



新增功能指南

---

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

## 16.0

文档 ID: DC01057-01-1600-01

最后修订日期: 2014 年 3 月

©2014 SAP 股份公司或其关联公司版权所有, 保留所有权利。

未经 SAP 股份公司明确许可, 不得以任何形式或为任何目的复制或传播本文的任何内容。本文包含的信息如有更改, 恕不另行事先通知。

由 SAP 股份公司及其分销商营销的部分软件产品包含其它软件供应商的专有软件组件。各国的产品规格可能不同。

上述资料由 SAP 股份公司及其关联公司(统称“SAP 集团”)提供, 仅供参考, 不构成任何形式的陈述或保证, 其中如若存在任何错误或疏漏, SAP 集团概不负责。与 SAP 集团产品和服务相关的保证仅限于该等产品和服务随附的保证声明(若有)中明确提出之保证。本文中的任何信息均不构成额外保证。

SAP 和本文提及的其它 SAP 产品和服务及其各自标识均为 SAP 股份公司在德国和其他国家的商标或注册商标。

如欲了解更多商标信息和声明, 请访问: <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>。

# 目录

<b>版本 16.0 概述</b> .....	<b>1</b>
<b>分区锁定提高数据可用性</b> .....	<b>3</b>
分区锁 .....	3
分区锁名称 .....	4
启用分区锁定 .....	4
通过 <code>sp_lock</code> 查看分区锁 .....	4
通过 <code>sp_familylock</code> 查看分区锁 .....	5
通过 <code>monLocks</code> 和 <code>monDeadLock</code> 查看分区锁 .....	6
死锁和锁超时 .....	6
分区锁升级 .....	6
设置分区锁升级阈值 .....	7
删除分区锁升级阈值 .....	7
锁兼容性和锁充分性 .....	7
分区锁兼容性 .....	7
分区锁充分性 .....	8
模式锁兼容性 .....	8
模式锁充分性 .....	9
分区级联机操作的并发性得以改进 .....	9
分区级联机操作语法 .....	9
分区级联机操作的并发性 .....	10
包含全局索引的分区级联机操作 .....	10
模式锁 .....	11
<b>HANA 服务器的组件集成服务支持</b> .....	<b>13</b>
配置 HANA 的 CIS .....	13
在 Windows 上创建 SAP HANA 作为 ODBC 数 据源 .....	13
将 SAP HANA 添加到 SAP ASE Interfaces 文件 中 .....	13
配置 PCI Bridge 和 PCA/ODBC .....	14
添加服务器类 HANAODBC .....	16
SAP ASE 与 HANA 之间的数据类型映射 .....	16

限制 .....	18
<b>宽泛查询限制 .....</b>	<b>21</b>
<b>使用星型连接优化查询计划 .....</b>	<b>23</b>
星型连接提示 .....	24
<b>use fact_table</b> 提示下的星型连接查询计划 .....	25
<b>查询性能改进 .....</b>	<b>33</b>
动态线程分配 .....	33
<b>SORT</b> 运算符性能改进 .....	33
<b>Hash Join</b> 运算符性能改进 .....	35
<b>全文本审计 .....</b>	<b>39</b>
<b>存储过程内部授权检查的审计 .....</b>	<b>41</b>
<b>替换对象定义 .....</b>	<b>43</b>
安装脚本更改 .....	44
数据和日志段更改 .....	44
<b>HTML 格式的查询计划和执行统计信息 .....</b>	<b>47</b>
已生成文件的前缀选项 .....	49
<b>索引压缩 .....</b>	<b>51</b>
启用索引压缩 .....	52
创建按索引压缩的表 .....	52
创建压缩索引 .....	53
更改压缩状态 .....	53
<b>SAP JVM 支持 .....</b>	<b>55</b>
<b>完全数据库加密 .....</b>	<b>57</b>
完全数据库加密与加密列 .....	57
创建数据库加密密钥 .....	57
更改数据库加密密钥 .....	59
删除数据库加密密钥 .....	59
创建加密数据库 .....	60
加密现有数据库 .....	61
加密状态和进度 .....	63
性能注意事项 .....	63
挂起加密进程 .....	66
<b>quiesce database</b> 命令和完全加密数据库 .....	66
恢复加密进程 .....	67

解密加密数据库 .....	67
恢复完全加密数据库 .....	68
备份（转储）完全加密数据库 .....	68
备份数据库加密密钥 .....	68
恢复（装载）完全加密数据库的备份 .....	69
加密数据库的装载行为 .....	69
删除正在进行加密的数据库 .....	70
装入和卸下完全加密数据库 .....	70
存档数据库和完全加密 .....	71
完全数据库加密和系统更改 .....	71
<b>可伸缩性增强和功能 .....</b>	<b>73</b>
运行时日志记录增强功能 .....	73
锁管理增强 .....	74
元数据和门锁管理增强 .....	74
<b>监控基于阈值的事件 .....</b>	<b>77</b>
<b>多个触发器 .....</b>	<b>79</b>
创建多个触发器 .....	79
更改触发器引发顺序 .....	80
Merge 语句中触发器的顺序 .....	80
多个触发器的事务型行为 .....	80
禁用和重新启用触发器 .....	81
@@trigger_name 全局变量 .....	81
<b>删除残留数据 .....</b>	<b>83</b>
<b>配置历史记录跟踪 .....</b>	<b>85</b>
配置 SAP ASE 以跟踪配置更改 .....	85
捕获的更改 .....	86
查询 ch_events 以查看更改 .....	89
<b>对 dump database 的循环冗余检查 .....</b>	<b>93</b>
<b>计算事务日志增长率 .....</b>	<b>95</b>
<b>系统更改 .....</b>	<b>97</b>
配置参数 .....	97
新配置参数 .....	97
更改过的配置参数 .....	101
内置函数 .....	101

dbencryption_status .....	101
命令 .....	102
alter database 用于完全数据库加密 .....	102
alter index .....	105
alter table 用于索引压缩 .....	105
alter table 用于多个触发器 .....	107
alter table 用于残留数据删除 .....	107
create archive database 用于完全数据库加密 ..	108
create database 用于完全数据库加密 .....	108
create default .....	109
create encryption key .....	111
create function .....	113
create function (SQLJ) .....	116
create index .....	118
create procedure .....	119
create procedure (SQLJ) .....	122
create rule .....	124
create table 用于索引压缩 .....	126
create table 用于残留数据删除 .....	128
create trigger 用于创建多个触发器 .....	130
create trigger 用于 or replace .....	131
create view .....	133
drop encryption key .....	136
drop trigger .....	137
dump database .....	137
kill .....	137
load database .....	138
select .....	138
select into .....	139
set .....	139
系统过程 .....	141
已更改的系统过程 .....	141
新的系统过程 .....	148
系统表 .....	160
更改过的系统表 .....	160

新的系统表 .....	162
更改过的监控表 .....	164
新的监控表 .....	167
实用程序 .....	168
ddlgen .....	169
sybmigrate .....	170
sybrestore .....	171
全局变量 .....	171

# 目录



## 版本 16.0 概述

SAP® Adaptive Server® Enterprise 16.0 属于主版本，包括以下方面的重要增强：纵向扩展和性能；大型数据管理；数据可用性提升；安全性和审计；管理和维护的易操作性。

- 纵向扩展和性能改进包括为最大限度减少争用而进行的缓冲区管理优化。
- 元数据和门锁管理改进考虑了高吞吐量的情况
- 通过分区级别锁定提高了数据可用性，允许对表执行并发 DDL 和 DML
- 大数据管理的改进基于 SAP® ASE 15.7 版本的相关功能，并在此基础上增加了宽泛查询限制、星型连接查询计划优化以及包含 SORT 运算符性能改进在内的相关查询性能增强。
- 新增了数据压缩方法，包括索引压缩、页级和列级压缩以及之前版本中增加的其他压缩技术。
- SCC 的增强包括：监控和管理改进；基于阈值的事件、HTML 格式的查询计划和执行统计信息的增强；顾问增强，可帮助您做出更明智的数据决策的顾问（尤其在使用压缩时）。

功能	说明
<b>可伸缩性和性能功能</b>	
线性递增	支持线性递增为 64 个内核
动态线程支持	并行查询计划的执行速度更快，使用的资源更少
索引压缩	更有效的数据存储，减少内存消耗，并且由于 I/O 需要降低而改善了性能。
支持多个触发器	创建多个触发器，并指定执行语句后触发器的引发顺序
分区锁	通过提供更细粒度的锁定来提高数据可用性，从而允许访问并发 DDL 和 DML 语句的其它分区。
配置历史记录跟踪	使用 <code>sp_confighistory</code> 系统过程跟踪服务器配置更改
基于阈值的事件监控表	配置、记录并列出阈值事件
增加执行时间监控	对于 SAP ASE 记录的每个事件，monThresholdEvent 监控表中分别有一行与之对应。
monCachedStatement 更新	针对已完成和正在进行的查询，大约每隔 5 秒更新所选列的指标
安装程序增强	为 SAP ASE 流程指定用户
提高数据库限制	增加某些列和表的限制
<b>create or replace 命令</b>	创建新对象或替换同名的现有对象

功能	说明
<b>安全和审计功能</b>	
将整个数据库加密	加密整个数据库。
删除残留数据	将数据标记为敏感性数据，并将 SAP ASE 配置为在执行删除或更新操作后从磁盘删除残留数据。
全文本审计	<p>针对以下命令的全文本审计信息：</p> <ul style="list-style-type: none"> <li>• <b>select</b></li> <li>• <b>insert</b></li> <li>• <b>delete</b></li> <li>• <b>update</b></li> <li>• <b>select into</b></li> </ul>
<b>与 HANA 的互操作性</b>	
针对 HANA 的 CIS 支持	向组件集成服务 (CIS) 中添加本机 ODBC 接口，这样您就可以从 SAP ASE 直接连接到 HANA 服务器
<b>SCC 功能</b>	
安排数据库和事务备份	安排数据库和事务备份
压缩顾问	识别适合压缩的表并获取压缩估计信息和建议。
系统配置限制警报	配置特定资源超出所配置阈值时要引发的警报。
可伸缩性	最多管理 250 个服务器。
<b>工具支持</b>	
<b>sybrestore</b> 增强	主数据库损坏后恢复 SAP ASE 服务器。交互式和非交互式模式的新参数。

# 分区锁定提高数据可用性

分区级锁定功能通过创建允许访问其它分区以执行并发 DDL 和 DML 语句的更精细的锁定粒度来提高数据可用性。

## 分区锁

分区锁是分区表的一种属性，适用于所有类型的分区。

- 意图分区锁 - 表示表分区持有页级或行级锁。SAP ASE 将意图分区锁与一个共享分区锁或排它分区锁结合应用，因此意图锁可为排它锁或共享锁。设置意图锁可阻止其它事务获取表上相互冲突的分区级锁，因为这些分区锁对该事务均有效。
- 共享分区锁 - 与共享页锁或行锁相似，只是它影响整个分区。例如，如果带有 **holdlock** 子句的 **select** 命令未使用索引，则 SAP ASE 会为其应用一个共享分区锁。**create nonclustered index** 命令也将获取共享分区锁。
- 排它分区锁 - 与排它页锁或行锁相似，只是它影响整个分区。例如，SAP ASE 会在执行 **create clustered index** 命令期间应用排它分区锁。如果 **update** 和 **delete** 语句的搜索参数不参考对象的索引列，则在仅数据锁定分区上执行这些语句时需要排它分区锁。
- 覆盖分区锁 - 如果 DDL 和 DML 在获取精细锁时不知道 *partitionid*，则需要使用这种锁。

有关其它锁类型的信息，请参见《性能和调优系列：锁定和并发控制》。

以下示例表显示 SAP ASE 用于基本 SQL 语句的页锁或行锁的相应分区锁。对于这些示例，*acct\_number* 上有一个索引。

语句	所有页锁定表	数据行锁定表
<code>select balance from account where acct_number = 25</code>	共享意图分区锁 共享页锁	共享意图分区锁 共享行锁
<code>insert account values (34, 500)</code>	排它意图分区锁 数据页上的排它页锁 叶索引页上的排它页锁	排它意图分区锁 排它行锁
<code>delete account where acct_number = 25</code>	排它意图分区锁 数据页和叶级索引页上的更新页锁和随后的排它页锁	排它意图分区锁 数据行上的更新行锁和随后的排它行锁

语句	所有页锁定表	数据行锁定表
<pre>update account set balance = 0 where acct_num- ber = 25</pre>	排它意图分区锁 数据页和叶级索引页上的更新页锁和随后的排它页锁	排它意图分区锁 数据行上的更新行锁和随后的排它行锁。

## 分区锁名称

SAP ASE 结合数据库 ID (*dbid*)、对象 ID (*objid*) 和分区 ID (*ptnid*) 唯一标识分区锁。

### 未知分区锁

对于未知分区锁，SAP ASE 使用分区 ID 值为“-1”的特殊分区锁。它表示所有表分区上的分区锁。在特定情况下，SAP ASE 可能不知道数据行的分区 ID，因此会获取分区 ID 为“-1”的分区锁。

## 启用分区锁定

使用 **sp\_chgattribute** 启用或禁用分区级锁定。缺省情况下，分区锁定处于禁用状态。

只能在包含多个数据分区的用户表上启用分区级锁定。无法对系统表和临时表启用分区级锁定。

语法为：

```
sp_chgattribute objectname, 'ptn_locking', value
```

参数：

*objectname* - 要更改 **ptn\_locking** 的表的名称。

*value* - 要启用分区级锁定则设为 1，要禁用则设为 0。

权限：

只有对象所有者才能执行 **sp\_chgattribute**。

### 示例

此示例启用 *authors* 表的分区级锁定：

```
sp_chgattribute authors, "ptn_locking", 1
```

此示例禁用 *authors* 表的分区级锁定：

```
sp_chgattribute authors, "ptn_locking", 0
```

## 通过 **sp\_lock** 查看分区锁

使用 **sp\_lock** 查看当前由 SAP ASE 持有的分区锁。

在 **sp\_lock** 报告的 *partitionid* 列中查看已锁定的分区。

此示例显示当前由 SAP ASE 持有的锁，包括分区锁。

```

sp_lock
go

fid spid loid locktype table_id partitionid page row dbname class
context
-----
-----
0 13 26 Ex_intent 420193516 0 0 0 master Non Cursor Lock
0 13 26 Ex_intent_partition 420193516 452193630 0 0 master Non
Cursor Lock
0 13 26 Ex_page 420193516 452193630 4993 0 master Non Cursor Lock
0 14 28 Ex_intent 420193516 0 0 0 master Non Cursor Lock
0 14 28 Ex_intent_partition 420193516 468193687 0 0 master Non
Cursor Lock
0 14 28 Ex_page 420193516 468193687 5001 0 master Non Cursor Lock
0 16 32 Sh_intent 1006623598 0 0 0 master Non Cursor Lock

```

### 另请参见

- `sp_lock` (第 148 页)

## 通过 `sp_familylock` 查看分区锁

使用 `sp_familylock` 显示由系列持有的分区锁。

在 `sp_familylock` 报告的 `partitionid` 列中查看已锁定的分区。

此示例显示当前由系列持有的分区锁。 `class` 列显示与游标相关的锁的游标名称 (对于当前用户) 和游标 ID (对于其他用户)。

```

sp_familylock
go

Fid spid loid locktype table_id partitionid page dbname class context
-----
-----
-----
25 19 50 Sh_cpartition 672002394 -1 0 userdb Non Cursor Lock LOCK
CONTEXT VALUES
25 19 50 Sh_partition 672002394 688002451 0 userdb Non Cursor Lock
LOCK CONTEXT VALUES
25 20 50 Sh_cpartition 672002394 -1 0 userdb Non Cursor Lock LOCK
CONTEXT VALUES
25 20 50 Sh_intent_partition 672002394 688002451 0 userdb Non Cursor
Lock LOCK CONTEXT VALUES
25 20 50 Sh_partition 672002394 704002508 0 userdb Non Cursor Lock
LOCK CONTEXT VALUES
25 25 50 Sh_intent 672002394 0 0 userdb Non Cursor Lock LOCK CONTEXT
VALUES

(6 rows affected)
(return status = 0)

```

分区锁定提高数据可用性

另请参见

- `sp_familylock` (第 146 页)

## 通过 `monLocks` 和 `monDeadLock` 查看分区锁

在 `monLocks` 和 `monDeadLock` 监控表的 `partitionid` 列中查看已锁定的分区。

### `monLocks`

说明	数据类型	属性	说明
<code>partitionid</code>	<code>int</code>	Null	分区的唯一标识符。

### `monDeadLock`

说明	数据类型	属性	说明
<code>partitionid</code>	<code>int</code>	Null	分区的唯一标识符。

请参见《性能和调优系列： 监控表》。

## 死锁和锁超时

分区锁的死锁检测和锁超时采用与表锁相同的方式执行。

请参见《性能和调优系列： 锁定和并发控制》。

## 分区锁升级

表锁升级会将锁从精细锁升级为表锁。分区锁升级会将锁从精细锁升级为分区锁。

要执行分区锁升级，请将分区锁升级阈值设为非零值。

分区锁使用与表锁相同的语义。如果页锁或行锁属于在单次分区扫描中超出锁升级阈值的分区，则可触发向分区锁的锁升级。

如果获取的页锁或行锁的分区未知，则会完全禁用向分区锁的锁升级。在这些情况下，锁只能升级至表级别。

SAP ASE 支持两种不同类型的锁升级：

- 行锁或页锁升级为表锁 - 如果任务在单次表扫描中获取的锁所在的行或页的数量超出表级锁升级阈值，则 SAP ASE 会尝试获取相应表上的共享表锁或排它表锁，并替换该表中的全部现有行或页。如果获取的行锁或页锁属于覆盖分区，而所属分区对于锁升级未知，则也会触发表级锁升级。
- 行锁或页锁升级为分区锁 - 在分区锁升级期间，将获取相应的共享或排它分区锁，并释放所有已通过扫描（或 DML）获取且属于该分区的精细锁。
  - 分区锁升级会将共享精细锁升级为共享分区锁。

- 分区锁升级会将排它精细锁升级为排它分区锁。

## 设置分区锁升级阈值

**sp\_setpglockpromote\_ptn** 和 **sp\_setrowlockpromote\_ptn** 系统过程设置服务器、数据库和表级别的分区锁升级阈值。

另请参见

- **sp\_setpglockpromote\_ptn** (第 157 页)
- **sp\_setrowlockpromote\_ptn** (第 158 页)

## 删除分区锁升级阈值

**sp\_dropglockpromote\_ptn** 和 **sp\_droprowlockpromote\_ptn** 系统过程删除服务器、数据库和表级别的分区锁升级阈值。

另请参见

- **sp\_dropglockpromote\_ptn** (第 150 页)
- **sp\_droprowlockpromote\_ptn** (第 152 页)

## 锁兼容性和锁充分性

锁兼容性和锁充分性是支持锁定和并发问题的两个基本概念。

- 锁兼容性 - 如果任务持有任意资源（例如行、页、分区或表）上的锁，另一项任务能否同样持有该资源上的锁？
- 锁充分性，对于当前任务 - 如果任务需要再次访问任意资源（例如行、页、分区或表），当前在该资源上持有的锁是否充分？

请参见《性能和调优系列： 锁定和并发控制》>“锁定简介”。

## 分区锁兼容性

下表总结了关于分区锁兼容性的信息，说明了什么时候可以立即获取锁。

持有锁类型：	另一进程能否获取：					
	排它分区锁？	共享分区锁？	排它意图分区锁？	共享意图分区锁？	排它覆盖分区锁？	共享覆盖分区锁？
排它分区锁	否	否	否	否	否	否
共享分区锁	否	是	否	是	否	是
排它意图分区锁	否	否	是	是	否	否

## 分区锁定提高数据可用性

	另一进程能否获取：					
持有锁类型：	排它分区锁？	共享分区锁？	排它意图分区锁？	共享意图分区锁？	排它覆盖分区锁？	共享覆盖分区锁？
共享意图分区锁	否	是	是	是	否	是
排它覆盖分区锁	否	否	否	否	是	是
共享覆盖分区锁	否	是	否	是	是	是

## 分区锁充分性

分区锁充分性表格。

	如果任务需要以下锁，原有锁是否充分：					
持有锁类型：	排它分区锁？	共享分区锁？	排它意图分区锁？	共享意图分区锁？	排它覆盖分区锁？	共享覆盖分区锁？
排它分区锁	是	是	是	是	是	是
共享分区锁	否	是	否	是	否	是
排它意图分区锁	否	否	是	是	否	否
共享意图分区锁	否	否	否	是	否	否
排它覆盖分区锁	否	否	否	否	是	是
共享覆盖分区锁	否	否	否	否	否	是

## 模式锁兼容性

下表总结了关于模式锁兼容性的信息，说明了什么时候可以立即获取锁。

	另一进程能否获取：	
持有锁类型：	排它模式锁？	共享模式锁？
排它模式锁	否	否
共享模式锁	否	是



## 模式锁充分性

模式锁充分性表格。

	如果任务需要以下锁，原有锁是否充分：	
持有锁类型：	排它模式锁？	共享模式锁？
排它模式锁	是	是
共享模式锁	否	是

## 分区级联机操作的并发性得以改进

特定分区级操作可在表的不同分区上并发执行。DML 也可以在分区级联机操作运行期间在表上并发执行。

其中包括：

- **alter table ... split partition**
- **alter table ... merge partition**
- **alter table ... move partition**
- **alter table ... drop partition**
- **truncate partition**
- **dbcc checkindex**
- **dbcc checktable**
- **dbcc tablealloc**
- **dbcc indexalloc**

## 分区级联机操作语法

为提高数据可用性，**alter table** 和 **truncate partition** 命令在 **partition** 子句中包括 **with online** 子句。

必须先启用分区锁定。

**alter table** 的语法为：

```
alter table table_name
{merge | drop | move | split} partition partition_name
existing_clause
with online
```

其中：

*table\_name* - 要修改的表的名称。

*partition\_name* - 指定要合并、删除、移动或拆分的分区的名称。

*existing\_clause* - **partition** 子句，具体取决于指定的分区操作。

**with online** - 在联机模式下执行。启用对表的并发访问。

**truncate partition** 的语法为：

```
truncate table table_name  
    [partition partition_name]  
    [with online]
```

其中：

*table\_name* - 要截断的表的名称。

*partition\_name* - 指定要截断的分区的名称。

**with online** - 在联机模式下执行。启用对表的并发访问。

## 分区级联机操作的并发性

可在表的不同分区上并发执行多个分区级联机操作。

所有 DML - 即 **select**（但不是 **select into**）、**insert**、**update** 和 **delete** - 可在分区级联机操作过程中在表上执行。

DML 可在所有分区上执行，但不包括正处于分区级联机操作过程中的分区。

通过使 DML 合理使用谓词，SAP ASE 能够充分利用分区排除技术。这样，DML 便可在所需的最小分区集上执行。

若不进行分区排除，DML 则在表的所有分区上执行。

如果 DML 在正处于并发分区级操作过程中的分区上执行，则 DML 将被中止。

**注意：**要避免并发 DML 导致错误，可通过适当的谓词编写 DML 以利用分区排除技术。这样，DML 便可在最小分区集上执行。

对于拆分分区或移动分区等分区级联机操作，表必须具有唯一本地索引。允许并发 DML 对表中的所有分区执行移动命令，包括正在进行拆分的分区。

## 包含全局索引的分区级联机操作

包含全局索引的表的分区级操作行为。

如果正在对表执行分区级操作，则对表的并发 DML 访问不使用全局索引。SAP ASE 将使用替代本地索引扫描或表扫描进行并发 DML 扫描。

可以在不同分区上并发执行多项分区级操作。每项操作都不会执行全局索引重建。只有最后提交的分区级操作会执行全局索引重建。

导致中止的最后一项分区级操作可能会使全局索引被标记为可疑。在此情况下，必须显式重建全局索引。

## 模式锁

借助模式锁，通过实现与并发操作的隔离，增强的分区级操作可以更新表模式或元数据。

新的模式锁包括：

- 共享模式锁 - 表示任务使用表的当前模式执行查询。扫描和 **DML** 在查询开始执行前获取此锁。
- 排它模式锁 - 表示任务将更改表的模式。分区级操作获取此锁以更新表的模式。

分区锁定提高数据可用性

# HANA 服务器的组件集成服务支持

SAP ASE 16.0 为组件集成服务 (CIS) 添加了本机 ODBC 接口，这样便可直接从 SAP ASE 连接到 HANA 服务器。

有关使用 CIS 连接到远程服务器的信息，请参见《组件集成服务用户指南》。

## 配置 HANA 的 CIS

---

必须在运行 SAP ASE 的同一计算机上安装包含 ODBC 驱动程序的 SAP HANA 客户端软件包。

请参见 SAP HANA 数据库文档中的《客户端安装和更新指南》(Client Installation and Update Guide)。

## 在 Windows 上创建 SAP HANA 作为 ODBC 数据源

ODBC 管理程序可作为在 Windows 平台上创建和管理 ODBC 数据源的中心位置。

当使用 32 位 ODBC 驱动程序时，请使用 32 位 ODBC 管理程序来管理数据源；使用 64 位客户端时，请使用 64 位 ODBC 管理程序。

1. 从 Windows “开始” 窗口中选择 “设置” > “控制面板” > “管理工具” > “数据源” (ODBC)。
2. 单击 “添加”。
3. 从驱动程序列表中选择 “HDBODBC”。
4. 单击 “完成”。
5. 在 “ODBC 配置” (ODBC Configuration) 窗口中，输入 “数据源名称” (Data Source Name) 和 “服务器:端口” (Server:Port)。例如，要使用 TCP/IP 协议和端口 1870 在地址为 157.133.66.75 的主机 "hanaserver" 上连接到名为 "HANA\_DB" 的服务器，请选择 TCP/IP 然后输入：

```
host=hanaserver:1870
```

要使用主机网络地址，请输入：

```
host=157.133.66.75:1870
```

6. 单击 “确定”。

## 将 SAP HANA 添加到 SAP ASE Interfaces 文件中

必须先将 HANA 服务器条目添加到 SAP ASE Interfaces 文件中才能与其连接。

使用 vi 或类似的文本编辑器，将 SAP HANA 服务器的条目添加到 interfaces 文件或 sql.ini 文件中。

以下示例使用端口号 1870 和 libodbcHDB.so SAP HANA ODBC 驱动程序在 UNIX 上添加 HANA\_DB 服务器的条目：

```
HANA_DB
query tcp ether 157.133.66.75 1870 libodbcHDB.so
```

以下示例使用 HDBODBC（用于 Windows 32 位的 HDBODBC32）驱动程序在 Windows 上添加已在 Windows 注册表中注册的 HANA\_DB 服务器的条目：

```
[HANA_DB]
Query=NLWNSCK,157.133.66.75,1870,HDBODBC
```

---

**注意：** 由于 PCI Bridge 使用 HDBODBC 作为缺省值，因此不需要在 Windows 的 interfaces 文件中 (sql.ini) 包括 HDBODBC 条目。

---

### 配置 PCI Bridge 和 PCA/ODBC

当 SAP ASE 调用目标函数时，会使用可实现按需分配软件功能的可插入组件接口 (PCI) Bridge 来装载共享对象。

通常从配置有 PCI Bridge 的 ODBC 可插入组件适配器 (PCA/ODBC) 引导 ODBC 驱动程序管理器。PCA/ODBC 充当代理角色，管理 SAP ASE 和 ODBC 驱动程序管理器之间的服务请求。PCA/ODBC 可以双向转发和控制请求：在 SAP ASE 和 ODBC 驱动程序管理器之间双向进行。

ODBC 驱动程序管理器组件与 SAP ASE 不相关。可以随时更改或升级这些软件组件而无需升级或重建 SAP ASE。

---

**注意：** 必须先为 PCA/ODBC 启用 PCI Bridge，它才能使用 ODBC 驱动程序管理器。当您启用 PCI Bridge 时，SAP ASE 需要 sybpcidb 数据库。sybpcidb 包含 PCI Bridge 及其相关 PCA（如 PCA/ODBC）的所有配置数据。

---

1. 由系统管理员创建 sybpcidb 数据库，方法是使用 **srvbuild** 或 **sybconfig** 安装和配置实用程序或者通过运行 `${SYBASE}/${SYBASE_ASE}/scripts/installpcidb` 脚本（请参见《安装指南》）。

2. 启用 PCI Bridge:

```
sp_configure 'enable pci', 1
```

请参见《系统管理指南，卷 1》中的“设置配置参数”。

3. 重启 SAP ASE。

4. 使用 **sp\_odbcconfig** 配置这些 PCA/ODBC 参数：

- **pca\_odbc\_load\_dir** -（仅限 UNIX）设置到 ODBC 驱动程序位置的绝对路径。**pca\_odbc\_load\_dir** 创建多个位置的数组。也就是说，如果计算机上安装了多个 ODBC 驱动程序，**pca\_odbc\_load\_dir** 将创建它们的数组。

**pca\_odbc\_load\_dir** 的缺省值为 `/usr/sap/hdbclient`。

- a. 切换到 sybpcidb 数据库：

```
use sybpcidb
```

- b. 为 `pca_odbc_load_dir` 设置路径:

```
sp_odbcconfig 'add', 'pca_odbc_load_dir',
'path_to_driver_location'
```

- c. 对 `pca_odbc_load_dir` 进行更改后重启 SAP ASE。

要删除不再使用的驱动程序:

```
use sybpcidb
go
sp_odbcconfig 'delete', 'pca_odbc_load_dir',
'path_to_driver_location'
```

---

**注意:** 在 Windows 上, ODBC 驱动程序在驱动程序组件安装过程中由系统注册表管理和维护。请参见 Windows 平台文档和特定驱动程序供应商文档。

---

- `pca_odbc_root_dir` - UNIX 系统将 ODBC 数据源存储在 `odbc.ini` 文件中。`pca_odbc_root_dir` 指示包含 `odbc.ini` 的目录路径。`pca_odbc_root_dir` 的缺省值为 `$$SYBASE`。

`odbc.ini` 中 HANA\_DB 服务器所对应条目的格式为:

```
[HANA_DB]
DRIVER = libodbcHDB.so
SERVERNODE = hanaserver:1870
```

---

**注意:** 在 Windows 上, 使用 ODBC 注册表来配置 ODBC 数据源。请参见您的 Windows 文档。

---

在 UNIX 系统上, 使用 `sp_odbcconfig` 为 `pca_odbc_root_dir` 设置值:

- a. 切换到 `sybpcidb` 数据库:

```
use sybpcidb
```

- b. 输入新路径:

```
sp_odbcconfig 'update', 'pca_odbc_root_dir', 'old_path',
'new_path'
```

其中 `new_path` 指示 `$$SYBASE` 的相对位置。用正斜线作为 `new_path` 的前缀以指示到根目录的绝对位置。

---

**注意:** 如果 ODBC 驱动程序已经在 PCA/ODBC 的控制下装载, 则必须重启 SAP ASE。只有在启用 `pca_odbc_dsn_enabled` 后, SAP ASE 才会参考存储在 `odbc.ini` 中的 DSN 记录。

---

- `pca_odbc_dsn_enabled` - 为 PCA/ODBC 子系统启用或禁用 DSN 模式。值为:
    - 0 - 禁用 DSN (缺省值)。SAP ASE 从 `interfaces` 文件中自动收集来自 SAP 目录控制层 (DCL) 的 DSN 信息。
    - 1 - 启用 DSN。SAP ASE 参考存储在 `odbc.ini` 中的 DSN 记录。
5. (仅限 UNIX) 将 `$$SYBASE/DataAccess64/ODBC/dm/lib64` 添加到 `LD_LIBRARY_PATH`。

## 添加服务器类 HANAODBC

---

SAP ASE 16.0 版本添加了服务器类 HANAODBC，这样您可使用 ODBC 连接到 SAP HANA 服务器。

必须在运行 SAP ASE 的同一计算机上安装 HANA ODBC 驱动程序。

1. 要添加 HANAODBC 服务器类，请执行：

```
sp_addserver logical_server_name, HANAODBC, ODBC_DSN_name
```

本示例添加 big\_hana\_server 作为 Big\_company DSN：

```
sp_addserver big_hana_server, HANAODBC, Big_company
```

2. 使用 **sp\_addexternlogin** 来映射 SAP ASE 登录帐号和密码。

本示例配置系统管理员可以使用名称 “HANA\_login” 和 “HANA\_login\_password” 来登录名为 “big\_hana\_server” 的远程服务器：

```
sp_addexternlogin big_hana_server, sa, HANA_login,  
HANA_login_password
```

## SAP ASE 与 HANA 之间的数据类型映射

---

创建代理表时，系统会将 SAP ASE 数据类型映射到对应的 HANA 数据类型。

使用 **create table** 创建代理表时，HANAODBC 将下面的 SAP ASE 数据类型映射到对应的 HANA 数据类型：

SAP ASE 数据类型	HANA 数据类型
tinyint	tinyint
smallint	smallint
int	integer
integer	integer
bigint	bigint
unsigned smallint	integer
unsigned int	bigint
unsigned bigint	不支持
numeric (precision, scale)	decimal (precision, scale)
decimal (precision, scale)	decimal (precision, scale)



SAP ASE 数据类型	HANA 数据类型
float (precision)	float (precision)
double (precision)	float
real	real
smallmoney	decimal (10,4)
money	decimal (19,4)
smalldatetime	timestamp
datetime	timestamp
date	date
time	time
char (n)	char (n)
varchar (n)	varchar (n)
unichar (n)	nchar (n)
univarchar (n)	nvarchar (n)
nchar (n)	char (n)
nvarchar (n)	varchar (n)
text	clob
unitext	nclob
binary (n)	binary (n)
varbinary (n)	varbinary (n)
image	blob
bit	tinyint
bigdatetime	timestamp
bigtime	不支持

将远程 HANA 列映射到本地代理表列时，允许使用以下数据类型：

远程 HANA 数据类型	SAP ASE 数据类型
tinyint	tinyint

远程 HANA 数据类型	SAP ASE 数据类型
smallint	smallint
int	integer
integer	integer
bigint	bigint
decimal (precision, scale)	decimal (precision, scale)
real	real
double	double precision
float (precision)	double precision
date	date
time	time
seconddate	datetime
timestamp	bigdatetime
varchar (n)	varchar (n)
nvarchar (n)	univarchar (n)
alphanum (n)	univarchar (n)
char (n)	char (n)
nchar (n)	unichar (n)
varbinary (n)	varbinary (n)
clob	text
binary (n)	binary (n)
blob	image
nclob	unitext

## 限制

HANA 的 CIS 支持存在以下限制。

- 只有在针对线程化内核模式进行了配置的 SAP ASE 服务器中才支持 HANA 的 CIS。不能在针对进程模式进行了配置的 SAP ASE 服务器中使用 HANA 的 CIS。

- HANA 的 CIS 不支持包括 **where current of** *cursor\_name* 子句的 **update** 或 **delete** 命令。
- HANA 的 CIS 不支持敏感游标。
- HANA 的 CIS 不支持大对象 (LOB) 参数。
- 由于 HANA LOB 不支持文本指针，因此不能将包括文本指针的查询发送到远程 HANA 服务器。
- 如果发出 **create index...desc** 来创建代理表的索引，不会为 **create index** 语句生成 **desc** 参数。而是使用 **asc** 参数创建远程 HANA 表中的索引。
- 远程表的 **alter table** 语句包括以下限制。您不能：
  - 包括 **partition** 子句
  - 包括参照完整性约束
  - 添加约束
  - 替换列
  - 更改锁定方案
  - 添加带 **check** 条件的列
  - 添加带 **unique** 或 **primary key** 约束的列
- 单个 **alter table** 命令不能包括多个以下参数：
  - **add column\_name**
  - **modify column\_name**
  - **drop column\_name**
- 对于定位型 **update** 和 **delete** 语句，必须为基表定义一个唯一索引，该唯一索引必须至少有一个非空列。如果该唯一索引不可用，SAP ASE 会发出一条错误消息。
- 不能在 HANA 代理表上使用包括两阶段提交的分布式事务管理 (DTM)。
- 不能在涉及 HANA 代理表的事务中包括 **save tran** 和 **rollback to savepoint** 语句。
- 对涉及 HANA 代理表的事务发出的 **rollback tran** 操作会使所有语句回退到当前事务中的第一个语句。**rollback tran** 操作之后的所有语句将正常执行。
- 不能对包括 HANA 代理表的触发器发出 **rollback trigger**。
- 对于使用已映射到外部 HANA 过程的代理表的查询，搜索子句只能包含输入参数 (不能包含结果列)。例如，如果创建此表：

```
create existing table rpc_table
(
    col1 int,
    col2 int
) external procedure at "odbcana..SYSTEM.MYPROC"
```

然后发出下面的查询；搜索结果的范围不会缩小，因为查询指定了 `col1`：

```
select * from rpc_table where col1 > 100
```

- 对于已映射到外部 HANA 过程的代理表，不能将 **null** 用作输入参数的值。例如，不允许使用如下语句：
- ```
select * from rpc_table where _p1 = null
```
- HANA 的 CIS 不支持批量操作。
  - 对于映射到 HANA **timestamp** 数据类型的 SAP ASE **bigdatetime** 和 **bigtime** 数据类型，隐式转换可能会导致 **millisecond** 值发生精度损失。SAP ASE 使用 6 位

数存储 bigdatetime 和 bigtime 值的 millisecond 部分（例如 hh.mi:ss.zzzzzz）。但 HANA 在其 timestamp 数据类型中使用 7 位数存储 bigdatetime 和 bigtime 值的 millisecond 部分。使用 ODBC 传输值时，精度最高达 3 位数。将值从 ODBC 隐式转换为 SAP ASE 内部格式可能会进一步降低精度，因为可用的转换格式最多只支持 3 位数。

- 在搜索子句中指定二进制值时，必须填满这些值。在下面的第一个 select 语句中，c1 的值 (0x123) 未填满，因此不会返回结果。第二个 select 语句使用了 c1 的填满值 (0x012300)，因此返回有效的结果：

```
1> create table cb(c1 binary(3)) at 'odbchana..SYSTEM.cb'  
2> go  
1> insert into cb values(0x123)  
2> go  
(1 row affected)  
  
1> select * from cb where c1 = 0x123  
2> go  
c 1  
-----  
(0 rows affected)  
  
3> select * from cb where c1 = 0x012300  
c1  
-----  
0x012300  
(1 row affected)
```

使用 HANA 的 CIS 时，对带引号标识符的限制有：

- DDL 语句中包括的对象定义内的对象名必须至少包括一个大写字母。
- DML 语句中的带引号标识符必须至少包括一个大写字母。
- 对于所包括的对象定义仅包含小写字符的 DDL 语句，HANA 的 CIS 将引发错误。例如，下面这条语句将引发错误：<hanaserver>..<schema>.table1。
- SAP ASE 在代理表中创建不带引号的对象名。因此，如果远程带引号的标识符只包含小写字符（如 column1），则在执行 DML 语句时，SAP ASE 无法确定是否为带引号的标识符添加引号。也就是说，SAP ASE 按照为 COLUMN1 创建对象名的方式为 column1 创建对象名，因为 SAP ASE 缺省使用小写字母。解决办法是使用跟踪标志 11242，为所有带引号的小写标识符加上引号。
- HANA 的 CIS 不支持带中括号（即 [ 和 ]）的标识符。

## 宽泛查询限制

SAP ASE 16.0 增加了一些列限制和表限制。

| 限制                         | 16.0 之前版本中的限制  | 16.0 及更高版本中的限制  |
|----------------------------|----------------|-----------------|
| <b>select</b> 查询中表的数量      | 50 个表, 46 个工作表 | 250 个表, 46 个工作表 |
| <b>select</b> 查询中子查询的数量    | 50             | 250             |
| <b>order by</b> 子句所允许的最大列数 | 31             | 400             |

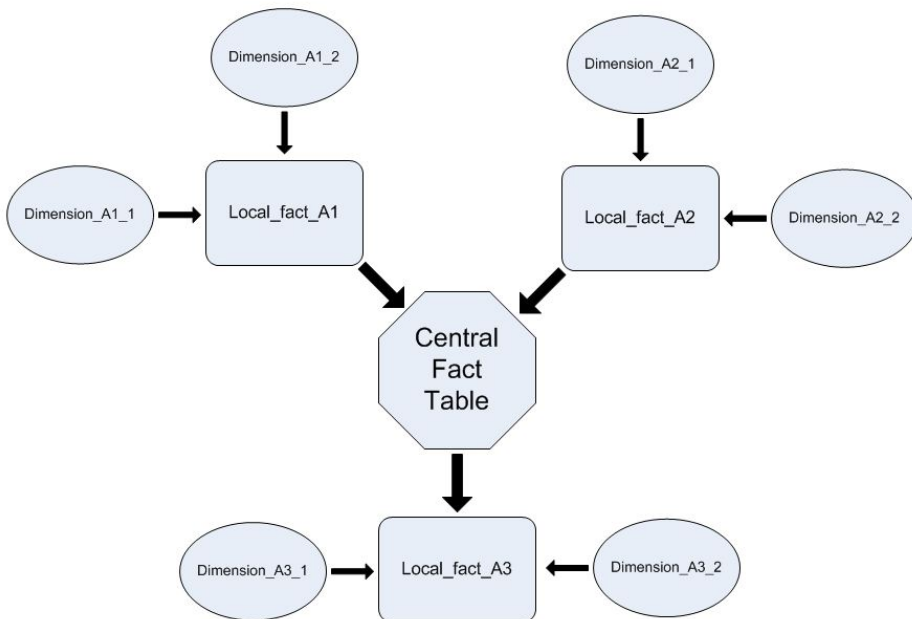
宽泛查询限制

## 使用星型连接优化查询计划

SAP ASE 16.0 提高了星型连接的性能。

星型连接是一种常用的数据仓库查询，针对星型模式数据库运行，该数据库包括一个由维度表环绕的大型表（也称为事实表）。事实表通常包括两种类型的列：一种包括与业务流程相关的度量、指标或事实，另一种类型的列包括维度表的外键。维度表通常包括描述性属性。例如，事实表可能包括某一天售出的某一作者的徒步旅行类书籍的数量信息。相应的维度表包括作者的地址、年龄和性别。

星型连接通过外键将事实表与一个或多个维度表连接在一起。星型连接可能包括维度表的过滤谓词（例如，**where gender = 'M'**），但不包含维度表之间的连接。



雪花模式允许每个维度表充当本地事实表，然后连接另一组维度表，从而扩展了星型连接。

下面的星型连接将 Orders 事实表与 Time、Customer 和 Location 维度表相连接：

```

select C.dattr, sum(O.measure1), count(*)
from Orders O
  join Time T on O.key_dim1 = T.key_col
  join Customer C on O.key_dim2 = C.key_col
  
```

```
join Location L on O.key_dim3 = L.key_col
where T.dattr between 1 and 1000
      and C.dattr between 1001 and 2000
      and L.dattr between 901 and 1900
group by C.dattr
```

为提高星型连接查询评估的性能，在 16.0 版本中进行了以下更改：

- 对于并行计划，**bloom** 过滤器探查移到了 **xchg** 运算符的下方。
- 新的星型连接提示使用抽象计划，能够更好地帮助查询处理器检测到星型连接查询。
- 现在强制所有星型连接的最终计划使用 **use fact\_table** 提示下的 **bloom** 过滤器。
- 强制散列连接和连接顺序基于 **use fact\_table** 提示下星型连接查询的事实表和维度表之间的选择性。查询处理器调整了查询计划，以便充分体现 **bloom** 过滤器的优势。

查询优化程序会在适当的时候使用星型连接。无需将 SAP ASE 配置为使用星型连接。但 SAP 建议在使用星型连接时启用 **set join\_bloom\_filter** 选项和并行查询处理。

---

**注意：** 将优化目标设置为 **allrows\_dss** 时，查询处理器将启用 **set join\_bloom\_filter** 和并行查询处理。

---

## 星型连接提示

---

SAP ASE 16.0 引入了 **use fact\_table** 抽象计划提示，用于指定星型连接查询中的中心事实表并为查询触发特殊查询计划优化策略。

语法为：

```
PLAN '(use fact_table fact_table_name_or_alias_name)'
```

其中：

- **fact\_table** - 指示查询包括星型连接中的一个事实表。
- **fact\_table\_name\_or\_alias\_name** - 指定用于中心事实表的名称或别名。在下面的示例中，**F** 是别名：

```
use fact_table 'F'
```

下面的示例用于将 Orders 事实表连接到 Time、Customer 和 Location 维度表：

```
select C.dattr, sum(O.measure1), count(*)
from Orders O
join Time T on O.key_dim1 = T.key_col
join Customer C on O.key_dim2 = C.key_col
join Location L on O.key_dim3 = L.key_col
where T.dattr between 1 and 1000
      and C.dattr between 1001 and 2000
      and L.dattr between 901 and 1900
group by C.dattr
plan '(use fact_table O)'
```

**use fact\_table** 提示对查询处理器的星型查询检测算法和计划生成进行了如下更改：



- 检测算法从 *fact\_table\_alias\_name* 开始。该表始终是雪花模式的中心。
- 允许使用 **or** 子句。
- 允许事实表和维度表之间的连接成为多列等值连接。不需要在主键和外键之间加入显式约束定义。
- 在为星型连接生成查询计划时，如果查询处理器无法为嵌套循环连接找到好的计划（通常是由于缺少索引或者是对选择性的限制不够严格），它会选择带有强制连接顺序的散列连接计划。该连接顺序基于维度表和事实表之间的选择性。选择性越高（即比值越低），采用该连接顺序的维度表就越接近事实表。

如果查询未满足以下条件，查询处理器将忽略 **use fact\_table** 提示：

- 查询至少有三个表。
- 查询没有外连接或嵌套子查询。
- 带提示的事实表是其查询块中的最大连接表。
- 维度表与事实表通过等值连接进行连接。
- 维度表之间没有连接谓词。

## use fact\_table 提示下的星型连接查询计划

**use fact\_table** 抽象计划提示允许查询处理器为星型连接查询选择并行散列连接计划。通过并行计划，查询处理器能够将 **bloom** 过滤器探查（支持维度表与事实表间的快速连接）推送到 EXCHANGE 运算符的下方，从而进一步减少事实表中符合条件的行的数量。

在 SAP ASE 16.0 之前的版本中，**bloom** 过滤器位于 **sp\_showplan** 输出中 EXCHANGE 运算符的上方。

下面的示例在提示下包括一个并行计划，其中包括表 F 作为带提示的中心事实表：

```

QUERY PLAN FOR STATEMENT 1 (at line 1).
Optimized using Parallel ModeExecuted in parallel by coordinating
process and 66 worker processes.
STEP 1
The type of query is SELECT.

29 operator(s) under root

|ROOT:EMIT Operator (VA = 29)
|
| |HASH VECTOR AGGREGATE Operator (VA = 28)
| | GROUP BY
| | Evaluate Grouped COUNT AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.

```

## 使用星型连接优化查询计划

```
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Using Worktable10 for internal storage.
| | Key Count: 3
| |
| | |HASH JOIN PROBE Operator (VA = 27) (Join Type: Inner Join)
| | | Using Worktable1 for internal storage.
| | | Key Count: 1
| | |
| | |HASH JOIN BUILD Operator (VA = 1) (Join Type: Inner Join)
| | | | Using Worktable1 for internal storage.
| | | | Key Count: 1
| | |
| | | |SCAN Operator (VA = 0)
| | | | FROM TABLE
| | | | /B49/DBENCH01P
| | | | DP
| | | | Index : /B49/DBENCH01P~010
| | | | Forward Scan.
| | | | Positioning by key.
| | | | Keys are:
| | | | SID_0CHNGID ASC
| | | | SID_0RECORDTP ASC
| | | | SID_0REQUID ASC
| | | | Using I/O Size 16 Kbytes for index leaf pages.
| | | | With LRU Buffer Replacement Strategy for index leaf pages.
| | | | Using I/O Size 16 Kbytes for data pages.
| | | | With LRU Buffer Replacement Strategy for data pages.
| | |
| | | |HASH JOIN PROBE Operator (VA = 26) (Join Type: Inner Join)
| | | | Using Worktable2 for internal storage.
| | | | Key Count: 1
| | | |
| | | |HASH JOIN BUILD Operator (VA = 3) (Join Type: Inner Join)
| | | | | Using Worktable2 for internal storage.
| | | | | Key Count: 1
| | | |
| | | | |SCAN Operator (VA = 2)
| | | | | FROM TABLE
| | | | | /B49/DBENCH01U
| | | | | DU
| | | | | Index : "/B49/DBENCH01U~020"
| | | | | Forward Scan.
| | | | | Positioning at index start.
| | | | | Using I/O Size 16 Kbytes for index leaf pages.
| | | | | With LRU Buffer Replacement Strategy for index leaf pages.
| | | | | Using I/O Size 16 Kbytes for data pages.
| | | | | With LRU Buffer Replacement Strategy for data pages.
```



## 使用星型连接优化查询计划

```
| | | | | | | | | | D1
| | | | | | | | | | Table Scan.
| | | | | | | | | | Forward Scan.
| | | | | | | | | | Positioning at start of table.
| | | | | | | | | | Using pushdown bloom filter (bv7).
| | | | | | | | | | Using I/O Size 128 Kbytes for data pages.
| | | | | | | | | | With LRU Buffer Replacement Strategy for data pages.
| | | | | | | | | |
| | | | | | | | | | |EXCHANGE Operator (VA = 22) (Merged)
| | | | | | | | | | |Executed in parallel by 64 Producer and 1 Consumer
processes.

| | | | | | | | | |
| | | | | | | | | | |EXCHANGE:EMIT Operator (VA = 21)
| | | | | | | | | |
| | | | | | | | | | |HASH JOIN PROBE Operator (VA = 20) (Join Type:
Inner Join)
| | | | | | | | | | |Using Worktable9 for internal storage.
| | | | | | | | | | |Key Count: 1
| | | | | | | | | |
| | | | | | | | | | |EXCHANGE Operator (VA = 13) (Replicated)
| | | | | | | | | | |Executed in parallel by 1 Producer and 64
Consumer processes.

| | | | | | | | | |
| | | | | | | | | | |EXCHANGE:EMIT Operator (VA = 12)
| | | | | | | | | |
| | | | | | | | | | |HASH JOIN BUILD Operator (VA = 11) (Join Type:
Inner Join)
| | | | | | | | | | |Using Worktable6 for internal storage.
| | | | | | | | | | |Building pushdown bloom filter (bv11)
| | | | | | | | | | |Key Count: 1
| | | | | | | | | |
| | | | | | | | | | |SCAN Operator (VA = 10)
| | | | | | | | | | |FROM TABLE
| | | | | | | | | | |/B49/SSALESORG
| | | | | | | | | | |S1
| | | | | | | | | | |Index : "/B49/SSALESORG~0"
| | | | | | | | | | |Forward Scan.
| | | | | | | | | | |Positioning by key.
| | | | | | | | | | |Keys are:
| | | | | | | | | | |/B49/S SALESORG ASC
| | | | | | | | | | |Using I/O Size 16 Kbytes for index leaf
pages.
| | | | | | | | | | |With LRU Buffer Replacement Strategy for
index leaf pages.
| | | | | | | | | | |Using I/O Size 16 Kbytes for data pages.
| | | | | | | | | | |With LRU Buffer Replacement Strategy for
data pages.
| | | | | | | | | |
| | | | | | | | | | |HASH JOIN PROBE Operator (VA = 19) (Join Type:
Inner Join)
| | | | | | | | | | |Using Worktable8 for internal storage.
| | | | | | | | | | |Key Count: 1
| | | | | | | | | |
| | | | | | | | | | |EXCHANGE Operator (VA = 17) (Replicated)
```



## 使用星型连接优化查询计划

```
| | Evaluate Grouped COUNT AGGREGATE.
| | Evaluate Grouped SUM OR AVERAGE AGGREGATE.
| | Using Worktable4 for internal storage.
| | Key Count: 1
| |
| | |HASH JOIN Operator (VA = 6) (Join Type: Inner Join)
| | | Using Worktable3 for internal storage.
| | | Building pushdown bloom filter (bv6)
| | | Key Count: 1
| | |
| | | |SCAN Operator (VA = 0)
| | | | FROM TABLE
| | | | Location
| | | | L
| | | | Table Scan.
| | | | Forward Scan.
| | | | Positioning at start of table.
| | | | Using I/O Size 2 Kbytes for data pages.
| | | | With LRU Buffer Replacement Strategy for data pages.
| | |
| | | |HASH JOIN Operator (VA = 5) (Join Type: Inner Join)
| | | | Using Worktable2 for internal storage.
| | | | Building pushdown bloom filter (bv5)
| | | | Key Count: 1
| | | |
| | | | |SCAN Operator (VA = 1)
| | | | | FROM TABLE
| | | | | Customer
| | | | | C
| | | | | Table Scan.
| | | | | Forward Scan.
| | | | | Positioning at start of table.
| | | | | Using I/O Size 2 Kbytes for data pages.
| | | | | With LRU Buffer Replacement Strategy for data pages.
| | | |
| | | | |HASH JOIN Operator (VA = 4) (Join Type: Inner Join)
| | | | | Using Worktable1 for internal storage.
| | | | | Building pushdown bloom filter (bv4)
| | | | | Key Count: 1
| | | | |
| | | | | |SCAN Operator (VA = 2)
| | | | | | FROM TABLE
| | | | | | Time
| | | | | | T
| | | | | | Table Scan.
| | | | | | Forward Scan.
| | | | | | Positioning at start of table.
| | | | | | Using I/O Size 2 Kbytes for data pages.
| | | | | | With LRU Buffer Replacement Strategy for data pages.
| | | | |
| | | | | |SCAN Operator (VA = 3)
| | | | | | FROM TABLE
| | | | | | Orders
| | | | | | O
| | | | | | Table Scan.
| | | | | | Forward Scan.
```

```
| | | | | Positioning at start of table.  
| | | | | Using pushdown bloom filter (bv4, bv5, bv6).  
| | | | | Using I/O Size 2 Kbytes for data pages.  
| | | | | With LRU Buffer Replacement Strategy for data pages.
```





# 查询性能改进

SAP ASE 16.0 版本引入了对查询的改进。

## 动态线程分配

---

SAP ASE 16.0 及更高版本使用动态线程分配来代替静态线程分配。

利用动态线程分配，SAP ASE 能够以更少的资源更快速地执行并行查询计划。SAP ASE 将动态线程分配应用到为 **select** 查询生成的并行 **lava** 查询计划。但以下命令将继续使用静态线程分配：

- **select into**
- 包括数据复制的 **reorg** 命令
- 包括数据复制的 **alter table** 命令
- **create index**

动态线程分配通过以下方式提高性能：

- 以更少的线程并行执行查询计划（静态线程分配要求执行串行查询）。
- 在工作线程间执行动态负载平衡：如果执行查询计划的线程数少于工作单元数，执行最小工作单元的线程将以更快的速度完成自身任务，从而释放自身空间来执行剩余的可用工作单元，与此同时，执行大工作单元的线程也会完成自身任务。
- 更加有效地使用连接中的现有语义分区。在连接已基于连接列进行了分区的两个表时，动态线程分配允许分区到分区的连接；与其相反，静态线程分配则在一个操作中外部表的所有分区连接到内部表的所有分区。动态线程分配允许单个工作线程将外部表的第一个分区连接到内部表的第一个分区，然后重新执行查询计划片段以连接第二个分区，依此类推。

使用 **number of worker processes** 和 **max parallel degree** 参数配置 SAP ASE 并行度时，启用动态线程分配。请参见《系统管理指南，卷 1》中的设置配置参数。

## SORT 运算符性能改进

---

SAP ASE 16.0 及更高版本改进了某些包含 **sort** 运算符的并行查询的性能。

要充分利用这些更改带来的好处，并行查询必须满足以下条件：

- **sort** 运算符必须位于 **Exchange** 运算符上方
- **Exchange** 运算符必须为包含一个生产者和多个使用者的 **Replicated Exchange**。

## 查询性能改进

如果查询不满足此条件，查询处理器将使用 **SAP ASE 16.0** 之前版本中所包含的标准 **SORT** 运算符（也称为 **DEFAULT** 排序）。

本例所示为 **SAP ASE 16.0** 之前版本，不包含 **SORT** 运算符更改：

```
| | | | |MERGE JOIN Operator (Join Type: Inner Join) (VA = 5)
| | | | |  Using Worktable2 for internal storage.
| | | | |   Key Count: 1
| | | | |   Key Ordering: ASC
| | | | |
| | | | |   |SORT Operator (VA = 3)
| | | | |   |  Using Worktable1 for internal storage.
| | | | |
| | | | |   |EXCHANGE Operator (VA = 2) (Replicated)
| | | | |   |  Executed in parallel by 1 Producer and 8
Consumer processes.
| | | | |
| | | | |   |EXCHANGE:EMIT Operator (VA = 1)
| | | | |
| | | | |   |SCAN Operator (VA = 0)
| | | | |   |  FROM TABLE
| | | | |   |  t2
| | | | |   |  Table Scan.
| | | | |   |  Forward Scan.
| | | | |   |  Positioning at start of table.
| | | | |   |  Using I/O Size 16 Kbytes for data
pages.
| | | | |   |  With LRU Buffer Replacement Strategy
for data pages.
```

但是 **SAP ASE 16.0** 及更高版本将 **SORT** 运算符的构建部分移到了 **Exchange Operator** 的下方，而将排序表的读取部分留在 **Exchange** 的上方，这样同一查询会产生以下查询执行计划

```
| | | | |MERGE JOIN Operator (Join Type: Inner Join) (VA = 6)
| | | | |  Using Worktable3 for internal storage.
| | | | |   Key Count: 1
| | | | |   Key Ordering: ASC
| | | | |
| | | | |   |SORT (GETSORTED) Operator (VA = 4)
| | | | |   |  Using Worktable2 for internal storage.
| | | | |
| | | | |   |EXCHANGE Operator (VA = 3) (Replicated)
| | | | |   |  Executed in parallel by 1 Producer and 8
Consumer processes.
| | | | |
| | | | |   |EXCHANGE:EMIT Operator (VA = 2)
| | | | |
| | | | |   |SORT (SORTBUILD) Operator (VA = 1)
| | | | |   |  Using Worktable1 for internal storage.
| | | | |
| | | | |   |SCAN Operator (VA = 0)
| | | | |   |  FROM TABLE
| | | | |   |  t2
| | | | |   |  Table Scan.
| | | | |   |  Forward Scan.
```

```
| | | | | | | | | | | Positioning at start of table.
| | | | | | | | | | | Using I/O Size 16 Kbytes for data
pages.
| | | | | | | | | | | With LRU Buffer Replacement
Strategy for data pages.
```

新增的并行 SORT 运算符的好处在于：

- 可将 SORT 运算符拆分为两个运算符：SORTBUILD 和 GETSORTED。GETSORTED 的多个实例共用 SORTBUILD 构建的一个排序表来并行读取已排序的行，从而减少 I/O。DEFAULT SORT 运算符计划将创建多个 SortTables 副本（第一个示例中为 8 Consumer processes）。
- 将包含元数据的单个行从 SORTBUILD 运算符发送至 GETSORTED 运算符的多个实例，允许通过引用直接访问排序表，从而避免通过 Exchange 复制并传播多个行。

并行 SORT 运算符存在以下限制：

- 不支持 Repartitioned Exchange。但是，位于 Repartitioned Exchange 上方的 SORT 运算符在查询执行计划中将使用 DEFAULT 排序。
- 抽象计划并不显式支持强制使用 GETSORTED 和 SORTBUILD，但可通过确保 SORT 运算符位于 Replicated Exchange 上方来强制执行这些计划。
- 对于包含 GETSORTED 和 SORTBUILD 运算符的查询执行计划，SAP ASE 会隐式禁用动态线程分配 (DTA)。

## Hash Join 运算符性能改进

SAP ASE 16.0 以及更高版本针对使用 HASH JOIN 运算符的某些并行查询计划进行了性能改进并减少了使用的资源。

HASH JOIN 运算包含以下两步：

1. 构建一个包含要连接的行外部数据流的连接列的散列表
2. 使用要连接的内部数据流中每行的连接列值探查散列表

SAP ASE 16.0 使用 HASH PROBE 和 HASH BUILD 运算符替代了 HASH JOIN 运算符，并在这些运算符之间增加了复制的 EXCHANGE 运算符。HASH BUILD 运算符用于构建散列表，HASH PROBE 运算符用于读取内部数据流并探查散列表以查找匹配行（在早期版本的 SAP ASE 中，由 HASH JOIN 运算符执行所有这些步骤）。

由一个工作线程执行 HASH BUILD 运算符，构建一个散列表。查询引擎将该散列表经由内存管道传递到所有执行 HASH PROBE 运算符的生产者。这些生产者共用该散列表，并在其中探查与其内部数据流中连接列匹配的条目。早期版本的 SAP ASE 需要使用多个生产者来执行 HASH JOIN 运算符。

以下是使用 HASH JOIN 运算符的查询计划的 **showplan** 输出示例：

```
|
| |EXCHANGE Operator (VA = 6) (Merged)
```

## 查询性能改进

```
| |Executed in parallel by 4 Producer and 1 Consumer processes.
|
| |EXCHANGE:EMIT Operator (VA = 5)
| | |
| | | |HASH JOIN Operator (VA = 4) (Join Type: Inner Join)
| | | | |Using Worktable1 for internal storage.
| | | | |Key Count: 1
| | | |
| | | |EXCHANGE Operator (VA = 2) (Replicated)
| | | | |Executed in parallel by 1 Producer and 4 Consumer
processes.
| | | |
| | | | |EXCHANGE:EMIT Operator (VA = 1)
```

查询计划必须具有以下属性，查询处理器才能使用 HASH JOIN 改进功能：

- 查询处理器可以并行执行 HASH JOIN 运算符 (VA = 4) (上例的 EXCHANGE Operator (VA = 6) 中包含四个 Producer)。
- HASH JOIN 运算符的左侧子 Operator 是包含一个 Producer 的 Replicated EXCHANGE Operator (VA = 2)

如果查询不符合该条件，查询处理器将使用单个 HASH JOIN 运算符。

SAP ASE 16.0 查询计划的优势在于：

- 不必将所有行经由内存管道复制到执行 HASH JOIN 运算符的生产者，因而性能不会有所降低。相反，查询引擎只需通过管道发送一行即可，该行含有对 HASH PROBE 运算符构建的散列表的引用。
- 每个生产者不需要占用额外的内存和磁盘资源来构建各自的散列表。而是共享一个散列表。

以下是使用改进的散列连接处理的查询计划 showplan 输出示例：

```
| |EXCHANGE Operator (VA = 7) (Merged)
| | |Executed in parallel by 4 Producer and 1 Consumer processes.
|
| | |EXCHANGE:EMIT Operator (VA = 6)
| | | |
| | | | |HASH JOIN PROBE Operator (VA = 5) (Join Type: Inner Join)
| | | | | |Using Worktable2 for internal storage.
| | | | | |Key Count: 1
| | | | |
| | | | |EXCHANGE Operator (VA = 3) (Replicated)
| | | | | |Executed in parallel by 1 Producer and 4 Consumer
processes.
| | | | |
| | | | | |EXCHANGE:EMIT Operator (VA = 2)
| | | | | |
| | | | | | |HASH JOIN BUILD Operator (VA = 1) (Join Type:
```

```
Inner Join)
| | | | | | | Using Worktable1 for internal storage.
| | | | | | | Key Count: 1
| | | | | | |
```



# 全文本审计

当启用 DML 审计选项（包括 *table\_access* 和 *view\_access*）时，SAP ASE 16.0 及更高版本支持全文本 DML 审计，在屏蔽敏感参数的情况下打印参数名和参数值。

有关全文本审计的信息，请参见《安全性管理指南》>“审计”>“管理审计系统”>“配置审计系统”>“设置全局审计选项”>“审计 DML 语句”。





## 存储过程内部授权检查的审计

审计选项 `sproc_auth` 允许对在系统存储过程内部执行的授权检查执行审计。

| 细化权限 | 事件  |
|------|-----|
| 启用   | 146 |
| 禁用   | 80  |

启用 `security` 审计选项，或启用 `sproc_auth` 审计选项时，将对审计事件 80 进行审计。仅当启用 `sproc_auth` 选项时，才会对审计事件 146 进行审计。



# 替换对象定义

用新定义替换现有编译对象定义，但保留原始名称、对象 ID、安全属性（如审计选项和权限）和复制属性。

**create or replace** 功能可创建新对象（如果该对象不存在）或以相同名称替换现有对象。**or replace** 子句将在数据库内隐式删除现有对象并重新创建具有相同名称和类型的对象，从而更改了对象定义，但保留现有安全和复制属性。如果编译对象的文本在替换前是隐藏的，则替换后的文本仍将保留隐藏状态。该功能没有引入新的关键字。

**or replace** 子句已添加到以下命令中：

- **create default**
- **create function**
- **create function (SQLJ)**
- **create procedure**
- **create procedure (SQLJ)**
- **create rule**
- **create trigger**
- **create view**

仅不包含数据的对象支持 **or replace** 功能。

如果进行替换时对象正在使用中，会引发错误 3702：

```
"Cannot drop or replace the %S_MSG '%.*s' because it is currently in use."
```

如果替换了某个对象，SAP ASE 将替换以下系统表中该对象的定义：`sysprocedures`、`syscomments`、`sysdepends` 和 `syscolumns`。还将更新 `sysobjects` 表中的部分字段。先对对象的查询树执行规范化，然后再在 `sysprocedures` 中进行替换。

替换后的对象可用于其它对象定义中。在使用替换后的对象时，SAP ASE 将对其进行重新编译，但在某些情况下，当替换后的对象的接口与调用对象中使用的接口不匹配时，可能需要替换调用对象。可对替换后的对象运行 `sp_depends` 以验证是否存在调用对象，然后执行替换。有关详细信息，请参见 `create procedure`、`create view` 和 `create function` 中的与替换后的对象相关的对象部分。

无论是否启用细化权限，您只有成为对象所有者才能替换编译对象。您无法通过别名或 `setuser` 充当对象所有者来替换编译对象。但如果您通过 `set proxy` 成为所有者，则可以替换编译对象。

---

**注意：** `create or replace` 功能将在同一事务中先执行隐式 `drop` 再执行 `create`。因此需要更多的事务日志空间。如果使用 `create or replace` 来代替删除对象再创建对象的过程，可能需要增加事务日志的大小。

---

另请参见

- create default (第 109 页)
- create function (第 113 页)
- create function (SQLJ) (第 116 页)
- create procedure (第 119 页)
- create procedure (SQLJ) (第 122 页)
- create rule (第 124 页)
- create trigger 用于 or replace (第 131 页)
- create view (第 133 页)

## 安装脚本更改

---

为使用 **create or replace**，对安装脚本中使用的存储过程进行了更改。

在 SAP ASE 16.0 之前版本的安装脚本中使用的存储过程已被删除并重新创建。自 SAP ASE 16.0 起，这些安装脚本中所用的存储过程将使用 **or replace** 功能：

|                 |               |                 |               |
|-----------------|---------------|-----------------|---------------|
| installmaster   | installmodel  | installmsgsvss  | installpcidb  |
| installjconnect | installcommit | installdbccalt  | installdbccdb |
| installupgrade  | installjsb    | installsecurity | installoledb  |
| installodbc     | installhasvss | installpubs2    | installpubs3  |

## 数据和日志段更改

---

对各种数据库的数据和日志段的大小实施了更改。

为执行 **create or replace**，设备需要足够大才能符合额外空间要求。

缺省数据库大小更改如下：

| 数据库      | 页大小  | 缺省大小 - 16.0 之前版本 | 缺省大小 - 16.0 和更高版本 |
|----------|------|------------------|-------------------|
| master   | 2KB  | 13MB             | 18MB              |
|          | 4KB  | 26MB             | 26MB              |
|          | 8KB  | 52MB             | 52MB              |
|          | 16KB | 104MB            | 104MB             |
| sybpcidb | 2KB  | 24MB             | 48MB              |
|          | 4KB  | 48MB             | 96MB              |
|          | 8KB  | 96MB             | 192MB             |

| 数据库 | 页大小  | 缺省大小 - 16.0 之前版本 | 缺省大小 - 16.0 和更高版本 |
|-----|------|------------------|-------------------|
|     | 16KB | 192MB            | 384MB             |

| 数据库            | 缺省大小 - 16.0 之前版本 | 缺省大小 - 16.0 和更高版本 |
|----------------|------------------|-------------------|
| sybsystemprocs | 172MB            | 196MB             |
| sybmgmtdb      | 75MB             | 76MB              |
| dbccdb         | 5MB              | 33MB              |
| dbcalt         | 5MB              | 33MB              |

有关配置 SAP ASE 服务器的缺省值和最低要求的详细信息，请参见《安装指南》> “安装 SAP ASE” > “最低限度地配置 SAP ASE 服务器”。

替换对象定义

# HTML 格式的查询计划和执行统计信息

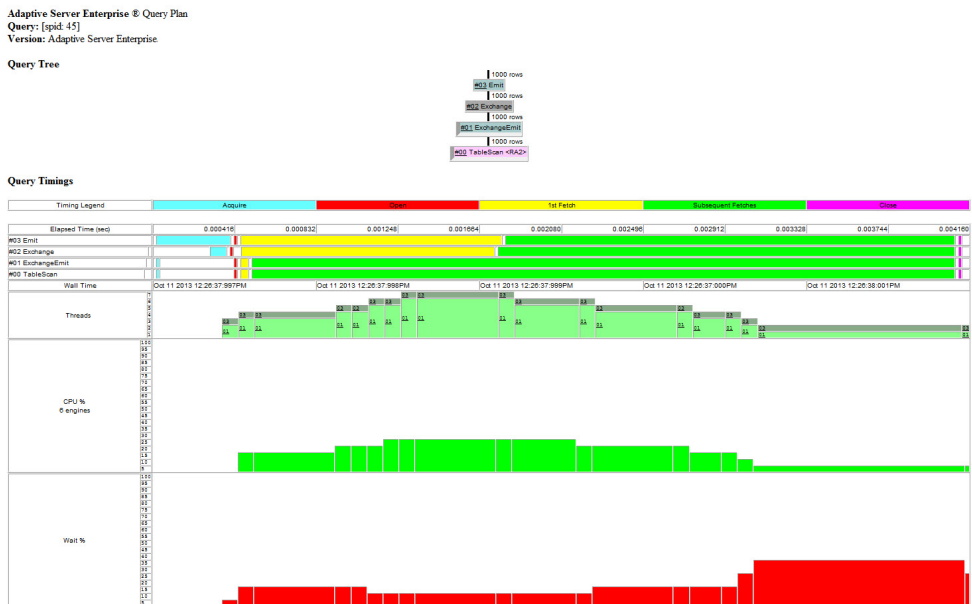
新增的动态线程分配 (DTA) 模型将增强工作线程的使用。

15.7 SP100 版本引入了一种能够以 HTML 格式生成图形查询计划的功能，从而能够使用静态线程分配 (STA) 工作线程模型查看 Web 浏览器中的数据。使用 STA，计划片段和工作线程间存在直接关系：因为许多计划片段的生成是由于分配了用于执行查询的工作线程。计划片段到工作线程的分配是静态且唯一的：工作线程只能执行所分配的计划片段。使用这种模型时，HTML 表示形式将打印每个线程及其单个计划片段执行的相关详细信息。

在 16.0 版本中，实现了 DTA 工作线程模型。随着 DTA 的引入，“HTML 格式的查询计划和执行统计信息”功能得到了更新，从而正确反映新的并行执行模型。有关 DTA 工作线程模型的详细信息，请参见查询性能改进。

HTML 表示形式报告每个计划片段执行（或工作单元）的执行统计信息，从而允许分别报告同一计划片段的多次执行。为了更好地识别新的 DTA 模型后的工作单元执行，对于每个计划片段执行来说，输出都由“工作单元执行”表示，并提供线程 SPID 以及其它值，如 Root 运算符标识符（对于计划片段）和生产者 ID。

下面是使用多个工作线程并行执行查询的示例。



# HTML 格式的查询计划和执行统计信息

## Query Text

```
select * from R2 parallel 41
```

## Threads

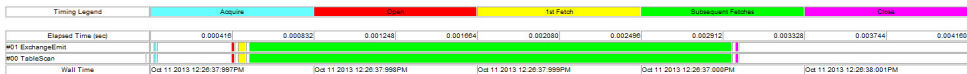
| SPID | Thread ID | Rows produced |
|------|-----------|---------------|
| 34   | 2606097   | 168           |
| 35   | 2606824   | 168           |
| 36   | 2424851   | 168           |
| 37   | 2293778   | 168           |
| 38   | 2162765   | 162           |
| 39   | 2031632   | 168           |

## Work unit execution:

Thread SPID: 34 Root operator: 1 Producer ID: 0



## Query Timings

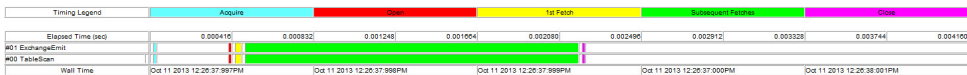


## Work unit execution:

Thread SPID: 35 Root operator: 1 Producer ID: 1



## Query Timings

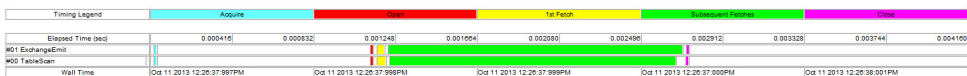


## Work unit execution:

Thread SPID: 36 Root operator: 1 Producer ID: 2



## Query Timings

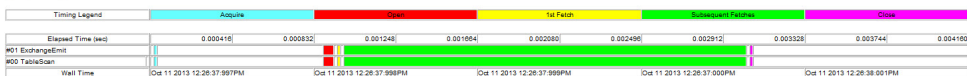


## Work unit execution:

Thread SPID: 37 Root operator: 1 Producer ID: 3



## Query Timings

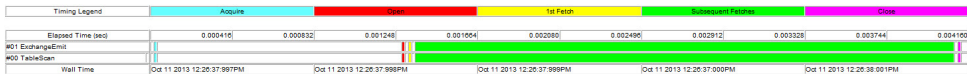


## Work unit execution:

Thread SPID: 38 Root operator: 1 Producer ID: 4



## Query Timings

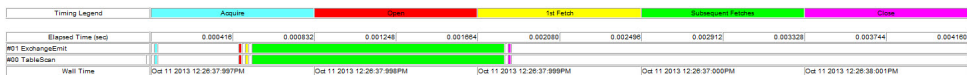


## Work unit execution:

Thread SPID: 39 Root operator: 1 Producer ID: 5



## Query Timings





**另请参见**

- 查询性能改进（第 33 页）

## 已生成文件的前缀选项

---

`SET STATISTICS QUERY_NAME_HTML` 命令有助于区分或识别与同一查询的多次执行相关的文件。

如果给定查询名称，该查询名称用作将要在其中写入 **HTML** 输出内容的内部生成文件的前缀。

```
SET STATISTICS QUERY_NAME_HTML [queryname | ON | OFF]
```



# 索引压缩

利用关系数据库中的索引压缩，能够更有效地存储数据，减少内存耗用量，同时由于 I/O 需求降低而使性能得到提高。

索引压缩支持：

- 索引叶页压缩
- DOL 和 APL 索引叶页格式
- 表、索引和本地索引分区级别的压缩

可在服务器级别或会话级别使用 **sp\_configure** 和 **set compression** 启用或禁用索引压缩。

要在表、索引或本地索引分区级别指定索引压缩，请使用以下命令：

- **create table**
- **alter table**
- **create index**
- **alter index**
- **select into**

在索引级别指定压缩将优先于在表级别指定的索引压缩。本地索引分区级别指定优先于索引级别指定。

支持

- **reorg rebuild** 已扩展为支持索引压缩。
- DML 已扩展为支持索引压缩。
- 触发器支持索引压缩。
- 对于索引定义为压缩索引的表，批量复制到该表中的数据将会自动进行压缩。
- **dbcc checktable** 和 **dbcc checkstorage** 都能够检查索引的完整性。这两个命令已得到增强，可通过扫描表的所有页检查各页的完整性。**dbcc checktable** 检查的信息更多，因此检查功能更为强大。
- 如果数据库包含具有压缩索引的表，则允许使用 **dump database**、**dump transaction** 和 **load transaction**。但不允许使用 XPD L。

对于复制索引来说，即使在创建表期间为所有索引指定了压缩，仍将它们创建为未压缩索引。

不支持按索引压缩的表上的 APL 聚簇索引。

只有一列的唯一索引不进行压缩。

## 启用索引压缩

---

在服务器级别或数据库级别启用索引压缩。

### 在服务器级别启用索引压缩

要对服务器上所有数据库中的所有索引启用压缩，请在服务器级别设置索引压缩。

语法为：

```
sp_configure "enable compression", 0 | 1
```

缺省值为 0。

如果未启用索引压缩，则当试图创建按索引压缩的表或创建索引时将引发错误。

### 在会话级别启用索引压缩

要对某个会话所有数据库中的所有索引启用压缩，请在数据库级别设置索引压缩。

语法为：

```
set {compression
    [= {default | ON | OFF} ]
    |index_compression
    [= {default | ON | OFF} ] }
```

缺省值为 `off`。此命令仅影响在执行此命令后针对压缩索引生成的叶行。

- **set index\_compression** 设置为 `off` 后，新插入到压缩索引中的所有行都不是按索引压缩的行。
- **set index\_compression** 设置为 `on` 后，新插入到压缩索引中的所有行都是按索引压缩的行。

## 创建按索引压缩的表

---

要为表指定索引压缩，请使用 **create table** 或 **select into** 命令。

除系统目录和工作表外，可为任何其它表或索引（包括临时表）指定索引压缩。

在索引级别指定压缩将优先于在表级别指定的索引压缩。本地索引分区级别指定优先于索引级别指定。

### 使用 **create table** 命令

**with index\_compression** 子句提供以下压缩选项：

- **NONE** - 不压缩指定表的索引。压缩专门使用 **index\_compression = PAGE** 创建的索引。

- **PAGE** - 压缩指定表的所有索引。不压缩专门使用 **index\_compression = NONE** 创建的索引。

如果未在表 DDL 中的任何位置指定压缩，则不压缩表索引。

使用 **select into** 命令

选择一个现有表，然后使用 **select into** 创建按索引压缩的表。with **index\_compression** 子句的语法与 **create table** 命令的语法相同。

另请参见

- **create table** 用于索引压缩（第 126 页）
- **select into**（第 139 页）

## 创建压缩索引

---

要为索引或本地索引分区级别指定索引压缩，请使用 **index\_compression** 子句。

仅压缩叶页。压缩和未压缩索引行可共同存在于单个索引叶页上。如果未在表或索引 DDL 中的任何位置指定压缩，则不压缩索引。索引压缩不支持 APL 聚簇索引。只有一列的唯一索引不进行压缩。

在索引级别指定压缩将优先于在表级别指定的索引压缩。本地索引分区级别指定优先于索引级别指定。

使用 **create index** 命令

**WITH index\_compression** 子句提供以下压缩选项：

- **NONE** - 不压缩指定索引的索引页。压缩专门使用 **index\_compression = PAGE** 创建的本地索引分区。
- **PAGE** - 当该页已满时，使用页前缀压缩来压缩现有索引行。当添加一行时，系统将执行检查以确定该行是否适合压缩。

另请参见

- **create index**（第 118 页）

## 更改压缩状态

---

要更改表的压缩状态以便将来进行索引插入或更新，请使用 **alter table** 或 **alter index**。

无论现有索引页是否进行了压缩，它们都不会受到影响。要更改表的压缩状态，必须具备该表的独占访问权限。

更改本地索引分区的压缩状态仅影响分区中新插入或更新的索引行。

新建索引的缺省行为取决于表的压缩设置：

- 对于按索引压缩的表，新建索引的索引压缩设置为 `on`。
- 对于未按索引压缩的表，新建索引将保持不压缩状态。

### 使用 `alter table` 命令

`alter table` 命令允许模式修改和属性修改的多种组合。其中有些命令仅需要目录更新，而其他命令则需要数据移动和任何现有索引的重建。如果索引需要重建且索引压缩设置为 `on`，则在索引重建过程中将压缩索引页。索引重建后，根据索引压缩状态，产生的索引将包括压缩或未压缩索引行。

`set index_compression` 子句指定对表、索引或本地索引分区启用还是禁用索引压缩。如果将表修改为按索引进行压缩，则将压缩新建索引。

`modify partition partition_name` 子句将按照后面的 `set compression` 子句的指定对要修改压缩状态的本地索引分区进行命名。

### 使用 `alter index` 命令

缺省情况下，`alter index` 权限授予索引所有者，并且不能移交，除非移交给数据库所有者（数据库所有者可以通过运行 `setuser` 命令来充当索引所有者）。系统管理员也可以改变用户索引。

### 限制

索引压缩不支持 APL 聚簇索引。

只有一列的唯一索引不进行压缩。

### 从索引中删除压缩

要从索引中删除压缩，必须先删除索引，然后使用 `set index_compression off` 重新创建索引。

### 另请参见

- `alter table` 用于索引压缩（第 105 页）
- `alter index`（第 105 页）

# SAP JVM 支持

SAP ASE 使用 SAP JRE 来支持 Java 应用程序。

缺省情况下，SAP JRE 安装在以下位置：

```
$SYBASE/shared/SAPJRE-7_*
```

安装程序自动设置 *SAP\_JRE7*、*SAP\_JRE7\_32* 和 *SAP\_JRE7\_64* 环境变量。

---

**注意：**（仅限 IBM AIX）使用任意 Java 应用程序时，必须将数据大小资源限制设置为 unlimited：

```
limit datasize unlimited
```

请参见操作系统文档。

---





# 完全数据库加密

SAP ASE 16.0 版中引入了完全加密数据库的功能。

SAP ASE 的早期版本允许对列进行加密。而在 16.0 版中，您可以完全加密整个数据库，为整个数据库提供保护。对数据库进行完全加密时，数据库的所有数据、索引以及事务日志都将加密。该加密是透明的，因此用户可照常对表和索引等执行操作，不会发现有任何差异。

SAP ASE 在页级加密数据。将数据库设置为“加密”后，将自动进行加密和解密。数据加密发生在将页写入磁盘之前，数据解密发生在数据页装载到内存之后。

## 完全数据库加密与加密列

---

根据您的需要选择加密整个数据库或仅加密列。

SAP ASE 验证和访问控制机制可确保只有正确识别和授权的用户能够访问数据。数据加密可进一步保护敏感数据，以防失窃或出现安全问题。

加密列和完全加密数据库这两项功能均能符合安全性和隐私要求，但由于用法的不同，二者在部署的便捷性方面有所差异。如果：

- 可以轻松确定哪个列包含敏感数据，则选择加密列。
- 必须对敏感数据列执行范围搜索或因不了解数据模型而无法确定敏感数据列（例如，在包含数千个表的应用程序包中），请选择加密数据库。此外，敏感数据的定义（如个人信息）因地点（如州或国家）而异；加密整个数据库可帮您满足这些不同的数据安全性要求。

## 创建数据库加密密钥

---

数据库加密密钥是 Master 数据库中创建的用于加密数据库的 256 位对称密钥。

### 前提条件

在创建数据库加密密钥 (DEK) 前，必须执行以下操作：

- 验证您是否具备有效的 SAP ASE 加密功能许可证 (ASE\_ENCRYPTION)
- 设置 **enable encrypted columns** 配置参数
- 在 master 数据库中创建主密钥和双主密钥（可选）；这些密钥可保护数据库加密密钥。请参见《加密列用户指南》中的“使用数据库级主密钥和双主密钥”。
- 确保已具备相应的权限：

- 如果细分权限处于启用状态，则需具备名为 `manage database encryption key` 的系统权限才能创建密钥。
- 如果细分权限处于禁用状态，则必须具备 `sso_role`、`keycustodian_role` 或 **create encryption key** 权限。

### 过程

在 master 数据库中使用 **create encryption key** 命令创建数据库加密密钥。语法为：

```
create encryption key keyname
  [for algorithm]
  for database encryption
  [with
    { [master key]
      [key_length 256]
      [init_vector random]
      [[no] dual_control]}]
```

其中：

- *keyname* - 在 master 数据库的用户表、视图和过程命名空间中必须唯一。
- *for algorithm* - 指定算法。当前仅支持高级加密标准 (AES) 一种算法。
- *for database encryption* - 显式指定您正在创建加密整个数据库而非加密列的加密密钥。
- *master key* - 对于完全数据库加密是必需的。如果主密钥尚未存在，SAP ASE 将返回错误。
- *key\_length 256* - 是要创建的密钥的大小（以位为单位）。数据库加密密钥的唯一有效长度是 256 位；如果使用其它大小，SAP ASE 将返回错误消息。
- *init\_vector random* - 对于完全数据库加密是必需的。如果像创建列加密密钥一样也指定 *init\_vector null*，SAP ASE 将返回错误。
- *[no] dual control* - 指示是否必须使用双重控制加密数据库加密密钥。缺省情况下不配置双重控制。

本示例创建由主密钥保护的数据库加密密钥：

```
sp_configure 'enable encrypted columns', 1
create encryption key master with passwd "testpassword"
set encryption passwd 'testpassword' for key master
create encryption key dbkey for database encryption
```

### 另请参见

- 更改数据库加密密钥（第 59 页）
- 删除数据库加密密钥（第 59 页）
- 备份数据库加密密钥（第 68 页）
- `create encryption key`（第 111 页）
- `drop encryption key`（第 136 页）

## 更改数据库加密密钥

要更改保护数据库加密密钥的方式及其所有者，请使用 **alter encryption key** 命令。

无法为数据库重新生成数据库加密密钥。

- 要更改数据库加密密钥：
  1. 解密由数据库加密密钥保护的数据库。
  2. 删除并重新创建数据库加密密钥。

---

**注意：**无法将列加密密钥转换为数据库加密密钥。如果使用 **for database encryption** 选项将不同的加密密钥类型更改为数据库加密密钥，则 SAP ASE 将显示错误消息。

---

- 如果只是要更改保护数据库加密密钥的方式，而不同时更改数据库加密密钥，则使用以下语法：

```
alter encryption key key_name
for database encryption
modify encryption with {[master key]
                        [[no] dual_control]}
```

- 要更改数据库加密密钥的所有者：

```
alter encryption key [[database.][owner].]dek_name
modify owner user_name
```

运行此选项的权限与 **alter encryption key** 的权限相同。

### 另请参见

- 删除数据库加密密钥（第 59 页）
- 备份数据库加密密钥（第 68 页）
- **create encryption key**（第 111 页）
- **drop encryption key**（第 136 页）
- 创建数据库加密密钥（第 57 页）

## 删除数据库加密密钥

要删除数据库加密密钥，请使用 **drop encryption key** 命令。

语法为：

```
drop encryption key key_name
```

此命令删除 **master** 数据库中 **sysencryptkeys** 表的数据库加密密钥。

---

**注意：**如果要删除的数据库加密密钥仍用于加密某个数据库，该命令将失败。

---

### 另请参见

- 更改数据库加密密钥（第 59 页）
- 备份数据库加密密钥（第 68 页）

- `create encryption key` (第 111 页)
- `drop encryption key` (第 136 页)
- 创建数据库加密密钥 (第 57 页)

## 创建加密数据库

---

要创建完全加密数据库，请使用 `create database` 命令。

指定在创建数据库时是否对该数据库加密，如果加密该数据库，则插入到该数据库中的数据也会自动加密。数据库在加密后，其大小不会发生变化；另外，无论数据库是否加密，其所有存储访问功能的工作方式保持不变。支持加密的数据库有以下几种类型：

- 常规用户数据库
- 临时数据库
- 存档数据库

不能加密内存数据库。

要创建加密数据库，请使用：

```
create [temporary] database database_name
    encrypt with key_name
```

其中：

- `database_name` 是要创建的加密数据库的名称。
- `key_name` 是数据库加密密钥的名称。

要创建加密存档数据库，请使用：

```
create archive database database_name
    encrypt with key_name
```

其中：

- `database_name` 是要创建的存档数据库的名称
- `key_name` 是加密备份数据库所用的密钥。SAP ASE 在数据库装载过程中验证 `key_name` 是否匹配。如果不匹配，恢复将失败。

### 示例

使用加密密钥 `dbkey` 创建一个名为 `demodb` 的加密数据库，其数据位于设备 `demodev` 上，日志位于设备 `demologdev` 上：

```
create database demodb on demodev log on demologdev encrypt with
dbkey
```

### 用法

使用 `create database` 命令的 `encrypt with` 选项不需要特殊权限。但是，用户需要具备对数据库加密密钥的 `select` 权限以能够引用其作为 `key_name`。

### 另请参见

- `create archive database` 用于完全数据库加密（第 108 页）
- `dbencryption_status`（第 101 页）
- `create database` 用于完全数据库加密（第 108 页）

## 加密现有数据库

---

在 SAP ASE 16.0 版中，您可以使用 `alter database` 命令加密未加密的数据库。

根据数据库大小的不同，加密可能会花费一段时间。为此，在数据库标记为加密后命令立即返回。加密将在后台进行，且进程对用户透明。要检查数据库加密的状态和进程，请运行 `sp_helpdb` 系统过程、`dbencryption_status()` 内置函数或 SAP Control Center 用户界面。记住：

- 数据库加密在数据库联机时进行。这表示数据库在进行加密时，其他用户可以对其进行访问，无需将数据库置于单用户模式。
- 加密进程不会中断对数据库执行的任何用户查询、更新或插入操作。
- 可暂停及恢复数据库加密，以便在重新启动 SAP ASE 后恢复数据库加密。
- 加密操作逐页执行。
- 无法更改存档数据库以进行加密和解密。
- SAP ASE 记录数据库的加密进程并提供实用程序报告其状态。

限制：

- 不能加密 `master`、`model`、`dbccdb` 以及 `dbccalt` 数据库。
- 不能解密正在进行加密的数据库，也不能加密正在进行解密的数据库。
- 不能卸下正在进行加密的数据库。
- 不能基于某个正在加密的数据库加载另外一个数据库。
- 在进行数据库加密的过程中，不要执行缩减数据库大小的命令。

语法为：

```
alter database database_name
{encrypt with key_name [parallel degree_of_parallelism]
| resume encryption [parallel degree_of_parallelism]
| suspend encryption
}
```

其中：

- `encrypt with key_name` 指示 SAP ASE 使用 `key_name` 加密数据库。具体来说，该命令从 `master` 数据库的 `sysencryptkeys` 系统表中检索相应的密钥 ID，并在其关联的 `sysdatabases` 行中设置 `enckeyid` 列。数据库处于以下状态时，SAP ASE 无法运行 `alter database` 并显示错误消息：
  - 已使用另一密钥加密。

- 正在加密。

如果对当前未进行加密的部分加密数据库运行此命令，且密钥名称与先前指定的密钥相同，则 **SAP ASE** 会将该命令视为已指定 `resume encryption` 选项。

- `parallel degree_of_parallelism` 确定为执行任务所要启动的工作线程数。

如果数量等于或少于“`number of worker processes`”配置，则将为每个数据库存储虚拟设备创建一个线程。`degree_of_parallelism` 值不应大于数据库设备数，因为额外的工作线程不会提高加密性能。如果不指定 `degree_of_parallelism`，**SAP ASE** 将基于联机引擎数以及数据库在不同设备中的分布方式在内部定义该参数的值。

- `resume encryption` 从之前加密被挂起的页恢复加密进程。

如果出现以下情况，该命令失败：

- **SAP ASE** 已有正在运行的加密进程。
- 从未对数据库启动加密。
- 加密进程已完成。

可以将 `parallel degree_of_parallelism` 与 `resume encrypt` 一起使用。

- `suspend encryption` 终止正在加密数据的所有加密工作线程。**SAP ASE** 将记录加密进度，以便 `resume encryption` 能够在前一个加密进程停止处重新开始加密。如果没有正在进行的加密进程，**SAP ASE** 将忽略此命令。

此示例使用加密密钥 `dbkey` 对现有数据库 `existdb` 进行加密：

```
alter database existdb encrypt with dbkey
```

此示例未指定并行度，而是由 **SAP ASE** 来决定并行加密 `existdb` 所应启动的工作线程数。

除并行度外，另一个影响数据库加密性能的主要因素是缓冲池大小。充足的缓冲区高速缓存和适当大小的缓冲池可确保 **SAP ASE** 将大量页面装载到内存中，供每个磁盘读取、执行加密并将其写回。

下面的示例显示了为要加密的数据库 `demodb` 配置缓冲区高速缓存和缓冲池大小可采取的步骤：

1. 为 `demodb` 创建特定数据高速缓存：

```
sp_cacheconfig demodb_cache, '10M'
```

这将创建一个名为 `demodb_cache` 的指定缓冲区高速缓存，数据库页面空间为 **10MB**。

2. 创建特定大小的缓冲池。缓冲池的大小应该为数据库页面大小的 8 倍。例如，数据库页面缺省大小为 **2K**，则缓冲池的大小应为  $8 \times 2 = 16K$ ：

```
sp_poolconfig demodb_cache, '10M' , '16k'
```

这将在名为 `demodb_cache` 的指定高速缓存中创建 10MB 的缓冲池，其中每个缓冲区的大小为 16K。

### 3. 将数据库绑定到缓冲区高速缓存：

```
sp_bindcache demodb_cache, demo_db
```

#### 另请参见

- `dbencryption_status`（第 101 页）
- `sp_helpdb`（第 146 页）
- 挂起加密进程（第 66 页）
- 恢复加密进程（第 67 页）
- `alter database` 用于完全数据库加密（第 102 页）
- 解密加密数据库（第 67 页）

## 加密状态和进度

---

可通过以下两种方法来查明某个数据库的加密状态。

要获得关于数据库是否加密以及正在加密数据库的加密进度方面的信息，请使用：

- `sp_helpdb` 系统过程。下面列出了语法，其中 *database\_name* 是数据库名称：  

```
sp_helpdb database_name
```
- `dbencryption_status` 内置函数。使用 `status` 获取关于数据库是否加密的信息，使用 `progress` 查明加密进程的进度：
  - ```
select dbencryption_status( "status" , db_id( "exisddb" ))
```
  - ```
select dbencryption_status( "progress" , db_id( "exisddb" ))
```

## 性能注意事项

---

现有数据库在进行加密时仍保持联机状态。请考虑性能问题，以减轻对用户访问数据库以及 SAP ASE 的总响应时间的影响。

为确保良好的数据库加密性能，需要考虑的因素包括：

- 多处理器计算机上的 SAP ASE 引擎数
- 存储数据库的磁盘数
- 与数据库关联的缓冲池的大小

在 `alter database` 中指定加密或解密的并行度值，本质上来说是指示 SAP ASE 执行操作时要启动的工作线程数。由于工作线程并发运行，因此最好将其分配到多个 CPU。同时应避免过度使用 CPU 资源，从而缩短 SAP ASE 的总响应时间。因此，决定并行度值时应考虑 SAP ASE 引擎数。

设备 I/O 是数据库加密过程中的主要瓶颈。SAP ASE 可以从两个角度应对该问题：

- 如果每个单独的设备都分配了一个工作线程，则可以独立并发执行设备 I/O 以获得最佳吞吐量。因此，决定并行度值时应考虑存储数据库的磁盘数。
- 如果每次设备读/写可以处理大量页面，那么性能将有所提高。在加密/解密的过程中，数据库必须处于联机状态。因此，必须利用现有缓冲区管理器机制来解决同步问题，而不是分配专有缓冲区。为此，可以创建足够大的缓冲区高速缓存和 I/O 较大的缓冲池，以帮助 SAP ASE 提高其加密性能。

本示例显示如何配置缓冲区高速缓存和缓冲池的大小，以完全加密名为 demodb 的数据库（该数据库的数据和日志分配到 11 个设备）：

```
> select dbid, segmap, lstart, size, vstart, vdevno from sysusages
where dbid=db_id('demodb')
```

| dbid | segmap | lstart   | size    | vstart | vdevno |
|------|--------|----------|---------|--------|--------|
| 4    | 3      | 0        | 92160   | 0      | 1      |
| 4    | 4      | 92160    | 30720   | 0      | 2      |
| 4    | 3      | 122880   | 184320  | 92160  | 1      |
| 4    | 4      | 307200   | 61440   | 30720  | 2      |
| 4    | 3      | 368640   | 419840  | 276480 | 1      |
| 4    | 4      | 788480   | 61440   | 92160  | 2      |
| 4    | 3      | 849920   | 122880  | 696320 | 1      |
| 4    | 4      | 972800   | 153600  | 153600 | 2      |
| 4    | 3      | 1126400  | 819200  | 819200 | 1      |
| 4    | 3      | 1945600  | 1638400 | 0      | 3      |
| 4    | 3      | 3584000  | 1638400 | 0      | 4      |
| 4    | 3      | 5222400  | 1638400 | 0      | 5      |
| 4    | 3      | 6860800  | 1638400 | 0      | 6      |
| 4    | 3      | 8499200  | 1638400 | 0      | 7      |
| 4    | 3      | 10137600 | 1638400 | 0      | 8      |
| 4    | 3      | 11776000 | 1638400 | 0      | 9      |
| 4    | 3      | 13414400 | 1638400 | 0      | 10     |
| 4    | 3      | 15052800 | 1638400 | 0      | 11     |
| 4    | 4      | 16691200 | 204800  | 307200 | 2      |

#### 1. 配置缓冲区高速缓存和缓冲池的大小：

- a. 为 demodb 创建特定数据高速缓存：

```
sp_cacheconfig demodb_cache, '100M'
```

这将创建数据库页面空间为 100MB 的名为 demodb\_cache 的缓冲区高速缓存。

- b. 创建特定大小的缓冲池，其中缓冲池的大小为数据库页面大小的 8 倍：

```
sp_poolconfig demodb_cache, '100M', '16k'
```

因为缺省数据库页面大小为 2K，所以缓冲池大小应为  $8 \times 2 = 16\text{KB}$ 。

这将在指定高速缓存 demodb\_cache 中创建 100MB 的缓冲池，其中的缓冲区为 16K。

- c. 将数据库绑定到缓冲区高速缓存：

```
sp_bindcache demodb_cache, demo_db
```



这会将数据库 `demo_db` 绑定到创建的缓冲区高速缓存 `demodb_cache`。

2. 确定要使用的并行度。在本示例中，共配置了 8 个 SAP ASE 引擎：

```
[Thread Pool:syb_default_pool]
```

线程数 = 8

最大工作线程数不应超过 8。

同时，由于 SAP ASE 使用 11 个数据库设备，因此最多需要 11 个工作线程来并行执行设备 I/O。因为 11 个工作线程会过度占用八个引擎，所以并行度应设置为 8。但是，为使 SAP ASE 维持其响应时间并执行其它操作，应将并行度选择为 6 以避免占用所有 CPU 资源。

- a. 确保配置足够多的工作线程：

```
sp_configure 'number of worker processes', 6
```

- b. 更改数据库 `demodb` 以进行加密：

```
alter database demodb encrypt with dbkey parallel degree 6
```

`sp_who` 显示 6 个工作线程：

```
>sp_who
fid      spid      status      loginame      origname
-----
      hostname      blk_spid      dbname
      tempdbname      cmd
      block_xloid      threadpool
-----
.....
      0      16      sleeping      NULL      NULL      NULL      0
master
      master DB      ENCRYPTION CONTROLLER      0
NULL
      16      1      sleeping      NULL      NULL      NULL      0
master
      master      WORKER PROCESS      0      NULL
      16      17      sleeping      NULL      NULL      NULL      0
master
      master      WORKER PROCESS      0      NULL
.....
```

`sp_helpdb` 可以报告加密进程和状态：

```
1> sp_helpdb demodb
2> go
name      db_size      owner      dbid      created      durability
lobcomplvl inrowlen status
-----
-----
demodb      33000.0 MB sa      4      Sept 27, 2013 full      0
NULL      encryption in progress: 18%
```

也可以使用 `dbencryption_status` 函数获取加密状态和进程：

```
1> select dbencryption_status("status", db_id('demodb'))
2> go
-----
2
1> select dbencryption_status("progress", db_id('demodb'))
```

```
2> go
-----
21
```

这表示已加密 21% 的数据库页面。

还可以使用 **dbencryption\_status** 查找特定段的进度：

```
1> select dbencryption_status("progress", db_id('demodb'),
92160)
2> go
-----
83
```

这表示在逻辑页面开头为 92160 的段中，83% 的页面已加密。

与未加密数据库相比，加密数据库需占用更多的缓冲区来执行加密和解密操作。如果因执行加密和解密操作，导致没有干净的缓冲区可用，请执行以下操作：

- 增加缓冲池大小和缓冲池清洗大小
- 将 **housekeeper free write percent** 配置为一个允许管家任务更频繁清洗缓冲区的值

## 挂起加密进程

---

要停止加密正在进行加密的数据库，请使用 **alter database** 命令的 **suspend encrypt** 选项：

```
alter database database_name
suspend encryption
```

### 另请参见

- 加密现有数据库（第 61 页）
- **dbencryption\_status**（第 101 页）
- **sp\_helpdb**（第 146 页）
- 恢复加密进程（第 67 页）
- **alter database** 用于完全数据库加密（第 102 页）

## quiesce database 命令和完全加密数据库

对正在进行加密的数据库运行 **quiesce database** 命令时，SAP ASE 将暂停加密进程。

运行 **quiesce database** 后，无需运行 **alter database** 的 **suspend encryption** 选项；**quiesce database** 自动挂起对数据库的 I/O 操作。

释放 **quiesce** 模式后，加密（或解密）任务将自动恢复；无需运行 **alter database** 中的 **resume encryption** 选项。

## 恢复加密进程

---

要恢复加密进程被中断或被挂起的数据库，请使用 **alter database** 命令的 `resume encryption` 选项：

```
alter database database_name
    resume encryption [parallel degree_of_parallelism]
```

### 另请参见

- 加密现有数据库（第 61 页）
- `dbencryption_status`（第 101 页）
- `sp_helpdb`（第 146 页）
- 挂起加密进程（第 66 页）
- `alter database` 用于完全数据库加密（第 102 页）

## 解密加密数据库

---

要解密完全加密的数据库，请使用 **alter database** 命令。

语法为：

```
alter database database_name
    {decrypt [with key_name] [parallel degree_of_parallelism]
    | resume decryption [parallel degree_of_parallelism]
    | suspend decryption}
```

其中：

- *database\_name* - 要解密的完全加密数据库的名称。
- *key\_name* - （可选）与用于加密数据库的数据库加密密钥相同。如果指定其它密钥名称，该命令将失败，同时 SAP ASE 会显示一条错误消息。
- `resume decryption` - 恢复之前被挂起的数据库的解密进程。如果 *database\_name* 已完全解密，SAP ASE 将忽略此命令。
- *parallel degree\_of\_parallelism* - 指定为执行任务所要启动的工作线程数。
- `suspend decryption` - 终止解密进程。SAP ASE 会记录进程停止的位置，以便 `resume decrypt` 可从数据库的正确位置重新启动解密进程。

要使用该命令，必须对数据库的 *key\_name* 具有 **select** 权限。

## 恢复完全加密数据库

---

如果由于主密钥或双主密钥不可用，SAP ASE 无法在启动期间检索数据库加密密钥，则 SAP ASE 将忽略加密数据库。

SAP ASE 需要访问数据库加密密钥以了解如何处理完全加密数据库。数据库加密密钥本身也进行了加密，由主密钥解密。

要在重新启动 SAP ASE 后连接到服务器，主密钥或双主密钥的口令持有者可以设置加密口令：

```
set encryption passwd for key [dual] master
```

这样主密钥便能够解密数据库加密密钥，同时数据库加密密钥可以使完全加密数据库联机：

```
online database encrypted_database_name
```

随后会在服务器重新恢复到联机状态时进行数据库恢复。

还可设置自动恢复；请参见《加密列用户指南》中的“在无人值守启动模式下启动 Adaptive Server”。

## 备份（转储）完全加密数据库

---

由于以透明方式执行加密过程，因此备份完全加密的数据库与备份普通数据库（非加密数据库）相同。

要装载转储加密数据库得来的副本，必须使用加密转储所使用的加密密钥。

数据库加密密钥存储于所备份数据库外部的 master 数据库中。为此，在执行 **dump database** 命令时，备份过程并不自动应用到数据库加密密钥；必须在备份数据库的同时单独备份数据库加密密钥和主密钥。

要备份密钥值，可使用以下任意一种方法：

- 使用 **ddlgen** 实用程序生成 DDL 语句；或者，
- 直接进行备份。

## 备份数据库加密密钥

---

要重新获得可恢复性，必须备份数据库加密密钥、主密钥或双主密钥以及加密数据库。

本示例使用 **ddlgen** 实用程序生成针对数据库加密密钥的 SQL 语句：

```
ddlgen -Usa -P -Sserver -TEK -Nmaster.owner.dek_name
```

针对主密钥或双主密钥生成 SQL 语句的语法与此类似。

**另请参见**

- 更改数据库加密密钥 (第 59 页)
- 删除数据库加密密钥 (第 59 页)
- create encryption key (第 111 页)
- drop encryption key (第 136 页)
- 创建数据库加密密钥 (第 57 页)

## 恢复 (装载) 完全加密数据库的备份

按照恢复普通未加密数据库的方式恢复完全加密数据库。

要装载加密数据库转储，必须先执行以下操作：

1. 恢复主密钥和数据库加密密钥。
2. 使用用于要装载数据库的数据库加密密钥创建要加密的目标数据库。

使用该命令恢复加密数据库，其中 *database\_name* 为要恢复的加密数据库的名称：

```
load database database_name
```

**注意：**不能将验证选项 (load database database\_name with verify only = full) 用于加密数据库。指定该选项时，Backup Server 读取所有行并检查行格式是否有效。由于 Backup Server 不能理解加密文本，命令将失败，Backup Server 将显示一条错误消息。

执行 **load database** 恢复加密数据库时，SAP ASE 将验证目标数据库：

- 是否为加密数据库。如果不是加密数据库，SAP ASE 将显示一条错误消息，**load database** 命令将失败。
- 是否拥有正确的数据库加密密钥。如果数据库加密密钥不匹配，SAP ASE 将显示一条错误消息。

## 加密数据库的装载行为

装载行为因目标数据库和要恢复的数据库或事务日志的加密状态而异。

| 装载行为     | 未加密目标数据库 | 加密目标数据库                                   | 部分加密目标数据库                               | 部分解密目标数据库                           |
|----------|----------|-------------------------------------------|-----------------------------------------|-------------------------------------|
| 未加密数据库转储 | 允许。      | 仅在使用 with override 子句时允许。转储安全状态反映在目标数据库中。 | 仅在使用 with override 子句时允许。转储状态反映在目标数据库中。 | 仅在使用 with override 子句时允许。将反映转储安全状态。 |

| 装载行为      | 未加密目标数据库 | 加密目标数据库                                 | 部分加密目标数据库                                 | 部分解密目标数据库                                 |
|-----------|----------|-----------------------------------------|-------------------------------------------|-------------------------------------------|
| 未加密事务转储   | 允许。      | 允许。将目标数据库标记为部分加密。                       | 允许。目标数据库将保留其部分加密状态。                       | 允许。目标数据库将保留其部分加密状态。                       |
| 加密数据库转储   | 不允许。     | 允许。                                     | 允许。转储安全状态反映在目标数据库中。                       | 仅在使用 with override 子句时允许。转储安全状态反映在目标数据库中。 |
| 加密事务转储    | 不允许。     | 允许。                                     | 允许。目标数据库将保留其部分加密状态。                       | 不允许。                                      |
| 部分加密数据库转储 | 不允许。     | 允许。转储安全状态反映在目标数据库中。                     | 允许。目标数据库将保留其部分加密状态。                       | 仅在使用 with override 子句时允许。转储状态反映在目标数据库中。   |
| 部分加密事务转储  | 不允许。     | 允许。转储安全状态反映在目标数据库中。                     | 允许。目标数据库将保留其部分加密状态。                       | 不允许。                                      |
| 部分解密数据库转储 | 不允许。     | 仅在使用 with override 子句时允许。转储状态反映在目标数据库中。 | 仅在使用 with override 子句时允许。转储安全状态反映在目标数据库中。 | 允许。目标数据库将保留其部分解密状态。                       |
| 部分解密事务转储  | 不允许。     | 不允许。                                    | 不允许。                                      | 允许。目标数据库将保留其部分解密状态。                       |

## 删除正在进行加密的数据库

可以删除正在进行加密或解密的数据库。

对正在进行加密或解密的数据库执行 **drop database** 命令时，**drop database** 会终止加密/解密进程，搜索 `sysattributes` 系统表，清除所有进程信息，然后删除数据库。

## 装入和卸下完全加密数据库

不能装入正在进行加密或解密的数据库。

不能装入加密数据库。

请不要对加密数据库使用 **unmount database** 命令；命令会失败，SAP ASE 显示如下消息：

```
Could not unmount encrypted database 'mydatabase'.
```

要卸下加密数据库，请先将其解密。

### 另请参见

- 解密加密数据库（第 67 页）

## 存档数据库和完全加密

---

存档数据库是只读的。加密语法指示存档数据库可以装载加密数据库转储。

要进行数据库备份和装载，需要恢复主密钥和数据库加密密钥，然后将数据库加密密钥与存档数据库相关联。

对于完全加密的存档数据库，其转储或装载步骤与普通数据库相同。

要创建存档数据库，请使用：

```
create archive database database_name  
    encrypt with key_name
```

其中：

- *database\_name* 是要创建的存档数据库的名称
- *key\_name* 是加密存档数据库所用的密钥。SAP ASE 在数据库转储过程中检验 *key\_name* 是否匹配。如果不匹配，恢复将失败。

与普通数据库不同，存档数据库提供修改页面区域，该区域存储由于重做/撤消操作和事务装载操作所引起的页面修改或分配信息。加密存档数据库时，还需要使用存档数据库的数据库加密密钥对修改页面区域的数据进行加密。

使用 **create archive database** 命令的 **encrypt with** 选项并不需要特殊权限。但是，用户需要具备对数据库加密密钥的 **select** 权限以引用其作为 *key\_name*。

### 另请参见

- 备份（转储）完全加密数据库（第 68 页）
- 恢复（装载）完全加密数据库的备份（第 69 页）

## 完全数据库加密和系统更改

---

下面是与 SAP ASE 16.0 的完全数据库加密功能相关联的系统更改。

- **alter database** 命令

- **alter encryption key** 命令
- **create archive database** 命令
- **create database** 命令
- **create encryption key** 命令
- **dbencryption\_status()** 内置函数
- **ddlgen** 实用程序
- **sp\_encryption** 系统过程
- **sybmigrate** 实用程序
- **sysdatabases** 系统表

### 另请参见

- **create archive database** 用于完全数据库加密 (第 108 页)
- **dbencryption\_status** (第 101 页)
- **sp\_helpdb** (第 146 页)
- **sp\_encryption** (第 145 页)
- **ddlgen** (第 169 页)
- **sybmigrate** (第 170 页)
- 更改过的系统表 (第 160 页)
- **alter database** 用于完全数据库加密 (第 102 页)
- **create database** 用于完全数据库加密 (第 108 页)
- **create encryption key** (第 111 页)
- **drop encryption key** (第 136 页)



# 可伸缩性增强和功能

SAP ASE 16.0 版通过改善运行时记录和锁、元数据以及闩锁管理提高了可伸缩性。

## 运行时日志记录增强功能

---

SAP ASE 16.0 包括运行时日志记录增强功能。

将日志记录传输到 `syslogs` 时要求 SAP ASE 获得日志锁，在繁忙的 OLTP 系统中，该日志锁可能会成为争用点。通过对每个事务使用用户日志高速缓存，SAP ASE 能够减少争用并提高运行时日志记录性能。用户日志高速缓存先缓冲日志记录，然后将这些记录传输到 `syslogs`，同时允许 SAP ASE 以批量方式向 `syslogs` 发送日志记录，而非单独发送。

为充分利用用户日志高速缓存，在事务完成时，SAP ASE 仅将其日志记录传输到日志中一次。但在使用数据行锁定表的繁忙 OLTP 系统中，当并发打开的事务对同一个数据页进行更改时，SAP ASE 可能需要在事务完成前，将日志记录从用户日志高速缓存传输到 `syslogs`，这样可以减少批处理量但增加了对日志锁的争用。在使用多个并发事务更新同一个表的繁忙 OLTP 系统中，如果 SAP ASE 在事务完成前频繁将日志记录从用户日志高速缓存传输到日志，那么将日志记录缓冲在用户日志高速缓存所带来的优势可能会被全盘否定。

为减少争用，SAP ASE 16.0 及更高版本将每个用户日志高速缓存分成许多小块，每个块的大小为服务器逻辑页的大小。SAP ASE 16.0 版本将日志记录直接添加到用户日志高速缓存中的当前活动块（与之前版本将日志记录直接添加到用户日志高速缓存的方式相同）。当并发打开的事务对同一个数据页进行更改时，SAP ASE 将用户日志高速缓存中的当前块添加（或链接）到全局队列的末尾，而不是将日志记录直接从用户日志高速缓存中移到 `syslogs`；SAP ASE 稍后可将日志记录从全局队列中传输到 `syslogs`。这样 SAP ASE 便能够高效地向 `syslogs` 进行批量添加，改善运行时记录的性能，而不受与数据行已锁定的表相关问题的影响。

将当前块添加到全局队列后，用户日志高速缓存中新的可用块变成当前块，并继续接受来自事务的日志记录。

使用配置参数 `user log cache queue size` 来启用和禁用此项功能。将 `user log cache queue size` 设置为以下值时：

- 1 - SAP ASE 使用队列策略（缺省值）
- 0 - SAP ASE 使用 16.0 之前版本的行为

如果配置 SAP ASE 使用队列策略，则必须将 `user log cache size` 设置成服务器逻辑页大小的至少四倍。也就是说，每个用户日志高速缓存必须至少有 4 个用户日志高速缓存块（每个块的大小为逻辑页的大小）。如果值为 `DEFAULT`，则 SAP ASE 16.0 版本会将值设置为服务器逻辑页大小的 4 倍。

在设置 **user log cache size** 和 **user log cache queue size** 的值时，请切记以下两点：

- 如果将 **user log cache queue size** 设置成大于 0 的值，则不能将 **user log cache size** 设置成小于逻辑页大小 4 倍的值。
- 如果对 **user log cache size** 设置的值小于或等于逻辑页大小的 4 倍，则不能将 **user log cache queue size** 的值从 0 更改为 1。

## 锁管理增强

---

SAP ASE 16.0 及更高版本涵盖了锁管理的多项增强。

在 SAP ASE 16.0 之前的版本中，螺旋锁偶尔会导致锁争用。SAP ASE 16.0 版通过优化以下方面减少了锁争用现象：

- 引擎锁传送 - 改善了全局池与引擎本地高速缓存之间锁的传送。
- 引擎锁高速缓存 - 优化了引擎可在本地进行高速缓存的锁的数量，方式为：
  - 增加引擎本地高速缓存的缺省大小
  - 增加从全局池向本地高速缓存传送的锁的大小。
  - 使用锁块而非单个锁改善本地与全局间的锁传送
  - 降低清除或回收本地到全局高速缓存的锁的频率
- LOCK\_VERIFY 操作 - 优化 LOCK\_VERIFY 操作以部署优化方法（如有可能）并避免使用锁。
- 锁升级 - 跟踪反复失败的锁升级尝试，并在达到一定次数后针对造成锁升级失败的 DML 语句禁用锁升级。
- 日志信号锁 - 避免使用日志信号锁，因为 `semawait` 只能持有一个锁。
- 死锁检查 - 确保针对 **deadlock checking period** 所指定的较小值永远不会产生小于 1 的值，并且死锁检查要在专用的服务线程上进行
- 热 DOL 表 - 允许 DOL 表成为热表，这将减轻对表锁的大量争用

这些功能旨在提高性能：当 SAP ASE 使用这些增强时，不会对性能有任何影响。锁管理增强在缺省情况下处于开启状态，无需执行任何配置进行启用。

## 元数据和门锁管理增强

---

SAP ASE 16.0 及更高版本针对高事务发生率环境下的门锁冲突降低了 CPU 利用率。

其中包括减少对以下内容的争用：

- 对于事务发生率非常高的跨数据库查询和事务，减少对内部结构的争用，包括由于隐式或显式跨数据库引用而导致的对锁、门锁以及数据高速缓存的争用
- 对于事务发生率高的操作，减少对含多个分区的表的争用
- 在需要频繁创建和删除表的情况下，减少对内部结构的争用，其中包括减少对某些系统表的锁和门锁争用

这些功能旨在提高性能：当 SAP ASE 使用这些增强时，不会对性能有任何影响。元数据和闩锁管理增强在缺省情况下处于开启状态，无需执行任何配置进行启用。

SAP ASE 16.0 及更高版本针对高事务发生率环境下的闩锁冲突降低了 CPU 利用率。

在事务发生率极高的情况下，对单个页执行并发操作会引起争用，表现为对单个高速缓存分区的数据高速缓存螺旋锁争用。通常情况下，通过跟踪该争用可追溯到针对单个数据页的大量闩锁请求。**sp\_chgattribute exp\_row\_size** 参数用于管理服务器上的空间。SAP ASE 将根据您为 **exp\_row\_size** 设置的大小（可限制每页的行数）在数据页上预留空间。

SAP ASE 16.0 及更高版本允许您针对只含有固定长度列的表设置 **exp\_row\_size** 的值。要减少争用，则需减少每页的行数，通过将 **exp\_row\_size** 设置为逻辑页大小（例如，8,192）即可。

高负载的情况下会增加对过程高速缓存的争用，SAP ASE 通过为每个引擎留出本地内存来缓解争用。由于该内存为本地内存，因此对其进行访问并不会引起争用。

在 16.0 之前的版本中，SAP ASE 为引擎本地高速缓存 (ELC) 留出 25% 的过程高速缓存以供特定大小的请求使用。通过使用跟踪标志 758 可将大小增加至过程高速缓存的 50%，并可使 ELC 适用于所有请求大小。

缺省情况下，SAP ASE 16.0 及更高版本中的 ELC 处于启用状态。过程高速缓存的 50% 用于 ELC，这可满足所有请求大小。**engine local cache percent** 配置参数用于确定 ELC 的大小。缺省值 50 表示本地高速缓存使用百分之五十的过程高速缓存，而每个引擎会分到如下数量的缓存：

```
((0.50 * procedure_cache_size) / number_of_online_engines)
```

增加 ELC 的大小可减少过程高速缓存的争用。例如，要将 ELC 的大小从 50% 提高至 60%，请发出以下命令：

```
sp_configure 'engine local cache percent', 60
```

另外，SAP ASE 16.0 及更高版本增加了下列配置参数以控制引擎本地高速缓存：

- **enable large chunk elc**（取代跟踪标志 758）
- **large allocation auto tune**（取代跟踪标志 753）

缺省情况下，这些配置参数处于启用状态。



## 监控基于阈值的事件

SAP ASE 16.0 及更高版本中包含配置、记录和列出阈值事件的功能。

要配置查询使用 CPU 花费的时间，请使用 `sp_add_resource_limit cpu_time` 限制类型。

要将事件记录到 `monThresholdEvent` 表，请将 `sp_add_resource_limit action` 参数设置为 5（请参见《参考手册：过程》）。

语法为：

```
sp_add_resource_limit name, appname, rangename, limittype,  
limitvalue [, enforced [, action [, scope ]]]
```

在本示例中，如果执行查询批处理需要的 CPU 时间超过 120 秒，SAP ASE 会发出警告并将事件记录到 `monThresholdEvent`：

```
sp_add_resource_limit NULL, payroll, tu_wed_7_10,cpu_time, 120, 2,  
5, 2
```

通过 `monThresholdEvent` 可以记录与已配置资源限制的所有冲突，还可确定以下各项的阈值：

- 每个事务的 `tempdb` 使用率
- 查询或事务的总运行时
- 由查询获取或编写的行数
- 由查询读取的行数
- 查询的估计计划开销
- 查询的逻辑/物理 I/O 总数
- 查询的总 CPU 时间



# 多个触发器

SAP ASE 16.0.版中引入了创建多个触发器及指定在语句执行后引发触发器的顺序的功能。

## 创建多个触发器

---

对于每个命令 (**insert**、**update** 以及 **delete**)，最多可为某个表创建 50 个不同的触发器。在 **create trigger** 命令中使用新的 **order** 参数并指定引发触发器的顺序。您可以在不使用 **order** 子句的情况下创建多个触发器。

语法为：

```
create [or replace] trigger [owner.]trigger_name
  on [owner.]table_name
  for {insert | update | delete}
  [order integer]
  as sql_statements
```

**order integer** 指定引发触发器的部分顺序或完整顺序：

- 使用 **order** 子句创建所有触发器时，为完整顺序。
- 部分触发器未使用 **order** 子句指定时，为部分顺序。未使用 **order** 子句创建的触发器隐式采用序列号 0，且不具备定义顺序，只是会先于使用 **order** 创建的触发器引发。

---

**注意：** 只能将 **order integer** 子句与 **for {insert | update | delete}** 一起使用；不能将其与 **instead of {insert | update | delete}** 触发器一起使用。

---

如果您为 **order** 指定重复编号，SAP ASE 将返回错误。**order** 编号不必连续；实际上，非连续编号可能更可取，因为这样允许在序列中间插入新的触发器。

使用 **sp\_helptrigger** 列出下列内容：

- **tablename** 指定的表上创建的所有触发器
- 提供触发操作的 **insert**、**update** 或 **delete** 命令
- 触发器序列号

另请参见

- **create trigger** 用于创建多个触发器（第 130 页）

## 更改触发器引发顺序

---

要更改触发器的引发顺序，请在 **create trigger** 命令中使用 **or replace** 选项，其中所使用的触发器名称、操作以及触发器主体与源触发器相同，但需指定新的 **order** 编号。

## Merge 语句中触发器的顺序

---

merge 语句按照特定顺序引发触发器。

**merge** 语句按照以下顺序在目标表上引发触发器：

1. **insert**
2. **update**
3. **delete**

这意味着，即使 **update** 语句的序号低于 **insert** 的序号，也会先引发 **insert** 语句的触发器。

但是，同一操作的多个触发器是按顺序引发的。即，首先按照序号顺序引发 **insert** 的所有触发器，然后按照顺序引发 **update** 的所有触发器，最后是 **delete**。

在本示例中，**dbo** 在 **GlobalSales** 表上创建了以下触发器：

- **trigger1 (delete)**，序号为 1
- **trigger2 (delete)**，序号为 4
- **trigger3 (insert)**，序号为 1
- **trigger4 (insert)**，序号为 5

**merge** 语句将数据合并到 **GlobalSales**。如果 **merge** 语句包含对 **GlobalSales** 的 **insert** 和 **delete** 操作，则在语句执行后，**SAP ASE** 将按照以下顺序引发触发器：

- **trigger3**
- **trigger4**
- **trigger1**
- **trigger2**

## 多个触发器的事务型行为

---

在触发器中执行的回退事务将导致引发触发器的 **insert**、**update** 或 **delete** 语句以及触发器所执行的任何工作的回退。

对于多个触发器，一个触发器的回退事务还将回退其它已引发触发器的工作，并拒绝为当前的 **insert**、**update** 或 **delete** 命令引发任何剩余的触发器。



## 禁用和重新启用触发器

---

可以使用 `alter table` 命令禁用和重新启用多个触发器。

使用 **alter table** 命令分别禁用或重新启用多个触发器：

要禁用或重新启用触发器，请使用：

```
alter table table_name {disable | enable} trigger trigger_name
```

## @@trigger\_name 全局变量

---

`@@trigger_name` 全局变量返回正在执行的触发器的名称。

可将下列内容放置到触发器主体或通过触发器调用（在任意嵌套级别）的存储过程的主体中。

```
select @@trigger_name
```

如果已引发嵌套触发器，`@@trigger_name` 将保留最近一次引发的触发器的名称。

多个触发器

# 删除残留数据

删除残留数据可增强数据库数据的安全性。

某些删除空间的数据库操作不一定会实际删除数据。使用 **dbcc** 实用程序的用户可能会看到这些残留数据，因此会带来安全威胁。SAP ASE 16.0 版本引入了这样一种功能，即如果启用该功能，则当用户执行数据库操作时系统会自动将残留数据清零。

SAP ASE 16.0 之前的版本在将磁盘页清零时实际上并没有删除残留数据，这样磁盘上可能会留有敏感数据。在 16.0 及更高版本中，可将数据标记为敏感，然后将 SAP ASE 配置为在执行删除或更新操作后删除残留数据。

---

**注意：**您可在用户执行此类数据库操作时自动将残留数据清零。但您无法从系统数据库（如 master、sybsystemdb 和 sybsystemprocs）以及这些数据库的系统表中删除残留数据。

---

有关产生残留数据的操作以及如何启用残留数据删除的信息，请参见《安全性管理指南》中的删除残留数据。

删除残留数据

## 配置历史记录跟踪

SAP ASE 16.0 版本添加了对服务器的配置更改历史记录的跟踪功能。

**sp\_confighistory** 系统过程管理配置更改的历史记录，并在 `sybsecurity` 数据库中存储更改的相关数据。

跟踪的配置属性包括

- 服务器范围的配置参数
- 数据库选项
- 数据高速缓存和数据高速缓存池属性
- 引擎线程
- 对服务器配置文件的更改。

必须安装 `sybsecurity` 审计数据库以跟踪这些属性。

**sp\_confighistory** 显示 SAP ASE 配置更改，包括已更改的配置选项、旧值和新值、进行更改的用户以及更改时间。SAP ASE 在 `sybsecurity` 数据库中存储配置更改的记录。查询 `ch_events` 视图或运行 **sp\_confighistory** 以访问这些记录。

例如，以下的输出包括的更改有“启用审计”以及“将 **max online engines** 的值从 5 更改为 7”：

```
area type target element oldvalue newvalue mode timestamp username
instanceid
-----
AUDIT global auditing NULL NULL off on NULL Jul 15 2013 2:22PM sa
NULL
SERVER sp_configure NULL max online engines 5 7 static Jul 15 2013
2:23PM sa NULL
```

## 配置 SAP ASE 以跟踪配置更改

要安装 **sp\_confighistory**，请运行 `installsecurity` 脚本。

### 前提条件

**注意：** `ch_events` 从所有审计表中收集信息，在添加或删除审计表时将失去同步。如果出现这种情况，`ch_events` 可能不会包括某些配置历史记录更改，或者在查询 `ch_events` 时可能会看到错误消息 208（未找到表）和 4413（视图不可用）。

在添加或删除审计表时，使用 **sp\_confighistory create\_view** 来更新 `ch_events`。如果存在该视图，**sp\_confighistory create\_view** 会将其删除并创建与当前审计表对应的新视图。

安装并启用审计系统。请参见《安全性管理指南》。

### 过程

1. 启用配置历史记录跟踪（需要 `sa_role`、`sso_role` 或者在已启用细化权限时需要 `manage auditing`）：

```
sp_audit "config_history", "all", "all", "on"
```

---

**注意：**发出 `sp_audit` 记录在配置历史记录中。

---

2. 启用审计：

```
sp_configure 'auditing', 1
```

3. 转到 `sybsecurity` 数据库：

```
use sybsecurity
```

4. 创建 `ch_events` 视图：

```
sp_confighistory create_view
```

## 捕获的更改

---

当启用配置历史记录审计时，SAP ASE 将捕获多个事件。

如果新值与旧值相同，`ch_events` 不记录更改。

`ch_events` 记录的配置更改（下面进行详细介绍）：

- 配置文件的更改
- 读取配置文件
- **sp\_configure** 更改
- 服务器选项的更改
- 数据库选项的更改
- 高速缓存配置的更改
- 跟踪标志和开关的更改
- 引擎数的更改
- SAP ASE Startup 和 Shutdown 事件
- 启用或禁用审计

### 启动配置更改

如果在 SAP ASE 关闭时修改 SAP ASE 配置文件，则 SAP ASE 将在 `ch_events` 表中记录其启动的任何配置更改（将这些更改的 `mode` 和 `username` 值记录为 `NULL`）。

### 读取配置文件

`ch_events` 记录您读取、写入、验证和恢复配置文件时的事件，但不记录配置值更改。例如，如果更改 **number of user connections** 的值然后发出：

```
sp_configure "configuration file", 0, "read", "srv.config"
```

`ch_events` 将记录您读取配置文件的操作，但不记录配置值更改。

#### *sp\_configure 更改*

SAP ASE 记录所有由 `sp_configure` 所做的更改，包括：

- 配置参数的名称
- 旧配置值
- 新配置值
- 参数是动态的还是静态的
- 更改日期和时间的戳
- 执行更改的用户登录

由读取配置文件而导致的配置更改未被记录。也就是说，SAP ASE 记录读取、写入、验证和恢复操作，但不会记录由读取操作导致的配置更改。也可以通过读取其它配置文件或手动修改配置文件来更改配置值。尽管 SAP ASE 记录其读取文件的操作，但是不会记录单个参数更改。

#### *服务器选项的更改*

`ch_events` 记录所有由 `sp_serveroption` 所做的更改，包括：

- 受影响的服务器的名称
- 更改的选项的名称
- 旧选项值
- 新选项值
- 更改日期和时间的戳
- 执行更改的用户登录

#### *数据库选项的更改*

`ch_events` 记录由 `sp_dboption` 所做的以下更改：

- 受影响的数据库的名称
- 更改的选项的名称
- 旧选项值
- 新选项值
- 更改日期和时间的戳
- 执行更改的用户登录

#### *高速缓存配置的更改*

`ch_events` 记录所有由 `sp_cacheconfig` 和 `sp_poolconfig` 对高速缓存配置所做的更改。

记录的 `sp_cacheconfig` 更改包括：

- 受影响的高速缓存的名称

## 配置历史记录跟踪

- 旧高速缓存大小
- 新高速缓存大小
- 属性（高速缓存类型、高速缓存替换策略、分区号），适用时含此项
- 更改日期和时间的戳
- 执行更改的用户登录
- （仅限于 **Cluster Edition**）此更改应用的实例

记录的 **sp\_poolconfig** 更改包括：

- 受影响的高速缓存的名称
- 配置池
- 旧高速缓存池大小
- 新高速缓存池大小
- 更改的属性的名称（受影响的池、清洗大小、异步预取 (APF) 百分比）
- 更改日期和时间的戳
- 执行更改的用户登录
- （仅限于 **Cluster Edition**）此更改应用的实例

### *跟踪标志和开关的更改*

**ch\_events** 记录对 **dbcc traceon | off** 和 **set switch on | off** 的更改。

记录的 **dbcc traceon | off** 更改包括：

- 受影响的跟踪标志
- 会话 ID
- 旧跟踪标志状态（**on** 或 **off**）
- 新跟踪标志状态（**on** 或 **off**）
- 更改日期和时间的戳
- 执行更改的用户登录
- （仅限于 **Cluster Edition**）此更改应用的实例

记录的 **set switch on | off** 更改包括：

- 受影响的开关编号
- 旧开关状态（**on** 或 **off**）
- 新开关状态（**on** 或 **off**）
- 更改的属性的名称（服务器范围或特定于会话、**with override**、**with no\_info**）
- 更改日期和时间的戳
- 执行更改的用户登录

### *引擎数的更改*

由 **create thread pool**、**alter thread pool** 和 **drop thread pool** 跟踪的更改包括：

- 名称受影响的池的名称



- 旧的池大小
- 新的池大小
- 更改的属性的名称（新的池名称、idle timeout 等）
- 更改日期和时间的戳
- 执行更改的用户登录
- （仅限于 Cluster Edition）此更改应用的实例

### SAP ASE Startup 和 Shutdown 事件

ch\_events 为 SAP ASE 和 Cluster Edition 的实例记录 startup、shutdown、shutdown with nowait 和 abrupt（非预定）shutdown 事件的以下信息：

- 操作名称（startup、shutdown、shutdown with nowait、abrupt shutdown）。
- 等待关闭花费的时间。不适用于 shutdown with no\_wait。
- 启动服务器的主机的名称。
- 更改日期和时间的戳。
- 执行更改的用户登录。
- （仅限于 Cluster Edition）此更改应用的实例。

---

**注意：** 由于 ch\_events 在您发出 shutdown 命令时记录关闭，因此，如果多次发出命令，ch\_events 会在一个正常关闭期间记录多个关闭。

---

### 启用或禁用审计

ch\_events 记录有关跟踪、全局审计和配置历史记录审计的以下信息：

- 操作名称（enable 或 disable）
- 更改日期和时间的戳
- 执行更改的用户登录

## 查询 ch\_events 以查看更改

---

SAP ASE 引入了 ch\_events 视图作为 sybsecurity 数据库的一部分。

ch\_events 以易于读取的格式显示配置更改历史记录数据。可以直接查询 ch\_events 或使用 sp\_confighistory 系统过程来生成配置更改历史记录的报告。这两种方法可提供相同的信息。

使用 select 命令可灵活运用 Transact-SQL™ 语言来限定结果集（必须先转到 sybsecurity 数据库才能从 ch\_events 视图进行选择）。sp\_confighistory 可提供更为精简的结果集。

例如，如果在 SAP ASE 中进行以下配置更改：

```
sp_dboption subsystemprocs, "delayed commit", false
sp_cacheconfig pub_cache, '10M'
sp_cacheconfig pub_log_cache, '2000K', logonly
```

然后关闭并重新启动服务器，`sp_confighistory` 将返回：

```
sp_confighistory
```

| area     | type            | target          | element             | oldvalue            |
|----------|-----------------|-----------------|---------------------|---------------------|
|          | newvalue        | mode            | timestamp           | username instanceid |
| AUDIT    | global auditing | NULL            | NULL                | off                 |
|          | on              | NULL            | Aug 22 2013 11:56AM | sa NULL             |
| DATABASE | sp_dboption     | sybssystemprocs | delayed commit      | true                |
|          | false           | NULL            | Aug 22 2013 3:16PM  | sa NULL             |
| CACHE    | sp_cacheconfig  | pub_cache       | NULL                | 10240               |
|          | not changed     | NULL            | Aug 22 2013 3:18PM  | sa NULL             |
| CACHE    | sp_cacheconfig  | pub_log_cache   | cache type: logonly | 2000                |
|          | not changed     | NULL            | Aug 22 2013 3:19PM  | sa NULL             |
| SUSD     | shutdown        | NULL            | NULL                | NULL                |
|          | NULL            | NULL            | Aug 22 2013 3:49PM  | sa NULL             |
| SUSD     | startup         | NULL            | tigger              | NULL                |
|          | NULL            | NULL            | Aug 22 2013 3:50PM  | sa NULL             |

在使用 `sp_confighistory` 时包含日期可选择一段时期内的更改。本示例显示 2013 年 8 月 23 日之后进行的所有更改：

```
sp_confighistory "Aug 23 2013"
```

| area | type     | target     | element | oldvalue | newvalue | mode | timestamp           |
|------|----------|------------|---------|----------|----------|------|---------------------|
|      | username | instanceid |         |          |          |      |                     |
| SUSD | shutdown | NULL       | NULL    | NULL     | NULL     | NULL | Aug 23 2013 9:00AM  |
|      | sa       |            | NULL    |          |          |      |                     |
| SUSD | startup  | NULL       | tigger  | NULL     | NULL     | NULL | Aug 23 2013 10:38AM |
|      | NULL     |            | NULL    |          |          |      |                     |

发出 `select` 命令可提供以下结果集：

```
use sybsecurity
go
select * from ch_events
go
```

| area     | type            | target          | element             |
|----------|-----------------|-----------------|---------------------|
| oldvalue | newvalue        | mode            | timestamp           |
|          |                 |                 | username            |
|          |                 |                 | instanceid          |
| AUDIT    | global auditing | NULL            | NULL                |
| off      | on              | NULL            | Aug 22 2013 11:56AM |
|          |                 |                 | sa                  |
|          |                 |                 | NULL                |
| DATABASE | sp_dboption     | sybssystemprocs | delayed commit      |

|       |                |               |             |        |      |                     |
|-------|----------------|---------------|-------------|--------|------|---------------------|
| true  | false          | NULL          | Aug 22 2013 | 3:16PM | sa   | NULL                |
| CACHE | sp_cacheconfig | pub_cache     |             |        | sa   | NULL                |
| 10240 | not changed    | NULL          | Aug 22 2013 | 3:18PM | sa   | NULL                |
| CACHE | sp_cacheconfig | pub_log_cache |             |        | sa   | cache type: logonly |
| 2000  | not changed    | NULL          | Aug 22 2013 | 3:19PM | sa   | NULL                |
| SUSD  | shutdown       | NULL          |             |        | sa   | NULL                |
| NULL  | NULL           | NULL          | Aug 22 2013 | 3:49PM | sa   | NULL                |
| SUSD  | startup        | NULL          |             |        | sa   | tiger               |
| NULL  | NULL           | NULL          | Aug 22 2013 | 3:50PM | NULL | NULL                |

包括 **last** 参数可查看最新更改的项目：

```
sp_confighistory last

area type      target element oldvalue newvalue mode timestamp
-----
-----
-----
-----
-----
SUSD startup NULL  tigger  NULL    NULL    NULL Aug 22 2013 3:50PM
      NULL    NULL
```



## 对 **dump database** 的循环冗余检查

SAP ASE 针对使用压缩选项创建的数据库或事务转储的原始数据意外更改增加了循环冗余检查，目的是检查并验证压缩块是否可正确读取和解压缩。

语法为：

```
dump database database_name to dump_device with
compression=n,verify={crc | read_after_write}
load database database_name from dump_device with verify[only]=crc
```

其中：

- **verify=crc** - 指示正在执行循环冗余检查。
- **verify=read\_after\_write** - Backup Server 将在写入并解压缩每个压缩块后对其进行重新读取。如果 Backup Server 发现错误，将在发现该错误的文件中打印偏移。  
**verify=read\_after\_write** 仅适用于 **dump database** 命令。

本示例演示在将数据库 `new_dump` 转储至设备 `mydumpdev` 之前对其进行验证：

```
dump database new_dump to mydumpdev with
compression=x,verify=read_after_write
```

本示例演示在装载 `new_dump` 数据库转储时执行循环冗余检查：

```
load database new_dump from mydumpdev with verify[only]=crc
```

本示例演示在将数据库 `new_dump` 转储至 `mydumpdev` 设备之前将执行循环冗余检查并重新读取每个压缩块：

```
dump database new_dump to mydumpdev with
compression=x,verify=read_after_write,verify=crc
```

用法：

- 对于未使用 **verify=crc** 参数创建的转储，其所使用的格式与 Backup Server 早期版本相同。
- 如果最初并未使用 **verify=crc** 转储数据库，则 SAP ASE 将忽略 **verify=crc** 选项。
- 对于不包含循环冗余检查功能的 Backup Server 版本，不能装载包含该功能的转储。
- **verify={crc | read\_after\_write}** 检查仅适用于使用 **with compression** 参数创建的文件。
- **verify=crc** 适用于所有类型的文件，其中包括：原始设备、磁盘文件、磁带、管道或 API。但是，**verify=read\_after\_write** 需要使用“seek back”重新读取块，且仅适用于原始设备和磁盘文件。
- 如果加入不适用的 **verify={crc | read\_after\_write}** 参数，SAP ASE 将忽略该参数且不会引发错误消息。

对 **dump database** 的循环冗余检查

## 计算事务日志增长率

SAP ASE 16.0 版增加了计算指定时间段内事务日志增长率的功能。

**sp\_logging\_rate** 将显示系统过程运行期间内事务日志增长的最小、最大以及平均速率 (千兆字节/每小时)，提供的结果为计算值总和的平均值或迭代结果。

本示例显示的是 24 小时内事务日志的增长情况摘要 (计算间隔为 1 小时)：

```
sp_logging_rate 'sum', '1,00:00:00', '01:00:00'
=====
Total Summary Information
=====
Transaction Log Growth Rate      Min GB/h      Max GB/h      Avg
GB/h
-----
1.566053                        0.000000     1.970084
```

计算事务日志增长率



# 系统更改

SAP ASE 16.0 添加了对配置参数、命令、系统过程、函数、监控表以及实用程序的更改。

## 配置参数

---

SAP ASE 16.0 版引入了对配置参数的更改。

### 新配置参数

SAP ASE 16.0 纳入了新的配置参数。

*enable utility lvl 0 scan wait*

表 1. 摘要信息

|       |                 |
|-------|-----------------|
| 缺省值   | 0               |
| 值的范围  | 0 (关闭) , 1 (打开) |
| 状态    | 动态              |
| 显示级别  | Comprehensive   |
| 要求的角色 | 系统管理员           |
| 配置组   | 应用程序功能          |

用于在 Adaptive Server 运行隔离级别 0 扫描时运行 **alter table ... add | drop partition** 命令。

**注意：** **enable utility lvl 0 scan wait** 的缺省值取决于为 **enable functionality group** 设置的值。如果将 **enable functionality group** 设置为：

- 0 - **enable utility lvl 0 scan wait** 的缺省值为 0。
- 1 - **enable utility lvl 0 scan wait** 的缺省值为 1。

**注意：** 但如果将 **enable utility lvl 0 scan wait** 设置为 1，则无论 **enable functionality group** 的值设置为多少，它都使用值 1。请参见 **enable functionality group**。

(仅 UNIX) *max network peek depth*

表 2. 摘要信息

|       |                |
|-------|----------------|
| 缺省值   | 0              |
| 值的范围  | 0 到 2147483647 |
| 状态    | 动态             |
| 显示级别  | Comprehensive  |
| 要求的角色 | 系统管理员          |
| 配置组   | 网络通信           |

指定 **SAP ASE** 为达到待执行的取消所要查看的连接操作系统接收缓冲区的深度级别。例如，如果客户端在 **SAP ASE** 完成当前命令的处理之前发送了一个后接取消的新命令，则 **SAP ASE** 将在操作系统的接收缓冲区中查看至由 **max network peek depth** 指定的深度。如果取消出现在指定深度中，系统将同时放弃当前命令和后接取消的命令，**SAP ASE** 等待执行下一条命令。

*aggressive task stealing*

表 3. 摘要信息

|       |                 |
|-------|-----------------|
| 缺省值   | 1 (打开)          |
| 值的范围  | 0 (关闭) , 1 (打开) |
| 状态    | 动态              |
| 显示级别  | Comprehensive   |
| 要求的角色 | 系统管理员           |
| 配置组   | SQL Server 管理   |

启用 **aggressive task stealing** 会将 **SAP ASE** 调度程序任务挪用策略设置为积极。

*enable large chunk elc*

表 4. 摘要信息

|      |                 |
|------|-----------------|
| 缺省值  | 1 (打开)          |
| 值的范围 | 0 (关闭) , 1 (打开) |
| 状态   | 静态              |

|       |               |
|-------|---------------|
| 显示级别  | Comprehensive |
| 要求的角色 | 系统管理员         |
| 配置组   | 元数据高速缓存       |

在引擎本地高速缓存中启用较大分配。

#### *engine local cache percent*

表 5. 摘要信息

|       |               |
|-------|---------------|
| 缺省值   | 50            |
| 值的范围  | 0-100         |
| 状态    | 静态            |
| 显示级别  | Comprehensive |
| 要求的角色 | 系统管理员         |
| 配置组   | 元数据高速缓存       |

用于修改引擎本地高速缓存占过程高速缓存的百分比。

#### *large allocation auto tune*

表 6. 摘要信息

|       |                 |
|-------|-----------------|
| 缺省值   | 1 (打开)          |
| 值的范围  | 0 (关闭) , 1 (打开) |
| 状态    | 静态              |
| 显示级别  | Comprehensive   |
| 要求的角色 | 系统管理员           |
| 配置组   | 元数据高速缓存         |

配置 SAP ASE 为执行查询预分配的大量内存，以减少过程高速缓存争用。

#### *threshold event max messages*

表 7. 摘要信息

|     |   |
|-----|---|
| 缺省值 | 0 |
|-----|---|

|       |               |
|-------|---------------|
| 值的范围  | 0-2147483647  |
| 状态    | 动态            |
| 显示级别  | Comprehensive |
| 要求的角色 | 系统管理员         |
| 配置组   | 内存用户、监控       |

确定 SAP ASE 存储在 monThresholdEvent 表中的事件数。一旦 monThresholdEvent 监控表中的事件数超过此值，SAP ASE 将用新事件覆盖最早的未读事件。

### *threshold event monitoring*

**表 8. 摘要信息**

|       |                 |
|-------|-----------------|
| 缺省值   | 0 (关闭)          |
| 值的范围  | 0 (关闭) , 1 (打开) |
| 状态    | 动态              |
| 显示级别  | Comprehensive   |
| 要求的角色 | 系统管理员           |
| 配置组   | 监控              |

允许或禁止 SAP ASE 记录阈值事件。

### *user log cache queue size*

**表 9. 摘要信息**

|       |                 |
|-------|-----------------|
| 缺省值   | 1 (打开)          |
| 值的范围  | 0 (关闭) , 1 (打开) |
| 状态    | 静态              |
| 显示级别  | Comprehensive   |
| 要求的角色 | 系统管理员           |
| 配置组   | 用户环境            |

确定是否将排队策略用于日志记录。将 **user log cache queue size** 设置为：

- 1 - 允许将排队用于用户日志高速缓存。用户日志高速缓存分为多个 `cachelet`，其数量取决于 `user log cache size` 的值。
- 0 - 禁止将排队用于用户日志高速缓存。无论将 `user log cache size` 的值设置为多少，用户日志高速缓存都不会分为多个 `cachelet`。

## 更改过的配置参数

SAP ASE 16.0 版本包括配置参数的更改。

### *allow nested triggers*

多个触发器的功能不更改 `sp_configure "allow nested triggers"` 配置参数的行为。

### *stack guard size*

(仅限 UNIX 平台) 当您将 `stack guard size` (堆栈保护大小) 配置为非缺省值时，SAP 建议您额外增加 4096 个字节以扩展堆栈保护区的可用部分。

## 内置函数

SAP ASE 16.0 版引入了一个新的内置函数。

### dbencryption\_status

SAP ASE 16.0 引入了一个新的内置函数 `dbencryption_status()`，该函数能够报告数据库的加密/解密状态及进度，从而支持完全数据库加密功能。

报告数据库加密/解密状态及进度。语法为：

```
dbencryption_status ('status'|'progress', dbid[,
                    lstart])
```

其中：

- `status` - 返回您在 `dbid` 中指定的数据库的加密状态。只有提供 `dbid` 才能使用 `status`。返回的值为：
  - 0 - 表示常规数据库。
  - 1 - 表示数据库已加密。
  - 2 - 表示数据库正在加密。
  - 3 - 表示数据库已部分加密（但不是正在进行加密）。
  - 4 - 表示数据库正在解密。
  - 5 - 表示数据库已部分解密（但不是正在进行解密）。
- `progress` - 报告加密/解密进度的百分比。如果提供：
  - `dbid` - `progress` 返回已处理页面占整个数据库的百分比。
  - `dbid` 和 `lstart` (逻辑起始页) - `progress` 返回已处理页面占 `lstart` 所指示的片段的百分比。

使用“progress”时，如果 SAP ASE 未找到进度信息（如，未发生加密或解密操作，或者加密/解密进程已完成），SAP ASE 将返回“-1”。

- *dbid* - 是数据库 ID。

### 另请参见

- 创建加密数据库（第 60 页）
- `create archive database` 用于完全数据库加密（第 108 页）
- `create database` 用于完全数据库加密（第 108 页）
- 加密现有数据库（第 61 页）
- `sp_helpdb`（第 146 页）
- 挂起加密进程（第 66 页）
- 恢复加密进程（第 67 页）
- `alter database` 用于完全数据库加密（第 102 页）
- 完全数据库加密和系统更改（第 71 页）
- `sp_encryption`（第 145 页）
- `ddlgen`（第 169 页）
- `sybmigrate`（第 170 页）
- 更改过的系统表（第 160 页）
- `create encryption key`（第 111 页）
- `drop encryption key`（第 136 页）

## 命令

---

SAP ASE 16.0 版引入了对命令的更改。

### **alter database 用于完全数据库加密**

在 SAP ASE 16.0 版本中，可使用 `alter database` 命令加密和解密数据库。

#### 语法

```
alter database database_name
{[encrypt with key_name | decrypt [with key_name]] [parallel
degree_of_parallelism]
| resume [encryption | decryption [parallel degree_of_parallelism]]
| suspend [encryption | decryption]
}
```

#### 参数

- *database\_name* - 要进行加密或解密的数据库的名称。

- **encrypt with *key\_name*** - 指示 SAP ASE 完全加密数据库。

具体来说，该命令从 master 数据库的 sysencryptkeys 系统表中检索相应的密钥 ID，并在其关联的 sysdatabases 行中设置 encrkeyid 列。

*key\_name* 是用于加密数据库的数据库加密密钥。如果指定其它密钥名称，该命令将失败，同时 SAP ASE 会显示一条错误消息。

数据库处于以下状态时，SAP ASE 无法运行 **alter database** 并显示错误消息：

- 已使用另一密钥加密。
- 正在加密。

如果对当前未进行加密的部分加密数据库运行此命令，且密钥名称与先前指定的密钥相同，则 SAP ASE 会将该命令视为已指定 resume encryption 选项。

- **decrypt [with *key\_name*]** - 指示 SAP ASE 解密数据库。解密数据库时，[with *key\_name*] 是可选的，因为 SAP ASE 将在 sysdatabases 系统表中查找密钥 ID。但如果为 *key\_name* 指定的密钥名称不同于之前加密数据库时所使用的密钥名称，该命令将失败。
- **resume decryption** - 指示 SAP ASE 恢复之前被挂起的数据库解密进程。如果 *database\_name* 已完全解密，SAP ASE 将忽略此命令。
- **parallel degree\_of\_parallelism** - 确定为执行任务所要启动的工作线程数。

如果数量等于或少于“number of worker processes”配置，则将为每个数据库存储虚拟设备创建一个线程。*degree\_of\_parallelism* 值不应大于数据库设备数，因为额外的工作线程不会提高加密性能。如果不指定 *degree\_of\_parallelism*，SAP ASE 将基于联机引擎数以及数据库在不同设备中的分布方式在内部定义该参数的值。

- **resume encryption** - 从之前加密被挂起的页恢复加密进程。

如果出现以下情况，该命令失败：

- SAP ASE 已有正在运行的加密进程。
- 从未对数据库启动加密。
- 加密进程已完成。

可以将 *parallel degree\_of\_parallelism* 与 *resume encrypt* 一起使用。如果不指定 *parallel degree\_of\_parallelism*，SAP ASE 将基于数据库在不同引擎中的分布方式确定该参数的值。

- **suspend encryption** - 终止正在加密数据的所有加密工作线程。SAP ASE 将记录加密进度，以便 *resume encryption* 能够在前一个加密进程停止处重新开始加密。如果没有正在进行的加密进程，SAP ASE 将忽略此命令。

## 示例

- **示例 1** - 此示例使用加密密钥 dbkey 对现有数据库 existdb 进行加密：

```
alter database existdb encrypt with dbkey
```

此示例未指定并行度，而是由 **SAP ASE** 来决定并行加密 existdb 所应启动的工作线程数。

- **示例 2** - 此示例将暂停对数据库 existdb 执行的加密操作：

```
alter database existdb suspend encryption
```

- **示例 3** - 此示例将恢复数据库 existdb 的暂停加密：

```
alter database existdb resume encryption
```

### 用法

- 数据库加密在数据库运行期间进行。这表示数据库在进行加密时，其他用户可以对其进行访问；无需将数据库置于单用户模式。
- 加密进程不会中断对数据库执行的用户查询、更新或插入操作。
- 可暂停及恢复数据库加密，以便在重新启动 **SAP ASE** 后恢复数据库加密。
- 加密操作是非事务性的。
- 可对存档数据库和临时数据库进行加密和解密操作。
- **SAP ASE** 记录数据库的加密进程并提供实用程序报告其状态。

如果出现以下情况，该命令失败：

- 用于已加密数据库。
- 用于正在加密的数据库。如果对部分加密的数据库使用该命令，而且 **SAP ASE** 中当前未运行任何加密进程，则只要使用的数据库加密密钥名称与上次加密数据库时所用的加密密钥名称相同，该命令就会从上次暂停的位置恢复加密。

限制：

- 不能加密 master 和 model 数据库。
- 不能解密正在进行加密的数据库，也不能加密正在进行解密的数据库。
- 不能在加密数据库时删除或卸载该数据库。
- 不能基于某个正在加密的数据库加载另外一个数据库。
- 不能在加密数据库时备份（转储）该数据库。

### 另请参见

- 加密现有数据库（第 61 页）
- dbencryption\_status（第 101 页）
- sp\_helpdb（第 146 页）
- 挂起加密进程（第 66 页）
- 恢复加密进程（第 67 页）
- create database 用于完全数据库加密（第 108 页）
- create encryption key（第 111 页）
- drop encryption key（第 136 页）
- 完全数据库加密和系统更改（第 71 页）



- `create archive database` 用于完全数据库加密 (第 108 页)
- `sp_encryption` (第 145 页)
- `ddlgen` (第 169 页)
- `sybmigrate` (第 170 页)
- 更改过的系统表 (第 160 页)

## alter index

使用 `index_compression` 子句更改将来进行插入或更新的索引的压缩状态。

### 语法

```
alter index [[database.][owner].table_name.index_name
set index_compression [= {none | page} ]
```

### 参数更改

**index\_compression** - 更改本地索引分区的压缩状态仅对分区中新插入或更新的索引行产生影响。

- `none` - 不压缩指定索引的索引页。压缩专门使用 `index_compression = page` 创建的索引。
- `page` - 当该页已满时, 使用页前缀压缩来压缩现有索引行。当添加一行时, 系统将执行检查以确定该行是否适合压缩。

### 示例

将索引 `idx_char` 的压缩状态设置为打开:

```
alter index order_line.idx_char
set index_compression = page
```

### 另请参见

- 更改压缩状态 (第 53 页)
- `alter table` 用于索引压缩 (第 105 页)

## alter table 用于索引压缩

使用 `index_compression` 子句更改将来进行插入或更新的索引的压缩状态。

### 语法

```
alter table [[database.][owner].table_name
{add column_name datatype}
[default {constant_expression | user | null}]
{identity | null | not null}
[off row | in row]
[[constraint constraint_name]
{{unique | primary key}
[clustered | nonclustered] [asc | desc]
[with {fillfactor = pct,
max_rows_per_page = num_rows,
```

```

        reservepagegap = num_pages}}
    [on segment_name]
| references [[database.]owner.]ref_table
    [(ref_column)] [match full]
| check (search_condition)
[encrypt [with key_name] [decrypt_default value]],
[[not] compressed]
[, next_column]...
| add {[constraint constraint_name]
    {unique | primary key}
    {clustered | nonclustered}
    (column_name [asc | desc]
    [, column_name [asc | desc]...])
    [with {fillfactor = pct,
        max_rows_per_page = num_rows,
        reservepagegap = num_pages}}]
    [on segment_name]
| foreign key (column_name [{, column_name}...])
    references [[database.]owner.]ref_table
    [(ref_column [{, ref_column}...])]
    [match full]
| check (search_condition)}
| set { [ dml_logging = {full | minimal | default}]
    | [,compression = {NONE | PAGE | ROW | ALL}]
    | [,index_compression = {NONE | PAGE}]
    }
| drop {column_name [, column_name]...
| constraint constraint_name}
| modify column_name
    [datatype [null | not null]]
    [[[encrypt [with key_name]
        [decrypt_default value]]
    | decrypt
    ]
    [[not] compressed ]
[, next_column]...
| replace column_name
    default {constant_expression | user | null}
| decrypt_default {constant_expression | null}
| drop decrypt_default
| lock {allpages | datarows | datapages}
| with exp_row_size=num_bytes
| partition number_of_partitions
| unpartition
| partition_clause
| alter_partition_clause

    alter_partition_clause ::=
{add_partition_clause
| drop_partition_clause
| modify partition partition_name
    [, partition_name ...]
    set compression [= {none | row | page | ALL}]
    set index_compression [= {none | page}] }

```

### 参数更改

**index\_compression** - 指定对表、索引或本地索引分区启用还是禁用索引压缩。如果将表修改为按索引进行压缩，则将压缩新建索引。

- NONE - 不压缩指定表的索引。
- PAGE - 压缩指定表的所有索引。

### 示例

#### 示例 1

更改现有表 `order_line`，将压缩状态更改为 NONE：

```
alter table order_line set index_compress = NONE
```

#### 示例 2

更改现有表 `sales`，将本地索引分区 `Y2009` 的压缩状态更改为 PAGE。

```
alter table sales
modify partition Y2009 set index_compression = PAGE
```

### 另请参见

- 更改压缩状态（第 53 页）
- `alter index`（第 105 页）

## alter table 用于多个触发器

如果某个表具有多个触发器，表所有者可禁用为该表定义的部分或全部触发器。

## alter table 用于残留数据删除

**alter table** 命令支持删除在 SAP ASE 中进行删除后残留的数据。

### 语法

在现有表中指定此命令的语法为：

```
alter table table_name
    set "erase residual data" {on | off}
```

### 用法

针对某表设置此选项时，对该表执行产生残留数据的操作（**drop table**、**delete row**、**alter table** 和 **drop index**）会自动清除已释放空间。缺省设置为 **off**。

### 权限

仅表所有者或具有 **alter any table** 权限的用户可以使用“`erase residual data`”选项。

## **create archive database** 用于完全数据库加密

**create archive database** 命令支持完全数据库加密功能。

### 语法

```
create archive database database_name  
    encrypt with key_name
```

### 参数

- *database\_name* 是要创建的存档数据库的名称。
- *key\_name* 是加密存档数据库所用的密钥。SAP ASE 在数据库转储过程中检验 *key\_name* 是否匹配。如果不匹配，恢复将失败。

### 权限

使用 **create archive database** 命令的 `encrypt with` 选项并不需要特殊权限。但是，用户需要具备对数据库加密密钥的 **select** 权限以能够引用其作为 *key\_name*。

### 另请参见

- 创建加密数据库 (第 60 页)
- `dbencryption_status` (第 101 页)
- `create database` 用于完全数据库加密 (第 108 页)
- 完全数据库加密和系统更改 (第 71 页)
- `sp_helpdb` (第 146 页)
- `sp_encryption` (第 145 页)
- `ddlgen` (第 169 页)
- `sybmigrate` (第 170 页)
- 更改过的系统表 (第 160 页)
- `alter database` 用于完全数据库加密 (第 102 页)
- `create encryption key` (第 111 页)
- `drop encryption key` (第 136 页)

## **create database** 用于完全数据库加密

可使用 **create database** 命令创建完全加密的数据库。

指定在创建数据库时是否对该数据库加密，如果加密该数据库，则插入到该数据库中的所有数据也会自动加密。数据库在加密后，其大小不会发生变化；另外，无论数据库是否加密，其所有存储访问功能的工作方式保持不变。支持加密的数据库有以下几种类型：

- 常规用户数据库

- 临时数据库
- 存档数据库

不能加密内存数据库。

### 语法

```
create [temporary] database database_name
    encrypt with key_name
```

### 参数

- ***database\_name*** - 要创建的加密数据库的名称。
- ***key\_name*** - 数据库加密密钥的名称。

### 示例

- **从头创建加密数据库** - 使用加密密钥 `dbkey` 在计算机 `demologdev` 上创建一个名为 `demodb` 的加密数据库，日志为 `demodev`：

```
create database demodb on demodev log on demologdev encrypt with
dbkey
```

### 另请参见

- 创建加密数据库 (第 60 页)
- `create archive database` 用于完全数据库加密 (第 108 页)
- `dbencryption_status` (第 101 页)
- `alter database` 用于完全数据库加密 (第 102 页)
- `create encryption key` (第 111 页)
- `drop encryption key` (第 136 页)
- 完全数据库加密和系统更改 (第 71 页)
- `sp_helpdb` (第 146 页)
- `sp_encryption` (第 145 页)
- `ddlgen` (第 169 页)
- `sybmigrate` (第 170 页)
- 更改过的系统表 (第 160 页)

## create default

在 `create default` 中使用 `or replace` 子句可替换缺省值定义。

### 语法

更改之处以粗体显示。

```
create [or replace] [owner.] default_name
as constant_expression
```

**create or replace default** 的参数更改

- **create** - 如果缺省值尚不存在，将创建一个缺省值。
- **or replace** - 如果缺省值已存在，则使用新定义替换缺省值定义。
- **default\_name** - 如果指定的缺省名称已存在，则将其替换为新的缺省值定义。对象名称和 ID 保持不变。
- **constant\_expression** - 替换缺省值时，可更改缺省值的定义。新的缺省值会覆盖旧的缺省值。

## 示例

以下示例将创建一个缺省值，其中电话号码定义为 UNKNOWN：

```
create default phonedflt as "UNKNOWN"

select object_id("phonedflt")
-----
1001051571
```

通过使用 **or replace** 子句，以下缺省值将替换之前创建的缺省值。电话号码已更改，但缺省值的对象 ID 保持不变：

```
create or replace default phonedflt as "999-999-9999"

select object_id("phonedflt")
-----
1001051571
```

## 与替换后的缺省值相关的对象

- 许多列都可能会绑定到一个替换后的缺省值。
- 用户定义的数据类型可能会绑定到替换后的缺省值。

在替换缺省值后执行过程时，将重新编译访问这些列的过程。

**create or replace default** 的权限更改

通过别名或 **setuser** 充当缺省值所有者的任何用户都不能替换此缺省值。

替换缺省值发生的更改以粗体显示。

|        |                                                                                                                                  |
|--------|----------------------------------------------------------------------------------------------------------------------------------|
| 启用细化权限 | 在启用细化权限的情况下，您必须具有 <code>create default</code> 特权。要为另一用户创建缺省值，您必须具有 <code>create any default</code> 特权。 <b>您必须是缺省值所有者才能替换缺省值。</b> |
| 禁用细化权限 | 在禁用细化权限的情况下，您必须是数据库所有者、具有 <code>sa_role</code> 的用户或是具有 <code>create default</code> 特权。 <b>您必须是缺省值所有者才能替换缺省值。</b>                 |

**create or replace default** 的审计更改

更改之处以粗体显示。

| 事件 | 审计选项   | 审计的命令或访问权限     | extrainfo 中的信息                                                                                                                                                                                                                                               |
|----|--------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14 | create | create default | <ul style="list-style-type: none"> <li>• 角色 - 当前活动角色</li> <li>• 关键字或选项 - NULL</li> <li>• 以前值 - NULL</li> <li>• 其它信息 - NULL</li> <li>• 当前值 - NULL</li> <li>• 代理信息 - set proxy 有效时的初始登录名</li> <li>• <b>or replace</b> - 用于 <b>create or replace</b></li> </ul> |

## create encryption key

**create encryption key** 命令支持完全数据库加密功能。

数据库加密密钥是 Master 数据库中创建的用于加密数据库的 256 位对称密钥。

### 语法

```
create encryption key keyname
    [for algorithm]
    for database encryption
    [with
        { [master key]
          [key_length 256]
          [init_vector random]
          [[no] dual_control]}]
```

### 参数

- **keyname** - 在当前数据库的用户表、视图和过程命名空间中必须唯一。如果该密钥位于另一数据库中，请指定数据库名，如果数据库中有多个具有该名称的密钥，请指定所有者的名称。owner 的缺省值是当前用户，而 database 的缺省值是当前数据库。只有系统安全员才能为其他用户创建密钥。
- **algorithm** - 指定算法。仅支持高级加密标准 (AES) 一种算法。AES 支持的密钥大小为 128、192 和 256 位，支持的块大小为 16 个字节。在加密单元中，块大小即为字节数。大型数据将拆分以进行加密。
- **for database encryption** - 指示您正在创建加密整个数据库而非加密列的加密密钥。
- **master key** - 在 master 数据库中创建主密钥并指示 SAP ASE 使用此密钥保护数据库加密密钥。缺省情况下，SAP ASE 使用此主密钥（如果存在）保护数据库加密密钥。
- **key\_length 256** - 是要创建的密钥的大小（以位为单位）。数据库加密密钥的唯一有效长度是 256 位；如果使用其它大小，将显示错误消息。
- **init\_vector random** - 指定在加密过程中使用初始化矢量。如果加密算法使用初始化矢量，则两块相同明文的密文是不同的，这样便可防止检测数据模式。通过使用初始化矢量，可以提高数据的安全性。与列加密相比，数据库加密进一

步增强了安全性；如果像创建列加密密钥一样也指定 `init_vector null`，SAP ASE 将返回错误。

- **[no] dual control** - 指示是否必须使用双重控制加密新密钥。缺省情况下不配置双重控制。要使用 `dual control`，主密钥和双主密钥均必须均位于 `master` 数据库中。

## 示例

- **示例 1** - 创建主密钥以保护 “testkey”：

```
create encryption key testkey for database encryption
with master key
```

- **示例 2** - 创建双主密钥以保护 “testkey”：

```
create encryption key testkey for database encryption
with dual_control
```

- **示例 3** - 创建主密钥和双主密钥以保护 “testkey”：

```
create encryption key testkey for database encryption
with master key dual_control
```

- **示例 4** - 创建主密钥以保护 “testkey”，同时显式排除双主密钥：

```
create encryption key testkey for database encryption
with master key no dual_control
```

- **示例 5** - 创建主密钥以保护 “testkey”，同时显式排除双主密钥：

```
create encryption key testkey for database encryption
with no dual control
```

- **示例 6** -

```
sp_configure 'enable encrypted columns', 1
create encryption key master with passwd "testpassword"
set encryption passwd 'testpassword' for key master
create encryption key dbkey for database encryption
```

## 用法

- 数据库加密密钥不支持 **create encryption key** 命令中的 `pad` 选项。
- 数据库加密密钥不能为缺省的列加密密钥。
- 成功创建的数据库加密密钥存储于 `master` 数据库的 `sysencryptkeys` 表中，并用下面的密钥类型表示：

```
#define EK_DBENCKEY 0x1000
```

## 标准

符合 ANSI SQL 的级别：Transact-SQL 扩展。



## 权限

对 **create encryption key** 的权限检查因您的细分权限设置而异：

|        |                                                                                                           |
|--------|-----------------------------------------------------------------------------------------------------------|
| 启用细化权限 | SAP ASE 创建一个名为 “manage database encryption key” 的新权限。必须具有相应的权限才能创建数据库加密密钥。                                |
| 禁用细化权限 | 您必须是具有 <code>sso_role</code> 或 <code>keycustodian_role</code> 角色的用户或必须具有 <b>create encryption key</b> 特权。 |

## 另请参见

- `drop encryption key` (第 136 页)
- 更改数据库加密密钥 (第 59 页)
- 删除数据库加密密钥 (第 59 页)
- 备份数据库加密密钥 (第 68 页)
- 创建数据库加密密钥 (第 57 页)
- `alter database` 用于完全数据库加密 (第 102 页)
- `create database` 用于完全数据库加密 (第 108 页)
- 完全数据库加密和系统更改 (第 71 页)
- `create archive database` 用于完全数据库加密 (第 108 页)
- `dbencryption_status` (第 101 页)
- `sp_helpdb` (第 146 页)
- `sp_encryption` (第 145 页)
- `ddlgen` (第 169 页)
- `sybmigrate` (第 170 页)
- 更改过的系统表 (第 160 页)

## create function

在 **create function** 中使用 **or replace** 子句可替换用户定义函数的定义。

### 语法

更改之处以粗体显示。

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

**create or replace function** 的参数更改

- **create** - 如果函数尚不存在，将创建一个函数。
- **or replace** - 重新定义现有函数。使用此子句可更改现有用户定义函数的定义，而不必对之前已授予该函数的对象特权进行删除、重新创建以及重新授权。如果重新定义了函数，则在使用该函数时将重新编译该函数。
- **function\_name** - 尽管函数的定义已更改，但函数的名称将保持不变。
- **@parameter\_name** - 替换函数定义时，可更改参数的名称和数量。
- **parameter\_datatype** - 可更改函数参数的数据类型。
- **with recompile** - 可将选项更改为每次替换函数时都重新编译，或不重新编译。
- **return\_datatype** - 可更改函数的返回数据类型。
- **scalar\_expression** - 可更改函数返回的值。
- **function\_body** - 可更改用于定义函数值的 SQL 语句。

## 示例

以下示例定义一个用于并置 `firstname` 和 `lastname` 字符串的函数。

```
create function fullname(
    @firstname char(30),
    @lastname char(30))
returns char(61)
as
begin
declare @name char(61)
set @name = @firstname|| ' ' ||@lastname
return @name
end

select object_id("fullname")
-----
473049690
```

通过使用 **or replace** 子句，以下函数将替换之前创建的 `fullname` 函数。替换该函数后，删除了局部变量 `@name`。函数的对象 ID 保持不变。

```
create or replace function fullname(
    @firstname char(30),
    @lastname char(30))
returns char(61)
as
begin
return(@firstname|| ' ' ||@lastname)
end

select object_id("fullname")
-----
473049690
```

### 与替换后的函数相关的对象

如果另一函数调用替换后的函数，则在调用时将重新编译这两个函数。如果替换后函数的接口与调用函数中的接口不匹配，则必须替换该调用函数，否则该调用函数将引发错误。可对替换后的函数执行 **sp\_depends** 以检查是否存在任何调用对象。

例如，`testfun1` 被替换为具有两个参数而不是一个。因此，必须替换调用函数 `testfun2`，以便为第二个参数的出现做出解释。

```
create function testfun1 (@para1 int)
    returns int
    as
    begin
        declare @retval int
        set @retval = @para1
        return @retval
    end
create function testfun2 (@para int)
    returns int
    as
    begin
        declare @retval int
        select @retval= dbo.testfun1 (@para)
        return @retval
    end
create or replace function testfun1 (@para1 int,@para2 int)
    returns int
    as
    begin
        declare @retval int
        set @retval = @para1+@para2
        return @retval
    end
```

### 限制

如果在计算列或函数索引中引用了某一函数，将不能替换该函数。

### **create or replace function** 的权限更改

通过别名或 **setuser** 充当函数所有者的任何用户都不能替换此函数。

替换函数发生的更改以粗体显示。

|        |                                                                                                                                               |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 启用细化权限 | 在启用细化权限的情况下，您必须具有 <code>create function</code> 特权。您必须具有 <code>create any function</code> 特权才能为其他用户运行 <b>create function</b> 。您必须是函数所有者才能替换函数。 |
| 禁用细化权限 | 在禁用细化权限的情况下，您必须是数据库所有者或具有 <code>create function</code> 特权。您必须是函数所有者才能替换函数。                                                                    |

### **create or replace function** 的审计更改

更改之处以粗体显示。

| 事件 | 审计选项   | 审计的命令或访问权限      | extrainfo 中的信息                                                                                                                                                                                                                                               |
|----|--------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 97 | create | create function | <ul style="list-style-type: none"> <li>• 角色 - 当前活动角色</li> <li>• 关键字或选项 - NULL</li> <li>• 以前值 - NULL</li> <li>• 其它信息 - NULL</li> <li>• 当前值 - NULL</li> <li>• 代理信息 - set proxy 有效时的初始登录名</li> <li>• <b>or replace</b> - 用于 <b>create or replace</b></li> </ul> |

## **create function (SQLJ)**

在 **create function** 中使用 **or replace** 子句可替换用户定义的 SQLJ 函数的定义。

### 语法

更改之处以粗体显示。

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

### *create or replace function (SQLJ) 的参数更改*

- **create** - 如果 SQLJ 函数尚不存在，将创建一个 SQLJ 函数。
- **or replace** - 重新定义现有函数。使用此子句可更改现有用户定义 SQLJ 函数的定义，而不必删除、重新创建及重新授予先前授予该函数的对象特权。
- *sql\_function\_name* - 尽管函数的定义已更改，但其名称保持不变。
- *sql\_parameter\_name* - 替换函数定义时，可更改参数的名称和数量。
- *sql\_datatype* - 可更改函数参数的 Transact-SQL 数据类型。
- *returns sql\_datatype* - 可更改函数的结果数据类型。
- **external** - 可更改外部例程的名称。
- *java\_method\_name* - 可更改 Java 方法名称。

- `java_datatype` - 可更改 Java 数据类型。

### 示例

以下示例将创建一个名为 `sqlj_testfun` 的 SQLJ 函数。

```
create function sqlj_testfun (p1 int)
    returns int
    language java
    parameter style java
    external name 'UDFSample.sample(int)'
```

以下示例使用 **or replace** 子句替换之前创建的 SQLJ 函数。添加了参数 `p2` 并更改了外部 java 方法，但 SQLJ 函数的对象 ID 保持不变。

```
create or replace function sqlj_testfun (p1 int,p2 int)
    returns int
    language java
    parameter style java
    external name 'UDFSample.sample2(int,int)'
```

### 与替换后的函数相关的对象

如果另一函数调用替换后的 SQLJ 函数，则在调用时将重新编译这两个函数。

如果替换后函数的接口与调用函数中的接口不匹配，则必须替换该调用函数，否则该调用函数将引发错误。可对替换后的函数执行 **sp\_depends** 以检查是否存在任何调用对象。

### 限制

如果在计算列或函数索引中引用了某一函数，将不能替换该函数。

### **create or replace function (SQLJ) 的权限更改**

通过别名或 **setuser** 充当函数所有者的任何用户都不能替换此函数。

替换函数发生的更改以粗体显示。

|        |                                                                                                                                               |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 启用细化权限 | 在启用细化权限的情况下，您必须具有 <code>create function</code> 特权。您必须具有 <code>create any function</code> 特权才能为其他用户运行 <b>create function</b> 。您必须是函数所有者才能替换函数。 |
| 禁用细化权限 | 在禁用细化权限的情况下，您必须是数据库所有者或具有 <code>create function</code> 特权。您必须是函数所有者才能替换函数。                                                                    |

### **create or replace function (SQLJ) 的审计更改**

更改之处以粗体显示。

| 事件 | 审计选项   | 审计的命令或访问权限      | extrainfo 中的信息                                                                                                                                                                                                                                 |
|----|--------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 97 | create | create function | <ul style="list-style-type: none"> <li>角色 - 当前活动角色</li> <li>关键字或选项 - NULL</li> <li>以前值 - NULL</li> <li>其它信息 - NULL</li> <li>当前值 - NULL</li> <li>代理信息 - set proxy 有效时的初始登录名</li> <li><b>or replace</b> - 用于 <b>create or replace</b></li> </ul> |

## create index

可使用 **index\_compression** 子句压缩索引或索引分区。

### 语法

```
create [unique] [clustered | nonclustered] index index_name
on [[database.]owner.]table_name
(column_expression [asc | desc]
[, column_expression [asc | desc]]...)
[with {fillfactor = pct,
index_compression = { NONE | PAGE },
max_rows_per_page = num_rows,
reservepagegap = num_pages,
consumers = x, ignore_dup_key, sorted_data,
[ignore_dup_row | allow_dup_row],
statistics using num_steps values}]
[on segment_name]
[index_partition_clause]
Syntax to create index partitions
index_partition_clause::=
[local index
[partition_name [on segment_name]
with index_compression = { NONE | PAGE }]
[, partition_name [on segment_name]
with index_compression = { NONE | PAGE }]]...]]]
```

### 参数更改

#### **index\_compression**

- NONE - 不压缩指定索引的索引页。压缩专门使用 **index\_compression = PAGE** 创建的索引。
- PAGE - 当该页已满时，使用页前缀压缩来压缩现有索引行。当添加一行时，系统将执行检查以确定该行是否适合压缩。

### 示例

#### 示例 1

对列 `ol_delivery_d` 和 `ol_dist_info` 创建一个名为 `idx_order_line` 的压缩索引:

```
create index idx_order_line
  on order_line (ol_delivery_d, ol_dist_info)
with index_compression = page
```

如果该索引因其某个索引行长度过短从而不必执行压缩, 则会引发一条警告, 指示将不会压缩该索引。

## 示例 2

创建一个名为 `idx_sales` 的压缩索引。该索引包含可压缩的本地索引分区。索引前缀压缩应用到本地索引分区。当索引页已满时将应用页前缀压缩:

```
create index idx_sales
  on Sales(store_id, order_num)
  local index ip1 with index_compression = PAGE,
  ip2 with index_compression = PAGE,
  ip3
```

## 另请参见

- 创建压缩索引 (第 53 页)

## create procedure

在 `create procedure` 中使用 `or replace` 子句可替换过程定义。

保留之前授予替换后过程的特权。

### 语法

更改之处以粗体显示。

```
create [or replace] procedure

  [owner.]procedure_name[;number]
  [[(@parameter_name datatype [(length)
  (precision[,scale]])
  [= default][output]...)] ]
[with {recompile | execute as {owner | caller}} ]
as {SQL_statements | external name dll_name}
```

### **create or replace procedure** 的参数更改

- **create** - 如果仅指定 **create**, 则创建一个新过程。
- **or replace** - 如果指定过程不存在, 则创建一个新过程。如果该过程确实存在, 则更改过程定义; 保留现有权限、审计选项和复制属性。
- *procedure\_name* - 如果即将替换过程, 过程名称将保持不变, 并且该过程在所有相应目录中的对象标识符将保持不变。
- *number* - 如果未指定 **or replace** 但存在该过程, SAP ASE 将引发一条错误, 指出已使用该组编号创建该过程, 您必须使用其它组编号来创建该过程。如果指定了 **or replace** 子句而未指定组编号, 但存在的该过程具有其它组编号, 则会引发一条错

误，因为该过程是某个组的一部分，不能执行替换。您不能通过指定组编号来替换具有该组编号的现有过程。这与删除过程相类似，即您不能删除某个组中的个别过程。

- *parameter\_name* - 在替换过程定义时，可更改参数的名称和数量。
- *datatype [(length) (precision[,scale])]* - 可更改参数的类型、长度、精度和标度。
- *default* - 对于参数，可将缺省值更改为 NULL，或者在替换过程时设置一个不同的值。
- **output** - 可更改过程的返回参数。
- **with recompile** - 如果使用此选项创建了一个现有过程，则可使用 **or replace** 子句更改该过程，这样，SAP ASE 就不会在每次执行该过程时都创建一个新的计划。如果尚未使用 **with recompile** 创建现有过程，则将其替换为新定义，以便每次执行该过程时都创建计划。
- **with execute as** - 可将现有过程的 **with execute as** 子句从所有者更改为调用者，也可从调用者更改为所有者。还可使用 **or replace** 子句重新创建一个没有 **with execute as** 子句的过程。
- *SQL\_statements* - 可更改过程的主体，使其包含现有过程之外的那些语句。
- **external name** - 还可替换扩展存储过程。
- *dll\_name* - 可更改用于实现扩展存储过程的动态链接库的名称。

### 示例

以下示例基于一个产品信息表，其定义为：

```
create table Products (
    ProductID int,
    ProductName varchar(30),
    Discontinued varchar(10))

create procedure ProductType
    @product_ID int,
    @type char(10) output
as
declare @prod_name char(20)
select @prod_name = ProductName, @type =
    case @prod_name
        when 'Tee Shirt' then 'Shirt'
        when 'Sweatshirt' then 'Shirt'
        when 'Baseball Cap' then 'Hat'
        when 'Visor' then 'Hat'
        when 'Shorts' then 'Shorts'
        else 'UNKNOWN'
    end
from Products
where ProductID = @product_ID

select object_id("ProductType")
-----
425049519
```



接下来的命令使用 **or replace** 子句替换 ProductType 过程。Tee Shirt 和 Sweatshirt 的参数已更新，但过程的对象 ID 保持不变。

```
create or replace procedure ProductType
    @product_ID int,
    @type char(10) output
as
declare @prod_name char(20)
select @prod_name = ProductName, @type =
    case @prod_name
        when 'Tee Shirt' then ' T Shirt'
        when 'Sweatshirt' then 'Long Sleeve Shirt'
        when 'Baseball Cap' then 'Hat'
        when 'Visor' then 'Hat'
        when 'Shorts' then 'Shorts'
        else 'UNKNOWN'
    end
from Products
where ProductID = @product_ID

select object_id("ProductType")
-----
425049519
```

#### 与替换后的过程相关的对象

调用替换后的过程的过程将在执行时重新编译。如果替换过程使参数的数量或类型发生了更改，则必须替换调用过程。可对替换后的过程运行 **sp\_depends**，以验证是否存在受定义更改影响的调用过程。

#### **create or replace procedure** 的权限更改

通过别名或 **setuser** 充当过程所有者的任何用户都不能替换此过程。

替换过程发生的更改以粗体显示。

|        |                                                                                                                                                 |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 启用细化权限 | 在启用细化权限的情况下，您必须具有 <b>create procedure</b> 特权。您必须具有 <b>create any procedure</b> 特权才能为其他用户运行 <b>create procedure</b> 。<br><b>您必须是过程所有者才能替换过程。</b> |
| 禁用细化权限 | 在禁用细化权限的情况下，您必须是数据库所有者或具有 <b>create procedure</b> 特权。 <b>您必须是过程所有者才能替换过程。</b>                                                                   |

#### **create or replace procedure** 的审计更改

更改之处以粗体显示。

| 事件 | 审计选项   | 审计的命令或访问权限       | extrainfo 中的信息                                                                                                                                                                                                                                 |
|----|--------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11 | create | create procedure | <ul style="list-style-type: none"> <li>角色 - 当前活动角色</li> <li>关键字或选项 - NULL</li> <li>以前值 - NULL</li> <li>其它信息 - NULL</li> <li>当前值 - NULL</li> <li>代理信息 - set proxy 有效时的初始登录名</li> <li><b>or replace</b> - 用于 <b>create or replace</b></li> </ul> |

## **create procedure (SQLJ)**

在 **create procedure** 中使用 **or replace** 子句可替换 SQLJ 过程定义。

保留之前授予替换后过程的特权。

### 语法

更改之处以粗体显示。

```
create [or replace] procedure

    [owner_name.]sql_procedure_name
        ([in | out | inout] sql_parameter_name
        sql_datatype
        [(length) | (precision[, scale ])]
        [=default]
        ...))
    [, [in | out | inout] sql_parameter_name
    sql_datatype
    [(length) | (precision[, scale ])]
    [=default]
    ...))
    [modifies sql data]
    [dynamic result sets integer]
    [deterministic | not deterministic]
    language java
    parameter style java
    external name 'java_method_name
    [(java_datatype[, java_datatype
    ...]]]'
```

### **create or replace procedure (SQLJ) 的参数更改**

- **create** - 如果仅指定 **create**，则会创建新的 SQLJ 过程。
- **or replace** - 如果指定的 SQLJ 过程不存在，则会创建新的 SQLJ 过程。如果 SQLJ 过程不存在，则定义会发生更改。
- *sql\_procedure\_name* - 可更改名称和参数数量。
- **in | out | inout** - 可更改列出参数的模式。

- `sql_datatype [(length) (precision[,scale])]` - 可更改参数的类型、长度、精度和标度。
- `Default` - 可将参数的缺省值更改为 `NULL`，或者在过程发生更改时设置不同的值。
- **deterministic | not deterministic** - 可更改确定性值。
- **external** - 可更改外部例程的名称。
- `java_method_name` - 可更改 Java 方法名称。
- `java_datatype` - 可更改 Java 数据类型。

### 示例

此示例创建名为 `proc_name` 的 SQLJ 过程。

```
create procedure sqlj_proc (param int)
    language java
    parameter style java
    external name 'UDFSample.sample(int)'
```

此过程使用 **or replace** 子句替换先前创建的 SQLJ 过程。添加了参数 `p2` 并更改了外部 java 方法，但 SQLJ 过程的对象 ID 保持不变。

```
create or replace procedure sqlj_proc (p1 int, p2 int)
    language java
    parameter style java
    external name 'UDFSample.add(int,int)'
```

### 与替换后的过程相关的对象

调用替换后的 SQLJ 过程的 Transact-SQL 过程将在执行时重新编译。如果替换 SQLJ 过程使参数的数量或类型发生了更改，则必须替换调用过程。可对替换后的过程运行 **sp\_depends**，以验证是否存在受定义更改影响的调用过程。

### **create or replace procedure (SQLJ) 的权限更改**

通过别名或 **setuser** 充当过程所有者的任何用户都不能替换此过程。

替换 SQLJ 过程的更改以粗体显示。

|        |                                                                                                                                                             |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 启用细化权限 | 在启用细化权限的情况下，您必须具有 <code>create procedure</code> 特权。您必须具有 <code>create any procedure</code> 特权才能为其他用户运行 <b>create procedure</b> 。<br><b>您必须是过程所有者才能替换过程。</b> |
| 禁用细化权限 | 在禁用细化权限的情况下，您必须是数据库所有者或具有 <code>create procedure</code> 特权。 <b>您必须是过程所有者才能替换过程。</b>                                                                         |

### **create or replace procedure (SQLJ) 的审计更改**

更改之处以粗体显示。

| 事件 | 审计选项   | 审计的命令或访问权限       | extrainfo 中的信息                                                                                                                                                                                                                                 |
|----|--------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11 | create | create procedure | <ul style="list-style-type: none"> <li>角色 - 当前活动角色</li> <li>关键字或选项 - NULL</li> <li>以前值 - NULL</li> <li>其它信息 - NULL</li> <li>当前值 - NULL</li> <li>代理信息 - set proxy 有效时的初始登录名</li> <li><b>or replace</b> - 用于 <b>create or replace</b></li> </ul> |

## create rule

在 **create rule** 中使用 **or replace** 子句可替换规则定义。

### 语法

更改之处以粗体显示。

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

### **create or replace rule** 的参数更改

- **create** - 如果规则尚不存在，将创建一个规则。
- **or replace** - 如果规则已存在，则用新定义替换该规则定义。
- **access** - 可将访问规则从“and”规则更改为“or”规则，反之亦然。不能以同名域规则替换访问规则，反之亦然。
- **rule\_name** - 如果指定名称的规则已存在，则用新规则定义予以替换，但保留该名称。
- **constant\_expression** - 替换规则时，可更改该规则的定义。新规则值将覆盖旧规则值。

### 示例

#### 示例 1

以下示例创建一个名为 `limit` 的规则，该规则将 `advance` 的值限定为 \$1000:

```
create rule limit
  as @advance < $1000

select object_id("limit")
-----
1017051628
```

接下来的这条命令将替换所创建的规则。使用 **or replace** 子句更改限制值。规则的对象 ID 保持不变。

```
create or replace rule limit
as @advance < $2000

select object_id("limit")
-----
1017051628
```

## 示例 2

表所有者创建一个名为 `uname_acc_rule` 的 **AND** 访问规则：

```
create access rule uname_acc_rule
as @username = suser_name()

select object_id("uname_acc_rule")
-----
1033051685
```

用 **OR** 访问规则替换 `uname_acc_rule`：

```
create or replace or access rule uname_acc_rule
as @username = suser_name()

select object_id("uname_acc_rule")
-----
1033051685
```

与替换后的规则相关的对象

- 许多表中的列都可能会绑定到替换后的规则。
- 用户定义的数据类型可能会绑定到替换后的规则。

在替换规则后执行过程时，将重新编译访问这些列的过程。

### **create or replace rule** 的权限更改

通过别名或 **setuser** 充当规则所有者的任何用户都不能替换此规则。

替换规则发生的更改以粗体显示。

|        |                                                                                                                                                                   |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 启用细化权限 | 在启用细化权限的情况下，您必须具有 <code>create rule</code> 特权。您必须具有 <code>create any rule</code> 特权才能为其他用户使用 <b>create rule</b> 。您必须是规则所有者才能替换规则。                                 |
| 禁用细化权限 | 在禁用细化权限的情况下，您必须具有 <code>create rule</code> 特权、成为数据库所有者或是具有 <code>sa_role</code> 的用户。您必须是具有 <code>sa_role</code> 的用户才能为其他用户使用 <b>create rule</b> 。您必须是规则所有者才能替换规则。 |

### **create or replace rule** 的审计更改

更改之处以粗体显示。

| 事件 | 审计选项   | 审计的命令或访问权限  | extrainfo 中的信息                                                                                                                                                                                                                                               |
|----|--------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13 | create | create rule | <ul style="list-style-type: none"> <li>• 角色 - 当前活动角色</li> <li>• 关键字或选项 - NULL</li> <li>• 以前值 - NULL</li> <li>• 其它信息 - NULL</li> <li>• 当前值 - NULL</li> <li>• 代理信息 - set proxy 有效时的初始登录名</li> <li>• <b>or replace</b> - 用于 <b>create or replace</b></li> </ul> |

### create table 用于索引压缩

可使用 **index\_compression** 子句压缩指定表的索引。

#### 语法

```

create table [database.[owner].]table_name
(column_name datatype
[default {constant_expression | user | null}]
[{{identity | null | not null}}]
[off row | [in row [(size_in_bytes)]]]
[[constraint constraint_name]
{{unique | primary key}
[clustered | nonclustered] [asc | desc]
[with {fillfactor = pct,
max_rows_per_page = num_rows,}
reservepagegap = num_pages}]
[on segment_name
| references [[database.]owner.]ref_table
[(ref_column )]
[match full]
| check (search_condition)}}]
[match full]...
[encrypt [with key_name]
[decrypt default constant_expression | null]]
[[constraint [[database.[owner].]key_name]
{unique | primary key}
[clustered | nonclustered]
(column_name [asc | desc]
[[, column_name [asc | desc]]...])
[with {fillfactor = pct
max_rows_per_page = num_rows,
reservepagegap = num_pages}]
[on segment_name]
| foreign key (Column_name [{,column_name}...])
references [[database.]owner.]ref_table
[(ref_column [{, ref_column}...])]
[match full]
| check (search_condition) ...]
[{{, {next_column | next_constraint}}...]
[lock {datarows | datapages | allpages}]

```

```
[with {max_rows_per_page = num_rows,
      exp_row_size = num_bytes,
      reservepagegap = num_pages,
      identity_gap = value,
      transfer table [on | off],
      compression [= {NONE | ROW | PAGE}],
      index_compression [= {NONE | PAGE} ]
      }
]
[on segment_name]
  [[ external table ] at pathname ]
[partition_clause]
```

### 参数更改

#### **index\_compression**

- **NONE** - 不压缩指定表的索引。压缩专门使用 **index\_compression = PAGE** 创建的索引。
- **PAGE** - 压缩指定表的所有索引。不压缩专门使用 **index\_compression = NONE** 创建的索引。

### 示例

以下示例将创建按索引压缩的表 `order_line`，其中压缩了列 `ol_delivery_d` 和 `ol_dist_info`，且使用的压缩方法为页级压缩：

```
create table order_line (
  ol_o_id      int,
  ol_d_id      tinyint,
  ol_w_id      smallint,
  ol_number    tinyint,
  ol_i_id      int,
  ol_supply_w_id smallint,
  ol_delivery_d datetime,
  ol_quantity  smallint,
  ol_amount    float,
  ol_dist_info char(24) )
lock datapages
with index_compression = page
```

缺省情况下，将压缩在此表中创建的索引。不过，如果某个索引因其某个索引行长度过短从而不必执行压缩，则会引发一条警告，指示将不会压缩该索引。

### 另请参见

- 创建按索引压缩的表（第 52 页）
- `select into`（第 139 页）

## **create table** 用于残留数据删除

**create table** 命令支持删除在 SAP ASE 中进行删除后残留的数据。

### 语法

在新表中指定此命令的语法为：

```
create table table_name  
    with "erase residual data" {on | off}
```

### 示例

下列示例使用以下两个表：

- create table t1 (col1 int) with "erase residual data" on
- create table t2 (col1 int) with "erase residual data" off

### 示例 1

由于删除残留数据的选项是在数据库级别设置的，因此对于表 t1 该选项为启用，所以对 t1 执行 **drop table** 和 **truncate table** 命令都会导致清除其页面中的所有残留数据。

但是，由于创建表 t2 时使用了“**erase residual data off**”子句，所以其 **erase residual data** 选项为显式禁用。即使在数据库级别将“erase residual data”选项设置为 true，也不会删除残留数据。因此，即使在对 t2 运行 **drop table** 和 **truncate table** 后，残留数据仍将保留。

```
create database db1  
go  
sp_dboption db1, "erase residual data", true  
go  
use db1  
go  
create table t1 (col int)  
go  
insert t1 values ...  
go  
create table t2 (col1 int, col2 char(10)) with "erase residual data"  
off  
go  
truncate table t1  
go  
drop table t1  
go  
truncate table t2  
go  
drop table t2  
go
```

### 示例 2

在下面的示例中：



- 表 t1 未显式设置 “erase residual data off”，但是在数据库级别进行了设置，因此在运行 **truncate table t1** 时会删除 t1 中的残留数据。
- 表 t2 在创建时设置了 “erase residual data” 选项，这是因为该选项已在数据库级别设置。因此，运行 **truncate table t2** 时将删除 t2 中的残留数据。
- 表 t3 显式标记为 “erase residual data off”，因此即使 **sp\_dboption** 将 “erase residual data” 设置为 true，SAP ASE 运行 **truncate table t3** 时也不会删除残留数据。

```
create database db1
go
use db1
go
create table t1 (col int)
go
sp_dboption db1, "erase residual data", true
go
create table t2 (col1 int, col2 char(10))
go
create table t3 (col1 int, col2 char(10)) with "erase residual data"
off
go
truncate table t1
go
truncate table t2
go
truncate table t3
go
```

### 示例 3

在下面的示例中：

- 尽管缺省情况下 t1 和 t2 表的 “erase residual data” 选项均未设置，但是由于在执行 **truncate table** 命令前已在会话级别启用 “erase\_residual\_data”，所以 t1 和 t2 中的残留数据都将被删除。
- 尽管表 t3 的 “erase residual data” 选项显式设置为 off，但是由于已在会话级别设置 “erase\_residual\_data” 选项，所以执行 **truncate** 命令时仍将删除残留数据。

```
create database db1
go
use db1
go
create table t1(col int)
go
create table t2 (col1 int, col2 char(10))
go
create table t3 (col1 int, col2 char(10)) with "erase residual data"
off
go
set erase_residual_data on
go
truncate table t1
```

```
go
truncate table t2
go
truncate table t3
go
```

### 用法

针对某表设置此选项时，对该表执行产生残留数据的操作（**drop table**、**delete row**、**alter table** 和 **drop index**）会自动清除已释放空间。

### 权限

仅表所有者或具有 **create any table** 权限的用户可以使用“**erase residual data**”选项。

### 另请参见

- set (第 139 页)
- sp\_dboption (第 143 页)

## **create trigger** 用于创建多个触发器

使用 **create trigger** 命令创建多个触发器，并使用新的 `order` 参数指定引发触发器的顺序。

### 语法

```
create [or replace] trigger [owner.]trigger_name
on [owner.]table_name
{for | instead of} {insert | update | delete}
[order integer]
as sql_statements
```

### 参数

`order integer` 指定引发触发器的部分顺序或完整顺序：

- 使用 `order` 子句创建所有触发器时，为完整顺序。
- 部分触发器未使用 `order` 子句指定时，为部分顺序。未使用 `order` 子句创建的触发器隐式采用序列号 0，且不具备定义顺序，只是会先于使用 `order` 创建的触发器引发。

### 用法

---

**注意：**只能将 `order integer` 子句与 `for {insert | update | delete}` 一起使用；不能将其与 `instead of {insert | update | delete}` 触发器一起使用。

---

如果您为 `order` 指定重复编号，SAP ASE 将返回错误。`order` 编号不必连续；实际上，非连续编号可能更可取，因为这样允许在序列中间插入新的触发器。

另请参见

- 创建多个触发器 (第 79 页)

## **create trigger 用于 or replace**

在 **create trigger** 中使用 **or replace** 子句可替换触发器定义。

在 SAP ASE 16.0 之前的版本中，连续的 **create trigger** 命令会删除旧触发器，并替换为新的触发器定义。但同时还删除了触发器的审计选项。通过使用可选的 **or replace** 子句，不但替换了定义，同时还保留了审计选项。

*语法*

更改之处以粗体显示。

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

    SQL_statements] ...]
  [if update (column_name)
   [{and | or} update (column_name)]...
   SQL_statements]...]
```

### **create or replace trigger** 的参数更改

**create** - 如果触发器尚不存在，将创建一个触发器。

- 在 SAP ASE 16.0 及更高版本中，当存在多个触发器时，如果指定 **create** 但不使用 **or replace** 且触发器名称相同，则将引发错误。如果指定其它触发器名称，将创建新的触发器并保留旧触发器。另请参见 多个触发器
- 在 SAP ASE 16.0 之前的版本中，如果通过指定 **create** 但未使用 **or replace** 将现有触发器替换为新的触发器定义，则新触发器的名称不必与旧触发器的名称相同。
- **or replace** - 重新创建现有触发器。使用此子句可更改触发器的定义。指定 **or replace** 时，不删除触发器的审计选项。如果不存在具有您输入的名称的现有触发器，将创建新的触发器并保留旧触发器。此子句还适用于多个触发器。
- **trigger\_name** - 替换触发器定义时，不更改触发器的名称。新触发器定义的名称必须与将要替换的旧名称匹配。如果触发器名称与任何现有触发器的名称都不同，将创建新的触发器并且不会删除旧触发器。
- **table\_name** - 替换触发器时，不能更改表的名称。如果将现有触发器修改为与另一个表相关联，则会引发错误，指出另一个表上已存在触发器，无法进行替换。
- **for | instead of** - 不能将 “instead of” 触发器更改为 “for” 触发器，反之亦然。
- **insert,update,delete** - 使用 **or replace** 子句时可更改这些操作。例如，如果旧触发器定义指定所有子句，则替换定义也可指定所有子句，或指定操作的组合。

- *SQL\_statements* - 替换触发器定义时，可更改触发器条件和操作。
- **if update** - 可删除或添加 **if update**，及更改此子句引用的列名称。
- *order integer* - 替换触发器定义时，还可更改引发触发器的顺序。

*示例*

以下示例将创建这样一个触发器，即当有人试图在 titles 表中插入或更新数据时，输出一条消息：

```
create trigger reminder
on titles
for insert, update as
print "Don't forget to print a report for accounting."

select object_id("reminder")
-----
1312004674
```

接下来的命令使用 **or replace** 子句更改当有人试图更新 titles 表中的数据时所输出的触发器消息：

```
create or replace trigger reminder
on titles
for update as
print "Don't forget to give a report to accounting."

select object_id("reminder")
-----
1312004674
```

**create or replace trigger** 的权限更改

通过别名或 **setuser** 充当触发器所有者的任何用户都不能替换此触发器。

替换触发器发生的更改以粗体显示。

|        |                                                                                                                                                                                                                                                                       |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 启用细化权限 | 在启用细化权限的情况下，您必须是表所有者，并且不得撤消 create trigger 特权。您必须具有 create any trigger 特权才能对另一用户的表运行 <b>create trigger</b> 。您必须是触发器所有者才能替换触发器。                                                                                                                                        |
| 禁用细化权限 | 在禁用细化权限的情况下：<br>只有系统安全员才能授予或撤消创建触发器的权限。数据库所有者具有对任何用户表创建触发器的隐式权限。用户只能对他们所拥有的表创建触发器。<br><br>系统安全员可以撤消用户创建触发器的权限。撤消创建触发器的权限只影响系统安全员从中发出 <b>revoke</b> 命令的数据库。当系统安全员向被撤消权限的用户显式授予 create trigger 权限时，该用户运行 <b>create trigger</b> 命令的权限即得到恢复。<br><b>您必须是触发器所有者才能替换触发器。</b> |

**create or replace trigger** 的审计更改

更改之处以粗体显示。

| 事件 | 审计选项   | 审计的命令或访问权限     | extrainfo 中的信息                                                                                                                                                                                                                                 |
|----|--------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 12 | create | create trigger | <ul style="list-style-type: none"> <li>角色 - 当前活动角色</li> <li>关键字或选项 - NULL</li> <li>以前值 - NULL</li> <li>其它信息 - NULL</li> <li>当前值 - NULL</li> <li>代理信息 - set proxy 有效时的初始登录名</li> <li><b>or replace</b> - 用于 <b>create or replace</b></li> </ul> |

## create view

在 **create view** 中使用 **or replace** 子句可替换视图定义。

### 语法

更改之处以粗体显示。

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

### **create or replace view** 的参数更改

- **create** - 如果视图尚不存在，将创建一个视图。
- **or replace** - 在不更改视图任何安全属性的情况下替换现有视图定义。
- *view\_name* - 将要替换的视图的名称。只能替换现有视图。对象名称和 ID 保持不变。
- *column\_name* - 指定要用作视图中列标题的名称。通过 **or replace** 子句，可如下所示更改视图的列名称：
  - 如果之前的视图定义包含列标题名称，则新的视图定义可省略标题，或具有不同的列标题名称。
  - 如果之前的视图定义不包含列标题名称，则新定义可包含视图列标题名称。
  - 可根据 **select\_statement** 中的列名称更改列标题的数量。
- **distinct** - 允许重复行；如果原始视图定义未指定 **distinct** 子句，则可更改此参数，使新视图不能包含重复行。
- *select\_statement* - 在视图定义中指定不同的表和视图。可更改在替换后视图的 **select\_statement** 目标列表中指定的列，从而删除或添加列。
- **with check option** - 使用 **with check option** 子句创建的视图可不使用此子句进行替换，反之亦然。

### 示例

#### 示例 1

以下示例基于视图 `Current_Product_List`，该视图列出了表 `Products` 中的所有活动产品。该视图定义为：

```
create view Current_Product_List as
select ProductID, ProductName
from Products
where Discontinued = "No"

select object_id("Current_Product_List")
-----
889051172
```

接下来的命令使用 **or replace** 子句将 `Category` 列添加到 `Current_Product_List`。视图的对象 ID 保持不变：

```
create or replace view Current_Product_List as
select ProductID, ProductName, Category
from Products
where Discontinued = "No"

select object_id("Current_Product_List")
-----
889051172
```

## 示例 2

以下示例替换了 `V1`（具有相关对象的视图）：

```
create table T1(C1 int, C2 int)
create table T2(C1 int, C2 int)

create view V1 as select * from T1
create view V2 as select * from V1

create function fool
returns int
as
begin
    declare @number int
    select @number = C1 from V2
end
return @number
select object_id("V1")
-----
985051514
```

```
create or replace V1 as select * from T2

select * from V2

select dbo.fool()

select object_id("V1")
-----
985051514
```

v1 的替换版本参考的是 T2 而不是 T1。V2 和 foo1 都将重新编译。select \* from v2 将重新编译 v2，而非 foo1，当调用 UDF 时才会重新编译 foo1。

#### 与替换后的视图相关的对象

视图可包含在其它对象定义中。

- 如果被替换视图包含在另一视图中，则在访问该父视图时将自动重新编译该父视图。
- 如果由于替换更改了视图中列的数量，可能需要对参考该视图的其它视图和过程的定义进行修复。

在这种情况下，所有者以不同数量的列和列名称替换了 v1。还要必须替换 P。

```
create view V1 as select C1 from T1
create procedure P as select * from V1
```

```
create or replace V1 as select C1, C2 from T1
```

在接下来的这种情况下，所有者通过从定义中删除 C2 替换了 V2。执行 P 时将引发一条错误，因为 C2 不再是 V2 的一部分。因此，必须替换 P。

```
create view V2 as select C1, C2 from T2
create procedure P as select C2 from V2
```

```
create or replace V2 as select C1 from T2
```

在替换某个视图前，请运行 **sp\_depends** 来确定是否存在任何与要替换视图具有依赖关系的存储过程或父视图。如果存在此类存储过程或父视图，请在替换视图后根据需要替换存储过程或父视图。

- 替换视图时，将删除为视图定义的 Instead of 触发器
- 需要对依赖于替换后的视图的所有 PRS 执行完全刷新，将它们恢复到可用状态。在重新编译 PRS 之前，既不可以对其执行刷新，也不可以将其用于查询重写。

#### 限制

如果在列级别为某个视图授予了权限，则无法替换该视图，并将引发错误 2014。要替换该视图，必须先撤消权限，或者删除该视图，然后重新创建。

#### **create or replace view** 的权限更改

通过别名或 **setuser** 充当视图所有者的任何用户都不能替换此视图。

替换视图发生的更改以粗体显示。

|        |                                                                                                                                   |
|--------|-----------------------------------------------------------------------------------------------------------------------------------|
| 启用细化权限 | 在启用细化权限的情况下，您必须具有 <code>create view</code> 特权。您必须具有 <code>create any view</code> 特权才能为其他用户运行 <b>create view</b> 。您必须是视图所有者才能替换视图。 |
| 禁用细化权限 | 在禁用细化权限的情况下，您必须是数据库所有者或具有 <code>create view</code> 特权。您必须是视图所有者才能替换视图。                                                            |

### **create or replace view** 的审计更改

更改之处以粗体显示。

| 事件 | 审计选项   | 审计的命令或访问权限  | extrainfo 中的信息                                                                                                                                                                                                                                               |
|----|--------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13 | create | create view | <ul style="list-style-type: none"> <li>• 角色 - 当前活动角色</li> <li>• 关键字或选项 - NULL</li> <li>• 以前值 - NULL</li> <li>• 其它信息 - NULL</li> <li>• 当前值 - NULL</li> <li>• 代理信息 - set proxy 有效时的初始登录名</li> <li>• <b>or replace</b> - 用于 <b>create or replace</b></li> </ul> |

## drop encryption key

删除 master 数据库中 sysencryptkeys 表的数据库加密密钥。

### 语法

```
drop encryption key key_name
```

### 参数

- **key\_name** - 数据库加密密钥的名称。

### 用法

如果要删除的数据库加密密钥仍用于加密任何数据库，该命令将失败。

### 标准

符合 ANSI SQL 的级别：Transact-SQL 扩展。

### 权限

对 **drop encryption key** 的权限检查因您的细化权限设置而异：

|        |                                                                                 |
|--------|---------------------------------------------------------------------------------|
| 启用细化权限 | SAP ASE 创建一个名为 “manage database encryption key” 的新权限。必须具有该权限才能创建数据库加密密钥。        |
| 禁用细化权限 | 您必须是具有 sso_role 或 keycustodian_role 角色的用户或必须具有 <b>create encryption key</b> 特权。 |

### 另请参见

- create encryption key (第 111 页)



- 更改数据库加密密钥 (第 59 页)
- 删除数据库加密密钥 (第 59 页)
- 备份数据库加密密钥 (第 68 页)
- 创建数据库加密密钥 (第 57 页)
- `alter database` 用于完全数据库加密 (第 102 页)
- `create database` 用于完全数据库加密 (第 108 页)
- 完全数据库加密和系统更改 (第 71 页)
- `create archive database` 用于完全数据库加密 (第 108 页)
- `dbencryption_status` (第 101 页)
- `sp_helpdb` (第 146 页)
- `sp_encryption` (第 145 页)
- `ddlgen` (第 169 页)
- `sybmigrate` (第 170 页)
- 更改过的系统表 (第 160 页)

## drop trigger

可以使用 **drop trigger** 删除或替换现有触发器或多个触发器。

SAP ASE 16.0.版中引入了创建多个触发器及指定在语句执行后引发触发器的顺序的功能。

**drop trigger** 命令只能删除一个触发器。如果表上拥有多个触发器，可逐一删除。

## dump database

**dump database** 针对使用压缩选项创建的数据库或事务转储的原始数据意外更改增加了循环冗余检查，目的是检查并验证压缩块是否可正确读取和解压缩。

语法为：

```
dump database database_name to dump_device with
compression=n,verify={crc | read_after_write}
```

其中：

- **verify=crc** - 指示正在执行循环冗余检查。
- **verify=read\_after\_write** - Backup Server 将在写入并解压缩每个压缩块后对其进行重新读取。如果 Backup Server 发现错误，则其将在发现该错误的文件中打印偏移。不能将 **verify=read\_after\_write** 用于 **load database** 命令。

## kill

**kill** 命令添加了 **with force** 参数。

如果不能通过常规的 **killspid** 参数终止进程，则使用 **with force** 参数。

### 语法

语法为:

```
kill spid with force
```

其中 **with force** 指示您正在强制终止指定的 **spid**。

### 示例

以下是终止 **spid 16** 的示例:

```
kill 16 with force
```

### 用法

- 如果您使用 **with force** 来终止持有螺旋锁的 **spid**, **SAP ASE** 将发出以下消息:

```
You cannot kill spid 'spid_number' with force option as it is holding spinlock(s).
```

- 发出 **with force** 参数的权限与发出 **kill spid\_number** 的权限相同。
- **spid\_number** 必须是常量; 它不可作为参数传递到存储过程, 也不可作为局部变量使用。

## load database

**load database** 针对压缩数据库或事务转储的原始数据意外更改增加了循环冗余检查。

语法为:

```
load database database_name from dump_device with verify[only]=crc
```

其中:

- **verify=crc** - 指示正在执行循环冗余检查。

## select

在 **SAP ASE 16.0** 版以及更高版本中, 以链式模式发出仅引用 **@**变量、**@@**全局变量和常量的 **select** 语句不会启动新事务。

可以创建如下 **SQL** 结构:

```
set chained ON
<... Perform some DML or other commands ...>
select @@error, @@trancount, @@transtate
```

在某些情况下, 如果在执行 **DML** 或其它语句时出现错误, **SAP ASE** 可能会回退活动事务。服务器为链式模式时, 使用 **select** 语句检查事务状态 (不会自动启动新事务) 并收集触发回退的错误的发生日期和时间。

大部分功能在与 **select** 语句结合使用时都会启动新事务, 但以下诊断功能除外:

- **getdate**
- **getutcdate**

- **current\_date**
- **current\_time**
- **current\_bigdatetime**
- **current\_bigtime**
- **abs**
- **asehostname**
- **hostname**
- **switchprop**

## select into

**select into** 语法已扩展为通过从现有表中选择来创建按索引压缩的表。

### 语法

```
select [ all | distinct ] select_list
into [[database.][owner].table_name
[with {max_rows_per_page = num_rows,
      exp_row_size = num_bytes,
      reservepagegap = num_pages,
      identity_gap = value,
      transfer_table [on | off],
      compression [= {NONE | PAGE | ROW | ALL} ] ,
      index_compression [= {NONE | PAGE} ]
}
]
[on segment_name]
```

### 参数更改

- **NONE** - 不压缩指定表的索引。压缩专门使用 **index\_compression = PAGE** 创建的索引。
- **PAGE** - 压缩指定表的所有索引。不压缩专门使用 **index\_compression = NONE** 创建的索引。

### 另请参见

- 创建按索引压缩的表 (第 52 页)
- create table 用于索引压缩 (第 126 页)

## set

SAP ASE 16.0 版中引入了对 **set** 命令的更改。

### *set spinnlock\_aggregation {on | off}*

使用 **set spinnlock\_aggregation** 参数确定当结果集中多个行具有相同 SpinnlockName 值时，SAP ASE 是否会集合 monSpinnlockActivity 中所报告的螺旋锁指标。

缺省情况下，SAP ASE 会为所有拥有相同值的螺旋锁集合 Grabs、Waits 以及 Spins 的值。将 **set spinlock\_aggregation** 设置为 off 以将 SAP ASE 配置为针对 monSpinlockActivity 表中的每个螺旋锁实例单独返回一行。

本示例显示了 **set spinlock\_aggregation** 禁用时从 monSpinlockActivity 返回的结果集。

```

1> set spinlock_aggregation off
2> go
1> select * from monSpinlockActivity
2> where SpinlockName like "default data cache%"
3> order by Contention
4> go

```

| Grabs              | Spins        | Waits      | OwnerPID |
|--------------------|--------------|------------|----------|
| LastOwnerPID       | Contention   | InstanceID |          |
| SpinlockSlotID     | SpinlockName |            |          |
| 37697              | 978          | 1          | 0        |
| 1638413            | 0.000027     | 0          | 2338     |
| default data cache | 16396        | 15         |          |
| 2                  | 0            |            |          |
| 1638413            | 0.000122     | 0          | 2340     |
| default data cache |              |            |          |
| 17317              | 24           | 3          | 0        |
| 1638413            | 0.000173     | 0          | 2339     |
| default data cache |              |            |          |
| 27533              | 629          | 5          | 0        |
| 1638413            | 0.000182     | 0          | 2341     |
| default data cache |              |            |          |

选择此 monSpinlockActivity 实例的行数：

```

select count(*) from monSpinlockActivity
-----
2384

```

但是，如果您启用了 **set spinlock\_aggregation** 并且要执行相同的查询，则示例如下：

```

1> set spinlock_aggregation on
2> go
1> select * from monSpinlockActivity
2> where SpinlockName like "default data cache%"
3> order by Contention
4> go

```

| Grabs          | Spins        | Waits      | OwnerPID |
|----------------|--------------|------------|----------|
| LastOwnerPID   | Contention   | InstanceID |          |
| SpinlockSlotID | SpinlockName |            |          |



**sp\_audit**

SAP ASE 16.0 添加了 **config\_history** 审计选项。

| 选项                    | 说明              |
|-----------------------|-----------------|
| <b>config_history</b> | 启用或禁用对配置历史记录的审计 |

**sp\_chgattribute**

**pnt\_locking** 表属性通过 **sp\_chgattribute** 启用和禁用分区级锁定。缺省情况下，分区锁定处于禁用状态。

**语法**

```
sp_chgattribute objectname, 'pnt_locking', value
```

**参数**

- **objectname** - 要更改 **pnt\_locking** 的表的名称。
- **pnt\_locking** - 要启用分区级锁定则设为 1，要禁用则设为 0。

**示例**

- **示例 1** - 此示例启用 authors 表的分区级锁定：  

```
sp_chgattribute authors, "pnt_locking", 1
```
- **示例 2** - 此示例禁用 authors 表的分区级锁定：  

```
sp_chgattribute authors, "pnt_locking", 0
```

**权限**

只有对象所有者才能执行 **sp\_chgattribute**。

**sp\_clusterlockusage**

**sp\_clusterlockusage** 输出已得到增强，可打印特定于分区锁的集群锁使用信息。

| Lock Usage             | count    | % of total    |
|------------------------|----------|---------------|
| Total Locks            | 1479860  | n/a           |
| Free Locks             | 1469318  | 99.29 %       |
| Used Locks             | 10542    | 0.71 %        |
| Object Locks           | 7948     | 0.54 %        |
| Physical Locks         | 1864     | 0.13 %        |
| Table Locks            | 0        | 0.00 %        |
| <b>Partition Locks</b> | <b>8</b> | <b>0.00 %</b> |
| Page Locks             | 0        | 0.00 %        |
| Row Locks              | 42       | 0.00 %        |
| Others                 | 680      | 0.05 %        |

## **sp\_dboption**

**sp\_dboption** 系统过程支持删除 SAP ASE 中删除操作的残留数据。

### 语法

启用在数据库级别删除残留数据的语法为：

```
sp_dboption dbname, "erase residual data", true
```

### 示例

下列示例使用以下两个表：

- create table t1 (col1 int) with "erase residual data" on
- create table t2 (col1 int) with "erase residual data" off

### 示例 1

由于删除残留数据的选项是在数据库级别设置的，因此对于表 t1 该选项为启用，所以对 t1 执行 **drop table** 和 **truncate table** 命令都会导致清除其页面中的所有残留数据。

但是，由于创建表 t2 时使用了“**erase residual data off**”子句，所以其 **erase residual data** 选项为显式禁用。即使在数据库级别将“erase residual data”选项设置为 true，也不会删除残留数据。因此，即使在对 t2 运行 **drop table** 和 **truncate table** 后，残留数据仍将保留。

```
create database db1
go
sp_dboption db1, "erase residual data", true
go
use db1
go
create table t1 (col int)
go
insert t1 values ...
go
create table t2 (col1 int, col2 char(10)) with "erase residual data"
off
go
truncate table t1
go
drop table t1
go
truncate table t2
go
drop table t2
go
```

### 示例 2

在下面的示例中：

- 表 t1 未显式设置 “erase residual data off”，但是在数据库级别进行了设置，因此在运行 **truncate table t1** 时会删除 t1 中的残留数据。
- 表 t2 在创建时设置了 “erase residual data” 选项，这是因为该选项已在数据库级别设置。因此，运行 **truncate table t2** 时将删除 t2 中的残留数据。
- 表 t3 显式标记为 “erase residual data off”，因此即使 **sp\_dboption** 将 “erase residual data” 设置为 true，SAP ASE 运行 **truncate table t3** 时也不会删除残留数据。

```
create database db1
go
use db1
go
create table t1 (col int)
go
sp_dboption db1, "erase residual data", true
go
create table t2 (col1 int, col2 char(10))
go
create table t3 (col1 int, col2 char(10)) with "erase residual data"
off
go
truncate table t1
go
truncate table t2
go
truncate table t3
go
```

### 用法

在数据库级别启用此设置时，任何会导致释放的操作之后都会清理其页面。缺省情况下禁用此选项。

---

**注意：** 由于无法在细化级别进行控制，因此使用此选项会对性能产生显著影响。

---

### 权限

要使用 **sp\_dboption** 设置数据库级选项，用户必须为系统管理员或数据库所有者。

### 另请参见

- **create table** 用于残留数据删除（第 128 页）
- **set**（第 139 页）

### **sp\_depends**

在 SAP ASE 16.0 版中，您可以使用 **sp\_depends** 列出与表有关的多个触发器。

**sp\_depends** 无语法更改。使用此语法查看与每个 DML 操作有关的多个触发器的列表：

```
sp_depends table_name
```



**sp\_encryption**

**sp\_encryption** 系统过程报告名为“database encryption key”的新密钥类型以显示数据库是否完全加密。

例如：

```

1> create encryption key key1 as default for database encryption
2> go
1> sp_encryption helpkey, key1

```

| Key Name                          | Key Owner    | Key Length   | Key Algorithm | Pad | Initialization Vector |
|-----------------------------------|--------------|--------------|---------------|-----|-----------------------|
| Key Type                          | Protected By | Key Recovery |               |     |                       |
| # of Key Copies                   |              |              |               |     |                       |
| key1                              | dbo          | 256          | AES           |     |                       |
| symmetric database encryption key |              |              |               | 0   |                       |
| master key                        | 0            |              |               |     | 0                     |

```

1> create encryption key key2 for database encryption with master key
2> create encryption key key3 for database encryption with
dual_control
3> go
1> sp_encryption helpkey, 'key%'

```

| Key Name                                   | Key Owner    | Key Length   | Key Algorithm | Pad | Initialization Vector |
|--------------------------------------------|--------------|--------------|---------------|-----|-----------------------|
| Key Type                                   | Protected By | Key Recovery |               |     |                       |
| # of Key Copies                            |              |              |               |     |                       |
| key1                                       | dbo          | 256          | AES           |     |                       |
| symmetric database encryption key          |              |              |               | 0   |                       |
| master key                                 | 0            |              |               |     | 0                     |
| key2                                       | dbo          | 256          | AES           |     |                       |
| symmetric database encryption key          |              |              |               | 0   |                       |
| master key                                 | 0            |              |               |     | 0                     |
| key3                                       | dbo          | 256          | AES           |     |                       |
| symmetric database encryption key          |              |              |               | 0   |                       |
| dual_control(master key + dual master key) |              |              |               | 0   | 0                     |

```

1> create database encr_db1 encrypt with key1
2> create database encr_db2 encrypt with key2
3> create database encr_db3 encrypt with key3
4> go
1> sp_encryption helpkey, '%', "display_dbs"

```

| Key Name | Key Owner | Encrypted Database |
|----------|-----------|--------------------|
| key1     | dbo       | encr_db1           |
| key1     | dbo       | encr_db2           |
| key3     | dbo       | encr_db3           |

**另请参见**

- 完全数据库加密和系统更改 (第 71 页)
- `create archive database` 用于完全数据库加密 (第 108 页)
- `dbencryption_status` (第 101 页)
- `sp_helpdb` (第 146 页)
- `ddlgen` (第 169 页)
- `sybmigrate` (第 170 页)
- 更改过的系统表 (第 160 页)
- `alter database` 用于完全数据库加密 (第 102 页)
- `create database` 用于完全数据库加密 (第 108 页)
- `create encryption key` (第 111 页)
- `drop encryption key` (第 136 页)

**sp\_familylock**

`partitionid` 列已添加至 `sp_familylock` 存储过程的输出。

语法为：

```
sp_familylock
```

`partitionid` 的表锁值和精细锁值为 0。仅为分区级锁填充 `partitionid`。

| spid | locktype            | table_id   | partitionid | page | row... |
|------|---------------------|------------|-------------|------|--------|
| 0    | Ex_partition        | 576002052  | 576004423   | 0    | 0      |
| 0    | Sh_partition_intent | 1417053053 | 1417053053  | 0    | 0      |

**另请参见**

- 通过 `sp_familylock` 查看分区锁 (第 5 页)

**sp\_helpdb**

`sp_helpdb` 系统过程支持完全数据库加密功能。

针对完全加密数据库运行 `sp_helpdb` 时，将报告其加密状态：

- 已加密
- 正在进行加密
- 正在进行解密

如果数据库正在进行加密或解密，`sp_helpdb` 将报告已完成的工作百分比。

*示例*

**示例 1**

报告正在进行加密的数据库的状态：

```
>sp_helpdb
>go
```

```

name          db_size      owner dbid  created      durability
      lobcomplvl inrowlen
status
.....
test_db       6.0 MB       sa     4    Aug 07, 2013 full
              0 NULL
              encryption in progress: 35%
.....

```

## 示例 2

报告部分加密的数据库的状态:

```

>sp_helpdb
>go
name          db_size      owner dbid  created      durability
      lobcomplvl inrowlen
status
.....
test_db       6.0 MB       sa     4    Aug 07, 2013 full
              0 NULL
              encrypted partly
.....

```

## 示例 3

报告正在进行解密的数据库的状态:

```

>sp_helpdb
>go
name          db_size      owner dbid  created      durability
      lobcomplvl inrowlen
status
.....
test_db       6.0 MB       sa     4    Aug 07, 2013 full
              0 NULL
              decrypted partly
.....

```

## 另请参见

- 加密现有数据库 (第 61 页)
- `dbencryption_status` (第 101 页)
- 挂起加密进程 (第 66 页)
- 恢复加密进程 (第 67 页)
- `alter database` 用于完全数据库加密 (第 102 页)
- 完全数据库加密和系统更改 (第 71 页)
- `create archive database` 用于完全数据库加密 (第 108 页)
- `sp_encryption` (第 145 页)
- `ddlgen` (第 169 页)
- `sybmigrate` (第 170 页)

## 系统更改

- 更改过的系统表 (第 160 页)
- create database 用于完全数据库加密 (第 108 页)
- create encryption key (第 111 页)
- drop encryption key (第 136 页)

### **sp\_lock**

partitionid 列已添加至 **sp\_lock** 存储过程的输出。

partitionid 的表锁值和精细锁值为 0。仅为分区级锁填充 partitionid。

此示例显示当前由 SAP ASE 持有的全部锁，包括分区锁。

```
sp_lock
go

fid spid loid locktype table_id partitionid page row dbname class
context
-----
-----
0 13 26 Ex_intent 420193516 0 0 0 master Non Cursor Lock
0 13 26 Ex_intent_partition 420193516 452193630 0 0 master Non
Cursor Lock
0 13 26 Ex_page 420193516 452193630 4993 0 master Non Cursor Lock
0 14 28 Ex_intent 420193516 0 0 0 master Non Cursor Lock
0 14 28 Ex_intent_partition 420193516 468193687 0 0 master Non
Cursor Lock
0 14 28 Ex_page 420193516 468193687 5001 0 master Non Cursor Lock
0 16 32 Sh_intent 1006623598 0 0 0 master Non Cursor Lock
```

### 另请参见

- 通过 sp\_lock 查看分区锁 (第 4 页)

## **新的系统过程**

SAP ASE 16.0 版本引入了新的系统过程。

### **sp\_confighistory**

创建 ch\_events 视图并显示对 SAP ASE 配置所做的更改。

### **语法**

```
sp_confighistory create_view
    begin_date[, end_date]]
    last[items_num]
    {area | type | target | element}[, item_name]
    help
```

### **参数**

- **create\_view** - 指示正在创建 ch\_events 视图。

- **begin\_date, [end\_date** - 显示从 *begin\_date* 值到 *end\_date* 值的所有项目。
- **last** - 显示最新的配置历史记录项目。
- **items\_num** - 要从最新的配置历史记录项目列表中显示的项目数。
- **area | type | target | element** - 显示指定区域内的项目：
  - **area** - 可审计事件发生的区域。可以是以下各项之一：
    - server - 服务器级别事件。
    - database - 数据库级别事件。
    - cache - 高速缓存级别事件。
    - traceflag - **dbcc traceflag** 和 **set switch** 事件。
    - SUSU - startup/shutdown。
    - audit - 审计状态更改。
  - **type** - 可审计事件的类型。可以是以下各项之一：
    - **sp\_configure**
    - **sp\_serveroption**
    - **sp\_dboption**
    - **sp\_cacheconfig**
    - **sp\_poolconfig**
    - **create thread pool**
    - **alter thread pool**
    - **drop thread pool**
    - **dbcc traceflag**
    - **set switch**
    - configuration file change
    - startup
    - shutdown
    - shutdown with wait
    - shutdown with nowait
    - abrupt shutdown
    - global auditing
    - config history auditing
  - **target** - 更改应用的目标对象的名称（例如，服务器、高速缓存、线程池、数据库名称、跟踪标志编号等）。
  - **element** - 配置或其它选项的名称（例如，“enable monitoring”、“config pool: 4K, 选项: wash size”等）。
- **help** - 显示 **sp\_confighistory** 的使用信息。

## 示例

-

## 权限

- 只有系统管理员（具有 `sa_role` 的用户）可以使用此过程来创建 `ch_events` 视图。
- 只有系统管理员（具有 `sa_role` 的用户）和具有 `mon_role` 的用户可以使用此过程来查询 `ch_events` 视图。

权限检查因您的细化权限设置而异：

| 设置 | 说明                                                                                                                                                                                                                                                                                                                                                      |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 启用 | <p>仅限具有以下权限的用户：</p> <ul style="list-style-type: none"> <li>• <code>select any audit table</code> 权限可对 <code>ch_events</code> 视图进行查询。</li> <li>• <code>manage auditing</code> 权限可更改配置历史记录审计状态的选项。</li> <li>• <code>select any audit table</code> 权限可对 <code>ch_events</code> 视图进行查询。</li> <li>• <code>select any audit table</code> 权限可查询审计表。</li> </ul> |
| 禁用 | <p>仅限：</p> <ul style="list-style-type: none"> <li>• 系统安全员（具有 <code>sso_role</code> 的用户）可以更改配置历史记录审计状态的选项</li> <li>• 只有系统管理员（具有 <code>sa_role</code> 的用户）和具有 <code>mon_role</code> 的用户可以针对 <code>ch_events</code> 视图进行查询。</li> </ul>                                                                                                                     |

## sp\_dropglockpromote\_ptn

删除分区锁升级值。

### 语法

删除全服务器范围的分区锁升级设置的语法是：

```
sp_dropglockpromote_ptn "server"
```

删除数据库或表级别的分区锁升级阈值的语法是：

```
sp_dropglockpromote_ptn {"database" | "table"}, objname
```

### 参数

- **server** - 删除分区锁升级阈值的全服务器范围的值。
- **"database" | "table"** - 指定是为数据库还是为表删除分区锁升级阈值。这些是 Transact-SQL 关键字，因此需加引号。
- **objname** - 是要从其中删除分区锁升级阈值的表或数据库的名称。

### 示例

- **示例 1** - 从 `titles` 中删除分区锁升级值。 `titles` 的锁升级现在使用数据库或全服务器范围的值：

```
sp_dropglockpromote_ptn "table", titles
```

## 用法

使用 `sp_dropglockpromote_ptn` 时，还存在一些其它注意事项：

- 使用 `sp_dropglockpromote_ptn` 删除通过 `sp_setglockpromote_ptn` 设置的分区锁升级值。
- 删除数据库的分区锁升级阈值时，未配置分区锁升级阈值的表使用全服务器范围的值。
- 删除表的值之后，SAP ASE 服务器将使用数据库的锁升级阈值（如果已配置），或全服务器范围的值（如果未配置）。
- 删除全服务器范围的分区锁升级阈值后，将使用在表级别设置的分区锁升级阈值。否则将使用在数据库级别设置的分区锁升级阈值。如果既未在数据库级别也未在表级别设置分区锁升级阈值，则会禁用分区锁升级。可使用 `sp_setrowlockpromote_ptn` 重新启用。

## 权限

对 `sp_dropglockpromote_ptn` 的权限检查因细化权限设置而异。

| 设置 | 说明                                                                     |
|----|------------------------------------------------------------------------|
| 启用 | 在启用细化权限的情况下，您必须是具有 <code>manage lock promotion threshold</code> 特权的用户。 |
| 禁用 | 在禁用细化权限的情况下，您必须是具有 <code>sa_role</code> 的用户。                           |

## 审计

`sysaudits` 表中 `event` 和 `extrainfo` 列的值如下所示：

| 信息                    | 值                                                                                                                                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 事件                    | 38                                                                                                                                                                                                               |
| 审计选项                  | <code>exec_procedure</code>                                                                                                                                                                                      |
| 审计的命令或访问权限            | 执行过程                                                                                                                                                                                                             |
| <b>extrainfo</b> 中的信息 | <ul style="list-style-type: none"> <li>• 角色 - 当前活动角色</li> <li>• 关键字或选项 - NULL</li> <li>• 以前值 - NULL</li> <li>• 当前值 - NULL</li> <li>• 其它信息 - 所有输入参数</li> <li>• 代理信息 - <code>set proxy</code> 有效时的初始登录名</li> </ul> |

### 另请参见

- `sp_dropprowlockpromote_ptn` (第 152 页)
- 删除分区锁升级阈值 (第 7 页)

### **`sp_dropprowlockpromote_ptn`**

删除服务器、数据库或表级别的分区锁升级阈值。

### 语法

删除全服务器范围的分区锁升级设置的语法是：

```
sp_dropprowlockpromote_ptn "server"
```

删除数据库或表级别的分区锁升级阈值的语法是：

```
sp_dropprowlockpromote_ptn {"database" | "table"}, objname
```

### 参数

- **server** - 删除分区锁升级阈值的全服务器范围的值。
- **"database" | "table"** - 指定是为数据库还是为表删除分区锁升级阈值。这些是 Transact-SQL 关键字，因此需加引号。
- **objname** - 是要从其中删除分区锁升级阈值的表或数据库的名称。

### 示例

- **示例 1** - 从 `sales` 表中删除分区锁升级值。 `sales` 的分区锁升级现在使用数据库或全服务器范围的值：

```
sp_dropprowlockpromote_ptn "table", "sales"
```

### 用法

使用 `sp_dropprowlockpromote_ptn` 时，还存在一些其它注意事项：

- 使用 `sp_dropprowlockpromote_ptn` 删除通过 `sp_setrowlockpromote_ptn` 设置的分区锁升级值。
- 删除数据库的分区锁升级阈值时，未配置表级别的分区锁升级阈值的数据行锁定表使用全服务器范围的值。使用 `sp_configure` 检查分区锁升级配置参数的值。
- 删除表的分区锁升级值后，SAP ASE 服务器将使用数据库的分区锁升级阈值（如果已配置），或使用服务器范围的值（如果没有为该数据库设置阈值）。
- 要更改数据库的分区锁升级阈值，必须使用 `master` 数据库。要更改数据库中表的分区锁升级阈值，必须使用表所在的数据库。
- 删除全服务器范围的分区锁升级阈值后，将使用在表级别设置的分区锁升级阈值。否则将使用在数据库级别设置的分区锁升级阈值。如果既未在数据库级别也未在表级别设置分区锁升级阈值，则会禁用分区锁升级。可使用 `sp_setrowlockpromote_ptn` 重新启用。



## 权限

对 `sp_droprowlockpromote_ptn` 的权限检查因细化权限设置而异。

| 设置 | 说明                                                                     |
|----|------------------------------------------------------------------------|
| 启用 | 在启用细化权限的情况下，您必须是具有 <code>manage lock promotion threshold</code> 特权的用户。 |
| 禁用 | 在禁用细化权限的情况下，您必须是具有 <code>sa_role</code> 的用户。                           |

## 审计

`sysaudits` 表中 `event` 和 `extrainfo` 列的值如下所示：

| 信息                          | 值                                                                                                                                                                                                    |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 事件                          | 38                                                                                                                                                                                                   |
| 审计选项                        | <code>exec_procedure</code>                                                                                                                                                                          |
| 审计的命令或访问权限                  | 执行过程                                                                                                                                                                                                 |
| <code>extrainfo</code> 中的信息 | <ul style="list-style-type: none"> <li>角色 - 当前活动角色</li> <li>关键字或选项 - NULL</li> <li>以前值 - NULL</li> <li>当前值 - NULL</li> <li>其它信息 - 所有输入参数</li> <li>代理信息 - <code>set proxy</code> 有效时的初始登录名</li> </ul> |

## 另请参见

- `sp_droplockpromote_ptn` (第 150 页)
- 删除分区锁升级阈值 (第 7 页)

## `sp_helptrigger`

SAP ASE 16.0 版中引入了 `sp_helptrigger` 系统过程。

`sp_helptrigger` 列出

- `tablename` 指定的表上创建的所有触发器
- 引发触发器的命令 (`insert`、`update` 或 `delete`)
- 触发器序列号

## 语法

```
sp_helptrigger tablename
```

## 参数

- *tablename* - 表的名称。

## 权限

任何用户都可以执行 `sp_helptrigger`。

## sp\_jsconfigure

配置 Job Scheduler 代理。

## 语法

```
sp_jsconfigure [option [, value]]
```

## 参数

- *option* - 可以选择以下其中一个选项：
  - *interfaces path* - interface 文件的路径
  - *errorlog* - 错误日志的路径
  - *help* - 显示 `sp_jsconfigure` 的语法
- *value* - 指定设置 *option* 的值。

## 示例

- **示例 1** - 显示 `sp_jsconfigure` 语法：

```
sp_jsconfigure "help"
Usage: sp_jsconfigure [option [, value]]
       where option : 'interfaces path', 'errorlog ', 'help'
              value : value to set for the 'option'
```

- **示例 2** - 设置 `interfaces` 文件的路径：

```
sp_jsconfigure 'interfaces path', "/SAP_ASE/data"
```

- **示例 3** - 设置错误日志的路径：

```
1> sp_jsconfigure "errorlog", "/SAP_ASE/data/js.log"
```

- **示例 4** - 显示为 `sp_jsconfigure` 设置的值：

```
sp_jsconfigure
Parameter Name          Config
Value
-----
-----
interfaces path        /SAP_ASE/
data
errorlog                /SAP_ASE/data/
```

```
js.log
```

## 用法

- 使用 `installjsdb` 脚本安装 `sp_jsconfigure`（位于 `$SYBASE/$SYBASE_ASE/scripts`）。
- 必须重启 JS 代理，配置更改才能生效。
- 如果未提供 `interfaces path` 或 `errorlog` 的值，`sp_jsconfigure` 将使用缺省值。
- 如果为 `interfaces path` 或 `errorlog` 选项设置不存在的路径，Job Scheduler 不会启动。
- `interfaces path` 必须包含载明 Job Scheduler 主机和执行预定作业的目标服务器条目的 `interfaces` 文件（目标服务器和主机服务器可以是同一台计算机）。

## 权限

您必须具有 `js_admin_role` 才能执行 `sp_jsconfigure`。

## sp\_logging\_rate

计算指定时间段内的事务日志增长率。

## 语法

```
sp_logging_rate {'full'|'sum', '[day,]hh:mm:ss'}[,
interval='hh:mm:ss' | clear_option='y'|'n']
```

## 参数

- **full** - `sp_logging_rate` 提供了每个集合的详细报告。
- **sum** - `sp_logging_rate` 提供了摘要信息，其中包括平均值、最小值、最大值以及最高比率。如果不指定时间，`sp_logging_rate` 将会每隔 10 秒收集一次信息。
- **day, hh:mm:ss** - 指定 `sp_logging_rate` 运行的持续时间，格式为：*date, hour:minute:second*。
- **interval = 'hh:mm:ss'** - 间隔运行的时间期间，格式为：*hour:minute:second*。
- **clear\_option = 'y' | 'n'** - 确定是否在数据收集过程中清除监控计数器。

## 示例

- **sum 参数使用示例** - `sp_logging_rate` 将收集 1 天零 8 个小时内的信息，每隔 10 分钟进行一次采样，并且在间隔结束时打印摘要信息：

```
sp_logging_rate 'sum', '1,08:00:00', '00:10:00'
=====
Total Summary Information
=====
Transaction Log Growth Rate      Min GB/h      Max GB/h      Avg
GB/h
-----
```

```

-----
                                0.000000          2.870076
1.823028
• full 参数使用示例 - sp_logging_rate 将收集 3 分钟内的信息，每隔 10 秒（缺省
  值）进行一次采样，并在间隔结束时打印摘要信息：
sp_logging_rate 'full', '00:03:00'
Date Time                               Transaction Log Growth Rate GB/h
-----
Oct 22 2013  6:00:32:480AM                0.406779
Oct 22 2013  6:00:42:483AM                0.000000
Oct 22 2013  6:00:52:483AM                0.000000
Oct 22 2013  6:01:02:483AM                0.000000
Oct 22 2013  6:01:12:490AM                0.000000
Oct 22 2013  6:01:22:500AM                0.000000
Oct 22 2013  6:01:32:476AM                2.341870
Oct 22 2013  6:01:42:483AM                2.828132
Oct 22 2013  6:01:52:480AM                2.850305
Oct 22 2013  6:02:02:483AM                2.782750
Oct 22 2013  6:02:12:483AM                2.853574
Oct 22 2013  6:02:22:480AM                2.002917
Oct 22 2013  6:02:32:483AM                2.848995
Oct 22 2013  6:02:42:483AM                2.754143
Oct 22 2013  6:02:52:483AM                2.854949
Oct 22 2013  6:03:02:480AM                2.722928

```

```

Oct 22 2013  6:03:12:476AM                               2.870076

Oct 22 2013  6:03:22:480AM                               2.697094

=====
Total Summary Information
=====
Transaction Log Growth Rate      Min GB/h      Max GB/h      Avg
GB/h
-----
-----
1.823028                               0.000000      2.870076

```

## 用法

- **sp\_logging\_rate** 运行期间不能运行收集监控数据（例如，**sp\_sysmon**）的脚本或过程。由于 **sp\_logging\_rate** 在其运行期间会搜集并清除监控计数器，因此这些脚本或过程所搜集的监控计数器信息会不准确。
- 如果为 **interval = 'hh:mm:ss'** 指定的时间大于为 **'day,hh:mm:ss'** 指定的时间，则 **sp\_logging\_rate** 生成的结果不可靠。
- 为 **interval = 'hh:mm:ss'** 和 **'day,hh:mm:ss'** 指定值时，需要注意：
  - 如果为 **interval = 'hh:mm:ss'** 指定的值大于为 **'day,hh:mm:ss'** 指定的值，则 SAP ASE 将发出错误消息且 **sp\_logging\_rate** 不会生成结果集。
  - 如果 **'day,hh:mm:ss'** 与 **interval = 'hh:mm:ss'** 的比率过小，则 **sp\_logging\_rate** 生成的结果可能不可靠。例如，如果指定 **day,00:10:00**，和 **interval='00:04:00'**，则 **sp\_logging\_rate** 将只收集两个值，然后打印平均值，其中第一个值为最大值，第二个值为最小值。比率越大，所生成的结果集越可靠。

## 权限

必须具有系统管理员特权才能执行 **sp\_logging\_rate**。

## **sp\_setpglockpromote\_ptn**

**sp\_setpglockpromote\_ptn** 系统过程设置服务器、数据库和表级别的分区锁升级阈值。

## 语法

设置服务器级别的分区锁升级阈值的语法是：

```
sp_setpglockpromote_ptn "server", null, new_lwm, new_hwm, new_pct
```

设置数据库或表级别的分区锁升级阈值的语法是：

```
sp_setpglockpromote_ptn "database | table", objname, new_lwm,
new_hwm, new_pct
```

## 参数

- **server** - 为锁升级阈值设置全服务器范围的值。
- **"database" | "table"** - 指定是为数据库还是为表设置锁升级阈值。这些是 Transact-SQL 关键字，因此需加引号。
- **objname** - 是要为其设置锁升级阈值的分区、表或数据库的名称；如果要设置全服务器范围的值，则为空值。如果要设置全分区范围的值，则对 *objname* 使用 *table\_name.partition\_name* 格式。
- **new\_lwm** - 指定在 SAP ASE 获取分区锁之前必须获取的最小页锁数。
- **new\_hwm** - 指定 SAP ASE 尝试升级到分区锁之前对象上允许的最大页锁数。
- **new\_pct** - 指定一个已锁定页的百分比（基于表大小），如果超过此百分比，SAP ASE 便会在锁数量介于 *new\_hwm* 和 *new\_lwm* 锁升级之间时尝试获取分区锁。

## 示例

- **示例 1** - 将全服务器范围的分区锁升级阈值 LWM 设置为 200，将 HWM 设置为 300，将 PCT 设置为 50：

```
sp_setpglockpromote_ptn "server", NULL, 200, 300, 50
```

- **示例 2** - 为 master 数据库设置分区锁升级阈值：

```
sp_setpglockpromote_ptn "database", master, 1000, 1100, 45
```

- **示例 3** - 为 pubs2 数据库中的 titles 表设置分区锁升级阈值。必须从 pubs2 数据库发出此命令：

```
sp_setpglockpromote_ptn "table", "pubs2..titles", 500, 700, 10
```

## 权限

任何用户都可以执行 **sp\_setpglockpromote\_ptn**。

## 另请参见

- **sp\_setrowlockpromote\_ptn**（第 158 页）
- **设置分区锁升级阈值**（第 7 页）

## **sp\_setrowlockpromote\_ptn**

**sp\_setrowlockpromote\_ptn** 系统过程将设置服务器、数据库和表级别的分区锁升级阈值。

## 语法

设置服务器级别的分区锁升级阈值的语法是：

```
sp_setrowlockpromote_ptn "server", null, new_lwm, new_hwm, new_pct
```

设置数据库或表级别的分区锁升级阈值的语法是：

```
sp_setrowlockpromote_ptn "database | table", objname, new_lwm, new_hwm, new_pct
```

## 参数

- **server** - 为锁升级阈值设置全服务器范围的值。
- **"database" | "table"** - 指定是为数据库还是为表设置锁升级阈值。这些是 Transact-SQL 关键字，因此需加引号。
- **objname** - 是要为其设置锁升级阈值的分区、表或数据库的名称；如果要设置全服务器范围的值，则为空值。如果要设置全分区范围的值，则对 *objname* 使用 *table\_name.partition\_name* 格式。
- **new\_lwm** - 指定在 SAP ASE 获取分区锁之前必须获取的最小行锁数。
- **new\_hwm** - 指定 SAP ASE 尝试升级到分区锁之前对象上允许的最大行锁数。
- **new\_pct** - 指定一个已锁定行的百分比（基于表大小），如果超过此百分比，SAP ASE 便会在锁数量介于 *new\_hwm* 和 *new\_lwm* 锁升级之间时尝试获取分区锁。

## 示例

- **示例 1** - 将全服务器范围的分区锁升级阈值 LWM 设置为 200，将 HWM 设置为 300，将 PCT 设置为 50：

```
sp_setrowlockpromote_ptn "server", NULL, 200, 300, 50
```

- **示例 2** - 为 master 数据库设置分区锁升级阈值：

```
sp_setrowlockpromote_ptn "database", master, 1000, 1100, 45
```

- **示例 3** - 为 pubs2 数据库中的 titles 表设置分区锁升级阈值。必须从 pubs2 数据库发出此命令：

```
sp_setrowlockpromote_ptn "table", "pubs2..titles", 500, 700, 10
```

## 权限

任何用户都可以执行 **sp\_setrowlockpromote\_ptn**。

## 另请参见

- **sp\_setpglockpromote\_ptn**（第 157 页）
- **设置分区锁升级阈值**（第 7 页）

## 系统表

SAP ASE 16.0 纳入了新表以及更改过的表。

### 更改过的系统表

SAP ASE 16.0 版中引入了对系统表的更改。

#### *sysattributes*

*sysattributes* 系统表支持完全数据库加密功能。加密数据库功能引入了新的 *sysattributes* 类 43，用于指示完全数据库加密。对于正在进行加密的数据库，每次进行存储分配时，SAP ASE 都会向 *sysattributes* 中插入一行，行值如下：

| 列名                  | 值                    |
|---------------------|----------------------|
| <b>class</b>        | 43                   |
| <b>object</b>       | <i>dbid</i> (数据库 ID) |
| <b>object_info1</b> | 起始逻辑页 ID             |
| <b>object_info2</b> | 结束逻辑页 ID             |
| <b>int_value</b>    | 一次存储分配中最后加密的逻辑页 ID   |

SAP ASE 完成数据库加密后，会将该行删除。

#### *sysconstraints*

在低于 16.0 版的 SAP ASE 中，*sysconstraints* 保存检查约束、唯一约束和主键约束、参照约束以及规则和计算列定义的相关信息。而在 SAP ASE 16.0 中，*sysconstraints* 还会存储下列与表关联的多个触发器的相关信息：

| 名称              | 数据类型            | 说明              |
|-----------------|-----------------|-----------------|
| <i>colid</i>    | <i>smallint</i> | 触发器的序列号。缺省值为 0。 |
| <i>constrid</i> | <i>int</i>      | 触发器 ID。         |
| <i>tableid</i>  | <i>int</i>      | 声明触发器的表的 ID。    |
| <i>error</i>    | <i>int</i>      | 不用于触发器。         |



| 名称     | 数据类型 | 说明                                                                                                        |
|--------|------|-----------------------------------------------------------------------------------------------------------|
| status | int  | 0x0080 = <b>delete</b> 触发器<br>0x0100 = <b>insert</b> 触发器<br>0x0200 = <b>Update</b> 触发器<br>0x0400 = 触发器已禁用 |
| spare  | int  | 未使用                                                                                                       |

### *sysdatabases*

*sysdatabases* 系统表支持完全数据库加密功能。加密数据库功能引入了新列 *status5*，用于指示数据库加密状态。值为：

| 十六进制              | 说明                            |
|-------------------|-------------------------------|
| <b>0x00000001</b> | 指示数据库是否加密。                    |
| <b>0x00000002</b> | 正在对数据库进行加密，且加密仍在进行中。          |
| <b>0x00000004</b> | 正在对数据库进行解密，且解密仍在进行中。          |
| <b>0x00000008</b> | 由于某个错误或由于进程被用户挂起，数据库只进行了部分加密。 |
| <b>0x00000010</b> | 由于某个错误或由于进程被用户挂起，数据库只进行了部分解密。 |

### *sysobjects*

在低于 16.0 版的 SAP ASE 中，*sysobjects* 保存下列内容：

- 用于 **delete** 触发器的 *deltrig* 列
- 用于 **insert** 触发器的 *instrig* 列
- 用于 **update** 触发器的 *updtrig* 列

在 SAP ASE 16.0 中，针对 **delete**、**insert** 以及 **update** 操作而在表上创建的首个触发器（未使用 *order* 子句或 *order 0* 创建），通过 *sysobjects* 中的上述某一列与表相关联。

针对给定操作而创建的第二个和后续触发器通过 *sysconstraints* 中的行与表相关联。此外，使用 *order 1* 或更大值创建的触发器始终使用 *sysconstraints* 与表/触发器进行关联。

相关表的 *sysobjects* 行使用 *sysstat2* 字段中的位来指示触发器已禁用。在低于 16.0 版的 SAP ASE 中，不能存在一个以上的 **insert**、**delete** 或 **upgrade** 触发器，使用下列三个位执行语句：

- *disable\_instrig* 0x001000000
- *disable\_deltrig* 0x002000000

- `disable_updtrig 0x004000000`

如上所述，SAP ASE 16.0 中提供这三个位，并且在表触发器 ID 存储在 `sysobjects` 中时会使用这些位。

### 另请参见

- 完全数据库加密和系统更改 (第 71 页)
- `create archive database` 用于完全数据库加密 (第 108 页)
- `dbencryption_status` (第 101 页)
- `sp_helpdb` (第 146 页)
- `sp_encryption` (第 145 页)
- `ddlgen` (第 169 页)
- `sybmigrate` (第 170 页)
- `alter database` 用于完全数据库加密 (第 102 页)
- `create database` 用于完全数据库加密 (第 108 页)
- `create encryption key` (第 111 页)
- `drop encryption key` (第 136 页)

## 新的系统表

SAP ASE 16.0 版本引入了新的系统表。

### ch\_events

其中每个配置更改事件都对应一行。 `ch_events` 位于 `sysmgmtdb` 数据库。

`ch_events` 是基于 `extrainfo` 列的视图。 必须具有 `mon_role` 才能查看 `ch_events`。

### 列

`ch_events` 有以下各列：

| 名称                | 数据类型                              | 说明                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>area</code> | <code>varchar(10) not null</code> | 事件发生的区域。 可以是以下各项之一： <ul style="list-style-type: none"><li>• <code>server</code> - 服务器级别事件。</li><li>• <code>database</code> - 数据库级别事件。</li><li>• <code>cache</code> - 高速缓存级别事件。</li><li>• <code>traceflag</code> - <code>dbcc traceflag</code> 和 <code>set switch</code> 事件。</li><li>• <code>SUSD</code> - 用于 <code>startup/shutdown</code>。</li><li>• <code>audit</code> - 审计状态更改。</li></ul> |

| 名称        | 数据类型                 | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type      | varchar(30) not null | 可审计事件的类型。可以是以下各项之一： <ul style="list-style-type: none"> <li>• sp_configure</li> <li>• sp_serveroption</li> <li>• sp_dboption</li> <li>• sp_cacheconfig</li> <li>• sp_poolconfig</li> <li>• create thread pool</li> <li>• alter thread pool</li> <li>• drop thread pool</li> <li>• dbcc traceflag</li> <li>• set switch</li> <li>• configuration file change</li> <li>• startup</li> <li>• shutdown</li> <li>• shutdown with wait</li> <li>• shutdown with nowait</li> <li>• abrupt shutdown</li> <li>• global auditing</li> <li>• config history auditing</li> </ul> |
| target    | varchar(30) null     | 更改应用的对象的名称。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| element   | varchar(255) null    | 配置参数或其它选项的名称。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| oldvalue  | varchar(255) null    | 更改之前的事件值。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| newvalue  | varchar(255) null    | 更改后的事件值。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| mode      | varchar(10) null     | 配置参数的状态：静态或动态。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| timestamp | datetime not null    | 事件发生的日期和时间。对于配置文件更改和突然关闭的情况，timestamp 指示检测到事件的时间而不是事件发生的时间。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| username  | varchar(30) null     | 更改者的用户名。针对以下各项设为空值： <ul style="list-style-type: none"> <li>• Startup</li> <li>• Configuration file change</li> <li>• Abrupt shutdown</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                     |

| 名称              | 数据类型         | 说明                            |
|-----------------|--------------|-------------------------------|
| instan-<br>ceid | tinyint null | (仅限于 Cluster Edition) 实例的 ID。 |

## 更改过的监控表

SAP ASE 16.0 纳入了对现有监控表的更改。

### *monCachedStatement*

SAP ASE 16.0 及更高版本中的 monCachedStatement:

- 对已完成和正在进行的查询大约每隔 5 秒钟更新一次以下各列中的指标:

| 说明               | 数据类型   | 属性  | 说明                      |
|------------------|--------|-----|-------------------------|
| TotalLIO         | bigint | 计数器 | 累计逻辑 I/O                |
| TotalPIO         | bigint | 计数器 | 累计物理 I/O                |
| TotalCPUTime     | bigint | 计数器 | 该语句占用 CPU 的累计时间 (以秒为单位) |
| TotalElapsedTime | bigint | 计数器 | 执行该语句所用的累计时间量 (以秒为单位)   |

低于 16.0 版本的 SAP ASE 在语句完成时更新 monCachedStatement 中的指标。但 SAP ASE 16.0 及更高版本执行语句高速缓存时, 将在执行查询时定期更新以下值:

- TotalLIO
- MaxLIO
- TotalPIO
- MaxPIO
- TotalCPUTime
- MaxCPUTime
- TotalElapsedTime
- MaxElapsedTime

其它指标 (例如 MinLIO 和 AvgLIO) 则在查询执行完成后进行更新。

- 语句开始执行后, 递增 UseCount 列。UseCount 的值为  
(number of completed queries) + (number of ongoing queries)

CurrentUsageCount 列包括语句的活动查询数。语句的完成执行数为:

(Value of UseCount) - (value of CurrentUsageCount)

- 如果列 (例如 CpuTime) 所描述的指标超出中间更新期间使用的最大值, 则递增用于描述当前执行语句最大值 (本例中为 MaxCPUTime) 的各列的值。最大值列

反映最新的指标（包括活动查询的指标），用于确定当前执行的查询所消耗的资源是否超出之前的使用情况或正常使用情况。

### *monDeadLock*

| 说明          | 数据类型 | 属性   | 说明       |
|-------------|------|------|----------|
| partitionid | int  | Null | 分区的唯一标识符 |

### *monLocks*

| 说明          | 数据类型 | 属性   | 说明       |
|-------------|------|------|----------|
| partitionid | int  | Null | 分区的唯一标识符 |

### *monOpenObjectActivity*

SAP ASE 16.0 及更高版本向 monOpenObjectActivity 监控表中添加了以下各列。

| 说明             | 数据类型     | 属性  | 说明              |
|----------------|----------|-----|-----------------|
| Scans          | int      | 计数器 | 对此对象执行扫描的次数。    |
| LastScanDate   | datetime | 计数器 | 最近一次对此对象执行扫描的日期 |
| Updates        | int      | 计数器 | 对此对象执行更新的次数。    |
| LastUpdateDate | datetime | 计数器 | 最近一次对此对象执行更新的日期 |
| Inserts        | int      | 计数器 | 对此对象执行插入的次数。    |
| LastInsertDate | datetime | 计数器 | 最近一次对此对象执行插入的日期 |
| Deletes        | int      | 计数器 | 对此对象执行删除的次数。    |
| LastDeleteDate | datetime | 计数器 | 最近一次对此对象执行删除的日期 |

### *monOpenPartitionActivity*

SAP ASE 16.0 及更高版本向 monOpenPartitionActivity 监控表添加了以下各列。

| 说明             | 数据类型     | 属性  | 说明              |
|----------------|----------|-----|-----------------|
| Scans          | int      | 计数器 | 对此对象执行扫描的次数。    |
| LastScanDate   | datetime | 计数器 | 最近一次对此对象执行扫描的日期 |
| Updates        | int      | 计数器 | 对此对象执行更新的次数。    |
| LastUpdateDate | datetime | 计数器 | 最近一次对此对象执行更新的日期 |

| 说明             | 数据类型     | 属性  | 说明              |
|----------------|----------|-----|-----------------|
| Inserts        | int      | 计数器 | 对此对象执行插入的次数。    |
| LastInsertDate | datetime | 计数器 | 最近一次对此对象执行插入的日期 |
| Deletes        | int      | 计数器 | 对此对象执行删除的次数。    |
| LastDeleteDate | datetime | 计数器 | 最近一次对此对象执行删除的日期 |

### *monProcess*

| 说明                  | 数据类型      | 属性 | 说明                |
|---------------------|-----------|----|-------------------|
| ClientDriverVersion | varchar16 |    | 客户端程序使用的连接驱动程序的版本 |

### *monRepLogActivity*、*monRepScannersTotalTime* 和 *monRepSenders*

SAP ASE 16.0 及更高版本要求为 *monRepLogActivity*、*monRepScannersTotalTime* 和 *monRepSenders* 监控表启用 **activate monitoring** 配置参数，以启动监控数据收集。

### *monRepScanners*

SAP ASE 16.0 对 *monRepScanners* 监控表的 *Status* 列进行了以下更改：

- 删除值 *spawned*
- 添加值 *sync mode*、*async mode* 和 *near sync mode*

### *monRepScannersTotalTime*

SAP ASE 16.0 将 *MRPBootstrapTime* 列的名称更改为 *BootstrapTime*。

### *monSysExecutionTime*

SAP ASE 16.0 及更高版本将以下值加入到 *OperationName* 列中：

- *NetworkIO* - 报告发送和接收网络数据所用的时间
- *DeviceIO* - 报告执行磁盘 IO 操作所用的时间
- *CIPCIO* - 报告执行集群互连网络操作所用的时间（仅限 Cluster Edition）

### *monTables*

SAP ASE 16.0 及更高版本中的 *Description* 列支持 512 个字符。早期版本支持 255 个字符。

### *monTableColumns*

- SAP ASE 16.0 及更高版本中的 *Description* 列支持 512 个字符。早期版本支持 255 个字符。

- SAP ASE 16.0 及更高版本中的 Label 列支持 150 个字符。早期版本支持 50 个字符。

## 新的监控表

SAP ASE 16.0 添加了两个新的监控表。

### **monThresholdEvent**

对于 SAP ASE 记录的每个事件，monThresholdEvent 监控表中分别有一行与之对应。

启用 **allow resource limits** 配置参数以启用资源限制集合。要使该监控表收集数据，请为其启用 **enable monitoring**、**threshold event monitoring** 和 **set threshold event max messages** 配置参数。

monThresholdEvent 是有态历史监控表（请参见《性能和调优指南：监控表》）。通过 **threshold event max messages** 配置参数确定 monThresholdEvent 存储的事件数。

### 列

| 名称             | 数据类型        | 属性   | 说明                                                                                                  |
|----------------|-------------|------|-----------------------------------------------------------------------------------------------------|
| SPID           | int4        |      | 服务器进程 ID。                                                                                           |
| InstanceID     | int1        |      | 集群内实例的 ID。                                                                                          |
| KPID           | int4        |      | SAP ASE 内核进程 ID。                                                                                    |
| KTID           | int4        |      | 内核任务的 ID。                                                                                           |
| ServerUserID   | int4        |      | 执行此 SQL 文本的用户的服务器用户标识符 (SUID)。ServerUserID 与 syslogins.suid 中的值相符。使用 <b>suser_name</b> 函数可以获取相应的名称。 |
| FamilyID       | int4        | NULL | 父进程的 spid。                                                                                          |
| Login          | varchar(30) | NULL | 登录用户名。                                                                                              |
| Application    | varchar(30) | NULL | 应用程序名。                                                                                              |
| HostName       | varchar(30) | NULL | 客户端主机名。                                                                                             |
| ClientName     | varchar(30) | NULL | 通过 <b>set clientname</b> 设置的客户端名称。                                                                  |
| ClientHostName | varchar(30) | NULL | 由应用程序设置的 <b>clienthostname</b> 属性的值。                                                                |

| 名称             | 数据类型         | 属性      | 说明                                  |
|----------------|--------------|---------|-------------------------------------|
| ClientAppName  | varchar(30)  | NULL    | 由应用程序设置的 <b>clientappname</b> 属性的值。 |
| ClientIP       | varchar(64)  | NULL    | 客户端的 IP 地址。                         |
| Command        | varchar(30)  | NULL    | 进程或进程当前执行的命令的类别。                    |
| DBID           | int4         |         | 当前正在由进程使用的数据库的唯一标识符。                |
| DBName         | int4         | NULL    | 运行进程的数据库的名称                         |
| ProcedureID    | int4         |         | 过程的唯一标识符。                           |
| BatchID        | int4         |         | 包含正在执行的语句的 SQL 批处理的唯一标识符。           |
| LineNumber     | int4         |         | SQL 批处理中当前语句的行号。                    |
| BlockingSPID   | int4         | 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）。      |
| TempDbObjects  | int4         | Counter | 进程所创建的临时表的总数。                       |
| RangeID        | int2         |         | 限制范围 ID。                            |
| LimitType      | varchar(30)  |         | 限制类型。                               |
| LimitID        | int2         |         | 限制标识符。                              |
| LimitValue     | int4         |         | 违反的限制值。                             |
| Enforced       | int1         |         | 确定是否在执行查询前或执行查询期间强制应用限制。            |
| Action         | int1         |         | 超出限制时执行的操作。                         |
| Scope          | int1         |         | 限制范围。                               |
| ReportDatetime | datetime     |         | 因发生限制冲突而发出报告的日期和时间。                 |
| SQLText        | varchar(255) |         | 事件的 SQL 文本。                         |

## 实用程序

SAP ASE 16.0 版中引入了对某些现有实用程序的更改。

通过特定用户将 SAP ASE 安装到目录中时，其他用户可能由于缺少对 \$SYBASE 目录的写入权限而无法进行使用。



SAP ASE 16.0 允许您指定其他用户，以便在服务器安装期间或安装结束后配置新服务器，从而允许多个用户使用所安装的同一个人 SAP ASE 目录。

为支持这项新功能，SAP ASE 16.0 增加了 `-D data_directory` 选项，以便实用程序能够指定除缺省目录以外的目录。

在 UNIX 上：

- **auditinit**
- **sqlloc[res]**
- **sqlupgrade[res]**
- **srvbuild[res]**
- **updatease**

在 Windows 上：

- **auditinit.exe**
- **sybatch.exe**

请参见《实用程序指南》中的单个实用程序。

## **ddlgen**

在 SAP ASE 16.0 中，**ddlgen** 实用程序增加了新功能。

**ddlgen** 增加了对以下内容的支持：

- 创建完全加密数据库
- 创建数据库加密密钥

**ddlgen** 增加了针对 **-TEK** 参数使用以下值进行透明数据库加密的支持：

- **-XOCE** – 仅通过列加密密钥生成 DDL。
- **-XOMK** – 仅通过主密钥或双主密钥生成 DDL。
- **-XOCE** – 仅通过数据库加密密钥生成 DDL。

### **另请参见**

- 完全数据库加密和系统更改（第 71 页）
- `create archive database` 用于完全数据库加密（第 108 页）
- `dbencryption_status`（第 101 页）
- `sp_helpdb`（第 146 页）
- `sp_encryption`（第 145 页）
- `sybmigrate`（第 170 页）
- 更改过的系统表（第 160 页）
- `alter database` 用于完全数据库加密（第 102 页）
- `create database` 用于完全数据库加密（第 108 页）

- create encryption key (第 111 页)
- drop encryption key (第 136 页)

## **sybmigrate**

SAP ASE 16.0 版针对 sybmigrate 实用程序添加了新功能。

包括:

- 您可以使用 **sybmigrate** 实用程序按照迁移非加密数据库的方法迁移完全加密数据库。但是,要使用与加密源数据库所用的数据库加密密钥 (DEK) 加密目标数据库,则必须在迁移源数据库前执行下列操作:

1. 使用 **ddlgen** 为主密钥、双主密钥以及 DEK (这些密钥均位于 master 数据库中) 生成 DDL:

```
$SYBASE/ASE-16_0/bin/ddlgen -Username -Ppwd -Shostname:port -
TEK -Nmaster -Dmaster -XOD
$SYBASE/ASE-16_0/bin/ddlgen -Username -Ppwd -Shostname:port -
TEK -Ndualmaster -Dmaster -XOD
$SYBASE/ASE-16_0/bin/ddlgen -Username -Ppwd -Shostname:port -
TEK -Ndekname -Dmaster -XOD
```

**注意:** 如果仅通过主密钥加密 DEK, 则无需为双主密钥生成 DDL。

2. 对目标服务器执行生成的 DDL, 以便其生成相同原始密钥值的 DEK。
- 您可在 **source\_ase** 或 **target\_ase** 属性中添加 **ssl** 关键字, 从而能够连接到启用 SSL 的服务器。语法为:

- 资源文件模式 -

```
[server]
source_ase=ssl : host_name : port_number
source_ase_login=sa
source_ase_password=
```

- GUI 模式 - 在服务器文本字段中添加 “ssl : host\_name : port\_number”

### 另请参见

- 完全数据库加密和系统更改 (第 71 页)
- create archive database 用于完全数据库加密 (第 108 页)
- dbencryption\_status (第 101 页)
- sp\_helpdb (第 146 页)
- sp\_encryption (第 145 页)
- ddlgen (第 169 页)
- 更改过的系统表 (第 160 页)
- alter database 用于完全数据库加密 (第 102 页)
- create database 用于完全数据库加密 (第 108 页)
- create encryption key (第 111 页)
- drop encryption key (第 136 页)

## **sybrestore**

**sybrestore** 实用程序为在主数据库损坏后恢复 SAP ASE 服务器提供支持。

已为交互和非交互模式添加其它参数：

### 交互模式

```
sybrestore
  [-J character set ]
  [-R Restore from master database corruption ]
  [-d dump directory ]
  [-s list system databases except master database ]
  [-v version ]
  [-z language ]
  [-o Log output]
```

### 非交互模式

```
sybrestore
  [-o Log output]
```

请参见《实用程序指南》中的 **sybrestore**。

## 全局变量

---

SAP ASE 16.0 版增加了全局变量。

| 全局变量                       | 说明                 |
|----------------------------|--------------------|
| <b>@@trigger_<br/>name</b> | 返回当前处于执行状态的触发器的名称。 |

| 全局变量                  | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>@@tranrollback</b> | <p>返回遇到的回退的类型（如果有）。如果返回值：</p> <ul style="list-style-type: none"> <li>• &lt; 0 - 表示服务器引起的多语句事务隐式回退。 <b>@@tranrollback</b> 将存储导致隐式事务回退的错误号负值。</li> <li>• 0 - 表示当前处于活动状态的事务的本次会话未发生隐式回退。</li> <li>• &gt; 0 &lt; 10 - 表示最近发生的事务回退是由用户通过下列其中一个 SQL 命令发出 <b>rollback</b> 而引起的： <ul style="list-style-type: none"> <li>• <b>rollback tran</b>：在 SQL 批处理、过程或触发器中</li> <li>• <b>rollback trigger</b>：在触发器范围之外。</li> </ul> <b>@@transtate</b> 的返回值将说明用户发出了哪个 <b>rollback</b> 命令： <ul style="list-style-type: none"> <li>• 1 - 表示用户发出的是显式 <b>rollback tran</b> 命令</li> <li>• 2 - 用户发出的是 <b>rollback tran to savepoint</b>。事务仍处于活动状态。</li> </ul> </li> <li>• &gt; 100 - 表示最近发生的事务回退是基于单语句事务调用的。 <b>@@transtate</b> 将存储引起语句回退的错误号。</li> </ul> <p>直到发出下一个 <b>rollback tran</b> 或 <b>commit tran</b> 命令之后，SAP ASE 才会更改 <b>@@tranrollback</b> 的负值，表示会话已遇到隐式事务回退。在成功应用下一个 <b>rollback tran</b> 或 <b>commit tran</b> 后，SAP ASE 会将 <b>@@tranrollback</b> 的值重置为 0。本示例结束时，<b>@@tranrollback</b> 的值为 0：</p> <pre> set chained on go &lt;... Execute a DML statement ...&gt; if (@@error != 0) and (@@tranrollback &lt; 0) begin     rollback tran end go </pre> |