter table</code> 锁定方案命令来设置。
512	0x200	<code>create index</code> 语句中使用的 <code>sorted data</code> 选项。
2048	0x800	对主键的索引。
32768	0x8000	可疑索引；按照另一种排序顺序创建了索引。

表 1-13 列出了 status2 列的位表示形式。

**表 1-13: sysindexes 表的 status2 列中的 status 位**

十进制	十六进制	状态
1	0x1	索引支持外键约束
2	0x2	索引支持主键 / 唯一声明约束
4	0x4	索引包括 IDENTITY 列
8	0x8	未指定约束名
16	0x10	没有对表、索引或文本链启用大 I/O（预取）
32	0x20	没有对表、索引或文本链启用“最近使用最多” (MRU) 的高速缓存策略
64	0x40	为表打开升序插入
256	0x0100	对索引进行预排序，并且不需要将索引复制到新扩充中
512	0x0200	表是具有聚簇索引的 DOL 锁定表
8192	0x2000	DOL 锁定表中的索引是可疑的
32768	0x8000	索引是基于函数的

索引

- id、indid 上的唯一聚簇索引

## sysinstances

说明

报告实例状态的虚设表。集群配置中定义每个实例在 `sysinstances` 中都有相应的一行。`sysinstances` 中包含特定于该 Cluster Edition 的信息。

虽然 `sysinstances` 是虚设表，但是它不受 `set system_view` 设置的影响，并且无论 `system_view` 设置如何，它始终为每个实例返回一行。

列

`sysinstances` 的列为：

列名	数据类型	说明
<code>id</code>	<code>tiny int</code>	实例的 ID
<code>name</code>	<code>varchar(30)</code>	实例名
<code>state</code>	<code>char(17)</code>	实例的状态 ( <code>online</code> 、 <code>offline</code> 、 <code>joining</code> 、 <code>leaving</code> 和 <code>initiating</code> 中的一个)
<code>hostname</code>	<code>varchar(255)</code>	运行该实例的操作系统主机的名称
<code>starttime</code>	<code>datetime</code>	启动实例的日期和时间
<code>connections_active</code>	<code>int</code>	实例上的活动连接数
<code>engines_online</code>	<code>smallint</code>	该实例的联机引擎数

索引

无

## sysjars

### 所有数据库

#### 说明

数据库中保留的每个 Java 存档 (JAR) 文件在 `sysjars` 中都有相应的一行。有关 JAR 文件、Java 类和 Java 数据类型的详细信息，请参见《Adaptive Server Enterprise 中的 Java》。

#### 列

`sysjars` 的列为：

名称	数据类型	说明
<code>jid</code>	<code>int</code>	JAR 的 ID。
<code>jstatus</code>	<code>int</code>	内部状态信息。未使用。
<code>jname</code>	<code>varchar(255) null</code>	JAR 名。
<code>jbinary</code>	<code>image null</code>	JAR 的内容：Java 类。

#### 索引

- `jid` 上的唯一聚簇索引
- `jname` 上的唯一非聚簇索引

## syskeys

### 所有数据库

说明

每个主键、外键或公用键在 `syskeys` 中都有相应的一行。

列

`syskeys` 的列为：

名称	数据类型	说明
<code>id</code>	<code>int</code>	对象 ID
<code>type</code>	<code>smallint</code>	记录类型。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 主键</li> <li>• 2 = 外键</li> <li>• 3 = 公用键</li> </ul>
<code>depid</code>	<code>int null</code>	相关对象的 ID
<code>keycnt</code>	<code>int null</code>	非空键的数量
<code>size</code>	<code>int null</code>	保留
<code>key1 ... key8</code>	<code>smallint null</code>	列 ID
<code>depkey1 ... depkey8</code>	<code>smallint null</code>	列 ID
<code>spare1</code>	<code>smallint</code>	保留

索引

- `id` 上的聚簇索引

## syslanguages

### 仅限 master 数据库

**说明** Adaptive Server 可识别的每种语言在 `syslanguages` 中都有相应的一行。美式英语 (`us_english`) 虽不在 `syslanguages` 中，但 Adaptive Server 总是可以使用该语言。

**列** `syslanguages` 的列为：

名称	数据类型	说明
<code>langid</code>	<code>smallint</code>	唯一的语言 ID
<code>dateformat</code>	<code>char(3)</code>	日期顺序；如 “ <code>dmy</code> ”
<code>datefirst</code>	<code>tinyint</code>	每周的第一天 — 1 表示星期一，2 表示星期二，依此类推，7 表示星期日
<code>upgrade</code>	<code>int</code>	该语言的 Adaptive Server 最新升级版本
<code>name</code>	<code>varchar(30)</code>	正式的语言名称，如 “ <code>french</code> ”
<code>alias</code>	<code>varchar(30) null</code>	替代的语言名称，如 “ <code>français</code> ”
<code>months</code>	<code>varchar(251)</code>	用逗号分隔的月份全名列表，其顺序是从一月到十二月，每个名称的最大长度为 20 个字符
<code>shortmonths</code>	<code>varchar(119)</code>	用逗号分隔的月份名简称列表，其顺序是从一月到十二月，每个名称的最大长度为 9 个字符
<code>days</code>	<code>varchar(216)</code>	用逗号分隔的星期列表，其顺序是从星期一到星期日，每个名称的最大长度为 30 个字符

- 索引**
- `langid` 上的唯一聚簇索引
  - `name` 上的唯一非聚簇索引
  - `alias` 上的唯一非聚簇索引

## syslisteners

### 仅限 master 数据库

**说明** 每种可用来与当前 Adaptive Server 连接的网络协议在 `syslisteners` 中都有相应的一行。当用户或者客户端应用程序查询 `syslisteners` 时，Adaptive Server 将动态创建该表。

**列** `syslisteners` 的列为：

名称	数据类型	说明
<code>net_type</code>	<code>char(32)</code>	网络协议
<code>address_info</code>	<code>char(255)</code>	在网络上唯一地标识此 Adaptive Server 的信息；通常是当前 Adaptive Server 的名称和一个标识号（如协议的服务器端口号）
<code>spare</code>	<code>tinyint</code>	未使用
<code>nodeid</code>	<code>tinyint null</code>	留作将来使用（不能用于集群环境）
<code>instanceid</code>	<code>tinyint</code>	实例的 ID（只能用于集群环境）

## syslocks

### 仅限 master 数据库

**说明** syslocks 包含关于活动锁的信息，它是在用户进行查询时动态创建的。不允许更新 syslocks。

**列** syslocks 的列为：

名称	数据类型	说明
id	int	表 ID。
dbid	smallint	数据库 ID。
page	int	页码。
type	smallint	锁类型（type 列的位值在表 1-14 中列出）。
spid	smallint	持有锁的进程的 ID。
	对于 Cluster Edition, 为 int	
class	varchar(30)	与该锁关联的游标的名（如果有）。
fid	smallint（对于 Cluster Edition, 为 int）	锁所属的进程系列（协调进程及其工作进程）。fid 值在表 1-15 中列出。
context	tinyint	锁请求的环境类型。context 值在表 1-16 中列出。
row	smallint	行号。
loid	int	唯一的锁所有者 ID。
partitionid	int null	分区 ID。
nodeid	tinyint null	留作将来使用（不能用于集群环境）
instanceid	tinyint	实例的 ID（只能用于集群环境）

**注释** 由于 Cluster Edition 的数据类型的这一更改，Sybase 强烈建议您在升级之前存档并截断审计表。这会降低由于 sybsecurity 数据库中空间不足而导致升级失败的可能性。

表 1-14 列出了 type 列的位表示形式。



**表 1-14: syslocks 表中的 type 控制位**

十进制	十六进制	状态
1	0x1	排它表锁
2	0x2	共享表锁
3	0x3	排它意向锁
4	0x4	共享意图锁
5	0x5	排它页锁
6	0x6	共享页锁
7	0x7	更新页锁
8	0x8	排它行锁
9	0x9	共享行锁
10	0xA	更新行锁
11	0xB	共享下一键锁
256	0x100	锁阻塞另一个进程
512	0x200	请求锁

表 1-15 列出了 fid 列的值:

**表 1-15: syslocks 表中的 fid 列值**

值	解释
0	spid 代表的任务是串行执行语句的单个任务。
非 0	持有锁的任务 (spid) 是并行执行语句的任务系列中的一个任务。 如果值等于 spid, 则表明该任务是并行执行查询的进程系列中的协调进程。

表 1-16 列出了 context 列的值:

**表 1-16: syslocks 表中的 context 列值**

值	解释
null	持有该锁的任务在执行串行查询, 或者在事务隔离级别 1 上执行并行查询。
0x1	在完成查询之前, 持有锁的任务将一直持有锁。当出现以下情况时, 锁的环境可能是 FAM_DUR (0x1H): <ul style="list-style-type: none"> <li>锁是作为并行查询的一部分持有的表锁。</li> <li>锁是工作进程在隔离级别 3 上持有的。</li> <li>锁是并行查询中的工作进程持有的, 并且必须在事务的持续时间内持有。</li> </ul>
0x2	由可序列化的读取任务持有的域锁。
0x4	无限键锁。
0x8	在所有页锁定表的索引页上获取的锁。
0x10	在页或行上获取以删除行的锁。
0x20	执行收缩或拆分操作期间在索引页上获取的地址锁。
0x40	执行可重复的读取操作的事务持有的意图锁。仅对 DOL 表中的共享意图锁和排它意图锁有效。

## sysloginroles

### 仅限 master 数据库

#### 说明

拥有系统角色的服务器登录名的每个实例在 `sysloginroles` 中都有相应的一行。每当给一个角色授予一个登录名时，就会添加一行。例如，如果对单个服务器用户授予 `sa_role`、`sso_role` 和 `oper_role`，则会在与该用户的系统用户 ID (suid) 关联的 `sysloginroles` 中添加三行。

#### 列

`sysloginroles` 的列为：

名称	数据类型	说明
suid	int	服务器用户 ID
srld	int	服务器角色 ID；可以为以下值之一： <ul style="list-style-type: none"> <li>• 0 = sa_role</li> <li>• 1 = sso_role</li> <li>• 2 = oper_role</li> <li>• 4 = navigator_role</li> <li>• 5 = replication_role</li> <li>• 6 = 当前未使用</li> <li>• 7 = dtm_tm_role</li> <li>• 8 = ha_role</li> <li>• 8 = ha_role</li> <li>• 9 = 内部使用</li> <li>• 10 = mon_role</li> <li>• 11 = js_admin_role</li> <li>• 12 = messaging_role</li> <li>• 13 = js_client_role</li> <li>• 14 = js_user_role</li> <li>• 15 = webservices_role</li> </ul>
status	smallint	状态位，用于指示在登录时是否将各个服务器角色设置为其缺省值： <ul style="list-style-type: none"> <li>• 0 = 缺省情况下为关闭</li> <li>• 1 = 缺省情况下为打开</li> </ul>

**更改状态位** 当您使用 `sp_modifylogin` 更改状态位时，您必须先注销然后重新登录才能使该更改生效。要查看即时结果，请使用 `set role role_name off`。

#### 索引

- suid 上的聚簇索引

# syslogins

## 仅限 master 数据库

说明 每个有效的 Adaptive Server 用户帐户在 `syslogins` 中都有相应的一行。

列 `syslogins` 的列为：

名称	数据类型	说明
<code>suid</code>	<code>int</code>	服务器用户 ID。
<code>status</code>	<code>smallint</code>	帐户状态（请参见表 1-17）。
<code>accdate</code>	<code>datetime</code>	上次清除 <code>totcpu</code> 和 <code>totio</code> 的日期。
<code>totcpu</code>	<code>int</code>	登录所积累的 CPU 时间。
<code>totio</code>	<code>int</code>	登录所积累的 I/O。
<code>spacelimit</code>	<code>int</code>	保留。
<code>timelimit</code>	<code>int</code>	保留。
<code>resultlimit</code>	<code>int</code>	保留。
<code>dbname</code>	<code>sysname null</code>	建立连接时放置用户的数据库的名称。
<code>name</code>	<code>sysname</code>	用户的登录名。
<code>password</code>	<code>varbinary(128) null</code>	用户口令的单向散列。  <b>注释</b> <code>syslogins.password</code> 的内容取决于 <code>sp_passwordpolicy allow password downgrade</code> 的值。
<code>language</code>	<code>varchar(30) null</code>	用户的缺省语言。
<code>pwdate</code>	<code>datetime null</code>	上次更改口令的日期。
<code>audflags</code>	<code>int null</code>	用户的审计设置。
<code>fullname</code>	<code>varchar(30) null</code>	用户的全名。
<code>srvname</code>	<code>varchar(30) null</code>	服务器的名称，如果打开了 <code>AUTOCONNECT</code> 标志，则必须建立与该服务器的直通连接。
<code>logincount</code>	<code>smallint null</code>	失败的登录尝试次数；登录成功后将其重置为 0。
<code>procid</code>	<code>int null</code>	存储用 <code>sp_modifylogin</code> 中的 <code>login script</code> 选项注册的登录触发器
<code>lastlogindate</code>	<code>datetime</code>	用户上次登录的时间戳。
<code>crdate</code>	<code>datetime</code>	创建登录名时的时间戳。
<code>locksuid</code>	<code>int</code>	负责锁定登录的服务器用户 ID ( <code>suid</code> )。

名称	数据类型	说明
lockreason	int	锁定原因；下列原因之一： <ul style="list-style-type: none"> <li>• NULL — 帐户尚未被锁定</li> <li>• 0 — locksuid 通过执行 sp_locklogin 锁定了帐户</li> <li>• 1 — locksuid 通过执行 sp_locklogin 'all', 'lock', 'ndays' 锁定了不活动的帐户</li> <li>• 2 — 由于失败的登录尝试次数达到 max failed logins, Adaptive Server 锁定了帐户。</li> <li>• 3 — 由于口令降级阶段已结束，并且登录或角色未过渡到 SHA-256, locksuid 锁定了帐户</li> </ul>
lockdate	datetime	锁定登录名时的时间戳。

在 Adaptive Server 分发介质上，syslogins 包含一个条目，其名称为“sa”，suid 为 1，口令为空。它还包含带有未发布口令的条目“probe”。登录名“probe”和用户“probe”用于两阶段提交探查进程，该进程使用询问和响应机制来访问 Adaptive Server。

表 1-17 列出了 status 列的位表示形式：

**表 1-17: syslogins 表中的 status 控制位**

十进制	十六进制	状态
2	0x2	帐户已锁定。
4	0x4	口令已到期。
8	0x8	用户拥有 RepSrv 授权。
16	0x10	OMNI: 已启用 autoconnect 模式。
32	0x20	可以使用 Adaptive Server 内部鉴定机制 — syslogins。
64	0x40	可以使用 LDAP 外部鉴定。
128	0x80	可以使用 PAM 外部鉴定。
256	0x100	可以使用 Kerberos 外部鉴定

## 索引

- suid 上的唯一聚簇索引
- name 上的唯一非聚簇索引

## syslogs

### 所有数据库

#### 说明

`syslogs` 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。

不能对 `syslogs` 执行删除、插入或更新操作。将记录每个数据修改操作，因此，在更改 `syslogs` 之前，必须先记录更改。这意味着，在对 `syslogs` 执行更改操作时，将在 `syslogs` 中添加一行，随后必须将这一情况记录下来，因此又会在 `syslogs` 中添加一行，这样下去，将会造成无限循环。该循环将持续到填满数据库时为止。

#### 列

`syslogs` 的列为：

名称	数据类型	说明
<code>xactid</code>	<code>binary(6)</code>	事务 ID
<code>op</code>	<code>tinyint</code>	更新操作的次数

## syslogshold

### 仅限 master 数据库

**说明** syslogshold 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。不允许更新 syslogshold。

**列** syslogshold 的列为：

名称	数据类型	说明
dbid	smallint	数据库 ID。
reserved	int	未使用。
spid	smallint	拥有最早活动事务的用户的服务器进程 ID（对于 Replication Server，则始终为 0）。
		对于集群环境，则为 int
page	int	syslogs 中由最早事务定义的活动部分的起始页码（对于 Replication Server，则为 syslogs 中的截断页）。
xactid	binary(6)	最早活动事务的 ID（对于 Replication Server，则始终为 0x000000）。
masterxactid	binary(6)	对于多数据库事务，为事务的主事务（如果有）ID；对于其它事务，则为 0x000000（对于 Replication Server，始终为 0x000000）。
starttime	datetime	事务开始的日期和时间（或为 Replication Server 设置截断点的时间）。
name	char(67)	最早活动事务的名称。它是使用 begin transaction 定义的名称。如果没有使用 begin transaction 指定任何值，则为 “\$user_transaction”；对于由 ANSI 链式模式启动的隐式事务，它为 “\$chained_transaction”。如果 Adaptive Server 启动的内部事务的名称以美元符号 (\$) 开头并用操作来命名；对于 Replication Server，则命名为 “\$replication_truncation_point”。
xloid	int null	锁的所有权 ID。如果所有者是任务，则根据 spid 确定锁的所有权 ID；如果所有者是事务，则根据 xdes 确定锁的所有权 ID。

**注释** 由于 Cluster Edition 的数据类型的这一更改，Sybase 强烈建议您在升级之前存档并截断审计表。这会降低由于 sybsecurity 数据库中空间不足而导致升级失败的可能性。

## sysmessages

### 仅限 master 数据库

**说明** Adaptive Server 可返回的每个系统错误或者警告在 `sysmessages` 中都有相应的一行。Adaptive Server 将在用户屏幕上显示错误说明。

**列** `sysmessages` 的列为：

名称	数据类型	说明
<code>error</code>	<code>int</code>	唯一的错误号
<code>severity</code>	<code>smallint</code>	错误的严重级
<code>dlevel</code>	<code>smallint</code>	保留
<code>description</code>	<code>varchar(1024)</code>	带有参数占位符的错误解释
<code>langid</code>	<code>smallint null</code>	语言；对于 <code>us_english</code> ，则为 <code>null</code>
<code>sqlstate</code>	<code>varchar(5) null</code>	错误的 SQLSTATE 值

**索引**

- `error`、`dlevel` 上的聚簇索引
- `error`、`dlevel`、`langid` 上的非聚簇索引

## sysmonitors

### 仅限 master 数据库

说明

每个监控计数器在 `sysmonitors` 中都有相应的一行。

列

`sysmonitors` 的列为：

名称	数据类型	说明
<code>field_name</code>	<code>char(79)</code>	计数器的名称
<code>group_name</code>	<code>char(25)</code>	该计数器所属的组
<code>field_id</code>	<code>smallint</code>	行的唯一标识符
<code>value</code>	<code>int</code>	计数器的当前值
<code>description</code>	<code>varchar(255) null</code>	计数器的说明；未使用
<code>nodeid</code>	<code>tinyint null</code>	留作将来使用（不能用于集群环境）
<code>instanceid</code>	<code>tinyint</code>	实例的 ID（只能用于集群环境）



## sysobjects

### 所有数据库

#### 说明

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** 中都有相应的一行。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。

对于类型为 **EK**（加解密钥）的每个密钥，**sysobjects** 都有一个相应的条目。

对于跨数据库密钥引用，**syscolumns.enchrdate** 与 **sysobjects.cdate** 匹配。

**sysencryptkeys** 中的 **enckeyid** 与 **sysobjects** 中的 **id** 列匹配。

#### 列

**sysobjects** 的列为：

名称	数据类型	说明
name	varchar(255) not null	对象名。
id	int	对象 ID。
uid	int	对象所有者的用户 ID。
type	char(2)	可以为以下对象类型之一： <ul style="list-style-type: none"> <li>• C — 计算列</li> <li>• D — 缺省值</li> <li>• DD — 解密缺省值</li> <li>• F — SQLJ 函数</li> <li>• L — 日志</li> <li>• N — 分区条件</li> <li>• P — Transact-SQL 或 SQLJ 过程</li> <li>• PR — 准备对象（由动态 SQL 创建）</li> <li>• R — 规则</li> <li>• RI — 参照约束</li> <li>• S — 系统表</li> <li>• TR — 触发器</li> <li>• U — 用户表</li> <li>• V — 视图</li> <li>• XP — 扩展存储过程。</li> </ul>
userstat	smallint	与应用程序相关的类型信息（十进制 32768 [十六进制 0x8000] 向 Data Workbench <sup>®</sup> 表明过程是一个报告）。
sysstat	smallint	内部状态信息（十进制 256 [十六进制 0x100] 表示表是只读的）
indexdel	smallint	重新对对象模式中的更改进行计数并更新 <b>schemacnt</b> 。
schemacnt	smallint	对象模式中的更改计数（当添加规则或缺省值时增加）

名称	数据类型	说明
sysstat2	int	其它内部状态信息（请参见表 1-19）
sysstat3	unsigned smallint	其它内部状态信息（请参见表 1-20）
crdate	datetime	创建对象时的日期
expdate	datetime	保留
deltrig	int	当条目为表时，为删除触发器的存储过程 ID。当条目是触发器时，为表 ID。
instrig	int	当条目为表时，是表的插入触发器的存储过程 ID
updtrig	int	当条目为表时，是表的更新触发器的存储过程 ID
seltrig	int	保留
ckfirst	int	表上第一个检查约束的 ID
cache	smallint	保留
audflags	int null	对象的审计设置
objspare	smallint	备用
versions	binary(6) null	此对象上一次模式更改的版本时间戳（由 Replication Server 使用）
loginame	varchar(30) null	创建对象的用户的登录名
identburnmax	numeric(17) null	此对象中标识列（如果有）的最大烧毁值
<p><b>注释</b> identburnmax 列是以内部格式存储的。如果您需要值，请使用 identity_burn_max() 函数。</p>		
spacestate	smallint null	仅留作内部使用
erlchgts	binary(8) null	仅留作内部使用

表 1-18 列出了 sysstat 列的位表示形式：

**表 1-18: sysobjects 表中的 sysstat 控制位**

十进制	十六进制	对象类型	说明
0	0x0	O_ANY	任何非法对象
1	0x1	O_SYSTEM	系统对象
2	0x2	O_VIEW	视图
3	0x3	O_USER	用户对象
4	0x4	O_PROC	存储过程
5	0x5	O_LOG	日志
6	0x6	O_DEFAULT	指定的缺省值
7	0x7	O_DOMAIN	域规则
8	0x8	O_TRIGGER	触发器过程
9	0x9	O_REFERENCE	参照完整性约束
10	A	O_CHECK	检查约束
11	B	O_XTYPE	扩展类型

十进制	十六进制	对象类型	说明
12	C	O_FUNC	存储函数
		O_TYPE_MAX	O_FUNC 在添加一个新的对象类型时更新对象类型的最大值。可在打印例程中使用 O_TYPE_MAX 打印此字段的 #define 值字符串。 参考：useful/statbits.c 和打印例程中宏 PRTYPESTR 的使用。
16	0x10	O_CLUST	具有簇索引
32	0x20	O_NONCLUST	具有非簇索引
		OBJ_FOR_SYSDEPENDS(obj_type)	检查在创建或删除以下对象类型时对象是否需要 sysdepends 中的条目： <ul style="list-style-type: none"> <li>• O_PROC</li> <li>• O_TRIGGER</li> <li>• O_VIEW</li> <li>• O_DEFAULT</li> <li>• O_DOMAIN</li> <li>• O_FUNC</li> </ul>
64	0x40	O_LOGGED	对象已记录。 后续位被重载，并且对于表和对于存储过程，它会具有不同的含义。此信息在打印例程 prOBJSTAT_OBJSYSSTAT() 中被解码。
64	0x40	O_PROC_SUBSCRIBABLE	该存储过程是可预定的 Replication Server 支持重载 O_LOGGED 位。 如果对象是存储过程，则 O_PROC_SUBSCRIBABLE 将用于表示该存储过程是否可预定。
128	0x80	O_IN_CREATE	正在创建对象
256	0x100	O_READONLY	对象包含可疑索引，并且在您运行 dbcc reindex 之前只能用于只读用途。
512	0x200	O_SUSPECT	对象被恢复过程标记为可能损坏；请运行 dbcc。通过打开表进行检查。
1024	0x400	O_FAKE	对象是“虚设的”；即，它驻留在 tempdb 中，并且会针对使用它的每个查询步骤进行重新定义。
2048	0x800	O_EXTTABLE	对象是外部表，如 Stratus VOS 文件
4096	0x1000	O_RAMBOIX	标识要重建其索引的系统表。
8192	0x2000	O_TEXTIMAGE	对象包含 text/image 字段
16384	0x4000	O_TABNOLOG	未使用
32768	0x8000	O_REPLICATED	已复制或过程

表 1-19 列出了 sysstat2 列的位表示形式：

**表 1-19: sysobjects 表中的 sysstat2 控制位**

十进制	十六进制	状态
1	0x1	表具有参照约束。
2	0x2	表具有外键约束。
4	0x4	表具有多个检查约束。
8	0x8	表具有主键约束。
16	0x10	存储过程只能在链式事务模式下执行。
32	0x20	存储过程可以在任何事务模式下执行。
64	0x40	表具有 IDENTITY 字段。
128	0x80	尚未使用。
256	0x100	尚未使用。
512	0x200	表中不包含可变长度的列。
1024	0x400	表是远程的。
2048	0x800	表是用 existing 关键字创建的代理表。
4096	0x1000	尚未使用。
8192	0x2000	表使用所有页锁定方案。
16384	0x4000	表使用数据页锁定方案。
32768	0x8000	表使用数据行锁定方案。
65536	0x10000	表是在 11.9 或更高版本的服务器中创建的。
131072	0x20000	表具有聚簇索引。
262144	0x40000	对象表示一个嵌入式 SQL 过程。
524288	0x80000	尚未使用。
16777216	0x1000000	对象表示一条访问规则。
33554432	0x2000000	对象表示 SQLJ 存储过程。
67108864	0x4000000	对象表示 OR 访问规则。
1073741824	0x40000000	表包含一个或多个基于函数的索引。
2147483648	0x80000000	对象具有扩展索引。

表 1-20 列出了 sysstat3 列的位表示形式：

**表 1-20: sysobjects 表中的 sysstat3 控制位**

十进制	十六进制	状态
32768	0x8000	表参与增量传输

索引

- id 上的唯一聚簇索引
- name、uid 上的非聚簇索引

## sysoptions

### 所有数据库

**说明** sysoptions 是一个由 sp\_options 查询的新虚设表。查询 sysoptions 时，行的名称是区分大小写的。

**列**

名称	数据类型	属性	说明
spid	int		包含进程 ID。
name	varchar(100)		包含选项的名称。
category	varchar(100)		包含选项所属类别的名称。
currentsetting	varchar(100)	NULL	包含选项的当前设置。
defaultsetting	varchar(100)	NULL	包含选项的缺省设置。
scope	int		包含用于捕获有关选项的信息的位图。位的排序方式如下： <ul style="list-style-type: none"> <li>• 第 1 位 — 编译时选项</li> <li>• 第 2 位 — 存储过程特定的选项</li> <li>• 第 3 位 — 二进制选项</li> </ul>

使用 sp\_options 查询 sysoptions。当前和缺省值的数据类型是 varchar，因此，可以直接使用值为 varchar 的设置。可以在类型强制转化之后使用值为 integer 的设置。

您无需特殊权限即可查询 sysoptions。例如：

```
select * from sysoptions
where spid = 13
go
```

您还可以使用字符串处理或类型强制转化。例如，如果选项是数值，则可以通过输入下列内容来查询 sysoptions：

```
if (isnumeric(currentsetting))
    select@int_val = convert(int, currentsetting)
    ...
else
    select@char_val = currentsetting
    ...
```

## syspartitionkeys

### 所有数据库

#### 说明

表的散列、范围和列表分区的每个分区键在 `syspartitionkeys` 中都有相应的一行。所有列都为非空。

#### 列

`syspartitionkeys` 的列为：

名称	数据类型	说明
<code>indid</code>	<code>smallint</code>	索引类型。其值包括： <ul style="list-style-type: none"><li>• 0 = 表</li><li>• 1 = 聚簇索引</li><li>• &gt;1 = 非聚簇索引</li></ul>
<code>id</code>	<code>int</code>	分区表的对象 ID
<code>colid</code>	<code>smallint</code>	分区表的分区键的列 ID
<code>position</code>	<code>smallint</code>	键的位置

#### 索引

- `id`、`indid`、`colid` 上的唯一聚簇索引

## syspartitions

### 所有数据库

#### 说明

每个数据分区和每个索引分区在 `syspartitions` 中都有相应的一行。

对于每个数据库，以下每一项在 `syspartitions` 中都有相应的一行：

- 每个表分区。indid 为 0。
- 每个聚簇索引分区。indid 为 1。
- 每个非聚簇索引分区。indid 为 >1。
- 每个单分区（未分区）表。
- 每个单分区（未分区）聚簇或非聚簇索引。

如果是本地索引，则 `partitionid`（数据分区行）的值和 `data_partitionid`（关联索引行）的值相同。如果不是本地索引，则 `data_partitionid`（索引行）的值为 0，不等于 `partitionid`（数据分区行）的值。

---

**注释** Adaptive Server 15.0 之前版本中的 `syspartitions` 表已经重新命名为 `syssslices`，`syspartitions` 表已停用。对于 Adaptive Server version 15.0，`syspartitions` 经过完全重新定义，现在支持数据分区和索引分区。

---

#### 列

`syspartitions` 的列为：

名称	数据类型	说明
name	varchar(255)	分区名称。
indid	smallint	在所有页锁定表上索引 ID。其值包括： <ul style="list-style-type: none"> <li>• 0 = 数据页（表）</li> <li>• 1 = 所有页锁定表上的聚簇索引</li> <li>• &gt;1 且 &lt;255 = DOL 锁定表上的非聚簇索引或聚簇索引</li> <li>• 255 = 文本链</li> </ul>
id	int	表 ID。
partitionid	int	数据分区或索引分区的 ID。
segment	smallint	分区所在段的 ID。
status	int	内部状态信息。
datoampage	int	数据分区的对象分配映射的页码。
indoampage	int	索引分区的对象分配映射的页码。
firstpage	int	第一个数据页（叶页）的页码。

---

名称	数据类型	说明
rootpage	int	以下页的页码： <ul style="list-style-type: none"><li>• 根页（如果条目是索引分区）</li><li>• 最后一页（如果条目是数据分区）</li></ul>
data_partitionid	int	此索引跨越的数据分区的 ID。其值包括： <ul style="list-style-type: none"><li>• 0 = 对于跨越整个表的全局索引</li><li>• 1 = 本地索引的分区跨越的数据分区的分区 ID。</li></ul>
crdate	datetime	分区的创建日期。
cdataptname	varchar(255) null	数据分区的名称。

---

- 索引
- id、indid、partitionid 上的唯一聚簇索引
  - id、indid、name 上的唯一非聚簇索引
  - partitionid、indid 上的唯一非聚簇索引



## sysprocedures

### 所有数据库

**说明** 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` 中都有相应的条目。每个对象（包括计算列或基于函数的索引定义）的序列树都以二进制形式存储。如果不能将序列树放入一个条目中，则将其分解为多个行。 `sequence` 列标识子行。

**列** `sysprocedures` 的列为：

名称	数据类型	说明
<code>type</code>	<code>smallint</code>	对象类型（请参见表 1-21）
<code>qp_setting</code>	<code>varbinary(6)</code> <code>null</code>	仅供将来使用
<code>id</code>	<code>int</code>	对象 ID
<code>sequence</code>	<code>int</code>	使用多行来说明该对象时所用的序列号
<code>status</code>	<code>smallint</code>	内部系统状态
<code>number</code>	<code>smallint</code>	对过程进行分组时的子过程数（0 表示非过程条目）
<code>version</code>	<code>int null</code>	Adaptive Server 的版本，在此版本中为给定对象创建了存储在此目录中的序列树

表 1-21 列出了 `type` 列的位表示形式。

**表 1-21: `sysprocedures` 表中的 `type` 控制位**

十进制	十六进制	状态
1	0x1	描述计划的条目（保留）
2	0x2	描述树的条目

**索引**

- `id`、`number`、`type`、`sequence` 上的唯一聚簇索引

## sysprocesses

### 仅限 master 数据库

**说明** sysprocesses 包含有关 Adaptive Server 进程的信息，但它不是常规表。它在用户查询时动态建立。不允许更新 sysprocesses。请使用 kill 语句来注销进程。

**列** sysprocesses 的列为：

名称	数据类型	说明
spid	smallint	进程 ID。 对于 Cluster Edition，为 int
kpid	int	内核进程 ID。
enginenum	int	正在执行进程的引擎的编号。
status	char(12)	进程 ID 状态（请参见表 1-22）。
suid	int	发出命令的用户的服务器用户 ID。
hostname	varchar(30) null	主机名。
program_name	varchar(30) null	前端模块的名称。
hostprocess	varchar(30) null	主机进程 ID 号。
clientport	unsigned smallint	<ul style="list-style-type: none"> <li>显示客户端进程的客户端端口号</li> <li>对于系统进程，则显示 0</li> </ul>
cmd	varchar(30) null	当前执行的命令或进程。条件语句（例如，if 或 while 循环）求值返回 cond。
cpu	int	进程的累积 CPU 时间（以时钟周期表示）
physical_io	int	当前命令的磁盘读写次数。
memusage	int	分配给进程的内存量。
blocked	smallint	阻塞进程（如果有）的进程 ID。 对于 Cluster Edition，为 int
dbid	smallint	数据库 ID。
uid	int	执行命令的用户的 ID。
gid	int	执行命令的用户的组 ID。
tran_name	varchar(64) null	活动事务的名称。
time_blocked	int null	阻塞时间（以秒表示）。
network_pktsz	int null	当前连接的网络包大小。
fid	smallint	工作进程的父进程的进程 ID。 对于 Cluster Edition，为 int
execlss	varchar(30) null	进程绑定的执行类。
priority	varchar(10) null	与进程关联的基本优先级。
affinity	varchar(30) null	与进程相关的引擎的名称。

名称	数据类型	说明
id	int null	当前运行的过程的对象 ID（如果没有运行任何过程，则为 0）。
stmtnum	int null	正在运行的过程中的当前语句号（如果没有运行任何过程，则为 SQL 批处理语句号）。
linenum	int null	正在运行的存储过程中当前语句的行号（如果没有运行任何过程，则为当前 SQL 批处理语句的行号）。
origsuid	int null	原始服务器用户 ID。如果该值不为 NULL，则 suid 为 origsuid 的用户可执行 <code>set proxy</code> 或 <code>set session authorization</code> 来模拟执行该命令的用户。
block_xloid	int null	阻塞事务的锁的唯一锁所有者 ID。
clientname	varchar(30) null	可选 — 当前会话借以识别用户的名称。  <b>注释</b> Adaptive Server 会自动在 <code>clientname</code> 、 <code>clienthostname</code> 和 <code>clientappliance</code> 列中存储一个或多个空格。因此，使用包含 “ <code>is null</code> ” 的这三列中任一列的查询不会返回预期结果集。
clienthostname	varchar(30) null	可选 — 当前会话借以识别主机的名称。
clientappliance	varchar(30) null	可选 — 当前会话借以识别应用程序的名称。
sys_id	smallint null	协同服务器节点的唯一标识。
ses_id	int null	每个客户端会话的唯一标识。
loggedindatetime	datetime null	它显示客户端连接到 Adaptive Server 的时间和日期。有关详细信息，请参见《系统管理指南》的第 11 章“管理用户权限”中的“行级访问控制”。
ipaddr	varchar(64) null	它是进行登录的客户端的 IP 地址。有关详细信息，请参见《系统管理指南》的第 11 章“管理用户权限”中的“行级访问控制”。
nodeid	tinyint null	留作将来使用（不能用于集群环境）
instanceid	tinyint	实例的 ID（只能用于集群环境）
pad	smallint	为了对齐而添加的列（只能用于集群环境）
lcid	int	集群的 ID

**注释** 由于 Cluster Edition 的数据类型的这一更改，Sybase 强烈建议您在升级之前存档并截断审计表。这会降低由于 `sybsecurity` 数据库中空间不足而导致升级失败的可能性。

表 1-22 列出了 status 列的值：

**表 1-22: sysprocesses status 列的值**

状态	含义
alarm sleep	等待警报唤醒进程（用户执行了 <code>waitfor delay</code> 命令）
background	由 Adaptive Server 而不是用户进程运行的进程，例如阈值进程
infected	服务器已检测到严重的错误情况；极其少见
latch sleep	等待获取门锁
lock sleep	等待获取锁
PLC sleep	等待访问用户日志高速缓存
recv sleep	等待网络读取
remote I/O	使用远程服务器执行 I/O
runnable	在可运行的进程队列中
running	当前在一个服务器引擎上运行
send sleep	等待网络发送
sleeping	等待磁盘 I/O 或某些其它资源（通常表示正在运行但执行大量磁盘 I/O 的进程）
stopped	已停止的进程
sync sleep	等待系列中另一进程的同步消息

# sysprotects

## 所有数据库

说明 sysprotects 包含有关已为用户、组和角色授予或撤消的权限的信息。

列 sysprotects 的列为：

名称	数据类型	说明	名称
id	int	该权限适用的对象的 ID。在向用户授予运行 <code>create table</code> 或 <code>create default</code> 的能力时，会为这些用户提供 id 0。	
uid	int	该权限适用的用户、组或角色的 ID。	
action	smallint	可以为以下权限之一： 151 = references 167 = set proxy 或 set session authorization 187 = set statistics on 188 = set statistics off 193 = select 195 = insert 196 = delete 197 = update 198 = create table 203 = create database 205 = grant 206 = revoke 207 = create view 221 = create trigger 222 = create procedure 224 = execute 228 = dump database 233 = create default 235 = dump transaction 236 = create rule 253 = connect 282 = delete statistics 317 = dbcc 320 = truncate table 326 = update statistics 347 = set tracing	
protecttype	tinyint	可以为以下值之一： <ul style="list-style-type: none"> <li>0 = grant with grant</li> <li>1 = grant</li> <li>2 = revoke</li> </ul>	
columns	varbinary(133)	该 <code>select</code> 、 <code>update</code> 或 <code>references</code> 权限适用的列的位图。这些位表示以下信息： <ul style="list-style-type: none"> <li>0 = 表示所有列。</li> <li>1 = 表示权限适用于该列。</li> <li>NULL = 表示没有信息。</li> </ul> columns 也是一个由 <code>set session authorization</code> 权限所允许使用的角色组成的位图。	
grantor	int	授权者的用户 ID。如果授权者是系统管理员，则使用对象所有者的用户 ID。	

索引

- id、action、grantor、uid、protecttype 上的唯一聚簇索引

## sysquerymetrics

### 所有数据库

#### 说明

根据持久数据为单个查询提供集合的历史查询处理指标。除了监控表，还可以使用此目录中的性能指标信息。

#### 列

sysquerymetrics 的列为：

名称	数据类型	说明
uid	int	用户 ID
gid	int	组 ID
hashkey	int	SQL 查询文本上的散列键
id	int	唯一 ID
sequence	smallint null	在 SQL 文本需要多行的情况下行的序列号
exec_min	int null	最短执行时间
exec_max	int null	最长执行时间
exec_avg	int null	平均执行时间
elap_min	int null	最短经历时间
elap_max	int null	最长经历时间
elap_avg	int null	平均经历时间
lio_min	int null	最小逻辑 IO
lio_max	int null	最大逻辑 IO
lio_avg	int null	平均逻辑 IO
pio_min	int null	最小物理 IO
pio_max	int null	最大物理 IO
pio_avg	int null	平均物理 IO
cnt	int null	已经执行的查询次数。
abort_cnt	int null	由于超过资源限制资源管理器中止查询的次数
qtext	varchar(255) null	查询文本

Adaptive Server 版本 15.0.2 和更高版本增加了在用户 ID 之间共享的指标数，减少了 sysquerymetrics (sysqueryplans 的一个视图) 中的条目数，并自动集合不同用户 ID 中相同查询的指标。

当查询中未被用户名限定的所有表名都归 DBO 所有时，sysquerymetrics 的用户 ID (uid) 将为 0。

例如，如果表 `t1` 只归 `DBO` 所有并且由不同用户共享：

```
select * from t1 where c1 = 1
```

对于没有名为 `t1` 的专用表的执行此查询的所有用户，Adaptive Server 将使用 `0` 作为 `sysquerymetrics` 条目的 `uid`。

在此示例中，如果表 `t2` 归 “`user1`” 所有并由其限定，Adaptive Server 也会使用 `UID 0`：

```
selet * from user1.t2 where c1 = 1
```

但是，如果表 `t3` 仅归 “`user1`” 所有，不由 `DBO` 限定，也不归 `DBO` 所有，则将在 `sysquerymetrics` 条目中使用 “`user1`” 的 `UID`：

```
select * from t3 where c1 = 1
```

## sysqueryplans

### 所有数据库

**说明** 每个抽象查询计划在 `sysqueryplans` 中都有相应的两行或更多行。它使用数据行锁。

**列** `sysqueryplans` 的列为：

名称	数据类型	说明
<code>uid</code>	<code>int</code>	捕获抽象计划的用户的用户 ID。
<code>dbid</code>	<code>int null</code>	仅供将来使用
<code>qptime</code>	<code>datetime null</code>	仅供将来使用
<code>sprocid</code>	<code>int null</code>	仅供将来使用
<code>hashkey2</code>	<code>int null</code>	仅供将来使用
<code>key1</code>	<code>int null</code>	仅供将来使用
<code>key2</code>	<code>int null</code>	仅供将来使用
<code>key3</code>	<code>int null</code>	仅供将来使用
<code>gid</code>	<code>int</code>	保存抽象计划时使用的抽象计划组 ID。
<code>hashkey</code>	<code>int</code>	SQL 查询文本上的散列键。
<code>id</code>	<code>int</code>	如果是抽象计划，则为唯一 ID。
<code>type</code>	<code>smallint</code>	如果文本列中包含查询文本，则为 10；如果文本列中包含抽象计划文本，则为 100。
<code>sequence</code>	<code>smallint</code>	如果 SQL 查询或抽象计划的文本需要多个行，则为序列号。
<code>status</code>	<code>int null</code>	保留。
<code>text</code>	<code>varchar(255) null</code>	如果 <code>type</code> 为 10，则为 SQL 文本；如果 <code>type</code> 为 100，则为抽象查询计划文本。

**索引**

- `uid`、`gid`、`hashkey`、`id`、`type`、`sequence` 上的唯一聚簇索引
- `id`、`type`、`sequence` 上的非聚簇索引



## sysreferences

### 所有数据库

说明 在表或列上声明的每个参照完整性约束在 `sysreferences` 中都有相应的一行。

列 `sysreferences` 的列为：

名称	数据类型	说明
<code>indexid</code>	<code>smallint</code>	被引用列上的唯一索引的 ID
<code>constrid</code>	<code>int</code>	来自 <code>sysobjects</code> 的约束的对象 ID
<code>tableid</code>	<code>int</code>	引用表的对象 ID
<code>reftabid</code>	<code>int</code>	被引用表的对象 ID
<code>keycnt</code>	<code>smallint</code>	外键中的列数
<code>status</code>	<code>smallint</code>	选项和指示符
<code>frgnbid</code>	<code>smallint null</code>	包含被引用表（具有外键的表）的数据库的数据库 ID
<code>pmrydbid</code>	<code>smallint</code>	包含被引用表（具有主键的表）的数据库的数据库 ID
<code>spare2</code>	<code>int</code>	保留
<code>fokey1 ... fokey16</code>	<code>smallint</code>	第 1 至第 16 个引用列的列 ID
<code>refkey1 ... refkey16</code>	<code>smallint</code>	第 1 至第 16 个被引用列的列 ID
<code>frgndbname</code>	<code>varchar(30) null</code>	包含引用表（具有外键的表）的数据库的名称；如果引用表在当前数据库中，则为 NULL
<code>pmrydbname</code>	<code>varchar(30) null</code>	包含被引用表（具有主键的表）的数据库的名称；如果被引用表在当前数据库中，则为 NULL

表 1-23 显示了 `sysreferences` 中的 `status` 位。

**表 1-23: `sysreferences` 表中的 `status` 位**

十进制	十六进制	状态
2	0x2	参照约束具有 <code>match full</code> 选项

索引

- `tableid`、`frgndbname` 上的聚簇索引
- `constrid`、`frgndbname` 上的非聚簇索引
- `reftabid`、`indexid`、`pmrydbname` 上的非聚簇索引

## sysremotelogins

### 仅限 master 数据库

**说明** 每个可以在该 Adaptive Server 上执行远程过程调用的远程用户在 sysremotelogins 中都有相应的一行。

**列** sysremotelogins 的列为：

名称	数据类型	说明
remoteserverid	smallint	标识远程服务器
remoteusername	varchar(30) null	用户在远程服务器上的登录名
suid	int	本地服务器用户 ID
status	smallint	选项的位图

**索引**

- remoteserverid、remoteusername 上的唯一聚簇索引

## sysresourcelimits

### 仅限 master 数据库

**说明** Adaptive Server 定义的每个资源限制在 `sysresourcelimits` 中都有相应的一行。资源限制指定 Adaptive Server 登录名或应用程序执行查询、查询批处理或事务时可以使用的最大服务器资源数量。

**列** `sysresourcelimits` 的列为：

名称	数据类型	说明
<code>name</code>	<code>varchar(30) null</code>	登录名
<code>appname</code>	<code>varchar(30) null</code>	应用程序名
<code>rangeid</code>	<code>smallint</code>	<code>systemranges</code> 中的 id 列
<code>limitid</code>	<code>smallint</code>	<code>spt_limit_types</code> 中的 id 列
<code>enforced</code>	<code>tinyint</code>	<code>spt_limit_types</code> 中的 <code>enforced</code> 列的子集： <ul style="list-style-type: none"> <li>• 1 = 执行前</li> <li>• 2 = 执行中</li> <li>• 3 = 上述两种情况</li> </ul>
<code>action</code>	<code>tinyint</code>	出现冲突时执行的操作： <ul style="list-style-type: none"> <li>• 1 = 发出警告</li> <li>• 2 = 中止查询批处理</li> <li>• 3 = 中止事务</li> <li>• 4 = 注销会话</li> </ul>
<code>limitvalue</code>	<code>int</code>	限制值
<code>scope</code>	<code>tinyint</code>	用户限制的范围（表示以下一种或多种情况的位图）： <ul style="list-style-type: none"> <li>• 1 = 查询</li> <li>• 2 = 查询批处理</li> <li>• 4 = 事务</li> </ul>
<code>spare</code>	<code>tinyint</code>	保留

**索引**

- `name`、`appname` 上的聚簇索引

## sysroles

### 所有数据库

说明

sysroles 将服务器角色 ID 映射到本地角色 ID。

列

sysroles 的列为：

名称	数据类型	说明
id	int	服务器角色 ID (srid)
lrid	int	本地角色 ID
type	smallint	未使用
status	int	未使用

在将数据库权限授予角色时，如果 sysroles 中没有该角色的条目，Adaptive Server 将在 sysroles 中添加一个条目，用于将本地角色 ID (lrid) 映射到 sysserverroles 中的全服务器范围角色 ID (srid)。

索引

- lrid 上的唯一聚簇索引

## syssecmechs

### 仅限 master 数据库

说明

syssecmechs 包含可供 Adaptive Server 使用的每个安全机制所支持的安全服务的相关信息。syssecmechs 不是在安装期间创建的，而是在被用户查询时动态创建的。

列

syssecmechs 的列为：

名称	数据类型	说明
sec_mech_name	varchar(30)	安全机制的名称，如“NT LANMANAGER”
available_service	varchar(30)	安全机制所支持的安全服务的名称，如“unified login”

## syssegments

### 所有数据库

#### 说明

每个段（命名的磁盘区段集）在 `syssegments` 中都有相应的一行。在一个新创建的数据库中，这些条目为：段 0 (`system`) 用于系统表；段 2 (`logsegment`) 用于事务日志；段 1 (`default`) 用于其它对象。

#### 列

`syssegments` 的列为：

名称	数据类型	说明
<code>segment</code>	<code>smallint</code>	段号
<code>name</code>	<code>sysname</code>	段名
<code>status</code>	<code>smallint null</code>	表示哪个段是缺省段

## syssservers

### 仅限 master 数据库

**说明** 此 Adaptive Server 可在其上执行远程过程调用的每个远程 Adaptive Server™、Backup Server 或 Open Server™ 在 syssservers 中都有相应的一行。

**列** syssservers 的列为：

名称	数据类型	说明
srvid	smallint	远程服务器的 ID 号（仅供本地使用）。
srvstatus	smallint	选项的位图（请参见表 1-24）。
srvstatus2	unsigned int	选项的位图（请参见表 1-25）。
srvstat2	unsigned int	服务器选项的位图
srvname	varchar(30)	服务器名。
srvnetname	varchar(255)	服务器的 interfaces 文件名。
srvclass	smallint null	sp_addserver 的类参数所定义的服务器类别（请参见表 1-26）。
srvsecmech	varchar(30) null	安全机制。
svrcost	smallint null	它为通过网络访问服务器的操作提供网络开销（以毫秒为单位）。仅当访问代理表时，Adaptive Server 查询优化程序使用它来评估查询开销，缺省设置为 1,000 毫秒。

表 1-24 列出了 srvstatus 列的位表示形式：

**表 1-24: srvstatus 列的状态控制位**

十进制	十六进制	状态
0	0x0	超时已启用
1	0x1	超时已禁用
2	0x2	网络口令加密已启用
4	0x4	远程服务器为只读
8	0x8	使用 RPC 安全模式 A
16	0x10	使用 RPC 安全模式 B
64	0x40	使用消息保密性
128	0x80	使用消息完整性
256	0x100	相互鉴定

**表 1-25: syssservers 表中的 srvstatus2 控制位**

十进制	十六进制	状态
1	0x01	支持全限定表名
2	0x02	留作将来使用

表 1-26 列出了 `srvclass` 列的服务器类别：

**表 1-26: sys.servers 表中的服务器类别**

<b>srvclass</b>	<b>服务器类别</b>
0	本地服务器
1	sql_server 类服务器
3	direct_connect 类服务器
4	DB2 类服务器
6	sds 类服务器
7	Adaptive Server Enterprise 类服务器
8	Adaptive Server Anywhere 类服务器
9	ASIQ 类服务器

索引

- `srvid` 上的唯一聚簇索引
- `srvname` 上的非聚簇索引



## syssessions

### 仅限 master 数据库

#### 说明

syssessions 仅用于已在高可用性系统中将 Adaptive Server 配置为用于 Sybase 故障切换的情况。连接到具有故障切换属性的 Adaptive Server 的每个客户端在 syssessions 中都有相应的一行。当进行故障切换时，将在 syssessions 中具有条目的客户端移动到辅助协同服务器上。而在故障切换时，将在 syssessions 中没有条目的客户端删除。当进行故障恢复时，将在 syssessions 中具有条目的客户端移动到主协同服务器上。而在故障恢复时，将在 syssessions 中没有条目的客户端删除。

#### 列

syssessions 的列为：

名称	数据类型	说明
sys_id	smallint	协同服务器节点的唯一标识
ses_id	int	每个客户端会话的唯一标识
state	tinyint	说明会话是活动的还是不活动的
spare	tinyint	留作将来使用
status	smallint	留作将来使用
dbid	smallint	留作将来使用
name	varchar(30) null	与 syslogins 中指定的客户端登录名相同
nodeid	tinyint null	留作将来使用（不能用于集群环境）
instanceid	tinyint	实例的 ID（只能用于集群环境）
ses_data	image null	留作将来使用

## syslices

### 所有数据库

#### 说明

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。升级完成后将删除全部数据。

**注释** 在 15.0 以前的 Adaptive Server 版本中，`syspartitions` 是存储分区相关信息的目录的名称。在 Adaptive Server 15.0 中，此目录已经被重命名为 `syslices`，而 `syspartitions` 现在指的是用来跟踪 Adaptive Server 中所有与分区相关的数据的目录。

#### 列

`syslices` 的列为：

名称	数据类型	说明
<code>state</code>	<code>smallint</code>	有关分区状态的内部信息
<code>id</code>	<code>int</code>	分区表的对象 ID
<code>partitionid</code>	<code>int</code>	分区 ID 号
<code>firstpage</code>	<code>int</code>	分区第一页的页码
<code>controlpage</code>	<code>int</code>	分区控制页的页码
<code>spare</code>	<code>binary(32)</code>	保留

#### 索引

- `id`、`partitionid` 上的唯一聚簇索引

## sysrvroles

### 仅限 master 数据库

说明

每个系统角色或用户定义的角色在 `sysrvroles` 中都有相应的一行。

列

`sysrvroles` 的列为：

名称	数据类型	说明
<code>srid</code>	<code>int</code>	服务器角色 ID
<code>name</code>	<code>varchar(30)</code>	角色名称
<code>password</code>	<code>varbinary(128)</code> <code>null</code>	角色的口令（已加密），该口令只有具有 <code>sso_role</code> 角色的用户才能读取。
<code>pwdate</code>	<code>datetime null</code>	上次更改口令的日期
<code>status</code>	<code>smallint null</code>	角色状态的位图（请参见表 1-27）
<code>logincount</code>	<code>smallint null</code>	失败的登录尝试次数；登录成功后将其重置为 0

表 1-27 列出了 `status` 列的位表示形式：

**表 1-27: `sysrvroles` 表中的 `status` 控制位**

十进制	十六进制	状态
2	0x2	角色已锁定
4	0x4	角色已到期

索引

- `srid` 上的唯一聚簇索引

## sysstatistics

### 所有数据库

#### 说明

用户表上的每个索引列和每个分区在 `sysstatistics` 中都有相应的一行或多行。未索引的列在其中也可能有相应的行。

#### 列

`sysstatistics` 的列为：

名称	数据类型	说明
<code>statid</code>	<code>smallint</code>	保留
<code>id</code>	<code>int</code>	表的对象 ID
<code>sequence</code>	<code>int</code>	序列号（如果这组统计信息需要多行）
<code>moddate</code>	<code>datetime</code>	上次修改该行的日期
<code>formatid</code>	<code>tinyint</code>	该行代表的统计信息类型
<code>usedcount</code>	<code>tinyint</code>	在该行中使用的字段 <code>c0</code> 到 <code>c79</code> 的个数
<code>colidarray</code>	<code>varbinary(100)</code>	列 ID 的有序列表
<code>c0...c79</code>	<code>varbinary(255)</code>	统计数据
<code>indid</code>	<code>smallint</code>	分区的索引 ID
<code>ststatus</code>	<code>smallint</code>	此统计信息行的状态位；根据行的类型的不同，可能的值会有所不同。
<code>partitionid</code>	<code>int</code>	分区 ID (Partition ID)
<code>spare2</code>	<code>smallint</code>	留作将来使用
<code>spare3</code>	<code>int</code>	留作将来使用

#### 索引

- `id`、`indid`、`partitionid`、`statid`、`colidarray`、`formatid`、`sequence` 上的唯一聚簇索引 `csysstatistics`

# systabstats

## 所有数据库

说明 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` 中都有相应的一行。

列 `systabstats` 的列为：

名称	数据类型	说明
<code>indid</code>	<code>smallint</code>	<ul style="list-style-type: none"> <li>0 = 如果是表</li> <li>1 = 如果是所有页锁定表上的聚簇索引</li> <li>&gt;1 = 如果是 DOL 锁定表上的非聚簇索引或聚簇索引</li> </ul> <p><code>systabstats</code> 并不维护关于 <code>text</code> 或 <code>image</code> 对象 (255) 的统计信息</p>
<code>id</code>	<code>int</code>	索引所属的表的 ID
<code>activestatid</code>	<code>smallint</code>	保留
<code>indexheight</code>	<code>smallint</code>	索引的高度；在 <code>indid</code> 大于 1 时保存
<code>leafcnt</code>	<code>int</code>	索引中的叶页数；在 <code>indid</code> 大于 1 时保存
<code>pagecnt</code>	<code>int</code>	表或索引中的页数
<code>rowcnt</code>	<code>float</code>	表中的行数；在 <code>indid</code> 为 0 或 1 时保存
<code>forrowcnt</code>	<code>float</code>	转移的行数；在 <code>indid</code> 为 0 或 1 时保存
<code>delrowcnt</code>	<code>float</code>	已删除的行数
<code>dpagcrcnt</code>	<code>float</code>	需要执行以读取整个表的扩充 I/O 数目
<code>ipagcrcnt</code>	<code>float</code>	需要执行以读取整个叶级非聚簇索引的扩充 I/O 数目
<code>drowcrcnt</code>	<code>float</code>	需要执行以读取整个表的页 I/O 数目
<code>oamapgcnt</code>	<code>int</code>	表的 OAM 页数，加上存储表信息的分配页数
<code>extent0pgcnt</code>	<code>int</code>	与分配页在相同扩充上的页数
<code>datarowsz</code>	<code>float</code>	数据行的平均大小
<code>leafrowsz</code>	<code>float</code>	DOL 锁定表的非聚簇索引和聚簇索引的叶行的平均大小
<code>status</code>	<code>int</code>	内部系统状态信息（请参见表 1-28）
<code>plljoindegree</code>	<code>int</code>	用于嵌套循环连接操作的并行度， <code>plljoindegree</code> 是表的并行扫描度，该表的 <code>systabstats</code> 具有此字段，该表是嵌套循环连接中的内部表。
<code>spare2</code>	<code>float</code>	保留
<code>rslastoam</code>	<code>int</code>	通过 <code>reorg reclaim_space</code> 或 <code>reorg compact</code> 命令访问的最后一个 OAM 页
<code>rslastpage</code>	<code>int</code>	通过 <code>reorg reclaim_space</code> 或 <code>reorg compact</code> 命令访问的最后一个数据页或叶页
<code>frlastoam</code>	<code>int</code>	通过 <code>reorg forwarded_rows</code> 命令访问的最后一个 OAM 页
<code>frlastpage</code>	<code>int</code>	通过 <code>reorg forwarded_rows</code> 命令访问的最后一个数据页
<code>conopt_thld</code>	<code>smallint</code>	并发优化阈值
<code>plldegree</code>	<code>int16</code>	关于数据操作语言 (DML) 表或索引的可能的最大并行度。值为 0 表示不存在最大值；查询处理器将配置最大并行度。

名称	数据类型	说明
emptypgcnt	int	分配给表或索引的扩充中的空白页数
spare4	float	保留
partitionid	int	分区 ID (Partition ID)
spare5	int	对齐方式的备用字段
statmoddate	datetime	行上一次被刷新到磁盘的时间
unusedpgcnt	int	未使用的页数
oampagecnt	int	对象分配映射中列出的分配页数

表 1-28 列出了 status 列的位表示形式：

**表 1-28: systabstats 表中的 status 位**

十进制	十六进制	状态
1	0x1	统计信息是升级的结果（而不是执行 update statistics 的结果）

索引

- id、indid、partitionid 上的唯一聚簇索引

## systhresholds

### 所有数据库

说明 为数据库定义的每个阈值在 `systhresholds` 中都有相应的一行。

列 `systhresholds` 的列为：

名称	数据类型	说明
<code>segment</code>	<code>smallint</code>	正在监控可用空间的段号。
<code>free_space</code>	<code>int</code>	阈值的大小，以逻辑页为单位。
<code>status</code>	<code>smallint</code>	对于日志段的最后机会阈值，第 1 位等于 1，而对于所有其它阈值，第 1 位等于 0。
<code>proc_name</code>	<code>varchar(255)</code>	当 <code>segment</code> 上的未使用页数少于 <code>free_space</code> 时所执行的过程的名称。
<code>suid</code>	<code>int null</code>	添加阈值或最近修改阈值的用户的服务器用户 ID。
<code>currauth</code>	<code>varbinary(255) null</code>	位屏蔽，表示在添加阈值或最近修改阈值时为 <code>suid</code> 激活的角色。当超过阈值时， <code>proc_name</code> 使用该角色集来执行，但要扣除在添加阈值或上次修改阈值后已失效的所有角色。

**根据 `currauth` 确定活动角色** 下表列出了在 `currauth` 列中可能看到（单个或组合形式）的可能位屏蔽。

十进制	十六进制	说明
1	0x1	<code>sa_role</code>
2	0x2	<code>sso_role</code>
4	0x4	<code>oper_role</code>
8	0x8	<code>sybase_ts_role</code>
16	0x10	<code>sybase_ts_role</code>
32	0x20	<code>navigator_ole</code>
128	0x80	<code>replication_role</code>
256	0x100	<code>dtm_tm_role</code>
1024	0x400	<code>ha_role</code>
2048	0x800	<code>mon_role</code>
4096	0x1000	<code>js_admin_role</code>
16384	0x4000	<code>messaging_role</code>
32768	0x8000	<code>web_services</code>

要确定 Adaptive Server 中哪一个角色 ID 与 currauth 中的位屏蔽输出相关联，请执行以下 select 语句：

```
1> select (c.number - 1) as role_id,role_name(c.number - 1) as role_name
2> from systhresholds ,master.dbo.spt_values c
3> where  convert(tinyint,substring(isnull(currauth,0x1), c.low,1)) &
4> c.high != 0
5> and c.type = "P"
6> and c.number <= 1024
7> and c.number >0
8> and role_name(c.number - 1) is not null
9> go
```

Adaptive Server 将返回类似于以下输出的内容：

role_id	role_name
0	sa_role
1	sso_role
2	oper_role
3	sybase_ts_role
4	navigator_role
7	dtm_tm_role
10	mon_role
11	js_admin_role
12	messaging_role
13	js_client_role



## systimeranges

### 仅限 master 数据库

**说明** `systimeranges` 存储指定的时间范围，Adaptive Server 将使用该时间范围来控制何时应用资源限制。

**列** `systimeranges` 的列为：

名称	数据类型	说明
name	varchar(255)	时间范围的唯一名称。
id	smallint	时间范围的唯一标识符。1 表示“所有时间”限制。
startday	tinyint	范围在一周中开始的那一天 (1 — 7)。星期一 = 1, 星期日 = 7。
endday	tinyint	范围在一周中结束的那一天 (1 — 7)。星期一 = 1, 星期日 = 7。
starttime	varchar(10)	范围在一天中开始的时间。
endtime	varchar(10)	范围在一天中结束的时间。

**索引**

- id 上的聚簇索引

# systransactions

## 仅限 master 数据库

**说明** systransactions 包含有关 Adaptive Server 事务的信息，但它不是常规表。表的某些部分是在用户查询时动态创建的，而其它部分则存储在 master 数据库中。不允许更新动态创建的 systransactions 列。

**列** systransactions 的列为：

名称	数据类型	说明
xactkey	binary(14)	唯一的 Adaptive Server 事务关键字
starttime	datetime	事务的启动日期
failover	int	表示事务故障切换状态的值（请参见表 1-29）
type	int	表示事务类型的值（请参见表 1-30）
coordinator	int	表示协调方法或协议的值（请参见表 1-31）
state	int	表示事务当前状态的值（请参见表 1-32）
connection	int	表示连接状态的值（请参见表 1-33）
status	int	内部事务状态标志
status2	int	附加的内部事务状态标志
spid	smallint	服务器进程 ID；0 表示进程分离
		对于 Cluster Edition，为 int
masterdbid	smallint	事务的起始数据库
loid	int	锁所有者 ID
namelen	smallint	xactname 的长度
xactname	varchar(255) null	事务名称或 XID
srvname	varchar(30) null	远程服务器的名称（对于本地服务器为 null）
nodeid	tinyint null	留作将来使用（不能用于集群环境）
instanceid	tinyint	实例的 ID（只能用于集群环境）

**注释** 由于 Cluster Edition 的数据类型的这一更改，Sybase 强烈建议您在升级之前存档并截断审计表。这会降低由于 sybsecurity 数据库中空间不足而导致升级失败的可能性。

**表 1-29: systransactions 中 failover 列的值**

failover 值	故障切换状态
0	Resident Tx
1	Failed-over Tx
2	Tx by Failover-Conn

**表 1-30: systransactions 中 type 列的值**

<b>type 值</b>	<b>事务类型</b>
1	本地
3	外部
98	远程
99	Dtx_State

**表 1-31: systransactions 中 coordinator 列的值**

<b>coordinator 值</b>	<b>协调方法或协议</b>
0	无
1	Syb2PC
2	ASTC
3	XA
4	DTC

**表 1-32: systransactions 中 state 列的值**

<b>state 值</b>	<b>事务状态</b>
1	Begun
2	Done Command
3	Done
4	Prepared
5	In Command
6	In Abort Cmd
7	Committed
8	In Post Commit
9	In Abort Tran
10	In Abort Savept
65537	Begun-Detached
65538	Done Cmd-Detached
65539	Done-Detached
65540	Prepared-Detached
65548	Heur Committed
65549	Heur Rolledback

**表 1-33: systransactions 中 connection 列的值**

<b>connection 值</b>	<b>连接状态</b>
1	Attached
2	Detached

## systypes

### 所有数据库

**说明** 每个系统提供的数据类型和用户定义的数据类型在 `systypes` 中都有相应的一行。域（由规则定义）和缺省值将在其存在时提供。

不能修改用于说明系统提供的数据类型的行。

**列** `systypes` 的列为：

名称	数据类型	说明
uid	int	数据类型创建者的用户 ID
usertype	smallint	用户类型 ID
variable	bit	当数据类型为可变长度时为 1；其它情况下为 0
allownulls	bit	表示是否允许该数据类型有空值
type	tinyint	物理存储数据类型
length	int	数据类型的物理长度
tdefault	int	生成该数据类型缺省值的系统过程的 ID
domain	int	包含该数据类型完整性检查的系统过程的 ID
name	varchar(255)	数据类型名称
printfmt	varchar(255) null	保留
prec	tinyint null	有效位数
scale	tinyint null	小数点右侧的位数
ident	tinyint null	当列具有 IDENTITY 属性时为 1，否则为 0
hierarchy	tinyint null	数据类型在混合模式算术中的优先级
xtypeid	int null	内部类 ID
xdbid	int null	安装类所在的 dbid: <ul style="list-style-type: none"> <li>• -1 = 系统数据库</li> <li>• -2 = 当前数据库</li> </ul>
accessrule	int null	sysprocedures 中的访问规则的对象 ID

表 1-34 列出了系统提供的每种数据类型的 name、hierarchy、type（不一定唯一）和 usertype（唯一）。数据类型按 hierarchy 排序。在混合模式算术中，具有最低 hierarchy 的数据类型优先执行：

**表 1-34: 数据类型的 name、hierarchy、type 和 usertype**

Name	hierarchy	type	usertype
floatn	1	109	14
float	2	62	8
datetimn	3	111	15
datetime	4	61	12
real	5	59	23
numericn	6	108	28
numeric	7	63	10
decimaln	8	106	27
decimal	9	55	26
moneyn	10	110	17
money	11	60	11
smallmoney	12	122	21
smalldatetime	13	58	22
intn	14	38	13
uintn	15	68	47
bigint	16	191	43
ubigint	17	67	46
int	18	56	7
uint	19	66	45
smallint	20	52	6
usmallint	21	65	44
tinyint	22	48	5
bit	23	50	16
univarchar	24	155	35
unichar	25	135	34
unitext	26	174	36
varchar	27	39	2
sysname	27	39	18
nvarchar	27	39	25
longsysname	27	39	42
char	28	47	1
nchar	28	47	24
varbinary	29	37	4

<b>Name</b>	<b>hierarchy</b>	<b>type</b>	<b>usertype</b>
timestamp	29	37	80
binary	30	45	3
text	31	35	19
image	32	34	20
date	33	49	37
time	34	51	38
daten	35	123	39
timen	36	147	40
extended type	99	36	-1

索引

- name 上的唯一聚簇索引
- usertype 上的唯一非聚簇索引

## sysusages

### 仅限 master 数据库

#### 说明

分配给数据库的每个**磁盘分配区段**在 `sysusages` 中都有相应的一行。每个数据库都包含指定数目的数据库（逻辑）页数。

`create database` 命令检查 `sysdevices` 和 `sysusages`，以找出可用的磁盘分配区段。将为数据库分配一个或多个连续的磁盘分配区段，并将映射记录在 `sysusages` 中。

有关 `sysusages` 的详细信息，请参见《系统管理指南》第 21 章“创建和管理用户数据库”中的“管理空间分配的系统表”。

**注释** 对于 Adaptive Server 15.0 版本，设备标识号存储在 `vdevno` 列中，而不再作为 `vstart` 列的一部分进行存储。因此，您可能需要对那些根据先前模式来确定设备标识号的脚本和存储过程进行修改。

#### 列

`sysusages` 的列为：

名称	数据类型	说明
<code>dbid</code>	<code>smallint</code>	数据库 ID
<code>segmap</code>	<code>int</code>	可能的段赋值位图
<code>lstart</code>	<code>int</code>	第一个数据库（逻辑）页的页码
<code>size</code>	<code>int</code>	相邻数据库（逻辑）页的数目
<code>vstart</code>	<code>int</code>	起始虚拟页的页码
<code>pad</code>	<code>smallint null</code>	未使用
<code>unreservedpgs</code>	<code>int null</code>	不属于已分配扩充的可用空间
<code>crdate</code>	<code>datetime null</code>	创建日期
<code>vdevno</code>	<code>int</code>	设备标识号

#### 索引

- `dbid`、`lstart` 上的唯一聚簇索引
- `vdevno`、`vstart` 上的唯一非聚簇索引

## sysusermessages

### 所有数据库

**说明** Adaptive Server 可返回的每个用户定义消息在 `sysusermessages` 中都有相应的一行。

**列** `sysusermessages` 的列为：

名称	数据类型	说明
error	varchar(1024)	唯一的错误编号。必须大于或等于 20,000。
uid	int	消息创建者的服务器用户 ID ( <code>suser_id</code> )。
description	varchar(1024)	带有参数可选占位符的用户定义消息。
langid	smallint null	该消息的语言 ID；对于美式英语 ( <code>us_english</code> ) 为 null。
dlevel	smallint null	存储 <code>with_log</code> 位，该位用于调用相应的例程来记录消息。

**索引**

- error 上的聚簇索引
- error、langid 上的唯一非聚簇索引



## sysusers

### 所有数据库

**说明** 数据库中允许的每个用户以及每个组或角色在 `sysusers` 中都有相应的一行。

**列** `sysusers` 的列为：

名称	数据类型	说明
<code>suid</code>	<code>int</code>	从 <code>syslogins</code> 中复制的服务器用户 ID。
<code>uid</code>	<code>int</code>	用户 ID，它在该数据库中是唯一的，用于授予或撤销权限。用户 ID 1 为 “ <code>dbo</code> ”。
<code>gid</code>	<code>int</code>	该用户所属的组 ID。如果 <code>uid = gid</code> ，则该条目将定义一个组。用户 ID ( <code>uid</code> ) 可使用负值。在 <code>sysusers</code> 中，将每个与组或角色相关联的 <code>suid</code> 设置为 <code>-2 (INVALID_SUID)</code> 。
<code>name</code>	<code>sysname</code>	用户名或组名，在该数据库中是唯一的。
<code>environ</code>	<code>varchar(255) null</code>	保留。

在 Adaptive Server 分发介质上，`master.sysusers` 包含以下初始用户：“`dbo`”，它的 `suid` 为 1，`uid` 为 1；“`guest`”，它的 `suid` 为 -1，`uid` 为 2；“`public`”，它的 `suid` 为 -2，`uid` 为 0。此外，在 `sysusers` 中还列出了系统定义的角色和用户定义的角色。

用户 “`guest`” 提供了一种机制，为 `sysusers` 中未显式列出的用户授予一组有限的权限来访问数据库。`master` 中的 “`guest`” 条目表示，在 Adaptive Server 中有帐户的任何用户（也就是在 `syslogins` 中有相应条目）都可以访问 `master`。

用户 “`public`” 指所有用户。当关键字 `public` 与 `grant` 和 `revoke` 命令一起使用时，表示为所有用户授予或撤销权限。

**索引**

- `suid` 上的唯一聚簇索引
- `name` 上的唯一非聚簇索引
- `uid` 上的唯一非聚簇索引

## sysxtypes

### 所有数据库

#### 说明

每种扩展 Java-SQL 数据类型在 `sysxtypes` 中都有相应的一行。

有关 Java-SQL 类和数据类型的详细信息，请参见《Adaptive Server Enterprise 中的 Java》。

#### 列

`sysxtypes` 的列为：

名称	数据类型	说明
<code>xtid</code>	<code>int</code>	系统生成的扩展类型 ID。
<code>xtstatus</code>	<code>int</code>	内部状态信息。未使用。
<code>xtmetatype</code>	<code>int</code>	未使用。
<code>xtcontainer</code>	<code>int</code>	包含类的 JAR 文件的 ID。可以为 NULL。
<code>xtname</code>	<code>varchar(255) null</code>	扩展类型的名称。
<code>xtsource</code>	<code>text null</code>	扩展类型的源代码。未使用。
<code>xtbinaryinrow</code>	<code>varbinary(255) null</code>	扩展类型的对象代码。对于 Java 类来说，它包含类文件。数据存储在行内，最大长度可为 255 个字节。
<code>xtbinaryoffrow</code>	<code>image</code>	扩展类型的对象代码。对于 Java 类来说，它包含类文件。数据作为图像列存储在行外。

#### 索引

- `xtid` 上的唯一聚簇索引
- `xtname` 上的唯一非聚簇索引

除了包含在所有数据库中的标准系统表外，dbcc 管理数据库 dbccdb 还包含七个表，这些表定义了 dbcc checkstorage 的输入和输出。它还至少包含两个工作空间。

主题	页码
dbccdb 工作空间	97
dbccdb 日志	98

## dbccdb 工作空间

工作空间是 dbccdb 中的特殊表，用于存储 dbcc checkstorage 操作的中间结果。工作空间与工作表的区别在于：

- 可通过连续预分配工作空间来提高 I/O 性能
- 工作空间是持久的
- 工作空间不在 tempdb 数据库中

当创建 dbccdb 时，将自动创建两个工作空间。它们的预分配方式如下：

- *scan 工作空间*— 目标数据库的每一页都在其中有相应的一行。该分配的大小约为数据库大小的 1%。每行都由单个 binary(18) 列构成。
- *text 工作空间*— 目标数据库中每个包含 text 或 image 列的表在其中都有相应的一行。该表的大小取决于目标数据库的设计，但它通常远小于 scan 工作空间。每行都由单个 binary(22) 列构成。

如果任一分配超过 dbcc checkstorage 所需的大小，该操作将只使用所需的部分。该分配并无变化。如果 text 工作空间分配太小，dbcc checkstorage 将报告这一情况，推荐新的大小并继续进行检查；但它不会检查所有文本链。如果 scan 工作空间分配太小，dbcc checkstorage 操作将立即失败。

您必须至少有一个 `scan` 和一个 `text` 工作空间，但也可以根据需要创建任意多个工作空间。这些工作空间在使用时是被锁定的，所以在任意给定的时间，只有一个 `dbcc checkstorage` 操作可以使用它们。可以执行并发的 `dbcc checkstorage` 操作，方法是每个操作单独提供一个 `scan` 和 `text` 工作空间。

有关创建工作空间的详细信息，请参见《系统管理指南》和 `Adaptive Server Reference Manual`（《Adaptive Server 参考手册》）。

在理想情况下，只应通过 `dbcc checkstorage` 来访问工作空间，但这并不是强制要求。`dbcc checkstorage` 将以独占方式锁定所用工作空间，且每次执行 `dbcc checkstorage` 时，都会重新生成工作空间的内容。工作空间不包含任何安全数据。

---

**注释** 虽然可以通过 SQL 访问工作空间的内容，但无法获得二进制值的解释。如果通过 SQL 进行访问，可能返回不同 `dbcc` 检查混杂在一起的数据。某一行在这些表中出现并不能保证它包含的数据有效。`dbcc` 只有在执行时才会跟踪有效的行。当操作结束时，该信息将会丢失。

---

`dbccdb` 中的大多数更新活动都是在 `text` 和 `scan` 工作空间中执行的。工作空间是预分配的，在任意给定的时间，只有一个 `dbcc checkstorage` 操作可以使用工作空间，所以与大多数用户表相比，工作空间更加不易损坏。工作空间损坏可能会使 `dbcc checkstorage` 操作失败或出现反常行为。如果发生这种情况，应删除损坏的工作空间并重新进行创建。

可以同时检查使用不同工作空间的数据库，但由于存在对磁盘吞吐量的争用，所以每个操作的性能会有所降低。

若要删除工作空间，请在 `dbccdb` 中输入：

```
drop table workspace_name
```

## dbccdb 日志

每个 `dbcc checkstorage` 操作的结果都记录在 `dbccdb` 日志中。该日志不记录对 `text` 和 `scan` 工作空间的更新。

`dbccdb` 日志的大小必须能够处理对表的更新。日志要求与目标数据库中的表和索引的数量有关。它与目标数据库的大小无关。

要最大限度地降低日志要求和减少恢复时间，请将 `truncate log on checkpoint` 选项用于 `dbccdb`。

## dbcc\_config

**说明** dbcc\_config 表说明当前正在执行的或者上次完成的 dbcc checkstorage 操作。它定义如下内容：

- dbcc checkstorage 操作的专用资源的位置
- dbcc checkstorage 操作的资源使用限制

**列** dbcc\_config 的列为：

列名	数据类型	说明
dbid	smallint	与 sysindatabases 行中的 dbid 相匹配。
type_code	int	与 <a href="#">dbcc_types</a> 表中某行的 type_code 相匹配。有效值为 1 到 9。
value	int null	指定 type_code 所标识项目的值。只有在 stringvalue 的值不为 null 时才能为 null。
stringvalue	varchar(255) null	指定 type_code 所标识项目的值。只有在 value 的值不为 null 时才能为 null。

**主键** dbid 和 type\_code 的组合

**另请参见** 有关初始化和更新 dbcc\_config 的信息，请参见《系统管理指南》。

## dbcc\_counters

**说明** dbcc\_counters 表存储 dbcc checkstorage 所执行的分析的结果。将为每个数据库、表、索引、分区、设备和 dbcc 调用保留相应的计数器。

**列** dbcc\_counters 的列为：

列名	数据类型	说明
dbid	smallint	标识目标数据库。
id	int	标识表。该值派生自 sysindexes 和 sysobjects。
indid	smallint	标识索引。该值派生自 sysindexes。
partitionid	int	标识已定义的对象页的关联。该值派生自 sysindexes 和 syspartitions。
devid	int	标识磁盘设备。该值是从 sysdevices 派生的。
opid	smallint	标识已执行的 dbcc 操作。
type_code	int	与 dbcc_types 表中某行的 type_code 列相匹配。有效值为 5000 至 5024。
value	real null	与给定 type_code 的相应 type_name 相匹配，如 dbcc_types 所述。

**主键** dbid、id、indid、partitionid、devid、opid 和 type\_code 的组合

## dbcc\_exclusions

**说明** 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** dbcc\_exclusionss 包括以下各列：

列名	数据类型	说明
dbid	smallint	标识目标数据库。
type	tinyint	排除类型代码。有效值包括： <ul style="list-style-type: none"> <li>• 1 — 故障</li> <li>• 2 — 表</li> <li>• 3 — 两者的组合</li> </ul>
fault_type	int null	当 type 为 1（故障）或 3（两者的组合）时要排除的故障类型。有关详细信息，请参见第 106 页的“dbcc 类型”。
table_name	varchar(30) null	当 type 为 2（表）或 3（两者的组合）时要排除的表名称。有关详细信息，请参见第 106 页的“dbcc 类型”。

**主键** dbid、fault\_type 和 table\_name 的组合

## dbcc\_fault\_params

**说明** dbcc\_fault\_params 表为在 dbcc\_faults 表中输入的故障提供附加的说明性信息。

**列** dbcc\_fault\_params 的列为：

列名	数据类型	说明
dbid	smallint	标识目标数据库。
opid	smallint	标识已执行的 dbcc 操作。
faultid	int	标识故障 ID。
type_code	int	定义值的解释，值是由“value”列提供的。有效值为 1000 到 1009。 <a href="#">dbcc_types</a> 中提供了相关的说明。
intvalue	int null	指定整数值。
realvalue	real null	指定实数值。
binaryvalue	varbinary(255) null	指定二进制值。
stringvalue	varchar(255) null	指定字符串值。
datevalue	datetime null	指定日期值。

每个“value”列（intvalue、realvalue、binaryvalue、stringvalue 和 datevalue）都可以包含空值。但至少有一列不能为 null。如果有多个列包含不为 null 的值，这些列将为同一值提供不同的表示形式。

**主键** dbid、opid、faultid 和 type\_code 的组合



## dbcc\_faults

**说明** dbcc\_faults 表提供 dbcc checkstorage 检测到的每个故障的说明。

**列** dbcc\_faults 的列为：

列名	数据类型	说明
dbid	smallint	标识目标数据库。
id	smallint	标识表。该值派生自 sysindexes 和 sysobjects。
indid	smallint	标识索引。该值派生自 sysindexes。
partitionid	int	标识分区。该值派生自 sysindexes 和 syspartitions。将为页范围维护计数器，因此，“分区”指的是已定义的对象页的相关性，而不是实际的对象页链。
devid	int	标识磁盘设备。该值是从 sysdevices 派生的。
opid	smallint	标识已执行的 dbcc 操作。
faultid	int	提供为操作记录的每个故障所分配到的唯一序列号。
type_code	int	标识故障的类型。有效值为 100000 到 100032。第 106 页的表 2-1 中提供了相关的说明。
status	int	对故障进行分类。有效值包括： <ul style="list-style-type: none"> <li>• 0 — 软故障，可能是瞬时故障。</li> <li>• 1 — 硬故障。</li> <li>• 2 — 证实是瞬时故障的软故障。</li> <li>• 3 — 已升级为硬故障的软故障。</li> <li>• 5 — 已修复的硬故障。</li> <li>• 7 — 已修复的升级硬故障。</li> <li>• 9 — 无法修复的硬故障。</li> <li>• 11 — 已升级为硬故障且无法修复的软故障。</li> <li>• 16 — 删除了对象（无法访问对象）的软故障。</li> <li>• 17 — 删除了对象（无法访问对象）的硬故障。</li> <li>• 18 — 删除了对象（无法访问对象）的瞬时软故障。</li> <li>• 19 — 已升级为硬故障且删除了对象（无法访问对象）的软故障。</li> </ul> 有关详细信息，请参见《系统管理指南》。

**主键** dbid、id、indid、partitionid、devid、opid、faultid 和 type\_code 的组合

## dbcc\_operation\_log

**说明** dbcc\_operation\_log 表记录 dbcc checkstorage 操作的使用情况。

**列** dbcc\_operaiton\_log 的列为：

列名	数据类型	说明
dbid	smallint	标识目标数据库。
opid	smallint	标识 dbcc checkstorage 操作的序列号。opid 是一种自动递增的编号，对于每个 dbid 和 finish 对来说，它是唯一的。
optype	smallint	optype 的有效值包括： <ul style="list-style-type: none"> <li>• 2 = checkstorage</li> </ul>
suid	int	标识正在执行命令的用户。
start	datetime	标识启动操作时的时间。
finish	datetime null	标识操作结束时的时间。
seq	smallint null	checkverify 操作的序列号。
id	int null	checkverify 操作的对象 ID（如果使用）。
maxseq	smallint null	checkstorage 操作的 checkverify 所使用的最大序列。

摘要结果记录在 dbcc\_operation\_results 表中。

**主键** dbid、opid 和 optype 的组合

## dbcc\_operation\_results

**说明** dbcc\_operation\_results 表为记录在 dbcc\_operation\_log 表中的操作提供附加的说明性信息。

**列** dbcc\_operation\_results 的列为：

列名	数据类型	说明
dbid	smallint	标识目标数据库。
opid	smallint	标识 dbcc 操作 ID。
optype	smallint	标识 dbcc 操作类型。
type_code	int	定义 dbcc 操作类型。有效值为 1000 到 1007。第 106 页的表 2-1 中提供了相关的说明。
intvalue	int null	指定整数值。
realvalue	real null	指定实数值。
binaryvalue	varbinary(255) null	指定二进制值。
stringvalue	varchar(255) null	指定字符串值。
datevalue	datetime null	指定日期值。
seq	smallint null	checkverify 操作的序列号。

每个“value”列（intvalue、realvalue、binaryvalue、stringvalue 和 datevalue）都可以包含 null 值。但至少有一列不能为 null。如果有多个列包含不为 null 的值，这些列将为同一值提供不同的表示形式。

dbcc checkstorage 操作的结果包括以下项目的数量：

- 找到的硬故障
- 找到的软故障
- 因硬错误而停止的操作

**主键** dbid、opid、optype 和 type\_code 的组合

## dbcc\_types

说明

`dbcc_types` 表提供 `dbcc checkstorage` 使用的数据类型的定义。`dbcc` 存储过程并不实际使用该表。提供该表是为了方便 `dbccdb` 中其它表的使用，并记录数据类型的语义。其中包括操作配置、报告的分析数据、故障分类和故障报告参数的类型代码。如果创建自己的存储过程来生成报告，则可以将 `type_name` 列中列出的值用作报告标题。

列

`dbcc_types` 的列如下所示。

**注释** 为了便于以后添加 `dbcc_types`，此时不使用某些 `type_code` 编号。

**表 2-1: dbcc 类型**

<code>type_code</code>	<code>type_name</code>	说明
1	max worker processes	可选 — 指定可以使用的最大工作进程数。该值也是使用的并发处理的最大级别。最小值为 1。
2	dbcc named cache	指定 <code>dbcc checkstorage</code> 所用高速缓存的大小（用 KB 表示）和该高速缓存的名称。
3	scan workspace	指定数据库扫描要使用的工作空间的 ID 和名称。
4	text workspace	指定要用于文本列的工作空间的 ID 和名称。
5	operation sequence number	指定标识最近启动的 <code>dbcc</code> 操作的编号。
6	database name	指定 <code>sysdatabases</code> 中数据库的名称。
7	OAM count threshold	指定 OAM 计数在被视作错误之前必须达到的变化百分比。
8	IO error abort	指定在 <code>dbcc</code> 停止检查磁盘页之前，该磁盘上允许的 I/O 错误数。
9	linkage error abort	指定在 <code>dbcc</code> 停止检查对象的页链之前，允许出现的链接错误数。相比于其它种类的页链损坏，某些种类的页链损坏可能在链接错误数更少的情况下就需要停止检查。
10	enable automatic workspace expansion	在估计大小超过实际工作空间大小时，该标志将启用或禁用工作空间的自动扩展。
1000	hard fault count	指定在一致性检查中发现的持续不一致处（硬故障）的数目。
1001	soft fault count	指定在一致性检查中发现的可疑情况（软故障）的数目。
1002	checks aborted count	指定在一致性检查中被停止的链接检查数。
1007	text column count	指定在一致性检查过程中发现的非 <code>null text/image</code> 列值的数量。
5000	bytes data	指定所检查的分区中存储的用户数据的数量（用字节表示）。
5001	bytes used	指定在所检查的分区中记录数据时使用的存储量（用字节表示）。 <code>bytes used</code> 和 <code>bytes data</code> 的差值表示，存储数据或对数据编制索引所需的开销量。
5002	pages used	指定链接到所检查的对象但实际上却用于保存该对象的页数。

<b>type_code</b>	<b>type_name</b>	<b>说明</b>
5003	pages reserved	指定为所检查对象保留但并未分配给该对象使用的页数。(8 * extents used) 和 (pages used + pages reserved) 的差值表示, 未提交的重新分配和未正确分配的页的总数。
5004	pages overhead	指定用于开销函数 (如 OAM 页或索引统计信息) 的页数。
5005	extents used	指定为所检查分区中的对象分配的扩充数。对于对象 99 (分配页), 该值是未分配给有效对象的扩充数。对象 99 包括未分配给其它对象的存储空间。
5006	count	指定在所检查对象特定部分的任何页上找到的组件项目 (行或键) 的数目。
5007	max count	指定在所检查对象特定部分的任何页上找到的组件项目的最大数量。
5008	max size	指定在所检查对象特定部分的任何页上找到的任何组件项目的最大大小。
5009	max level	指定索引中的最大级别数量。该数据类型不适用于表。
5010	pages misallocated	指定已分配给对象但未正确初始化的页数。它是故障计数器。
5011	io errors	指定遇到的 I/O 错误数。该数据类型是故障计数器。
5012	page format errors	指定所报告的页格式错误数。该数据类型是故障计数器。
5013	pages not allocated	指定通过其链链接到对象但未分配的页数。该数据类型是故障计数器。
5014	pages not referenced	指定已分配给对象但未通过其链到达的页数。该数据类型是故障计数器。
5015	overflow pages	指定遇到的溢出页数。该数据类型仅适用于聚簇索引。
5016	page gaps	指定未按升序序列链接到下一页的页数。该数字表示表段数量。
5017	page extent crosses	指定链接到其自身扩充之外的页的页数。当 page extent crosses 的数量随着 pages used 或 extents used 增加时, 大型 I/O 缓冲区的效率就会降低。
5018	page extent gaps	指定页扩展交叉数量, 其中后续扩充不是按升序序列排列的下一个扩充。当最大限度地减少 page extent gaps 数时, 可以在完全扫描上获得最佳 I/O 性能。对于每个间隔, 都可能执行搜索或全磁盘旋转。
5019	ws buffer crosses	指定在执行 dbcc checkstorage 操作的过程中, 链接到其工作空间缓冲区高速缓存之外的页数。该信息可用于调整高速缓存的大小, 从而在提供高性能的同时避免浪费资源。
5020	deleted rows	对象中已删除的行数。
5021	forwarded rows	对象中转移的行数。
5022	empty pages	已分配但不包含数据的页数。
5023	pages with garbage	可从碎片收集中受益的页数。
5024	non-contiguous free space	非连续可用空间的字节数。
10000	page id	指定当检测到故障时, 所检查的页在数据库中的位置。所有本地化的故障都包含该参数。

<b>type_code</b>	<b>type_name</b>	<b>说明</b>
10001	page header	指定当检测到故障时，所检查页标题的十六进制表示。该信息有助于评估软故障并确定页在检查后是否进行了更新。服务器将截断尾随零。
10002	text column id	指定一个 8 字节的十六进制值，以提供引用故障文本链的页、行和列的 ID。服务器将截断尾随零。
10003	object id	指定一个 9 字节的十六进制值，以提供所检查的页或分配的 object id（表）、相应的 partition id（表的分区）和 index id（索引）。 例如，如果某一页应属于表 T1（因为可通过 T1 链到达该页），而实际上却将该页分配给了表 T2。在这种情况下，将记录 T1 的 object id，同时还记录 T2 的 object id expected。服务器将截断尾随零。
10007	page id expected	指定在预期的页 ID 和实际遇到的页 ID 不一致时，链接页的预期页 ID。 例如，如果按照从 P1 到 P2 的顺序沿着链前进，那么在返回时，P1 应出现在 P2 之后。page id expected 的值应为 P1，page id 的值为 P2。当遇到实际值 P3 时，会将该值记录为 page id actual。
10008	page id actual	当遇到的页 ID 和预期的页 ID 不一致时，该值指定遇到的实际页 ID。（另请参见 type_code 10007。） 例如，如果按照从 P1 到 P2 的顺序沿着链前进，那么在返回时，P1 应出现在 P2 之后。page id expected 的值应为 P1，page id 的值为 P2。当遇到实际值 P3 时，会将该值记录为 page id actual。
10009	object id expected	指定一个 9 字节的十六进制值，以提供所检查页或分配的预期对象 ID（表）、适用的分区 ID（表的分区）和索引 ID（索引）。 例如，如果某一页应属于表 T1（因为可通过 T1 链到达该页），而实际上却将该页分配给了表 T2。在这种情况下，将记录 T1 的 object id，同时还记录 T2 的 object id expected。服务器将截断尾随零。
10010	data-only locked data page header	表示故障所在页的 44 字节的页标题。
10011	data-only locked b-tree leaf page header	表示故障所在页的 44 字节的页标题。
10012	data-only locked b-tree header	表示故障所在页的 44 字节的页标题。
20001	rerun checkstorage reco	重新运行 checkstorage。
20002	indexalloc reco	运行带 fix 选项的 dbcc indexalloc。
20003	tablealloc reco	运行带 fix 选项的 dbcc tablealloc。
20004	checktable fix_spacebits reco	运行带 fix_spacebits 选项的 dbcc tablealloc。
20005	checktable reco	运行 dbcc checktable。
20006	reorg reco	运行 reorg 命令
20007	no action reco	该故障是无害的，无需采取措施。
30000	drop object reco	删除对象并重新创建它。

<b>type_code</b>	<b>type_name</b>	<b>说明</b>
30001	bulk copy reco	将数据批量复制出来，然后再批量复制回去。
40000	check logs for hardware failure reco	检查操作系统日志，并纠正包含 Sybase 设备的磁盘上所有已报告的硬件问题。
40001	checkalloc reco	运行带 fix 选项的 dbcc checkalloc。
40002	reload db reco	使用干净备份重装数据库。
100000	IO error	表示无法从设备中读取已标识页的一部分。这通常是由操作系统或硬件故障造成的。
100001	page id error	表示在页上记录的标识 ID（页码）无效。这可能是由于以下原因造成的：将页写入错误的磁盘位置或从错误的磁盘位置读取页；写入页之前或写入页时发生页损坏；执行页分配后没有对该页进行初始化。
100002	page free offset error	表示页上的数据结尾无效。该事件影响对该页的插入和更新。它还可能影响对该页数据的某些访问。
100003	page object id error	表示看来将该页分配给了其它表，而不是预期的表。如果该故障一直存在，则可能是以下情况之一造成的： <ul style="list-style-type: none"> <li>• 页分配不正确，这可能只导致随后分配该页的效率下降；</li> <li>• 页链已损坏，这可能使受损链中的数据无法访问。</li> </ul>
100004	timestamp error	表示页的时间戳晚于数据库的时间戳。当对该页进行更改后，此错误可能会使恢复失败。
100005	wrong dbid error	表示数据库 ID dbid 存储在数据库分配页上。如果该 ID 不正确，分配页将损坏并且所有指定分配将变为可疑。
100006	wrong object error	表示页分配不一致。该页看来属于某一个表或索引，却被记录为分配给了分配页中的其它表或索引。该错误与 page object id error 的区别在于：分配是不一致的，但结果却是相似的。
100007	extent id error	表示为 dbcc checkstorage 不能识别的表或索引找到了分配。通常，这导致已分配的存储无法使用。
100008	fixed format error	表示该页错误地指出，它只包含具有单一固定长度的行。dbcc checkstorage 将报告此错误。虽然 dbcc checktable 不报告该错误，但会对其进行修复。
100009	row format error	表示该页上至少有一行的格式不正确。此错误可能导致该页上的部分或全部数据无法访问。
100010	row offset error	表示该页上至少有一行不在预期的页偏移处。此错误可能导致该页上的部分或全部数据无法访问。
100011	text pointer error	表示指向损坏的 text 或 image 数据的表行的位置。该信息有助于更正问题。
100012	wrong type error	表示页的格式不正确。例如，在索引或 text/image 列中找到了数据页。
100013	non-OAM error	此错误是 wrong type error 的一种特殊情形。在当前版本中，它并不作为单独情况来报告。

<b>type_code</b>	<b>type_name</b>	<b>说明</b>
100014	reused page error	表示通过多个链到达该页并且这些链属于不同的对象。此错误表示通过损坏的页链链接，非法地对页进行共享。它可能影响对其中一个表或两个表中的数据进行的访问。
100015	page loop error	表示在沿着对象的页链前进时第二次到达某一页，这说明页链中存在循环。当访问该对象中的数据时，可能会因循环而使会话无限期地挂起。
100016	OAM ring error	表示页已经分配，但无法通过对象的页链达到该页。通常，这导致已分配的存储空间无法使用。
100017	OAM ring error	表示 OAM 页环链接已经损坏。这可能不会影响到该对象数据的访问，但可能影响到该数据的插入、删除和更新。
100018	missing OAM error	表示 dbcc checkstorage 发现了在 OAM 中未记录的对象的分配。此错误表示发生了损坏，该损坏可能会影响将来的存储分配，但可能不会影响到当前存储数据的访问。
100019	extra OAM error	表示在 OAM 中记录了该对象的分配，但未在分配页中对该分配进行验证。此错误表示发生了损坏，该损坏可能会影响将来的存储分配，但可能不会影响到当前存储数据的访问。
100020	check aborted error	表示 dbcc checkstorage 停止对表或索引进行检查。为了防止出现多个故障报告，可能会停止单个链上的检查操作而不报告该错误。当某个对象包含多个页链时，如果一个链的检查操作失败，只要未超过故障阈值，就不会妨碍在其它链上继续执行检查操作。
100021	chain end error	表示链的结尾已损坏。作为一种软故障，它可能仅表示在执行 dbcc checkstorage 操作时，有相当多的页扩展或截断了该链。
100022	chain start error	表示链的开头已损坏或不在预期的位置。如果该故障一直存在，则可能影响到存储在对象中的数据的访问。
100023	used count error	表示在 OAM 页中记录的已用页数与通过检查分配页确定的已用页数之间不一致。
100024	unused count error	表示在 OAM 页中记录的保留但未使用的页数与通过检查分配页确定的保留但未使用的页数之间不一致。
100025	row count error	表示在 OAM 页中记录的行数与通过 dbcc checkstorage 确定的行数不一致。
100026	serialloc error	表示违反适用于日志分配的串行分配规则。
100027	text root error	表示 text 或 image 索引的根页格式不符合规定。该检查类似于 dbcc textalloc 执行的根页检查。
100028	page misplaced	表示在检查系统表时所预期的位置未找到该对象的页。这通常表示在以前的某个时间使用了 sp_placeobject。在 dbcc_counters 表中，将计算所有放错位置的页的总数，而不是按设备和分区进行报告。
100029	page header error	表示页标题中的内部不一致性，而不是其它类型代码所描述的不一致性种类。此错误的严重级取决于页的类型和已发现的不一致性。
100030	page format error	表示页正文中的内部不一致性，而不是其它类型代码所描述的不一致性种类。此错误的严重级取决于页的类型和已发现的不一致性。



<b>type_code</b>	<b>type_name</b>	<b>说明</b>
100031	page not allocated	表示 dbcc checkstorage 沿着页链达到了未分配的页。这种情况可能会影响对存储在该对象中的数据的数据的访问。
100032	page linkage error	表示 dbcc checkstorage 检测到链内部页的下一个或前一个链接出现故障。如果该故障一直存在，则可能影响对存储在对象中的数据的数据的访问。
100033	non-contiguous free-space error	表示页上非连续可用空间的值无效或不一致。
100034	insert free space error	表示页上连续可用空间的值无效或不一致。
100035	spacebits mismatch	表示在页完整指示器中出现不一致性。
100036	deleted row count error	表示页上已删除行计数的值无效或不一致。
100037	forwarded rows error	表示页上转移行指示器和转移行数目不一致。
100038	page header type error	表示没有正确地设置页标题格式指示器。
100039	incorrect extent oampage	未正确设置扩充 OAM 页引用
100040	OAM page format error	非 OAM 首页包含非零的 OAM 首页特定数据。



## 监控表

本章按字母顺序介绍各个监控表。

“属性”列提供了有关 Adaptive Server 如何管理列的信息。“属性”值有：

- “Counter”表示该列中的值可以归零，即变为零并重新开始计数，其原因是该值超过允许的最大值  $2^{31}$ 。在未使用 `noclear` 选项的情况下运行 `sp_sysmon` 时，Adaptive Server 会重新设置监控计数器。在 Adaptive Server 15.0.1 及更高版本中，在缺省情况下已包括 `noclear` 选项作为 `sp_sysmon` 参数。在 15.0.1 之前的版本中，必须指定 `noclear` 以防止 Adaptive Server 重新设置监控计数器。

如果在同一 Adaptive Server 上既运行 `sp_sysmon` 又使用监控表，重新设置监控计数器可能会导致结果出现偏差。

- “Null”表示列值可以为空。
- “Reset”表示当您以某种导致 `sp_sysmon` 清空监控计数器的方式运行它时，会重新设置该列。请参见 [Performance and Tuning Series: Monitoring Adaptive Server with sp\\_sysmon](#)（《性能和调优系列：使用 `sp_sysmon` 监控 Adaptive Server》）。

## monCachedObject

**说明** 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。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

**列** monCacheObject 的列有：

名称	数据类型	属性	说明
CacheID	int		高速缓存的唯一标识符。
InstanceID			<i>特定于集群环境</i> — 实例在共享磁盘集群中的 ID。
DBID	int		数据库的唯一标识符。
IndexID	int		索引的唯一标识符。
PartitionID	int		分区的唯一标识符。对于未分区的对象，此值的等效值是 ObjectID。
CachedKB	int		对象在高速缓存中占用的千字节数。
CacheName	varchar(30)	Null	高速缓存的名称。
ObjectID	int	Null	对象的唯一标识符。如果已从服务器的元数据高速缓存中删除对象的描述符，则为 Null。遇此情况时，可以在指定数据库的 <code>syspartitions</code> 中查询 <code>PartitionID</code> 的值，进而确定对象标识符。
DBName	varchar(30)	Null	数据库的名称；如果已从服务器的元数据高速缓存中删除对象的描述符，则为 NULL。
OwnerUserID	int	Null	对象所有者的唯一标识符。
OwnerName	varchar(30)	Null	对象所有者的名称；如果已从服务器的元数据高速缓存中删除对象的描述符，则为 NULL。
ObjectName	varchar(30)	Null	对象的名称；如果已从服务器的元数据高速缓存中删除对象的描述符，则为 NULL。
PartitionName	varchar(30)	Null	对象分区的名称；如果已从服务器的元数据高速缓存中删除对象的描述符，则为 NULL。
ObjectType	varchar(30)	Null	对象类型；如果对象不再为打开状态，则为 NULL。
TotalSizeKB	int	Counter、null	分区大小（以千字节为单位）。
ProcessesAccessing	int	Counter、null	当前正在访问数据高速缓存中该对象的页面的进程数。

## monCachePool

说明 存储为所有数据高速缓存分配的所有池的统计信息。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

列 monCachePool 的列有：

名称	数据类型	属性	说明
CacheID	int		高速缓存的唯一标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
IOBufferSize	int		池的 I/O 缓冲区的大小（以字节为单位）
AllocatedKB	int		分配给池的字节数
PhysicalReads	int	Counter	从磁盘读取到池中的缓冲区数
Stalls	int	Counter	I/O 操作由于该数据高速缓存的清洗区中没有可用的干净缓冲区而被延迟的次数
PagesTouched	int	Counter	当前在池内使用的页数
PagesRead	int	Counter	读取到池中的页数
BuffersToMRU	int	Counter	在最近使用最频繁的池部分中读取和替换的缓冲区数。
BuffersToLRU	int	Counter	在最近使用最不频繁的池部分中读取和替换的缓冲区数：获取和放弃。
CacheName	varchar(30)	Null	高速缓存的名称

## monCachedProcedures

**说明** 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。

---

**注释** 要使该监控表收集数据，请为其启用 `per object statistics active` 配置参数。

---

列 `monCacheProcedures` 的列有：

名称	数据类型	属性	说明
ObjectID	int		过程的唯一标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
OwnerUID	int		对象所有者的唯一标识符
DBID	int		对象所在数据库的唯一标识符
PlanID	int		过程高速缓存中对象的查询计划的唯一标识符
MemUsageKB	int		过程所用内存的千字节数
CompileDate	datetime		编译过程的日期
ObjectName	varchar(30)	Null	过程名
ObjectType	varchar(32)	Null	过程的类型（例如，存储过程或触发器）
OwnerName	varchar(30)	Null	对象所有者的名称
DBName	varchar(30)	Null	数据库的名称
RequestCnt	int4		从高速缓存中请求该过程的次数
TempdbRemapCnt	int4		针对临时数据库的 ID 重新映射该过程的次数。
AvgTempdbRemapTime	int4		重新映射临时数据库的 ID 的平均用时（以毫秒为单位）。

## monCachedStatement

### 说明

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的信息。在进行故障排除以及确定要在高速缓存中保留哪些语句时，这些信息可能非常有用。

`monCachedStatement` 中的列允许两个属性：“counter”（如果列具有计数器值）和“reset”（如果可以使用 `sp_sysmon` 重新设置列）。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 和 `enable statement cache monitoring` 配置参数，并将 `statement cache size` 参数设置为大于 0 的值。

### 列

`monCacheStatement` 的列有：

名称	数据类型	属性	说明
SSQLID	int		每个高速缓存语句的唯一标识符。该值被视为 <code>monCachedStatement</code> 的主键，在函数中使用。 <code>show_cached_text</code> 使用 SSQLID 来引用高速缓存中的单个语句。
HashKey	int		高速缓存语句 SQL 文本的散列值。散列键是基于语句的文本生成的，可以用作搜索其它监控表的近似键。
UserID	int		启动已高速缓存语句的用户的用户 ID。
SUserID	int		启动高速缓存语句的用户的服务器 ID。
DBID	int		从中高速缓存语句的数据库的数据库 ID。
DBName	varchar (30)	Null	从中高速缓存语句的数据库的名称。
CachedDate	datetime		第一次高速缓存语句时日期和时间的时戳。
LastUsedDate	datetime		上次使用高速缓存语句时日期和时间的时戳。将此信息与 <code>CachedDate</code> 结合使用，可确定该语句的使用频率以及高速缓存该语句是否有用。
CurrentUsageCount	int	Counter	高速缓存语句的并发用户数。
StatementSize	int		高速缓存语句的大小（以字节为单位）。
MaxUsageCount	int	Counter	高速缓存语句的文本的最大同时访问次数。
SessionSettings			这些会话级设置与每个高速缓存语句相关联。
ParallelDegree			为该语句存储的查询所用的并行度
QuotedIdentifier			指定是否启用通过 <code>set quoted_identifier</code> 编译的计划。
TransactionIsolationLevel			编译语句时所依照的事务隔离级别。

名称	数据类型	属性	说明
TransactionMode			指定是否为语句启用了“链式事务模式”。
SAAuthorization			指定是否通过 sa_role 授权来编译计划。
SystemCatalogUpdates			指定是否在编译计划时启用了 allow catalog updates。
ExecutionMetrics			<p>使用高速缓存的计划时会收集执行开销。这些开销是根据逻辑 I/O (LIO) 值、物理 I/O (PIO) 值、执行情况和占用的时间计算的。</p> <p>ExecutionMetrics 报告的指标与 QP 指标报告的指标相同，不过，此表中报告的数据不需要您启用 QP 指标。monCachedStatement 可独立捕获高速缓存语句的指标，而不管 Adaptive Server 的 metrics capture 设置如何。</p>
MetricsCount			为此语句集合指标的次数。
MaxElapsedTime	int		此语句占用的最长执行时间。
MinElapsedTime	int		此语句占用的最短执行时间。
AvgElapsedTime	int		此语句占用的平均执行时间。
MaxLIO	int		在此语句任一执行过程中发生的最大逻辑 I/O 数。
MinLIO	int		在此语句任一执行过程中发生的最小逻辑 I/O 数。
AvgLIO	int		在此语句执行过程中发生的平均逻辑 I/O 数。
MaxPIO	int		在此语句任一执行过程中发生的最大物理 I/O 数。
MinPIO	int		在此语句任一执行过程中发生的最大物理 I/O 数。
AvgPIO	int		在此语句执行过程中发生的平均物理 I/O 数。
NumRecompilesPlanFlushes	int	Counter	由于在高速缓存中未找到计划而对高速缓存语句进行重新编译的次数。
NumRecompilesSchemaChanges	int	Counter	<p>由于模式更改而对语句进行重新编译的次数。对表运行 update statistics 可能会更改最佳计划。此更改被视为最小的模式更改。</p> <p>对某语句执行多次重新编译表示高速缓存该特定语句不产生作用，您可能需要从语句高速缓存中删除该语句，以便为其它一些更为稳定的语句留出空间。</p>
MaxPlanSize	int		计划使用时的大小（以千字节为单位）。
MinPlanSize	int		计划不使用时的大小（以千字节为单位）。
LastRecompiledDate	datetime		上次由于模式更改或在语句高速缓存中未找到语句而重新编译语句时的日期。
UseCount	int		高速缓存某语句后访问它的次数。
HasAutoParams	boolean		如果语句中有任何参数化的文字，则为“true”；否则为“false”。



## monCIPC

说明

特定于集群环境—提供从当前实例或所有实例查看的有关集群内部所有消息传送的摘要数字。

如果已将系统视图设置为 `cluster`，则会在 `monCIPC` 表中为集群中的每个实例返回一行；否则会为对其执行查询的实例返回单行。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列

`monCIPC` 的列有：

名称	数据类型	属性	说明
InstanceID	tinyint		集群内实例的 ID
ReceiveCount	int		此实例接收到的消息数
TransmitCount	int		此实例发送的消息数
Multicast	int		发送给集群中所有其它实例的消息数
Synchronous	int		同步发送的这些消息数
ReceiveSoftError	int		在此实例上接收到的可恢复错误数
ReceiveHardError	int		在此实例上接收到的不可恢复错误数
TransmitsSoftError	int		此实例上可恢复的传输错误数
TransmitHardError	int		此实例上不可恢复的传输错误数
Retransmits	int		此实例所执行的重新传输数
Switches	int		在主互连网络和次级互连网络之间进行切换的次数
FailedSwitches	int		在主互连网络和次级互连网络之间尝试进行切换的失败次数
RegularBuffersInUse	int4		当前分配的 CIPC 常规缓冲池中缓冲区数。
FreeRegularBuffers	int4		CIPC 常规缓冲池中可用缓冲区数。
MaxRegularBuffersInUse	int4		在自服务器启动以来的任意时间分配的 CIPC 常规缓冲池中缓冲区的最大数目。
LargeBuffersInUse	int4		当前分配的 CIPC 大型缓冲池中的缓冲区数。
FreeLargeBuffers	int4		CIPC 大型缓冲池中的可用缓冲区数。
MaxLargeBuffersInUse	int4		在自服务器启动以来的任意时间分配的 CIPC 大型缓冲池中缓冲区的最大数目。

## monCIPCEndpoints

**说明** 特定于集群环境 — 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。

为实例中的每个逻辑结束点返回一行。如果将系统视图设为 `cluster`，则会为集群中的每个节点返回一组行。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列 monCIPCEndpoints 的列有：

名称	数据类型	属性	说明
InstanceID	tinyint		集群内实例的 ID
ReceiveCount	int		此逻辑结束点在集群内接收到的消息数
TransmitCount	int		此逻辑结束点在实例内所发送的消息数
ReceiveBytes	int		此逻辑结束点在实例内接收到的字节数
TransmitBytes	int		此逻辑结束点在实例内所发送的字节数
ReceiveQ	int		排队等待此逻辑结束点的当前消息数
MaxReceiveQ	int		已观察到排队等待此逻辑结束点的最大消息数
DoneQ	int		此逻辑结束点的已经过处理并等待进一步操作的当前消息数
MaxDoneQ	int		所观察到此逻辑结束点的已经过处理并等待进一步操作的最大消息数
MaxRecvQTime	real4		某条消息在当前逻辑结束点队列中花费的最长时间（以毫秒为单位）。
AvgRecvQTime	real4		某条消息在当前逻辑结束点队列中花费的平均时间（以毫秒为单位）。
EndPoint	varchar		CIPC 结束点的名称

## monCIPCLinks

说明

特定于集群环境— 监控集群中实例间链接的状态。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列

monCIPCLinks 的列有：

名称	数据类型	属性	说明
InstanceID	tinyint		集群内实例的 ID。
LocalInterface	varchar30		链接的本地网络结束点的名称。作为服务器名称出现在 <i>hosts</i> 文件中的同一名称。
RemoteInterface	varchar30		链接的远程结束点的名称。作为服务器名称出现在 <i>hosts</i> 文件中的同一名称。
PassiveState	varchar10		链路上的通信中列出的最新状态。
PassiveStateAge	int		自 <b>PassiveState</b> 列更新后的时间，以毫秒为单位。
ActiveState	varchar10		所使用的最新状态，由活动监控确定（当链路上没有通信时）。
ActiveStateAge	int		自 <b>ActiveState</b> 列更新后的时间，以毫秒为单位。

## monCIPCMesh

### 说明

特定于集群环境—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。

四个连接中的每个连接都会向集群中每个其它节点返回一行，直到满足所配置的最大数目。如果将系统视图设为 **cluster**，则会为集群中每个活动实例返回一组行。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列 monCIPCMesh 的列有：

名称	数据类型	属性	说明
InstanceID	tinyint		集群内实例的 ID。
FarInstanceID	tinyint		集群中最末端实例的实例编号。
Received	int		此实例从 FarInstanceID 实例接收到的消息数。
Dropped	int		FarInstanceID 实例中由于资源不足而删除的消息数。
Transmitted	int		传输到 FarInstanceID 实例的消息数。
Resent	int		重新发送到 FarInstanceID 实例的消息数。
Retry	int		重新尝试发送到 FarInstanceID 实例的包数。
ControlRx	int		InstanceID 实例接收到的控制消息数。
ControlTx	int		此网络的 InstanceID 实例所发送的控制消息数。
SendQ	int		等待发送到此网络的 FarInstanceID 实例的当前消息数。
MaxSendQ	int		启动 InstanceID 实例以来此网络发送队列中的最大包数。
SentQ	int		由 InstanceID 实例发送给 FarInstanceID 实例但尚未经 FarInstanceID 实例确认的包数。
MaxSentQ	int		已发送但发送通知尚未被处理的最大消息数。
MaxSendQTime	real		处理此网络发送队列中的一条消息所需的最长时间。以毫秒为单位。
AvgSendQTime	real		处理此网络发送队列中的一条消息所需的平均时间。以毫秒为单位。
Mesh	varchar		连接的通道名称。以下状态之一： <ul style="list-style-type: none"> <li>• 带外</li> <li>• 消息</li> <li>• 大型消息</li> <li>• 直接内存访问 (DMA)</li> </ul>
MinRTT	int		观察到的消息的最小往返延迟（仅适用于用户数据报协议 (UDP) 传输）。
MaxRTT	int		观察到的消息的最大往返延迟（仅适用于 UDP 传输）。
AverageRTT	int		观察到的消息的平均往返延迟（仅适用于 UDP 传输）。

## monCLMObjectActivity

**说明** 特定于集群环境—收集集群锁信息。 monCLMObjectActivity:

- 仅跟踪 master 数据库和用户数据库中对象的活动。
- 在分区级别跟踪物理锁活动。

数据库的集群对象锁的 Object-PartitionID 为 0。

**注释** 此监控表不需要参数即可收集数据。

**列** monCLMObjectActivity 包含以下这些列:

列名	类型	属性	说明
InstanceID	int1		实例 ID。
DBID	int4		数据库 ID。
Object_PartitionID	int4		提出锁请求的对象的标识。
LockRequests	int4		集群锁请求数。
LocalMaster	int4		锁请求查找即将成为锁所有者的当前实例的次数。 集群中的一个实例将成为“锁所有者”。某个实例在需要集群锁时，会联系锁所有者以获取该锁。
Waited	int4		在具有远程实例争用情况下授予的锁请求数。
Granted	int4		在没有远程实例争用情况下授予的锁请求数。
RWConflictWaited	int4		由于向远程实例上的任务授予的读写冲突锁而等待的锁请求数。
AvgRWConflictWaitTime	flt4		执行 RWConflictWaited 所描述的等待所花费的平均时间。
MaxRWConflictWaitTime	flt4		执行 RWConflictWaited 所描述的等待所花费的最长时间。
WWConflictWaited	int4		由于向远程实例上的任务授予的写入 - 写入冲突锁而等待的锁请求数。
AvgWWConflictWaitTime	flt4		执行 WWConflictWaited 所描述的等待所花费的平均时间。
MaxWWConflictWaitTime	flt4		执行 WWConflictWaited 中所描述的等待所花费的最长时间。
ClusterMsgWaits	int4		由于集群消息传送而累积的等待数。
AvgClusterMsgWaitTime	flt4		由于集群消息传送而累积的平均等待时间。
MaxClusterMsgWaitTime	flt4		由于集群消息传送而累积的最长等待时间。

列名	类型	属性	说明
DowngradeReqRecv	int4		在集群锁所有者处接收到的降级服务请求数。
DowngradeReqRecvWithNoBlocker	int4		在集群锁所有者没有任何阻塞任务所有权的情况下接收到的降级服务请求数。
ClusterDeadlock	int4		多个实例在尝试同时获取同一集群锁时所引发的死锁数。
Locktype	varchar (20)		锁的类型。

## monClusterCacheManager

说明

特定于集群环境— 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。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列

monClusterCacheManager 的列有：

名称	数据类型	属性	说明
InstanceID	tinyint		集群内实例的 ID
RequestsQueued	int		排队等待集群高速缓存管理器守护程序的请求数
RequestsRequeued	int		重新排队等待集群高速缓存管理器守护程序的请求数
RequestsServiced	int		集群高速缓存管理器守护程序所支持的请求数
DiskWrites	int		集群高速缓存管理器守护程序所启动的磁盘写入数
SleepCount	int		集群高速缓存管理器守护程序进入休眠状态的次数
DaemonName	varchar		集群高速缓存管理器守护程序的名称
TransfersInitiated	int		集群高速缓存管理器守护程序所启动的传输数
Downgrades	int		集群高速缓存管理器守护程序所执行的降级数
Releases	int		集群高速缓存管理器守护程序所执行的释放数
AvgServiceTime	int		处理某个请求所花费的平均时间（以毫秒为单位）
MaxQSize	int		在自实例启动后的任何时间排队等待集群高速缓存管理器守护程序的最大请求数

## monCMSFailover

### 说明

特定于集群环境—跟踪集群成员资格服务 (CMS) 检测到故障的时间、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。每个实例都有一行。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

### 列

monCMSFailover 包含以下这些列：

列名	类型	属性	说明
InstanceID	tinyint1		执行故障切换的实例。
FailedInstanceID	varchar(96)		由逗号分隔的失败实例 ID 的列表。
FailDetectTime	datetime(8)		检测到集群故障的时间。
InitViewTime	datetime(8)		获取初始集群视图的时间。
FinalViewTime	datetime(8)		获取最终集群视图的时间。
ResynchHBTime	datetime(8)		重新同步集群范围心跳的时间。
NotifyFailTime	datetime(8)		发布失败事件的时间。
EventdoneTime	datetime(8)		完成最后一次失败事件的时间。



## monDataCache

说明 存储有关 Adaptive Server 数据高速缓存的统计信息。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

列 monDataCache 的列有：

名称	数据类型	属性	说明
CacheID	int		高速缓存的唯一标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
RelaxedReplacement	int		指定高速缓存是否正在使用宽松的高速缓存替换策略
BufferPools	int		高速缓存内的缓冲池数
CacheSearches	int	Counter、reset	定向到高速缓存的高速缓存搜索
PhysicalReads	int	Counter、reset	从磁盘读取到高速缓存中的缓冲区数
LogicalReads	int	Counter、reset	从高速缓存检索的缓冲区数
PhysicalWrites	int	Counter、reset	从高速缓存写入到磁盘的缓冲区数
Stalls	int	Counter、reset	I/O 操作由于清洗区中没有可用的干净缓冲区而被延迟的次数
CachePartitions	smallint		当前为高速缓存配置的分区数
CacheName	varchar(30)	Null	高速缓存名称

# monDBRecovery

**说明** 特定于集群环境 — 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。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列 monDBRecovery 的列有：

列名	类型	属性	说明
DBID	int4		数据库的唯一标识符
InstanceID	int1		执行恢复的实例（仅适用于 Cluster Edition）
MaxOpenXacts	int4		在恢复期间查看的最大已打开事务数
MaxPFTSEntries	int4		在恢复期间查看的最大 PFTS 条目数
Buckets	int4		表元的数量
LogBTotPages	int4		日志边界确定过程中的 log scan getpage 请求数。
LogBTotAPFWaited	int4		日志边界确定过程中找到正在进行的 I/O 的 log scan getpage 请求数
LogBTotIO	int4		日志边界确定过程中包含物理 I/O 的 log scan getpage 请求数
AnITotRec	int4		将由恢复进程扫描的日志记录的总数
AniPhase1Recs	int4		恢复进程第 1 阶段中的日志记录数
AniPhase1RedoRecs	int4		恢复进程第 1 阶段中要重做的日志记录数
AniPhase2Recs	int4		恢复进程第 2 阶段中的日志记录数
AniPhase2RedoRecs	int4		恢复进程第 2 阶段中要重做的日志记录数
AnITotPages	int4		分析过程中的 log scan getpage 请求数
AnITotAPFWaited	int4		分析过程中找到正在进行的 I/O 的 log scan getpage 请求数
AnITotIO	int4		分析过程中包含物理 I/O 的 log scan getpage 请求数
RedoOps	int4		考虑进行重做的总操作数
RedoOpsNotRedonePFTS	int4		不需要重做的操作（PFTS 检查）
RedoOpsRedonePFTS	int4		可能需要重做的操作（PFTS 检查）
RedoOpsRedoneTS	int4		需要重做的操作（时间戳检查）
RedoOpsNotRedoneTS	int4		不需要重做的操作（时间戳检查）
RedoLogTotPages	int4		重做过程中的 log scan getpage 请求数
RedoLogTotAPFWaited	int4		重做过程中找到正在进行的 I/O 的 log scan getpage 请求数
RedoLogTotIO	int4		重做过程中包含物理 I/O 的 log scan getpage 请求数

列名	类型	属性	说明
RedoRecTotPage	int4		重做过程中的 recovery pages getpage 请求数
RedoRecTotAPFWaited	int4		重做过程中找到正在进行的 I/O 的 recovery pages getpage 请求数
RedoRecTotIO	int4		重做过程中包含正在进行的物理 I/O 的 recovery pages getpage 请求数
UndoRecsUndone	int4		撤消的日志记录数
UndoLogTotPages	int4		撤消过程中的 log scan getpage 请求数
UndoLogTotAPFWaited	int4		撤消过程中找到正在进行的 I/O 的 log scan getpage 请求数
UndoLogTotIO	int4		撤消过程中包含物理 I/O 的 log scan getpage 请求数
UndoRecTotPages	int4		撤消过程中的 recovery pages getpage 请求数
UndoRecTotAPFWaited	int4		撤消过程中找到正在进行的 I/O 的 recovery pages getpage 请求数
UndoRedTotIO	int4		撤消过程中包含物理 I/O 的 recovery pages getpage 请求数
DBName	varchar(30)		数据库的名称
FailedInstanceID	int1		失败实例的 ID (仅适用于 Cluster Edition)
Command	varchar(30)		由正在运行恢复的进程执行的 load database、load transaction、online database、mount database 以及开始或故障切换命令中的一个
RecType	varchar(30)		恢复类型 — server start、load database、load transaction 或 node failover 之一
LobBStartTime	datetime(8)		日志边界确定过程的开始时间
LogBEndTime	datetime(8)		日志边界确定过程的结束时间
AnlStartTime	datetime(8)		分析过程的开始时间
AnlEndTime	datetime(8)		分析过程的结束时间
RedoStartTime	datetime(8)		重做过程的开始时间
RedoEndTime	datetime(8)		重做过程的结束时间
UndoStartTime	datetime(8)		撤消过程的开始时间
UndoEndTime	datetime(8)		撤消过程的结束时间

## monDBRecoveryLRTypes

### 说明

特定于集群环境—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相应的一行。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

### 列

monDBRecoveryLRTypes 包含以下这些列：

列名	类型	属性	说明
DBID	int4		数据库的唯一标识符
InstanceID	int1		执行恢复的实例（仅适用于 Cluster Edition）。
NumRecs	int4		恢复期间按类型查看的记录数
LogRecType	varchar(30)		日志记录类型

monDBRecoveryLRTypes 不需要参数。

## monDeadLock

**说明** 提供有关死锁的信息。使用 `deadlock pipe max messages` 可以调整返回消息的最大数。

`monDeadLock` 是一个历史监控表。请参见 Performance and Tuning: Monitoring Tables (《性能和调优: 监控表》)。

使用 `sp_monitor 'deadlock'` 可以检查当前死锁选项。`deadlock` 参数提供了许多基于 `monDeadLock` 的报告, 用于分析服务器死锁的历史记录。

---

**注释** 要使该监控表收集数据, 请为其启用 `enable monitoring`、`deadlock pipe active` 和 `deadlock max pipe messages` 配置参数。

---

列 `monDeadLock` 的列有:

名称	数据类型	属性	说明
DeadLockID	int		死锁的唯一标识符
VictimKPID	int		死锁牺牲品进程的内核进程 ID (kpid)
InstanceID	int		实例在共享磁盘集群中的 ID。
ResolveTime	datetime		解决死锁的时间
ObjectDBID	int		对象所驻留的数据库的唯一数据库标识符
PageNumber	int		为锁请求的页码 (如果适用)
RowNumber	int		为锁请求的行号 (如果适用)
HeldFamilyId	smallint		持有锁的父进程的 spid
HeldSPID	smallint		持有锁的进程的 spid
HeldKPID	int		持有锁的进程的 kpid
HeldProcDBID	int		导致锁被持有的存储过程所驻留的数据库的唯一标识符 (如果适用)
HeldProcedureID	int		导致锁被持有的存储过程的唯一对象标识符 (如果适用)
HeldBatchID	int		出现死锁时, 持有锁的进程所执行的 SQL 批处理的标识符
HeldContextID	int		持有锁的进程在它被另一进程阻塞时 (不是在它获得锁时) 的唯一环境标识符
HeldLineNumber	int		持有锁的进程在它被另一进程阻塞时 (不是获得锁时) 执行的成批语句内的行号
WaitFamilyId	smallint		等待锁的父进程的 spid
WaitSPID	smallint		等待锁的进程的 spid
WaitKPID	int		等待锁的进程的 kpid
WaitTime	int		在解决死锁之前等待进程被阻塞的时间 (以毫秒为单位)
ObjectName	varchar(30)	Null	对象名
HeldUserName	varchar(30)	Null	为其持有锁的用户的名称

名称	数据类型	属性	说明
HeldAppName	varchar(30)	Null	持有锁的应用程序的名称
HeldTranName	varchar(255)	Null	在其中获得锁的事务的名称
HeldLockType	varchar(20)	Null	被持有的锁的类型
HeldCommand	varchar(30)		进程被阻塞时所执行的进程或命令的类别
WaitUserName	varchar(30)	Null	为其请求锁的用户的名称
WaitLockType	varchar(20)	Null	所请求的锁的类型
HeldSourceCodeID	varchar(30)		仅供内部使用。
WaitSourceCodeID	varchar(30)		仅供内部使用。

## monDeviceIO

说明

返回与数据库设备上的活动有关的统计信息。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

列

monDeviceIO 的列有：

名称	数据类型	属性	说明
InstanceID	int		特定于集群环境—实例在共享磁盘集群中的 ID。
Reads	int	Counter、reset	从设备读取的次数
APFReads	int	Counter、reset	从设备进行异步预取 (APF) 读取的次数
Writes	int	Counter、reset	写入到设备的次数
DevSemaphoreRequests	int	Counter、reset	向镜像设备（如果已镜像）发出的 I/O 请求的次数
DevSemaphoreWaits	int	Counter、reset	被强制等待向镜像设备（如果已镜像）发出的 I/O 请求完成同步的任务数
IOTime	int	Counter	等待满足 I/O 请求所用的总时间（以毫秒为单位）
LogicalName	varchar(30)	Null	设备的逻辑名
PhysicalName	varchar(128)	Null	设备的完整层次文件名

# monEngine

说明

提供有关 Adaptive Server 引擎的统计信息。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

列 monEngine 的列有：

名称	数据类型	属性	说明
EngineNumber	smallint		引擎数。
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
CurrentKPID	smallint		当前执行的进程的内核进程标识符 (kpid)。
PreviousKPID	int		以前执行的进程的 kpid。
CPUTime	int	Counter、reset	引擎已运行的总时间（以秒为单位）。
SystemCPUTime	int	Counter、reset	引擎执行系统数据库服务的时间（以秒为单位）。
UserCPUTime	int	Counter、reset	引擎执行用户命令的时间（以秒为单位）。
IOCPUTime	int4		引擎等待系统完成发出的 IO 的时间（以秒为单位）。
IdleCPUTime	int	Counter、reset	引擎处于空闲旋转模式的时间（以秒为单位）。
Yields	int	Counter、reset	此引擎将控制权交给操作系统的次数。可使用 <code>runnable process search count</code> 修改空闲期间交出控制权的比率。
Connections	int	Counter	此引擎处理的连接数。
DiskIOChecks	int	Counter、reset	引擎检查是否存在异步磁盘 I/O 的次数。可使用 <code>i/o polling process count</code> 修改这些检查的频率。
DiskIOPolled	int	Counter、reset	引擎就以前未完成的异步磁盘 I/O 现在是否已完成而进行轮询的次数。每当磁盘 I/O 检查指示异步 I/O 已发布且尚未完成时，就会发生轮询。
DiskIOCompleted	int	Counter、reset	在引擎轮询未完成的异步磁盘 I/O 时已完成的异步磁盘 I/O 数。
MaxOutstandingIOs	int		由此引擎启动但尚未完成的当前 I/O 请求数。
ProcessesAffinitied	int		与此引擎关联的进程数。
ContextSwitches	int	Counter、reset	环境切换次数。
HkgcMaxQSize	int		Adaptive Server 可以让其排队等待在此引擎中进行管家碎片收集的最大项数。
HkgcPendingItems	int		尚待此引擎上的管家碎片收集器收集的项数。



名称	数据类型	属性	说明
HkgcHWMItems	int		在自服务器启动以来的任何时间排队等待管家碎片收集器的最大挂起项数。
HkgcOverflows	int		由于队列溢出而无法排队等待管家碎片收集器的项数。
Status	varchar(20)	Null	引擎的状态（联机、脱机等）。
Starttime	datetime	Null	引擎联机的日期。
StopTime	datetime		引擎脱机的日期。
AffinitiedToCPU	int	Null	引擎所密切连接的 CPU 数。
OSPID	int	Null	执行该引擎的操作系统进程的标识符。

## monErrorLog

### 说明

从 Adaptive Server 错误日志返回最新的错误消息。使用 `errorlog pipe max messages` 可以调整返回消息的最大数。请参见 *Performance and Tuning: Monitoring Tables*（《性能和调优：监控表》）。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring`、`errorlog pipe active` 和 `errorlog pipe max messages` 配置参数。

列 `monErrorLog` 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符 (spid)
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符 (kpid)
FamilyID	smallint		父进程的 spid
EngineNumber	smallint		进程在其上运行的引擎
ErrorNumber	int		错误消息号
Severity	int		错误的严重性
State	int		错误的状态
Time	datetime		出现错误时的时间戳
ErrorMessage	varchar(512)	Null	错误消息的文本

## monFailoverRecovery

### 说明

特定于集群环境 — 包含集群锁管理器 (CLM)、数据库恢复和集群成员资格服务 (CMS) 模块的已集合的故障切换恢复诊断信息。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列 `monFailoverRecovery` 包含以下这些列：

列名	类型	属性	说明
InstanceID	tinyint1		执行恢复的实例。
ModuleName	varchar(30)		模块名称。CML、CMS 或 Database 之一
FailedInstanceID	tinyint1		失败实例的 ID。
StartTime	datetime(8)		模块恢复的开始时间。
EndTime	datetime(8)		模块恢复的结束时间。

## monIOQueue

**说明** 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。

---

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

---

列 monIOQueue 的列有：

名称	数据类型	属性	说明
InstanceID	int		特定于集群环境—实例在共享磁盘集群中的 ID。
IOs	int	Counter	I/O 操作的总数
IOTime	int	Counter	等待满足 I/O 请求所用的时间（以毫秒为单位）
LogicalName	varchar(30)	Null	设备的逻辑名
IOType	varchar(12)	Null	用于对 I/O 进行分组的类别。UserData、UserLog、TempdbData、TempdbLog 之一。

## monLicense

说明 提供 Adaptive Server 当前检出的所有许可证的列表。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

列 monLicense 的列有：

名称	数据类型	属性	说明
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
Quantity	int		用于此功能的许可证的数量。
Name	varchar(30)	Null	功能许可证的名称。
Edition	varchar(30)	Null	其中此功能已获得许可的 Adaptive Server 版本。
Type	varchar(64)	Null	许可证类型。
Version	varchar(16)	Null	正在使用的功能许可证的版本
Status	varchar(30)	Null	此功能许可证的状态（即许可证是处于宽限期还是已过期）。
LicenseExpiry	datetime	Null	许可证过期日期（如果该许可证是会过期的许可证）。
GraceExpiry	datetime	Null	该许可证的过期日期（如果该许可证是在宽限模式下授予的）。请参见 <b>Status</b> 列以确定该许可证是否已被授予某个宽限期。
LicenseID	varchar(150)	Null	许可证标识符。如果许可证已被授予某个宽限期，则可能不会提供此标识符。
Filter	varchar(14)	Null	选择此功能许可证时所使用的过滤器。使用 <code>sp_lmconfig</code> 可以更改过滤器。
Attributes	varchar(64)	Null	许可证属性。这些属性是一些“名称=值”对，在被指定时用于限制 Adaptive Server 的某些特性。可能的限制符如下： <ul style="list-style-type: none"> <li>• ME = 最大引擎数</li> <li>• MC = 最大连接数</li> <li>• MS = 最大磁盘空间数</li> <li>• MM = 最大内存数</li> <li>• CP = 最大 CPU 数</li> </ul>

---

**注释** monLicense 不要求 mon\_role 权限；任何用户均可使用它。

---

## monLocks

说明

返回已授予的锁和挂起的锁请求的列表。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

列

monLocks 的列有：

名称	数据类型	属性	说明
SPID	smallint		持有或请求该锁的进程的会话进程标识符。
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符
DBID	int		此数据库对象的唯一标识符。
ParentSPID	smallint		父进程 ID。
LockID	int		锁对象 ID。
Context	int		锁定环境（位字段）。这些值与 <code>syslocks</code> 中 <code>context</code> 列的那些值相同。请参见《参考手册：表》以了解有关 <code>syslocks</code> 的信息。
DBName	varchar(30)		被锁定对象所在数据库的名称。查询 <code>monLocks</code> 时，如果数据库尚未打开，则此列为 NULL。
ObjectID	int	Null	对象的唯一标识符
LockState	varchar(20)	Null	表示是否已授予锁。其值包括： <ul style="list-style-type: none"> <li>• Granted</li> <li>• Requested</li> </ul>
LockType	varchar(20)	Null	锁的类型。其值包括： <ul style="list-style-type: none"> <li>• Exclusive</li> <li>• Shared</li> <li>• Update</li> </ul>
LockLevel	varchar(30)	Null	为其请求锁的对象的类型。其值包括： <ul style="list-style-type: none"> <li>• Row</li> <li>• 页码</li> <li>• 表</li> <li>• Address</li> </ul>
WaitTime	int	Null	未授予锁请求的时间（以秒为单位）。
PageNumber	int	Null	<code>LockLevel = 'PAGE'</code> 时锁定的页
RowNumber	int	Null	<code>LockLevel = 'ROW'</code> 时锁定的行
BlockedBy	int		如果锁请求被阻塞，则 <code>BlockedBy</code> 列为持有锁的进程的会话进程标识符，该锁会阻塞此锁请求。如果请求未阻塞，则为 Null。

名称	数据类型	属性	说明
BlockedState	varchar(64)		如果持有的锁阻塞了其它锁请求或者锁请求被阻塞，则为锁状态。其值包括： <ul style="list-style-type: none"><li>• Blocked</li><li>• Blocking</li><li>• Demand</li><li>• Detached</li><li>• Null（如果不存在阻塞条件）</li></ul>
SourceCodeID	varchar(30)		仅供内部使用。

## monLogicalCluster

说明

特定于集群环境— 显示有关系统上当前配置的逻辑集群的信息。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列

monLogicalCluster 的列有：

名称	数据类型	属性	说明
LCID	int		逻辑集群 ID。
Attributes	int		逻辑集群属性的位屏蔽。
ActiveConnections	int		使用此逻辑集群的活动连接数。
BaseInstances	tinyint		配置为此逻辑集群的基本实例的实例数。
ActiveBaseInstances	tinyint		此逻辑集群当前在其上处于活动状态的基本实例数。
FailoverInstances	tinyint		配置为此逻辑集群的故障切换实例的实例数。
ActiveFailoverInstances	tinyint		此逻辑集群当前在其上处于活动状态的故障切换实例数。
Name	varchar(30)		逻辑集群名称。
State	varchar(20)		当前状态。以下状态之一： <ul style="list-style-type: none"> <li>• Online</li> <li>• Offline</li> <li>• Failed</li> <li>• Inactive</li> <li>• Time_wait</li> </ul>
DownRoutingMode	varchar(20)		已关闭路由模式设置。以下状态之一： <ul style="list-style-type: none"> <li>• 系统</li> <li>• Open</li> <li>• 断开连接</li> </ul>
FailoverMode	varchar(20)		故障切换模式设置（instance 或 cluster）。
StartupMode	varchar(20)		启动模式设置（automatic 或 manual）。
SystemView	varchar(20)		系统视图设置（instance 或 cluster）。
Roles	varchar(20)		逗号分隔的此逻辑集群的特殊角色列表。“系统”逻辑集群始终具有系统角色。开放式逻辑集群具有“开放”角色。如果系统逻辑集群也具有开放角色，则该列的值为 system, open。没有任何特殊角色的逻辑集群会返回空值。
LoadProfile	varchar(30)		与此逻辑集群关联的负载配置文件。

名称	数据类型	属性	说明
ActionnRelease	varchar(20)		此逻辑集群的当前操作释放模式。其值包括： <ul style="list-style-type: none"><li>• Manual</li><li>• Automatic</li></ul> “Manual”表示用户必须执行操作释放命令才能释放此集群的操作。
Gather	varchar(30)		表示是否已将此逻辑集群配置为自动收集与此逻辑集群的可路由连接。其值包括： <ul style="list-style-type: none"><li>• Manual</li><li>• Automatic</li></ul>



## monLogicalClusterAction

**说明** 特定于集群环境— 显示从启动到释放这些操作的时间内对本地集群执行的所有管理操作。

monLogicalClusterAction 无需任何配置参数即可收集数据。

**列** monLogicalClusterAction 的列有：

名称	数据类型	属性	说明
Handle	int		用于取消此操作的唯一句柄。
State	varchar(20)		操作的状态：active、complete 或 canceled。
LCID	int		此操作应用到的逻辑集群 ID。
LogicalClusterName	varchar(30)		此逻辑集群（非规范化以减少连接）的逻辑集群名称。
Action	varchar(15)		所执行的操作。运行的命令及其作用范围的组合。例如，offline instance 或 failover cluster。
FromInstances	varchar(96)		此命令及操作的 from instances（要脱机的实例）的逗号分隔列表。
ToInstances	varchar(96)		此命令及操作的 to instances（要联机的实例）的逗号分隔列表。
InstancesWaiting	int		等待脱机的实例数（即处于 time_wait 状态的 FromInstances 的计数）。
WaitType	varchar(20)		此操作的当前等待状态。以下状态之一：wait、until 或 nowait。
StartTime	datetime		发出命令的日期和时间。
Deadline	datetime		必须完成命令的日期和时间（基于提供给 wait 或 until 选项的时间值）。
CompleteTime	datetime		命令和操作的完成日期和时间（当 InstancesWaiting 为零且操作从 active 状态转为 complete 状态时）。对于未完成的操作，将返回 NULL。
ConnectionsRemaining	int		执行此命令后剩下的要移动的连接数。
NonMigConnections	int		由于不支持迁移协议而将被终止的连接数。
NonHACConnections	int		不支持高可用性故障切换协议的连接数。这些连接已断开，不能在命令完成时进行故障切换。

## monLogicalClusterInstance

**说明** 特定于集群环境 — 显示有关实例与逻辑集群之间的多对多关系的信息。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

**列** monLogicalClusterInstance 的列有：

名称	数据类型	属性	说明
LCID	int		逻辑集群 ID
LogicalClusterName	varchar(30)		逻辑集群名称
InstanceID	tinyint		集群内实例的 ID
InstanceName	varchar(30)		实例名称
Type	varchar(20)		实例类型
FailoverGroup	tinyint		此实例（仅限故障切换实例）所属的故障切换组
State	varchar(20)		此实例（就逻辑集群而言）的状态
ActiveConnections	int		此实例上该逻辑集群的活动连接数
NonMigConnections	int		不支持连接迁移协议的活动连接数
NonHAConnections	int		不支持高可用性故障切换协议的活动连接数
LoadScore	real		此实例在使用与其逻辑集群关联的负载配置文件时的工作负载得分

## monLogicalClusterRoute

说明

特定于集群环境—显示有关已配置的路由（应用程序、登录和别名绑定）的信息。您不需要具有 `mon_role` 角色即可查询该监控表。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

列

monLogicalClusterRoute 的列有：

名称	数据类型	属性	说明
LCID	int		逻辑集群 ID
LogicalClusterName	varchar(30)		逻辑集群名称
RouteType	varchar(20)		路由类型。以下状态之一：application、login 或 alias
RouteKey	varchar(30)		与此路由关联的应用程序名称、登录名或别名。

## monNetworkIO

**说明** 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。

---

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

---

列 monNetworkIO 的列有：

名称	数据类型	属性	说明
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
PacketsSent	int	Counter、reset	发送出的包数量
PacketsReceived	int	Counter、reset	接收到的包数量
BytesSent	int	Counter、reset	发送出的字节数
BytesReceived	int	Counter、reset	接收到的字节数

## monOpenDatabases

**说明** 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。

如果 `number of open databases` 值过低，Adaptive Server 可能会从元数据高速缓存中刷新数据库描述符。如果发生这种情况，Adaptive Server 将失去数据库统计信息，但这些统计信息会在下一次将数据库描述符安装在元数据高速缓存中时重新初始化。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

列 monOpenDatabases 的列有：

名称	数据类型	属性	说明
DBID	int		数据库的唯一标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
BackupInProgress	int		指明数据库的备份当前是否在进行中
LastBackupFailed	int		指明数据库的上次备份是否失败
TransactionLogFull	int		指明数据库事务日志是否已满
AppendLogRequests	int	Counter	尝试附加到数据库事务日志时的信号请求数
AppendLogWaits	int	Counter	任务必须等待授予附加日志信号的次数
DBName	varchar(30)	Null	数据库的名称
BackupStartTime	datetime	Null	上一次完整数据库备份的开始日期
SuspendedProcesses	int	Null	由于数据库事务日志已满而导致的当前挂起的进程数
QuiesceTag	varchar(30)	Null	此数据库 <code>quiesce database</code> 命令中所使用的标记（如果该数据库处于受抑制状态）
LastCheckpointTime	datetime	Null	上次为此数据库运行检查点操作的日期和时间
LastTranLogDumpTime	datetime	Null	此数据库最近成功完成事务日志转储的日期和时间

## monOpenObjectActivity

说明 提供有关所有打开的表和索引的统计信息。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring`、`object lockwait timing` 和 `per object statistics active` 配置参数。

列 monOpenObjectActivity 的列有：

名称	数据类型	属性	说明
DBID	int		数据库的唯一标识符。
ObjectID	int		对象的唯一标识符。
IndexID	int		索引的唯一标识符。
InstanceID	int		<i>特定于集群环境</i> — 实例的唯一标识符。
DBName	varchar(30)	Null	对象所在的数据库的名称
ObjectName	varchar(30)	Null	对象名。
LogicalReads	int	Counter、 null	在不需要从磁盘读取的情况下从缓冲区高速缓存检索此对象缓冲区的总次数。
PhysicalReads	int	Counter、 null	从磁盘读取的缓冲区数。
APFReads	int	Counter、 null	从磁盘读取的 APF 缓冲区数。
PagesRead	int	Counter、 null	读取的总页数。
PhysicalWrites	int	Counter、 null	写入磁盘的缓冲区总数。
PagesWritten	int	Counter、 null	写入磁盘的总页数。
RowsInserted	int	Counter、 null	插入的行数。
RowsDeleted	int	Counter、 null	已删除的行数。
RowsUpdated	int	Counter、 null	更新的次数。
Operations	int	Counter、 null	对象被访问的次数。
LockRequests	int	Counter、 null	请求对象上的锁的次数。
LockWaits	int	Counter、 null	任务等待对象锁的次数。

名称	数据类型	属性	说明
OptSelectCount	int	Counter、 null	优化程序已选择要在查询计划中使用的此索引的次数。
LastOptSelectDate	datetime	Null	最近一次在编译过程中为计划选择索引的日期。
UsedCount	int	Counter、 null	对象在执行过程中用于某个计划的次数。
LastUsedDate	datetime	Null	最近一次在执行过程中将索引用于某个计划的日期。
HkgcRequests			排队等待某个对象的事件总数。该值大则意味着系统为指定对象生成了大量碎片。
HkgcPending			对象的挂起事件数。如果该值很大，则说明有很多碎片有待收集，尽管管家将会清理这些碎片。如果重新启动 Adaptive Server，则管家队列中的所有条目都将在重新启动 Adaptive Server 时丢失，并且这些页的碎片也不会被收集。
HkgcOverflows			溢出对象事件数。该值大则意味着管家队列已排满。由于管家无法调度作业，因此不会清除生成的碎片。
PhysicalLocks	int		特定于集群环境 — 每个对象所请求的物理锁数。
PhysicalLocksRetained	int		特定于集群环境 — 所保留的物理锁数。用于标识每个对象的锁命中率。命中率高则意味着此对象的分区十分平衡。
PhysicalLocksRetainWaitd	int4		特定于集群环境 — 在保留锁之前等待的物理锁请求数。
PhysicalLocksDeadlocks	int		特定于集群环境 — 请求的物理锁返回死锁的次数。在获取每个对象的物理锁时，sp_sysmon 的 Cluster Physical Locks 子部分会使用此计数器来报告死锁。
PhysicalLocksWaitd	int		特定于集群环境 — 实例等待物理锁请求的次数。
PhysicalLocksPageTransfer	int		特定于集群环境 — 在实例请求物理锁时所发生的页面传输数。sp_sysmon 的 Cluster Physical Locks 子部分使用此计数器以该对象的节点关联率的形式来报告节点到节点传输和物理锁获取情况
TransferReqWaitd	int4		特定于集群环境 — 在接收页面传输之前物理锁请求等待的次数。
AvgPhysicalLocksWaitTime	int4		特定于集群环境 — 授予物理锁之前客户端所花费的平均时间。

名称	数据类型	属性	说明
AvgTransferReqWaitTime	int4		特定于集群环境— 在接收页面传输之前物理锁请求的平均等待时间。
TotalServiceRequests	int4		特定于集群环境— 实例的集群高速缓存管理器所支持的物理锁请求数。
PhysicalLocksDowngraded	int4		特定于集群环境— 实例的集群高速缓存管理器所支持的物理锁降级请求数。
PagesTransferred	int4		特定于集群环境— 集群高速缓存管理器在某个实例上传输的页数。
ClusterPageWrites	int4		特定于集群环境— 实例的集群高速缓存管理器写入磁盘的页数。
AvgServiceTime	int4		特定于集群环境— 实例的集群高速缓存管理器所花费的平均服务时间。
AvgTimeWaitedOnLocalUsers	int4		特定于集群环境— 实例的集群高速缓存管理器由于该实例上的用户使用页面而等待的平均服务时间。
AvgTransferSendWaitTime	int4		特定于集群环境— 实例的集群高速缓存管理器为页面传输花费的平均服务时间。
AvgIOServiceTime	int4		特定于集群环境— 实例的集群高速缓存管理器用于页面传输的平均服务时间。
AvgDowngradeServiceTime	int4		特定于集群环境— 集群高速缓存管理器用于降级物理锁的平均服务时间。

---

**注释** 由于可以对存储过程或触发器多次使用该计划，OptSelectCount 列的值可能会小于 UsedCount 的值。此外，由于 Adaptive Server 可能会在执行期间决定不执行查询的某些部分，UsedCount 可能会小于 OptSelectCount。

---



## monOpenPartitionActivity

说明

提供有关服务器上每个打开的分区的使用信息。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 和 `per object statistics active` 配置参数。

列

monOpenPartitionActivity 的列有：

名称	数据类型	属性	说明
DBID	int		数据库的唯一标识符。
ObjectID	int		对象的唯一标识符。
IndexID	int		索引的唯一标识符。
PartitionID	int		分区的唯一标识符。
InstanceID	int		实例在共享磁盘集群中的 ID。
DBName	varchar (30)	Null	对象所在的数据库的名称。
ObjectName	varchar (30)	Null	对象名。
PartitionName	varchar (30)	Null	分区名。
LogicalReads	int	Counter、null	读取的缓冲区总数。
PhysicalReads	int	Counter、null	从磁盘读取的缓冲区数。
APFReads	int	Counter、null	读取的异步预取 (APF) 缓冲区数。
PagesRead	int	Counter、null	读取的总页数。
PhysicalWrites	int	Counter、null	写入磁盘的缓冲区总数。
PagesWritten	int	Counter、null	写入磁盘的总页数。
RowsInserted	int	Counter、null	插入的行数。
RowsDeleted	int	Counter、null	已删除的行数。
RowsUpdated	int	Counter、null	更新的次数。
OptSelectCount	int	Counter、null	编译期间为计划选择对象的次数。

名称	数据类型	属性	说明
LastOptSelectDate	datetime	Null	最近一次在编译过程中为计划选择索引的日期。
UsedCount	int	Counter、null	对象在执行过程中用于某个计划的次数。
LastUsedDate	datetime	Null	最近一次在执行过程中将索引用于某个计划的日期。
HkgcRequests			排队等待某个分区的事件总数。该值大则意味着系统为指定分区生成了大量碎片。
HkgcPending			某个分区的挂起事件数。如果该值很大，则说明有很多碎片有待收集，尽管管家将会清理这些碎片。如果重新启动 Adaptive Server，则管家队列中的所有条目都将在重新启动 Adaptive Server 时丢失，并且这些页的碎片也不会被收集。
HkgcOverflows			溢出分区事件数。该值大则意味着管家队列已排满。由于管家无法调度作业，因此不会清除生成的碎片。
PhysicalLocks	int		<i>特定于集群环境</i> — 每个对象所请求的物理锁数。
PhysicalLocksRetained	int		保留的物理锁数。用于标识每个对象的锁命中率。命中率高则意味着此对象的分区十分平衡。
PhysicalLocksRetainWaited	int4		<i>特定于集群环境</i> — 在保留锁之前等待的物理锁请求数。
PhysicalLocksDeadlocks	int		<i>特定于集群环境</i> — 请求的物理锁返回死锁的次数。在获取每个对象的物理锁时，sp_sysmon 的 Cluster Physical Locks 子部分会使用此计数器来报告死锁。
PhysicalLocksWaited	int		<i>特定于集群环境</i> — 实例等待物理锁请求的次数。
PhysicalLocksPageTransfer	int		<i>特定于集群环境</i> — 在实例请求物理锁时所发生的页面传输数。sp_sysmon 的 Cluster Physical Locks 子部分使用此计数器以该对象的节点关联率的形式来报告节点到节点传输和物理锁获取情况。
TransferReqWaited	int4		<i>特定于集群环境</i> — 在接收页面传输之前物理锁请求等待的次数。
AvgPhysicalLockWaitTime	int4		<i>特定于集群环境</i> — 授予物理锁之前客户端所花费的平均时间。
AvgTransferReqWaitTime	int4		<i>特定于集群环境</i> — 在接收页面传输之前物理锁请求的平均等待时间。
TotalServiceRequests	int4		<i>特定于集群环境</i> — 实例的集群高速缓存管理器所支持的物理锁请求数。

名称	数据类型	属性	说明
PhysicalLocksDowngraded	int4		特定于集群环境—实例的集群高速缓存管理器所支持的物理锁降级请求数。
PagesTransferred	int4		特定于集群环境—集群高速缓存管理器在某个实例上传输的页数。
ClusterPageWrites	int4		特定于集群环境—实例的集群高速缓存管理器写入磁盘的页数。
AvgServiceTime	int4		特定于集群环境—实例的集群高速缓存管理器所花费的平均服务时间。
AvgTimeWaitedOnLocalUsers	int4		特定于集群环境—实例的集群高速缓存管理器由于该实例上的用户使用页面而等待的平均服务时间。
AvgTransferSendWaitTime	int4		特定于集群环境—实例的集群高速缓存管理器为页面传输花费的平均服务时间。
AvgIOServiceTime	int4		特定于集群环境—实例的集群高速缓存管理器用于页面传输的平均服务时间。
AvgDowngradeServiceTime	int4		特定于集群环境—集群高速缓存管理器用于降级物理锁的平均服务时间。

**注释** 由于可以对存储过程或触发器多次使用该计划，OptSelectCount 列的值可能会小于 UsedCount 的值。此外，由于 Adaptive Server 可能会在执行期间决定不执行查询的某些部分，UsedCount 可能会小于 OptSelectCount。

## monPCIBridge

**说明** 包含有关 Java PCI Bridge 的信息。此表提供关于 Java 环境的信息。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

**列** monPCIBridge 的列有：

名称	数据类型	说明
InstanceID	tinyint	特定于集群环境 — 实例在共享磁盘集群中的 ID。
Status	char(10)	PCI Bridge 的当前状态。其值包括： <ul style="list-style-type: none"> <li>ACTIVE</li> <li>DOWN</li> </ul>
ConfiguredSlots	int	已配置的槽数。使用 <code>max pci slots</code> 配置参数进行设置。
ActiveSlots	int	当前处于活动状态的槽数。
ConfiguredPCIMemoryKB	int	使用 <code>pci memory</code> 配置参数为 PCI Bridge 配置的总内存。
UsedPCIMemoryKB	int	当前由 PCI 桥及其组件使用的总内存。

## monPCIEngine

**说明** 显示有关 PCI Bridge 及其插件的引擎信息。此表提供关于 Java 环境的信息。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

**列** monPCIEngine 的列有：

名称	数据类型	说明
InstanceID	tinyint	特定于集群环境 — 实例在共享磁盘集群中的 ID。
Engine	int	引擎号
Status	char(10)	引擎上插件的状态。其值包括： <ul style="list-style-type: none"> <li>ACTIVE</li> <li>INIT</li> </ul>

名称	数据类型	说明
PLBStatus	char(10)	PCI Launcher Boss 的状态。其值包括： <ul style="list-style-type: none"> <li>ACTIVE</li> <li>DOWN</li> </ul>
NumberOfActiveThreads	int	当前由 PCI Launcher Boss 控制的活跃线程数。
PLBRequests	int	要求 PCI Launcher Boss 为本机线程执行某项功能的请求数。
PLBwakeupRequests	int	PCI Launcher Boss 接收到的要求其为本机线程执行工作的唤醒次数。

## monPCISlots

说明

包含有关绑定到 PCI Bridge 中每个槽的插件的信息。此表提供关于 Java 环境的信息。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列

monPCISlots 的列有：

名称	数据类型	说明
InstanceID	tinyint	特定于集群环境 — 实例在共享磁盘集群中的 ID。
Slot	int	活动槽的编号：值为 1 到 31。
Status	char(10)	槽的状态。其值包括： <ul style="list-style-type: none"> <li>INIT</li> <li>IN USE</li> <li>STOPPED</li> </ul>
Modulename	varchar(30)	绑定到当前槽的逻辑模块名称。
engine	int	与槽关联的引擎。

## monPCM

说明

特定于集群环境—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相应的一行。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列

monPCM 的列有：

列名	类型	属性	说明
InstanceID	int1		收集的信息所针对的实例 ID
Sent	int4		每个模块发送的消息数
Fragments_sent	int4		每个模块发送的片段数
Fragments_received	int4		每个模块接收的片段数
Received	int4		每个模块接收的消息数
Reply	int4		每个模块接收的答复数
Unicast	int4		每个模块发送的单播消息数
Mulicat	int4		每个模块发送的多播消息数
Sync	int4		每个模块发送的同步消息数
Async	int4		每个模块发送的异步消息数
MinBytes	int4		每条消息传输的最小字节数
AvgBytes	int4		每条消息传输的平均字节数
MaxBytes	int4		每条消息传输的最大字节数
MinDialog	int4		对话的最小长度
AvgDialog	int4		对话的平均长度
MaxDialog	int4		对话的最大长度
Dialog	int4		对话数。
MinTimeSyncApi	flt4		每个模块在 PCM 层中花费在同步模式下的 PCM API 中的最短时间
AvgTimeSyncApi	flt4		每个模块在 PCM 层中花费在同步模式下的 PCM API 中的平均时间
MaxTimeSyncApi	flt4		每个模块在 PCM 层中花费在同步模式下的 PCM API 中的最长时间
MinTimeAsyncApi	flt4		每个模块在 PCM 层中花费在异步模式下的 PCM API 中的最短时间
AvgTimeAsyncApi	flt4		每个模块在 PCM 层中花费在异步模式下的 PCM API 中的平均时间
MaxTimeAsyncApi	flt4		每个模块在 PCM 层中花费在异步模式下的 PCM API 中的最长时间

列名	类型	属性	说明
MinTimeCIPCMsgAlloc	flt4		每个模块在 PCM 层中花费在 cipcmmsg 分配上的最短时间
AvgTimeCIPCMsgAlloc	flt4		每个模块在 PCM 层中花费在 cipcmmsg 分配上的平均时间
MaxTimeCIPCMsgAlloc	flt4		每个模块在 PCM 层中花费在 cipcmmsg 分配上的最长时间
MinTimeCIPCMsgSendCB	flt4		每个模块花费在 cipc_sendcb 上的最短时间
AvgTimeCIPCMsgSendCB	flt4		每个模块花费在 cipc_sendcb 上的平均时间
MaxTimeCIPCMsgSendCB	flt4		每个模块花费在 cipc_sendcb 上的最长时间
MinTimeCIPCUnicastmsg	flt4		每个模块在发送单播消息时花费在 CIPC 上的最短时间
AvgTimeCIPCUnicastmsg	flt4		每个模块在发送单播消息时花费在 CIPC 上的平均时间
MaxTimeCIPCUnicastmsg	flt4		每个模块在发送单播消息时花费在 CIPC 上的最长时间
MinTimeCIPCMulticastmsg	flt4		每个模块在发送多播消息时花费在 CIPC 上的最短时间
AvgTimeCIPCMulticastmsg	flt4		每个模块在发送多播消息时花费在 CIPC 上的平均时间
MaxTimeCIPCMulticastmsg	flt4		每个模块在发送多播消息时花费在 CIPC 上的最长时间
MinTimeClientRecvCB	flt4		每个模块在 PCM 层中花费在客户端接收回调上的最短时间
AvgTimeClientRecvCB	flt4		每个模块在 PCM 层中花费在客户端接收回调上的平均时间
MaxTimeClientRecvCB	flt4		每个模块在 PCM 层中花费在客户端接收回调上的最长时间
ModuleName	int4		PCM 客户端的名称

## monProcedureCache

说明 返回与 Adaptive Server 过程高速缓存有关的统计信息。

---

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

---

列 monProcedureCache 的列有：

名称	数据类型	属性	说明
Requests	int	Counter、reset	所请求的存储过程数
Loads	int	Counter、reset	装载到高速缓存中的存储过程数
Writes	int	Counter、reset	对过程进行规范化并将树写回到 <code>sysprocedures</code> 的次数
Stalls	int	Counter、reset	在向高速缓存中安装存储过程时进程必须等待可用的过程高速缓存缓冲区的次数
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。



## monProcedureCacheMemoryUsage

说明

每个过程高速缓存分配器在该表中都有一行。分配器是由分配器 ID 标识的，此 ID 是由 Adaptive Server 内部提供的。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列

monProcedureCacheMemoryUsage 的列有：

名称	数据类型	属性	说明
InstanceID	tinyint		特定于集群环境 — 实例在共享磁盘集群中的 ID。
AllocatorID	int		Allocator ID
ModuleID	int		模块 ID（由 Adaptive Server 内部提供）
Active	int		当前为此分配器分配的内存页 (2KB) 数
HWM	int		自服务器启动后分配的内存页的最大数量
ChunkHWM	int		自服务器启动后分配的连续内存页的最大数量
AllocatorName	varchar(30)		分配器名称
NumReuseCaused	int	Null	此分配器导致进行替换的次数

## monProcedureCacheModuleUsage

### 说明

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。使用模块 ID 标识的模块是 Adaptive Server 过程高速缓存管理功能内部的功能范围分类。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

### 列

monProcedureCacheModuleUsage 的列有：

名称	数据类型	属性	说明
InstanceID	tinyint		特定于集群环境 — 实例在共享磁盘集群中的 ID。
ModuleID	int		模块 ID
Active	int		当前为此模块分配的内存页 (2KB) 数
HWM	int		自服务器启动后分配的内存页的最大数量
NumPagesReused	int	Null	为此模块分配的页数
ModuleName	varchar(30)		模块名称

## monProcess

说明

提供有关当前正在执行的或等待的进程的详细统计信息。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 和 `wait event timing` 配置参数。

列 monProcess 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符
ServerUserID	int		与此进程相关联的用户的服务器用户 ID (SUID)。
BatchID	int		包含执行语句的 SQL 批处理的唯一标识符
ContextID	int		每次在执行查询引发存储过程、触发器、 <code>execute immediate</code> 、延迟编译或其它编译对象执行时生成的唯一标识符
LineNumber	int		SQL 批处理中当前语句的行号
SecondsConnected	int		自建立此连接以来经过的秒数
DBID	int		进程所使用数据库的唯一标识符
EngineNumber	smallint		进程在其上执行的引擎的唯一标识符
Priority	int		进程执行的优先级
FamilyID	int	Null	父进程的 <code>spid</code> （如果该进程为工作进程）
Login	varchar(30)	Null	登录用户名
Application	varchar(30)	Null	应用程序名。如果应用程序没有在其登录结构中设置名称，则可能为空。
Command	varchar(30)	Null	进程或进程当前执行的命令的类别
NumChildren	int	Null	子进程数（如果执行并行查询）
SecondsWaiting	int	Null	进程的等待时间（如果该进程当前被另一进程所持有的锁阻塞），以秒为单位。
WaitEventID	int	Null	进程等待的事件的唯一标识符（如果该进程当前处于等待状态）。
BlockingSPID	int	Null	持有此进程所请求锁的进程的会话进程标识符（如果正在等待锁）
BlockingXLOID	int	Null	此进程已经请求的锁的唯一锁标识符（如果正在等待锁）
DBName	varchar(30)	Null	进程当前所使用的数据库的名称
EngineGroupName	varchar(30)	Null	进程的引擎组
ExecutionClass	varchar(30)	Null	进程的执行类
MasterTransactionID	varchar(255)	Null	进程所打开的事务的名称

## monProcessActivity

说明 提供有关进程活动的详细统计信息。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring`、`object lockwait timing` 和 `wait event timing` 配置参数。

列 monProcessActivity 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符。
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符。
ServerUserID	int		运行此进程的用户的服务器用户标识符 (SUID)。 ServerUserID 中的值与 <code>syslogins.suid</code> 列相符。使用 <code>suser_name</code> 函数可以获取相应的名称。
CPUTime	int	Counter	进程占用的 CPU 时间（以毫秒为单位）。
WaitTime	int	Counter	进程等待所花费的时间（以毫秒为单位）。
PhysicalReads	int	Counter	从磁盘读取的缓冲区数。
LogicalReads	int	Counter	从高速缓存读取的缓冲区数。
PagesRead	int	Counter	读取的页数。
PhysicalWrites	int	Counter	写入磁盘的缓冲区数。
PagesWritten	int	Counter	写入的页数。
MemUsageKB	int		分配给进程的内存量（以字节为单位）。
LocksHeld	int		进程当前持有的锁数。
TableAccesses	int	Counter	Adaptive Server 在不使用索引的情况下检索到的已读页数。
IndexAccesses	int	Counter	Adaptive Server 在使用索引的情况下检索到的已读页数。
WorkTables	int	Counter	进程所创建的工作表的总数。
TempDbObjects	int	Counter	进程所创建的临时表的总数。
ULCBytesWritten	int	Counter	写入到进程的用户日志高速缓存的字节数。
ULCFlushes	int	Counter	刷新用户日志高速缓存的总次数。
ULCFlushFull	int	Counter	因用户日志高速缓存已满而刷新它的次数。
ULCMaxUsage	int		进程使用的最大量的用户日志高速缓存（以字节为单位）。
ULCCurrentUsage	int		进程当前使用的用户日志高速缓存量（以字节为单位）。
Transactions	int	Counter	进程启动的事务数。
Commits	int	Counter	进程提交的事务数。
Rollbacks	int	Counter	进程回退的事务数。

## monProcessLookup

**说明** 提供有关服务器上每个进程的标识信息。有关每个进程的活动的统计信息，请参见第 162 页的“[monProcessActivity](#)”。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

**列** monProcessLookup 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符
Login	varchar(30)	Null	登录用户名
Application	varchar(30)	Null	应用程序名
ClientHost	varchar(30)	Null	客户端的主机名
ClientIP	varchar(24)	Null	客户端的 IP 地址
ClientOSPID	varchar(30)	Null	客户端应用程序的操作系统进程标识符

## monProcessMigration

**说明** 特定于集群环境 — 显示有关当前迁移的连接的信息。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

**列** monProcessMigration 的列有：

名称	数据类型	属性	说明
SPID	int4		挂起的迁移会话进程 ID
KPID	int4		内核进程 ID
LogicalCluster	varchar(30)		当前逻辑集群
Instance	varchar(30)		当前实例。
MigrationLogicalCluster	varchar(30)		迁移逻辑集群。
MigrationInstance	varchar(30)		迁移实例。
Command	varchar(30)		迁移触发器。

## monProcessNetIO

说明 提供每个进程的网络 I/O 活动信息。

---

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

---

列 monProcessNetIO 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符
NetworkPacketSize	int		会话当前使用的网络包大小。
PacketSent	int	Counter	发送出的包数量
PacketsReceived	int	Counter	接收到的包数量
BytesSent	int	Counter	发送出的字节数
BytesRecieved	int	Counter	接收到的字节数
NetworkEngineNumber	smallint		此进程用作其网络引擎的引擎的编号。

## monProcessObject

说明

提供有关进程当前访问的对象的统计信息。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 和 `per object statistics active` 配置参数。

列

monProcessObject 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符
DBID	int		对象所驻留的数据库的唯一标识符
ObjectID	int		对象的唯一标识符
PartitionID	int		分区的唯一标识符
IndexID	int		索引的唯一标识符
OwnerUserID	int		对象所有者的用户标识符
LogicalReads	int	Counter	从高速缓存读取的缓冲区数
PhysicalReads	int	Counter	从磁盘读取的缓冲区数
PhysicalAPFReads	int	Counter	从磁盘读取的异步预取缓冲区数
DBName	varchar(30)	Null	数据库的名称
ObjectName	varchar(30)	Null	对象名
PartitionName	varchar(30)	Null	分区名
ObjectType	varchar(30)	Null	对象类型
PartitionSize	int	Counter、 null	分区大小（以千字节为单位）

## monProcessProcedures

说明 返回进程正在执行的所有过程的列表。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

列 monProcessProcedures 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符
DBID	int		对象的数据库的唯一标识符
OwnerUID	int		对象所有者的唯一标识符
ObjectID	int		过程的唯一标识符
PlanID	int		查询计划的唯一标识符
MemUsageKB	int		过程所使用的内存量（以 KB 为单位）
CompileDate	datetime		编译过程的日期
ContextID	int		每次在执行查询引发存储过程、触发器、execute immediate、延迟编译或其它编译对象执行时生成的唯一标识符
LineNumber	int		当前正在执行的过程中的行
DBName	varchar(30)	Null	包含该过程的数据库名称
OwnerName	varchar(30)	Null	对象所有者的名称
ObjectName	varchar(30)	Null	过程名
ObjectType	varchar(32)	Null	过程的类型（例如，存储过程或触发器）



## monProcessSQLText

说明

提供进程当前正在执行的 SQL 文本。使用 `max SQL text monitored` 可以调整 SQL 文本的最大大小。

`monProcessSQLText` 为进程（由 `SPID` 指定）所执行的 SQL 文本批处理的每一行返回一行。也就是说，如果批处理包含三行，`monProcessSQLText` 会在其结果集中返回三行。`LineNumber` 的值表示批处理中的行号。如果单行长度超过 255 字节，`monProcessSQLText` 会返回多个行且所有行的 `LineNumber` 值都相同，但每一行的 `SequenceInLine` 值都不同。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring`、`SQL batch capture` 和 `max SQL text monitored` 配置参数。

列

`monProcessSQLText` 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符。
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符。
ServerUserID	int		执行此 SQL 的用户的服务器用户标识符 (SUID)。 <code>ServerUserID</code> 与 <code>syslogins.suid</code> 列的值相符。使用 <code>suser_name</code> 函数可以获取相应的名称。
BatchID	int		包含 SQL 文本的 SQL 批处理的唯一标识符。
LineNumber	int		行的 SQL 文本的 SQL 批处理行号。
SequenceInLine	int		每个行都具有唯一的、递增的 <code>SequenceInLine</code> 值。如果 SQL 文本的长度超过了 255 字节，则将文本拆分为多行。
SQLText	varchar(255)	Null	正在执行的文本。

## monProcessStatement

说明 提供有关当前正在执行的语句的信息。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring`、`statement statistics active` 和 `wait event timing` 配置参数。

列 monProcessStatement 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符。
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符。
DBID	int		当前正在由进程使用的数据库的唯一标识符。
ProcedureID	int		存储过程的唯一标识符。
PlanID	int		进程正在执行的计划的唯一标识符。
BatchID	int		在其中执行语句的进程的批处理编号。
ContextID	int		过程的堆栈框架（如果是过程）。
LineNumber	int		SQL 批处理中语句的行号。
CPUTime	int	Counter	语句占用的 CPU 时间（以毫秒为单位）。
WaitTime	int	Counter	执行语句时任务的等待时间（以毫秒为单位）。
MemUsageKB	int		语句执行过程中所用内存的千字节数。
PhysicalReads	int	Counter	从磁盘读取的缓冲区数。
LogicalReads	int	Counter	从高速缓存读取的缓冲区数。
PagesModified	int	Counter	语句修改的页数。
PacketsSent	int	Counter	Adaptive Server 发送的网络包数目。
PacketsReceived	int	Counter	Adaptive Server 接收的网络包数目。
NetworkPacketSize	int		当前为会话配置的网络包大小（以字节为单位）。
PlansAltered	int	Counter	执行时改变的计划数。
RowsAffected	int		受当前语句影响的行数。使用低效率查询计划进行查询可能会为每个返回行显示大量逻辑 I/O。
DBName	varchar (30)		在其中执行此进程的数据库的名称。如果进程是在执行存储过程或其它编译对象，则数据库名为该对象的数据库的名称。
StartTime	datetime	Null	语句开始执行时的日期。

## monProcessWaits

**说明** 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。仅返回其 Waits 值大于零的等待事件。

---

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring`、`process wait events` 和 `wait event timing` 配置参数。

---

列 monProcessWaits 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符
ServerUserID	int		与此进程相关联的用户的服务器用户 ID (SUID)。
WaitEventID	smallint		等待事件的唯一标识符
Waits	int	Counter	进程已等待事件的次数
WaitTime	int	Counter	进程已等待事件的时间（以毫秒为单位）

WaitEventInfo 包含每个等待事件的说明。连接每个监控表的 WaitEventID 列可查看此数据。

请参见 Performance and Tuning: Monitoring Tables（《性能和调优：监控表》）以了解有关如何选择等待事件的说明。

## monProcessWorkerThread

**说明** 提供有关当前配置的每个工作进程的活动的统计信息。

---

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

---

**列** monProcessWorkerThread 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符
ThreadsActive	int		当前由进程使用的工作线程数目
MaxParallelDegree	smallint		此任务可以使用的最大并行度，该最大并行度是使用会话的 <code>set parallel_degree</code> 选项或 <code>max parallel degree</code> 的当前 <code>Run Value</code> 设置的。
MaxScanParallelDegree	smallint		此任务可以使用的扫描最大并行度。该最大并行度可以使用会话的 <code>set scan_parallel_degree</code> 进行设置；如果不设置此选项，该最大并行度也可以使用 <code>max scan parallel degree</code> 的当前 <code>Run Value</code> 进行设置。
ParallelQueries	int	Counter	此进程所执行的并行查询总数
PlansAltered	int	Counter	从进程的“最佳”项改变过来的计划数。如果 Adaptive Server 在使用最佳并行度执行查询时没有足够可用的工作线程数，则会改变计划。
FamilyID	int	Null	父进程的 <code>spid</code> （如果该进程为工作进程）

## monState

说明

提供有关 Adaptive Server 的总体状态的信息。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列

monState 的列有：

名称	数据类型	属性	说明
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
LockWaitThreshold	int		在被视为“已阻塞”并在 LockWaits 列中进行报告之前，进程必须等待锁的时间（以秒为单位）。LockWaitThreshold 的缺省值为 5 秒。如果不在查询的 where 子句中指定某个值（例如，LockWaitThreshold=30），则会使用缺省值。
LockWaits	int		等待锁的时间超过 LockWaitThreshold 值的进程数目。
DaysRunning	int		Adaptive Server 已运行的天数。
CheckPoints	int		指明当前是否有任何检查点在运行。
NumDeadlocks	int	Counter	已经出现的死锁总数。
Diagnostic Dumps	int		指明此服务器的共享内存转储当前是否正在进行。
Connections	int		活动入站连接数。
MaxRecovery	int		对于每个数据库，系统出现故障时 Adaptive Server 完成其恢复过程所用的最长时间（以分钟为单位）；此外，也指 recovery interval in minutes 配置选项的当前“运行值”。
Transactions	int4		服务器范围内运行的事务数目。
StartDate	datetime		启动 Adaptive Server 的日期和时间。
CountersCleared	datetime		上次清除监控器计数器的日期和时间。

## monStatementCache

**说明** 提供有关语句高速缓存的统计信息。必须先启用语句高速缓存，然后 monStatementCache 表才能收集数据。

---

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 和 `enable stmt cache monitoring` 配置参数并将 `statement cache size` 参数设置成大于 0。

---

列 monStatementCache 的列有：

名称	类型	属性	说明
InstanceID	tinyint		特定于集群环境 — 实例在共享磁盘集群中的 ID。
TotalSizeKB	int		语句高速缓存的配置大小 (KB)。
UsedSizeKB	int		当前正在使用的语句高速缓存量 (KB)。
NumStatements	int		语句高速缓存中的语句数。
NumSearches	int	Counter、reset	搜索语句高速缓存的次数。
HitCount	int	Counter、reset	搜索语句高速缓存并找到匹配项的次数。
NumInserts	int	Counter、reset	插入到语句高速缓存中的语句数。
NumRemovals	int	Counter、reset	从语句高速缓存中删除语句的次数。此值包括通过显式清除删除或从替换策略中删除的语句数。
NumRecompilesSchemaChanges	int	Counter、reset	由于高速缓存语句中所引用的表中的模式更改而进行的重新编译次数。
NumRecompilesPlanFlushes	int	Counter、reset	因高速缓存中的计划刷新而进行的重新编译次数。

## monSysLoad

说明

特定于集群环境—按引擎提供趋势统计信息。您不需要具有 `mon_role` 角色即可查询该监控表。

每个引擎的每项统计信息都有一行，但 `kernel run queue length` 除外（仅当引擎号为 0 时才报告该项）。

平均数是使用一种可以清除瞬时峰值和谷值并可提供总体趋势指示的算法来计算的。

`monSysLoad` 不需要任何配置参数。

列

`monSysLoad` 的列有：

名称	数据类型	属性	说明
<code>InstanceID</code>	<code>tinyint</code>		集群内实例的 ID。
<code>EngineNumber</code>	<code>smallint</code>		此行所属的引擎。
<code>SteadyState</code>	<code>real</code>		自 <code>Adaptive Server</code> 启动以来此统计信息的平均值。
<code>Avg_1min</code>	<code>real</code>		此统计信息的一分钟移动平均值。
<code>Avg_5min</code>	<code>real</code>		此统计信息的五分钟移动平均值。
<code>Avg_15min</code>	<code>real</code>		此统计信息的十五分钟移动平均值。
<code>Max_1min</code>	<code>real</code>		自启动以来的 1 分钟最大平均值。
<code>Max_5min</code>	<code>real</code>		自启动以来的 5 分钟最大平均值。
<code>Max_15min</code>	<code>real</code>		自启动以来的 15 分钟最大平均值。
<code>Max_1min_Time</code>	<code>datetime</code>		出现 <code>Max_1min</code> 时的 <code>datetime</code> 。
<code>Max_5min_Time</code>	<code>datetime</code>		出现 <code>Max_5min</code> 时的 <code>datetime</code> 。
<code>Max_15min_Time</code>	<code>datetime</code>		出现 <code>Max_15min</code> 时的 <code>datetime</code> 。
<code>Statistic</code>			此行所代表的统计信息名称： <ul style="list-style-type: none"> <li>• CPU 忙百分比</li> <li>• I/O 忙百分比</li> <li>• 运行队列长度</li> <li>• 内核运行队列长度</li> <li>• 未完成的磁盘 I/O 数</li> <li>• 每秒磁盘 I/O 数</li> <li>• 每秒网络 I/O 数</li> </ul>
<code>Sample</code>	<code>float</code>		最近一次采样间隔的指标值（即指标的当前值）。
<code>Peak</code>	<code>float</code>		自实例启动以来的 <code>Sample</code> 最高值（即 <code>Sample</code> 峰值）。
<code>Peak_time</code>	<code>datetime</code>		达到 <code>Peak</code> 值的日期和时间。
<code>StatisticID</code>	<code>int</code>		此统计信息的固定标识符。您可能希望将应用程序写入固定 <code>StatisticID</code> 而非本地化的 <code>Statistic</code> 名称。

## monSysPlanText

**说明** 提供当前执行的查询的查询计划的历史记录。

monSysPlanText 从运行的查询计划的每一行返回一行文本（与 `sp_showplan` 或 `set showplan on` 所返回的内容类似）。要确保 monSysPlanText 按正确顺序读取查询计划文本，请按 `SequenceNumber` 对查询结果排序。对于会返回多个查询或进程的数据的查询，请按 `SPID`、`KPID`、`BatchID`、`SequenceNumber` 对查询结果排序。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring`、`plan text pipe active` 和 `plan text pipe max messages` 配置参数。

列 monSysPlanText 的列有：

名称	数据类型	属性	说明
PlanID	int		计划的唯一标识符。
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
SPID	smallint		会话进程标识符。
KPID	int		内核进程标识符。
BatchID	int		为其创建了计划的 SQL 批处理的唯一标识符。
ContextID	int		过程的堆栈框架（如果是过程）。
SequenceNumber	int		指示 PlanText 列在整个计划文本内的位置的单调递增编号。
DBID	int		将过程存储在其中的数据库的唯一标识符（如果计划用于存储过程）。
ProcedureID	int		过程的唯一标识符（如果计划用于存储过程）。
DBName	varchar(30)	Null	在其中执行此计划所代表的语句的数据库的名称。查询 monSysPlanText 时，如果该数据库尚未打开，则此列为 NULL。如果进程是在执行存储过程或其它编译对象，则数据库名为该对象的数据库的名称。
PlanText	varchar(160)	Null	计划文本输出。

**注释** 通常情况下，每个查询计划在此表中有多个行。可以按升序对 `SequenceNumber` 列排序以排列各行。

monSysPlanText 是一个历史监控表。请参见 *Performance and Tuning Guide*（《性能和调优指南》）第 1 章“Introduction to Monitoring Tables”（监控表简介）中的“Stateful historical monitoring table”（有态历史监控表）。



## monSysSQLText

**说明** 提供最近执行的 SQL 文本或当前执行的 SQL 文本。可以使用 `sql text pipe max messages` 调整返回行的最大数。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring`、`SQL text pipe active`、`max SQL text monitored` 和 `SQL text pipe max messages` 配置参数。

`monSysSQLText` 是一个历史监控表。请参见 [Performance and Tuning: Monitoring Tables](#)（《性能和调优：监控表》）。

**列** `monSysSQLText` 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符。
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符。
ServerUserID	int		执行此 SQL 文本的用户的服务器用户标识符 (SUID)。 <code>ServerUserID</code> 与 <code>syslogins.suid</code> 中的值相符。使用 <code>suser_name</code> 函数可以获取相应的名称。
BatchID	int		包含 SQL 文本的 SQL 批处理的唯一标识符。
SequenceInBatch	int		表示此部分 SQL 文本在批处理中的位置（批处理的 SQL 文本可能跨多个行）。
SQLText	varchar(255)	Null	SQL 文本。

**注释** 在很多情况下，查询的文本会在此表中跨多个行。可以按升序对 `SequenceInBatch` 列排序来以适当的顺序排列各行。

## monSysStatement

### 说明

提供有关服务器上最近执行的语句的历史记录。使用 `statement pipe max messages` 可以调整返回语句统计信息的最大数。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring`、`per object statistics active`、`statement pipe active`、`statement pipe max messages` 和 `statement statistics active` 配置参数。

`monSysStatement` 是一个历史监控表。请参见 [Performance and Tuning: Monitoring Tables](#)（《性能和调优：监控表》）。

### 列

`monSysStatements` 的列有：

名称	数据类型	属性	说明
SPID	smallint		会话进程标识符。
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
KPID	int		内核进程标识符。
DBID	int		数据库的唯一标识符。
ProcedureID	int		过程的唯一标识符。
PlanID	int		过程的存储计划的唯一标识符。
BatchID	int		包含语句的 SQL 批处理的唯一标识符。
ContextID	int		过程的堆栈框架（如果是过程）。
LineNumber	int		SQL 批处理中语句的行号。
CpuTime	int	Counter	语句使用 CPU 的毫秒数。
WaitTime	int	Counter	语句执行过程中任务等待的毫秒数。
MemUsageKB	int		语句执行过程中所用内存的千字节数。
PhysicalReads	int	Counter	从磁盘读取的缓冲区数。
LogicalReads	int	Counter	从高速缓存读取的缓冲区数。
PagesModified	int	Counter	语句修改的页数。
PacketsSent	int	Counter	Adaptive Server 发送的网络包数目。
PacketsReceived	int	Counter	Adaptive Server 接收的网络包数目。
NetworkPacketSize	int		当前为会话配置的网络包大小（以字节为单位）。
PlansAltered	int	Counter	执行时改变的计划数。
RowsAffected	int		受当前语句影响的行数。使用低效率查询计划进行查询可能会为每个返回行显示大量逻辑 I/O。
ErrorStatus	int		语句的错误返回状态。
HashKey	int		语句的文本的散列值；该值不是唯一标识符。如果语句不是从语句高速缓存中执行的，则此列为零 (0)。

名称	数据类型	属性	说明
SsqlId	int		语句高速缓存中此语句的查询计划的 ID。如果语句不是从语句高速缓存中执行的，则此列为零 (0)。
ProcNestLevel	int		语句的嵌套级别。如果语句为即席查询，则此列为零 (0)。如果语句是在存储过程内，则此列表示该存储过程的嵌套级别。
StatementNumber	int		表示在进程的 SQL 批处理内执行此语句的顺序的编号。
DBName	varchar(30)		在其中执行语句的数据库的名称。查询 monSysStatement 时，如果数据库不再处于打开状态，则此列为 NULL。如果进程是在执行存储过程或其它编译对象，则数据库名为该对象的数据库的名称。
StartTime	datetime	Null	语句开始执行时的日期。
EndTime	datetime	Null	语句完成执行时的日期。

## monSysWaits

**说明** 针对进程等待的事件提供一个服务器范围的统计信息视图。

---

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 和 `wait event timing` 配置参数。

---

列 monSysWaits 的列有：

名称	数据类型	属性	说明
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
WaitEventID	smallint		等待事件的唯一标识符
WaitTime	int	Counter	任务等待事件所花费的时间（以秒为单位）
Waits	int		任务等待事件的次数

请参见 Performance and Tuning: Monitoring Tables（《性能和调优：监控表》）以了解详细信息

可以使用 WaitEventID 列作为连接列来连接 monSysWaits 表和 monWaitEventInfo，从而获取等待事件说明。例如：

```
select w.Waits, w.WaitTime, w.WaitEventID, i.Description
from master..monSysWaits w, master..monWaitEventInfo i
where w.WaitEventID = i.WaitEventID
```

## monSysWorkerThread

**说明** 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。

**注释** 要使该监控表收集数据，请为其启用 `enable monitoring` 配置参数。

列 monSysWorkerThread 的列有：

名称	数据类型	属性	说明
InstanceID	int		特定于集群环境 — 实例在共享磁盘集群中的 ID。
ThreadsActive	int		当前处于活动状态的工作进程数
TotalWorkerThreads	int		最大工作进程数（已通过设置 <code>number of worker processes</code> 进行配置）
HighWater	int	reset	曾使用的最大工作进程数
ParallelQueries	int	Counter、reset	尝试进行的并行查询数
PlansAltered	int	Counter、reset	由于工作进程不可用而改变的计划数
WorkerMemory	int		工作进程当前使用的内存量
TotalWorkerMemory	int		配置供工作进程使用的内存量
WorkerMemoryHWM	int	reset	工作进程曾使用的最大内存量
MaxParallelDegree	int		可以使用的最大并行度： <code>max parallel degree</code> 配置选项的当前 Run Value
MaxScanParallelDegree	int		可以用于扫描的最大并行度： <code>max scan parallel degree</code> 配置选项的当前 Run Value

## monTableColumns

**说明** 描述每个监控表的所有列。monTableColumns 可帮助用户确定监控表中存在哪些列。可以连接 monTableColumns 和 monTables，以便报告监控表的列和列属性。

此表的元数据视图对于共享磁盘集群中的所有实例都是相同的。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

**列** monTableColumns 的列有：

名称	数据类型	属性	说明
TableID	int		视图的唯一标识符
ColumnID	int		列的位置
TypeID	int		列的数据类型的标识符
Precision	tinyint		列的精度（如果列值为数字）
Scale	tinyint		列的标度（如果列值为数字）
Length	smallint		列的最大长度（以字节为单位）
Indicators	int		特定列属性的指示符（例如，如果列易于归零，而且应该进行采样） <sup>1</sup>
TableName	varchar(30)	Null	表名
ColumnName	varchar(30)	Null	列名
TypeName	varchar(20)	Null	列的数据类型名
Description	varchar(255)	Null	列的说明

<sup>1</sup>Indicators 列为位图。使用位屏蔽可以确定哪些位已打开。可能的值有：

- 1 — 如果值达到  $2^{32}$ （已将 Indicators 列值中的第 1 位打开的列可能会发生这种情况），则 Indicators 的值可能会迅速增大并导致计数器归零。要确定第 1 位是否已打开，请使用以下语法：

```
select TableName, ColumnName
from Master..monTableColumns
where Indicators & 1 != 0
```

- 2 — 计数器与 sp\_sysmon 共享并在您执行 sp\_sysmon...clear 的情况下重置。

要显示 sp\_sysmon 通过 clear 参数清除的所有列，请使用以下语法：

```
Select TableName, ColumnName
from master..monTableColumns
where Indicators & 2 != 0
```

## monTableParameters

**说明** 提供监控表中用于优化监控表查询性能的所有列的说明。  
此表的元数据视图对于共享磁盘集群中的所有实例都是相同的。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

**列** monTableParameters 的列有：

名称	数据类型	属性	说明
TableID	int		表的唯一标识符
ParameterID	int		参数位置
TypeID	int		参数数据类型的标识符
Precision	tiny_int		参数的精度（如果参数值是数字）
Scale	tiny_int		参数的标度（如果参数值是数字）
Length	small_int		参数的最大长度（以字节为单位）
TableName	varchar(30)	Null	表名
ParameterName	varchar(30)	Null	参数名
TypeName	varchar(20)	Null	参数的数据类型名
Description	varchar(255)	Null	参数说明

## monTables

### 说明

提供对所有监控表的说明。可以连接 `monTables` 和 `monTableColumns` 以获取每个监控表及其包含的列的说明。

此表的元数据视图对于共享磁盘集群中的所有实例都是相同的。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

### 列

`monTables` 的列有：

名称	数据类型	属性	说明
TableID	int		表的唯一标识符
Columns	tinyint		表中的总列数
Parameters	tinyint		可以指定的可选参数的总数
Indicators	int		特定表属性的指示符（例如，表是否保留会话上下文） <sup>1</sup>
Size	int		行大小最大值（以字节为单位）
TableName	varchar(30)	Null	表名
Description	varchar(368)	Null	表说明

<sup>1</sup>Indicators 列为位图。使用位屏蔽可以确定哪些位已打开。值为 1 则表示表为历史表。

要显示所有历史表，请使用以下语法：

```
Select TableName
from master..monTables
where Indicators & 1 != 0
```



## monTableTransfer

**说明** MonTableTransfer 提供 Adaptive Server 活动内存中表的历史传输信息。它不存储已完成传输的信息。MonTableTransfer 提供有关所有表当前正在进行的传输的传输信息（无论这些传输是否已标记为增量传输），以及有关表中标记为增量传输的以前传输的传输信息。

**列** monTableTransfer 的列有：

名称	数据类型	属性	说明
InstanceID	tinyint	NOT NULL	特定于集群环境 — 保存运行该命令的服务器的实例 ID。在非聚簇服务器中，始终保存零
DBID	smallint		表的数据库 ID
TableID	int		表的唯一标识符
TableName	varchar(255)	NULL	表名
SequenceID	int		Adaptive Server 生成的内部跟踪 ID
TrackingID	int	NULL	用户指定的跟踪 ID
PercentDone	smallint		已完成传输工作的百分比，以 0 到 100 之间的某个整数表示（所有完成的传输均显示 100）
BeginTime	datetime		传输开始的日期和时间
EndTime	datetime	NULL	传输结束的日期和时间。正在进行的传输显示 NULL
EndCode	smallint	NULL	传输的结束状态。 <ul style="list-style-type: none"> <li>• 0 — 传输成功。</li> <li>• NULL — 传输正在进行。</li> <li>• 错误代码 — 传输失败。</li> </ul>
TransferFloor	bigint		可以发送数据的时间戳
TransferCeiling	bigint		未能提交数据并因此无法发送数据的时间戳
RowsSent	bigint		发送的行数
BytesSent	bigint		发送出的字节数
Format	varchar(8)	NOT NULL	包含目标格式的名称：ase、bcp、csv 或 iq 之一

## monTempdbActivity

**说明** 特定于集群环境 — 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。

monTempdbActivity 需要 enable monitoring、per object statistics active 和 object lockwait timing 配置参数才能收集数据。

**列** monTempdbActivity 的列有：

名称	数据类型	说明
DBID	int	数据库的唯一标识符
InstanceID	tinyint	集群内实例的 ID
DBName	varchar(30)	数据库的名称
AppendLogRequest	int	实例在尝试附加到数据库事务日志时发出的信号请求数
AppendLogWaits	int	任务等待授予附加日志信号的次数
LogicalReads	int	缓冲区读取数总计
PhysicalReads	int	从磁盘读取的缓冲区数
APFReads	int	读取的异步预取 (APF) 缓冲区数
PagesRead	int	读取的总页数
PhysicalWrites	int	写入磁盘的缓冲区数总计
PagesWritten	int	写入磁盘的总页数
LockRequests	int	此临时数据库中对象锁的请求数
LockWaits	int	任务等待此临时数据库中对象锁的次数
CatLockRequests	int	请求系统目录上的锁的次数
CatLockWaits	int	任务等待系统表锁的次数
AssignedCnt	int	将此临时数据库分配给用户任务的次数
SharableTabCnt	int	创建的可共享表数

## monWaitClassInfo

### 说明

为所有等待类提供文本说明（例如，等待磁盘读取完成）。所有等待事件（请参见 `monWaitEventInfo` 的说明）都被归入等待类，这些等待类对进程等待的事件类型进行分类。

此表对共享磁盘集群中的所有实例都显示相同的信息。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

### 列

`monWaitClassInfo` 的列有：

名称	数据类型	属性	说明
WaitClassID	smallint		等待事件类的唯一标识符
Description	varchar(50)	Null	等待事件类的说明

## monWaitEventInfo

**说明** 为强制进程在 Adaptive Server 内等待的每种可能情况提供文本说明。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

**列** monWaitEventInfo 的列有：

名称	数据类型	属性	说明
WaitEventID	smallint		等待事件类型的唯一标识符
WaitClassID	smallint		等待事件类的唯一标识符
Description	varchar(50)	Null	等待事件类型的说明

连接 monWaitEventInfo 和 WaitEventID 列上的 monProcessWaits 或 monSysWaits 可获取这些表中列出的等待事件说明。

## monWorkload

说明

特定于集群环境— 根据负载配置文件显示每个实例上每个逻辑集群的工作负载得分。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列

monWorkload 的列有：

名称	数据类型	属性	说明
LCID	tinyint		逻辑集群 ID
InstanceID	tinyint		集群内实例的 ID
LoadProfileID	tinyint		用于生成负载得分的负载配置文件的 ID
LoadScore	int		此实例或逻辑集群的负载得分
ConnectionsScore	float		user connections 指标的加权值
CpuScore	float		cpu utilization 指标的加权值
RunQueueScore	float		run queue 指标的加权值
IoLoadScore	float		io load 指标的加权值
EngineScore	float		engine deficit 指标的加权值
UserScore	float		user 指标的加权值
LogicalClusterName	varchar(30)		逻辑集群名称
InstanceName	varchar(30)		实例名称
LoadProfileName	tinyint		用于生成负载得分的负载配置文件的名称

## monWorkloadPreview

### 说明

*特定于集群环境*— 提供有关负载配置文件在不启用时对工作负载得分的影响的估计值。monWorkload 为每个逻辑集群以及运行该逻辑集群的实例都设置了一行。负载得分和组件都基于该逻辑集群的当前配置文件。每个在系统上配置的实例和负载配置文件的组合在 monWorkloadPreview 表中都有相应的一行，从而使管理员可以查看每个配置文件的工作负载计算是如何执行的。您不需要具有 mon\_role 角色即可查询该监控表。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

列 monWorkloadPreview 的列有：

名称	数据类型	属性	说明
InstanceID	tinyint		集群内实例的 ID
LoadProfileID	smallint		负载配置文件 ID
LoadScore	int		此实例或逻辑集群的负载得分
ConnectionScore	float		user connections 指标的加权值
CpuScore	float		cpu utilization 指标的加权值
RunQueueScore	float		run queue 指标的加权值
IoLoadScore	float		io load 指标的加权值
EngineScore	float		engine deficit 指标的加权值
UserScore	float		user 指标的加权值
InstanceName	varchar(30)		实例名称
LoadProfileName	varchar(30)		用于生成负载得分的负载配置文件的名称

## monWorkloadProfile

**说明** 特定于集群环境— 显示当前配置的工作负载配置文件。您不需要具有 mon\_role 角色即可查询该监控表。

**注释** 无需启用任何配置参数，该监控表即可收集数据。

列 monWorkloadProfile 的列有：

名称	数据类型	属性	说明
ProfileID	smallint		工作负载配置文件 ID
ConnectionsWeight	tinyint		与 active connections 指标关联的权值
CpuWeight	tinyint		与 cpu utilization 指标关联的权值
RunQueueWeight	tinyint		与 run queue 指标关联的权值
IoLoadWeight	tinyint		与 io load 指标关联的权值
EngineWeight	tinyint		与 engine deficit 指标关联的权值
UserWeight	tinyint		与 user metric 指标关联的权值
LoginThreshold			登录负载分配的阈值。
DynamicThreshold	smallint		动态负载分配的阈值（即用于装载用途的登录后迁移）
Hysteresis	tinyint		允许进行重定向的最小负载得分。
Name	varchar(30)		工作负载配置文件名称
Type	varchar(30)		工作负载配置文件的类型。表示配置文件是由用户定义还是由系统定义。其值包括： <ul style="list-style-type: none"> <li>• 用户</li> <li>• 系统</li> </ul>

## monWorkloadRaw

**说明** 特定于集群环境— 提供每个实例的原始工作负载信息。您不需要具有 mon\_role 角色即可查询该监控表。

---

**注释** 无需启用任何配置参数，该监控表即可收集数据。

---

列 monWorkloadRaw 的列有：

名称	数据类型	属性	说明
InstanceID	tinyint		集群内实例的 ID
ConnectionsRaw	float		user connections 指标的原始值
CpuRaw	float		cpu utilization 指标的原始值
RunQueueRaw	float		run queue 指标的原始值
IoLoadRaw	float		io load 指标的原始值
EngineRaw	float		engine deficit 指标的原始值
UserRaw	float		user 指标的原始值
InstanceName	varchar(30)		实例名称



## ***sybpcidb* 表**

**sybpcidb** 数据库存储 Java PCI Bridge 和 PCA/JVM 插件的配置信息。本章按字母顺序介绍各个 **sybpcidb** 表。

针对 Java 配置服务器时，将创建 **sybpcidb**、安装其表并创建其系统存储过程。请参见针对所用平台的安装指南。有关如何使用 **sp\_jreconfig** 和 **sp\_pciconfig** 存储过程（使用它们可以配置和显示 **sybpcidb** 中的信息）的信息，另请参见《Adaptive Server Enterprise 中的 Java》。

## pca\_jre\_arguments

**说明** 存储有关配置 PCA/JVM 插件所用参数的信息。

**列** 位于 sybpcidb 中。pca\_jre\_arguments 的列包括：

名称	数据类型	说明
jre_args_directive_index	int	参数所属指令的索引。
jre_args_name	varchar(255)	参数的名称。
jre_args_units	varchar(255)	参数类型。其值包括： <ul style="list-style-type: none"> <li>• switch</li> <li>• string</li> <li>• number</li> <li>• array</li> </ul>
jre_args_number_value	int	如果 units=number，则保留与参数关联的数字。
jre_args_string_value	varchar(255)	如果 units=string 或 units=array，则保留与参数关联的字符串值。
jre_args_description	varchar(255)	参数的简短文本说明。
jre_args_enabled	int	其值包括： <ul style="list-style-type: none"> <li>• 0 — 未启用</li> <li>• 1 — 已启用（缺省值）</li> </ul>
jre_args_status	int	留作将来使用。

**索引**

- jre\_args\_directive\_index、jre\_args\_name、jre\_args\_string\_value 上的唯一聚簇索引

## pca\_jre\_directives

说明 存储有关配置 PCA/JVM 所用指令的信息。

列 位于 sybpcidb 中。pca\_jre\_directives 的列包括：

名称	数据类型	说明
jre_directives_index	int	指令的索引。
jre_directives_name	varchar(255)	指令的名称。
jre_directives_description	varchar(255)	指令的文本说明。
jre_directives_enabled	int	其值包括： <ul style="list-style-type: none"> <li>• 0 — 未启用</li> <li>• 1 — 已启用（缺省值）</li> </ul>
jre_directives_status	int	留作将来使用。

索引

- jre\_directives\_name 上的唯一聚簇索引。
- jre\_directives\_index 上的唯一非聚簇索引。

## pci\_arguments

**说明** 存储定义配置 PCI Bridge 所用每个参数的信息。

**列** 位于 sybpcidb 中。pci\_arguments 的列包括：

名称	数据类型	说明
pci_args_directive_index	int	参数所属指令的索引。
pci_args_name	varchar(255)	参数的名称。
pci_args_units	varchar(255)	单元类型。其值包括： <ul style="list-style-type: none"> <li>• switch</li> <li>• number</li> </ul>
pci_args_number_value	int	当 units=number 时，即为该数字的值。如果 units=switch，则值为零 (0)。
pci_args_string_value	varchar(255)	留作将来使用。
pci_args_description	varchar(255)	参数及其用途的简短文本说明。
pci_args_enabled	int	其值包括： <ul style="list-style-type: none"> <li>• 0 — 未启用</li> <li>• 1 — 已启用（缺省值）</li> </ul>
pci_args_status	int	留作将来使用。

**索引**

- pci\_args\_directive\_index 和 pci\_args\_name 上的唯一聚簇索引

## pci\_directives

说明 存储配置 PCI Bridge 的指令。

列 位于 sybpcidb 中。pci\_directives 的列包括：

名称	数据类型	说明
pci_directives_index	int	指令的索引。
pci_directives_name	varchar(255)	指令的名称。
pci_directives_description	varchar(255)	指令的说明。
pci_directives_enabled	int	其值包括： <ul style="list-style-type: none"> <li>• 0 — 未启用</li> <li>• 1 — 已启用（缺省值）</li> </ul>
pci_directives_status	int	留作将来使用。

索引

- pci\_directives\_name 上的唯一聚簇索引
- pci\_directives\_index 上的唯一非聚簇索引

## pci\_slotinfo

**说明** 包含介绍每个槽的信息，包括槽的指令和参数的表名。

**列** 位于 sybpcidb 中。pci\_slotinfo 的列包括：

名称	数据类型	说明
slot_number	int	槽的编号。
slot_name	varchar(255)	槽的名称，例如 JVM。
slot_pca_directives_table_name	varchar(255)	PCA 指令表的名称，例如 pca_jre_directives。
slot_pca_arguments_table_name	varchar(255)	PCA 参数表的名称，例如 pca_jre_arguments。
slot_status	varchar(255)	留作将来使用。

- 索引**
- slot\_name 上的唯一聚簇索引
  - slot\_number 上的唯一非聚簇索引

## pci\_slot\_syscalls

说明

列 位于 sybpcidb 中。pci\_slot\_syscalls 的列包括：

名称	数据类型	说明
syscall_slot_number	int	与系统调用关联的槽号。
syscall_system_call	varchar(255)	系统调用的名称。
syscall_dispatch_name	varchar(255)	系统调用的分派函数的名称。
syscall_enabled	int	其值包括： <ul style="list-style-type: none"> <li>• 0 — 未启用</li> <li>• 1 — 已启用（缺省值）</li> </ul>
syscall_status	int	留作将来使用。

索引

- syscall\_slot\_number、syscall\_system\_call 上的唯一聚簇索引





# 索引

## 符号

- ::= (BNF 表示法)
  - SQL 语句中 xv
- {} (大括号)
  - SQL 语句中 xv
- , (逗号)
  - SQL 语句中 xv
- [] (中括号)
  - SQL 语句中 xv
- () (小括号)
  - SQL 语句中 xv

## 英文

- allow updates to system tables** 配置参数 8
- Backus Naur Form (BNF) 表示法 xv
- dbid* 列, *sysusages* 表 93
- DTX 参与者 26
- ESP。请参见“扩展存储过程”
- grant** 命令
  - sysprotects* 表 67
- hierarchy
  - 数据类型 91
- ID, 服务器角色
  - sysroles* 表 74
- Java 项
  - sysjars* 表 42
  - sysxtypes* 表 96
- master* 数据库
  - 系统表 1-2
- monCachedObject** 表 114
- monCachedProcedures** 表 116
- monCachedStatement** 表 117
- monCachePool** 表 115
- monCIPC** 表 119
- monCIPCEndpoints** 表 120
- monCIPCLinks** 表 121
- monCIPCMesh** 表 122
- monClusterCacheManager** 表 125
- monDataCache** 表 127, 138
- monDeadLock** 表 131
- monDeviceIO** 表 133
- monEngine** 表 134
- monErrorLog** 表 136
- monIOQueue** 表 137
- monLicense** 表 138
- monLocks** 表 139
- monLogicalCluster** 表 141
- monLogicalClusterAction** 表 143
- monLogicalClusterRoute** 163
- monLogicalClusterRoute** 表 145
- monNetworkIO** 表 146
- monOpenDatabases** 表 147
- monOpenObjectActivity** 表 148, 151
- monOpenPartitionActivity** 表 151
- monPCIBridge* 154
- monPCIEngine* 154
- monPCISlots* 155
- monProcedureCache** 表 158
- monProcedureCacheMemoryUsage** 表 159
- monProcedureCacheModuleUsage** 表 160
- monProcess** 表 161
- monProcessActivity** 表 162
- monProcessLookup** 表 163
- monProcessNetIO** 表 164
- monProcessObject** 表 165
- monProcessProcedures** 表 166
- monProcessSQLText** 表 167
- monProcessStatement** 表 168
- monProcessWaits** 表 169, 170
- monProcessWorkerThread** 表 170
- monState** 表 171
- monStatementCache** 表 172
- monSysLoad** 表 173
- monSysPlanText** 表 174
- monSysSQLText** 表 175
- monSysStatement** 表 176

- monSysWaits** 表 178
- monSysWorkerThread** 表 179
- monTableColumns** 表 180
- monTableParameters** 表 181
- monTables** 表 182
- monTableTransfer 183
- monTempdbActivity** 表 184
- monWaitClassInfo** 表 185
- monWaitEventInfo** 表 186
- monWorkload** 表 187
- monWorkloadPreview** 表 188
- monWorkloadProfile** 表 189
- monWorkloadRaw** 表 190
- “probe” 登录帐户 50
- “public” 组 95
- revoke** 命令
  - sysprotects* 表 67
- segmap* 列, *sysusages* 表 93
- segment* 列, *syssegments* 表 76
- spid* 编号 64
  - 在 *sysaudits* 表中 16
  - syslogshold* 中 52
- SQL 语句中的 BNF 表示法 xv
- SQL 语句中的大括号 ({} ) xv
- suid* (服务器用户 ID)
  - sysalternates* 表列出 10
  - syslogins* 表列出 49
- sybdiagdb* 数据库 4
- syblicenseslog* 表 4, 9
- sybpcidb* 数据库
  - pca\_jre\_arguments* 192
  - pca\_jre\_directives* 193
  - pci\_arguments* 194
  - pci\_directives* 195
  - pci\_slot\_syscalls* 197
  - pci\_slotinfo* 196
- sybsecurity* 数据库
  - 中的系统表 2
- sybssystemdb* 数据库
  - 中的系统表 3
- sysalternates* 表 10
- sysaltusages* 表 11
- sysattributes* 表 13–14
- sysauditoptions* 表 15
- sysaudits\_01* — *sysaudits\_08* 表 16–17
- syscharsets* 表 18
- syscharsets* 系统表中的字符集 18
- syscolumns* 表 19–20
- syscomments* 表 21–22
- sysconfigures* 表 23–24
- sysconstraints* 表 25
- syscoordinations* 表 26
- syscurconfigs* 表 27
- sysdatabases* 表 29–31
- sysdepends* 表 32
- sysdevices* 表 33–34
- sysdevices* 中的 *status* 位 33
- sysengines* 表 36
- sysgams* 表 37
- sysindexes* 表 38–40
- sysinstances* 系统表 41
- sysjars* 表 42
- syskeys* 表 43
- syslanguages* 表 44
- syslisteners* 表 45
- syslocks* 表 46–47
- sysloginroles* 表 48
- syslogins* 表 49–50
- syslogs* 表 51
  - 无限循环, 如果更改 51
- syslogshold* 表 52
- sysmessages* 表 53
- sysmonitors* 表 54
- sysobjects* 表 55–58
- syspartitionkeys* 表 60
- sysprocedures* 表 63
- sysprocesses* 表 64–66
- sysprotects* 表 67
- sysquerymetrics* 表 68
- sysqueryplans* 表 70
- sysreferences* 表 71
- sysremotelogins* 表 72
- sysresourcelimits* 表 73
- sysroles* 表 74
- syssecmechs* 表 75
- syssegments* 表 76
- syssservers* 表 77–78
- sysessions* 表 79
- syslices* 表 80
- sysssrvroles* 表 81
- sysstatistics* 表 82
- sysstabstats* 表 83–84

*systhresholds* 表 85  
*systimeranges* 表 87  
*systransactions* 表 88–89  
*systypes* 表 90–92  
*sysusages* 表 93  
*sysusermessages* 表 94  
*sysusers* 表 95  
     *sysalternates* 表 10  
*sysxtypes* 表 96  
*tempdb* 数据库  
     系统表条目 55–58  
 timestamp 列 4  
 us\_english 语言 44

## B

保留列 8  
 标识值 5  
 表

- monCachedObject** 114
- monCachedProcedures** 表 116
- monCachePool** 115
- monCIPC** 119
- monCIPCEndpoints** 120
- monCIPCLinks** 121
- monCIPMesh** 122
- monClusterCacheManager** 125
- monDataCacher** 127
- monDeadILock** 131
- monDeviceIO** 133
- monEngine** 134
- monErrorLog** 136
- monIOQueue** 137
- monLicense** 138
- monLocks** 139
- monLogicalCluster** 141
- monLogicalClusterAction** 143
- monLogicalClusterRoute** 145
- monNetworkIO** 146
- monOpenDatabases** 147
- monOpenObjectActivity** 148
- monOpenPartitionActivity** 151
- monProcedureCache** 158
- monProcedureCacheMemoryUsage** 159
- monProcedureCacheModuleUsage** 160

- monProcess** 161
- monProcessActivity** 162
- monProcessLookup** 163
- monProcessNetIO** 164
- monProcessObject** 165
- monProcessProcedures** 166
- monProcessSQLText** 167
- monProcessStatement** 168
- monProcessWaits** 169
- monProcessWorkerThread** 170
- monState** 171
- monStatementCache** 172
- monSysLoad** 173
- monSysPlanText** 174
- monSysSQLText** 175
- monSysStatement** 176
- monSysWaits** 178
- monSysWorkerThread** 179
- monTableColumns** 180
- monTableParameters** 181
- monTables** 182
- monTempdbActivity** 184
- monWaitClassInfo** 185
- monWaitEventInfo** 186
- monWorkload** 187
- monWorkloadPreview** 188
- monWorkloadProfile** 189
- monWorkloadRaw** 190
- 对象相关性 32
- 系统表条目 19, 55–58
- 别名, 用户
  - sysalternates* 表 10
- 别名, 语言
  - syslanguages* 表 44

## C

参考信息

- dbcc** 表 97
- 系统表 3

参照完整性约束

- sysconstraints* 表 25
- sysobjects* 表 55–58
- sysreferences* 表 71

## 索引

### 查找

- 对象定义 21, 63
  - 对象依赖性 32
  - 角色 74
  - 配置参数 23, 27
  - 权限信息 67
  - 设备名 33
  - 数据库 ID 29
  - 数据库对象 55
  - 数据库名 29
  - 数据库设置 29
  - 数据库中的用户 95
  - 数据类型 90
  - 用户别名 10
  - 语言 44
  - 约束 25
  - 资源限制 73
  - 字符集 18
- 触发器
- 对象相关性 32
  - 系统表 8
  - 系统表条目 21, 55–58, 63
- 磁带转储设备
- sysdevices* 表 33
- 磁盘分配片 93
- 磁盘镜像
- sysdevices* 表中的状态 34
- 磁盘设备
- sysdevices* 表 33
- 存储过程
- 对象相关性 32
  - 系统表条目 21, 55–58, 63
- 存档数据库访问
- sysaltusages* 表 11
  - 空数据库 11
- 错误消息
- 系统表条目 53

## D

- 大小
  - 行 83
- 登录名
  - “probe” 50
  - syslogins* 表 49–50
  - sysremotelogins* 表 72
- 逗号 (,)
  - SQL 语句中 xv
- 段
  - syssegments* 表 76
  - syslices* 表 80
  - 系统表条目 76
- 对象分配映射 (OAM) 页
  - 数目 83

## F

- 分布式事务管理 (DTM) 26
- 分配单元
  - sysusages* 表 93
- 分区
  - 系统表条目 80
- 符号
  - SQL 语句中 xv

## G

- 高可用性
  - 重新连接信息 79
- 更新
  - 系统表 8
  - 系统过程 8
  - 直接对系统表 8
- 工作空间
  - 删除 98
- 公用键
  - syskeys* 表 43
- 规则
  - 系统表条目 21, 55–58, 63

**H**

- 行, 表
  - 大小 83
- 行, 数据
  - 数目 83
- 行, 索引
  - 大小 83
  - 叶大小 83

**J**

- 计划
  - 对象 63
- 加密
  - 角色口令 81
  - 用户口令 49
- 监控表 113–166
  - monLogicalClusterRoute 163
  - 描述 andisted. 列表和说明, 监控表 113
- 检查约束
  - sysconstraints 表 25
  - 系统表条目 55–58, 63
- 键, 表
  - syskeys 表 43
- 角色
  - sysroles 表 74
  - sysserverroles 表 81
- 角色, 系统
  - sysloginroles 表中 48
- 结构
  - 配置 27
- 进程 (服务器任务)
  - sysprocesses 表 64
  - 系统表条目 64

**K**

- 客户端
  - 在故障恢复时删除 79
- 空间分配
  - sysusages 表 93
  - 系统表条目 93
- 空数据库 11
- 括号。请参见 中括号 []
- 扩展存储过程
  - 系统表条目 21, 55–58

**L**

- 两阶段提交
  - 探查进程 50
- 列
  - 保留 8
- 列表
  - 系统表 1–4

**M**

- 美式英语, 请参见 us\_english 语言名称
  - 排序顺序 18
  - 字符集 18

**P**

- 排序顺序
  - syscharsets 系统表 18
- 配置参数
  - 系统表 23, 27

## 索引

### Q

- 区分大小写
  - 在 SQL 中 xvi
- 权限
  - sysprotects* 表 67
  - 系统表 6
  - 系统表条目 67
- 全局分配映射页 37
- 全局审计选项, *sysauditoptions* 系统表 15
- 缺省值
  - 系统表条目 21, 55–58, 63

### S

- 删除
  - 工作空间 98
- 设备
  - 系统表条目 33
- 审计
  - sysauditoptions* 表 15
  - sysaudits\_01* — *sysaudits\_08* 表 16, 192
- 时间范围
  - systemranges* 系统表 87
- 事务 88
- 事务日志
  - 系统表条目 55–58
- 视图
  - 对象相关性 32
  - 系统表条目 19, 21, 55–58, 63
- 数据库
  - 系统表条目 29
- 数据库对象
  - sysobjects* 表 55–58
  - 相关性 32
- 数据库设备
  - sysdevices* 表 33
  - 系统表条目 33
- 数据类型
  - hierarchy* 91
  - systypes* 表 90–92
  - 列表 91

- 数据行
  - 大小 83
- 数目 (数量)
  - OAM 页 83
  - 索引级 83
  - 索引叶页 83
  - 行 83
  - 页 83
  - 已删除的行 83
  - 转移的行 83
- 索引
  - 系统表条目 38
- 锁
  - 系统表条目 46

### T

- 探查进程, 两阶段提交 50
- 统计信息
  - 系统表 82, 83

### W

- 外键
  - syskeys* 表 43

### X

- 系统表 1–9
  - allow updates to system tables** 参数 8
  - master* 数据库 1–2
  - 触发器 8
  - 更新 8
  - 键 43
  - 权限 6
  - 直接更新 8
- 系统过程
  - 更新 8
- 系统角色
  - sysloginroles* 表 48
  - sysserverroles* 表 81

相关性, 数据库对象  
   *sysdepends* 表 32  
 消息  
   *sysmessages* 表 53  
   *sysusermessages* 表 94  
   用户定义的 94  
 小括号 ()  
   SQL 语句中 xv  
 信息 (服务器)  
   配置参数 23, 27  
   数据库 29–31  
 虚设表实现 5  
 序列树, 对象 63  
 循环  
   *syslogs* 更改和无限 51

## Y

页, 全局分配映射 37  
 页, 数据  
   数目 83  
 页, 索引  
   数目 83  
 已删除的行  
   数目 83  
 引擎  
   *sysengines* 表 36  
   系统表条目 36  
 映射  
   *sysusages* 表 93  
 用户  
   *syslogins* 表 49–50  
   *sysusers* 表 95  
   系统表条目 49–50, 95  
 用户定义角色  
   *sysrvroles* 表 81  
 语法约定, Transact-SQL xv  
 语言, 替代  
   *syslanguages* 表 44  
   系统表条目 44  
 阈值  
   *systhresholds* 表 85  
 远程登录名

*sysremotelogins* 表 72  
 系统表条目 72  
 远程服务器  
   *syssservers* 表 77  
   系统表条目 77  
 远程过程调用  
   *sysremotelogins* 表 72  
   *syssservers* 表 77  
 约定  
   Transact-SQL 语法 xv  
   另请参见 语法  
   在参考手册中使用 xv  
 约束  
   *sysconstraints* 表 25  
   *sysreferences* 表 71  
   系统表条目 21, 55–58  
 月份值  
   *syslanguages* 表 44  
   简称 (缩写) 44  
   替代语言 44

## Z

直接更新  
   系统表 8  
 指定的时间范围  
   *systimeranges* 系统表 87  
 中括号 []  
   SQL 语句中 xv  
 主键  
   *syskeys* 表 43  
 转储设备  
   *sysdevices* 表 33  
   系统表条目 33  
 转移的行  
   数目 83  
 资源限制  
   *sysresourcelimits* 表 73  
 阻塞进程 64  
 组  
   *sysusers* 表条目 95

