



管理：内存行级版本控制

---

# **SAP Sybase IQ 16.0 SP03**

文档 ID: DC02009-01-1603-01

最后修订日期: 2013 年 12 月

© 2013 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它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

关于内存行级版本控制 .....	1
内存行级版本控制术语 .....	1
内存行级版本控制体系结构 .....	2
内存行级版本控制限制 .....	3
行级版本控制 (RLV) 存储库 .....	4
RLV 存储库合并 .....	4
RLV 存储库分区 .....	5
表碎片和子碎片 .....	5
RLV 存储库持久性和耐久性 .....	6
RLV 存储库备份/恢复 .....	6
RLV Dbspace .....	7
RLV 存储库持久性日志 .....	7
数据和事务管理 .....	8
数据修改语言 (DML) .....	8
事务管理 .....	10
锁管理 .....	11
版本管理 .....	14
查询和 RLV 存储库 .....	15
行级版本控制对查询的影响 .....	15
RLV 表的 QP 输出详细信息 .....	16
配置内存行级版本控制 .....	19
配置前提条件 .....	19
了解 RLV 内存初始要求和使用情况 .....	20
配置 RLV 内存 .....	21
配置 RLV Dbspace .....	21
创建 RLV Dbspace .....	21
RLV 存储库允许的 ALTER DBSPACE 语法 .....	22
删除 RLV Dbspace .....	25
在 RLV Dbspace 中添加文件 .....	25
从 RLV Dbspace 中删除文件 .....	26
针对表配置 RLV 存储库 .....	26

使用 RLV 存储库设置创建新表 .....	27
针对现有表启用或禁用 RLV 设置 .....	27
为表配置缺省存储库 .....	28
配置快照版本控制 .....	28
行级快照版本控制 .....	29
教程：在表中使用行级版本控制 .....	31
<b>将 RLV 存储库合并到 IQ 主存储库 .....</b>	<b>33</b>
前台自动合并 .....	33
设置合并触发阈值 .....	34
运行手动合并 .....	35
查看合并历史记录 .....	36
IQMSG 文件中记录的合并阶段 .....	37
合并后的表碎片 .....	38
教程：在表中使用行级版本控制 .....	38
<b>监控锁和死锁 .....</b>	<b>41</b>
教程：监控意图写锁 .....	41
教程：监控行级锁 .....	44
教程：监控死锁 .....	47
在 Interactive SQL 中创建死锁报告事件 .....	49
<b>管理 RLV 存储库中的阻塞 .....</b>	<b>51</b>
启用连接阻塞 .....	51
禁用连接阻塞 .....	52
设置阻塞超时阈值 .....	52
事务阻塞死锁 .....	53
教程：监控阻塞 .....	54
<b>管理 RLV 存储库内存 .....</b>	<b>57</b>
配置 RLV 存储库内存大小 .....	57
监控 RLV 内存使用情况 .....	58
<b>附录：RLV 存储库疑难解答 .....</b>	<b>61</b>
RLV 存储库内存不足 .....	61
无法转换成 Multiplex .....	62
无法在 Multiplex 中创建 RLV Dbspace .....	62
RLV Dbspace 已存在 .....	62
无法将 RLV Dbspace 设置为只读 .....	63

无法在 RLV Dbspace 中创建表 .....	63
无法为表启用 RLV 存储库 .....	63
无法在启用 RLV 的表中使用外键 .....	64
无法在启用 RLV 的表中使用索引类型 .....	64
进行表级修改前需要合并 .....	64
无法执行 RLV 存储库合并 .....	65
RLV 存储库合并已在进行中 .....	65
在当前事务中无法打开请求的对象进行写入 .....	66
事务似乎挂起 .....	66
失败的 RLV 恢复 .....	66
<b>附录: SQL 参考 .....</b>	<b>69</b>
数据库选项 .....	69
AGGREGATION_PREFERENCE 选项 .....	69
ALLOW_SNAPSHOT_VERSIONING 选项 .....	70
BASE_TABLES_IN_RLV_STORE 选项 .....	71
BLOCKING 选项 .....	71
BLOCKING_TIMEOUT 选项 .....	72
ENABLE_ASYNC_IO 选项 .....	72
LOG_DEADLOCKS 选项 .....	73
RV_AUTO_MERGE_EVAL_INTERVAL 选项 .....	73
RV_MAX_ACTIVE_SUBFRAGMENT_COUNT 选项 .....	74
RV_MERGE_NODE_MEMSIZE 选项 .....	75
RV_MERGE_TABLE_MEMPERCENT 选项 .....	75
RV_MERGE_TABLE_NUMROWS 选项 .....	76
RV_RESERVED_DBSPACE_MB 选项 .....	76
SNAPSHOT_VERSIONING 选项 .....	77
过程 .....	77
sa_conn_info 系统过程 .....	78
sa_report_deadlocks 系统过程 .....	80
sa_server_option 系统过程 .....	81
sp_iqcolumn 过程 .....	92
sp_iqconnection 过程 .....	94
sp_iqdbsize 过程 .....	97

sp_iqdbspace 过程 .....	99
sp_iqfile 过程 .....	102
sp_iqlocks 过程 .....	104
sp_iqmergerlvstore 过程 .....	107
sp_iqrlvmemory 过程 .....	108
sp_iqspaceinfo 过程 .....	109
sp_iqspaceused 过程 .....	110
sp_iqstatistics 过程 .....	112
sp_iqstatus 过程 .....	115
sp_iqsysmon 过程 .....	118
sp_iqtable 过程 .....	138
sp_iqtablesize 过程 .....	142
sp_iqtransaction 过程 .....	143
sp_iqwho 过程 .....	146
服务器启动选项 .....	149
-iqrlvmem start_iq 服务器选项 .....	150
SQL 语句 .....	150
ALTER DBSPACE 语句 .....	150
ALTER TABLE 语句 .....	154
CREATE DBSPACE 语句 .....	169
CREATE TABLE 语句 .....	173
DELETE 语句 .....	188
DROP 语句 .....	190
INSERT 语句 .....	193
LOAD TABLE 语句 .....	201
LOCK TABLE 语句 .....	217
TRUNCATE 语句 .....	220
UPDATE 语句 .....	221
视图 .....	225
SYSIQDBSPACE 系统视图 .....	226
SYSIQRLVMERGEHISTORY 系统视图 .....	226
SYSIQRVLOG 系统视图 .....	227
SYSIQTAB 系统视图 .....	227
索引 .....	<b>229</b>

# 关于内存行级版本控制

通过 SAP® Sybase® IQ 的内存行级版本控制 (RLV)，多个用户可同时修改同一个表，用户可等待事务锁而不是必须重试，并且混合存储模式将优化数据写访问而不会牺牲读访问性能。

- **并发表写入** - 在早期版本的 SAP Sybase IQ 中，传入的流数据必须是成批的且是顺次运行的，这产生了转换开销并导致数据可用性方面的延迟。通过内存行级版本控制，IQ 服务器允许对表进行并发的低延迟修改。这意味着多个连接可以修改同一个表，只要它们正在添加或修改不同的行就可以。
- **阻塞和锁定** - 在使用早期版本的 SAP Sybase IQ 时，事务将锁定表，这样导致在事务开放期间将阻塞所有其它连接对表执行写操作。这些连接必须通过某种形式的循环来实现重试逻辑，这会影响到性能。对于行级的版本管理，内存行级版本控制支持多版本并发控制 (MVCC)。各连接可以等待锁（在表中或单个行中），从而消除了重试的必要性。
- **混合存储库** - 内存行级版本控制在 SAP Sybase IQ 中引进了行级版本控制 (RLV) 存储库。新的 RLV 存储库与现有的磁盘 IQ 主存储库合并以提供混合存储机制，此机制不但具备内存存储库的卓越性能和低延迟，还拥有磁盘存储库的稳健高性能和扩展能力。在优化写入的 RLV 存储库中，实现了即时数据修改（装载表/插入/更新/删除）。RLV 存储库通过异步数据传输定期合并到优化读取的 IQ 主存储库中。这样，通过索引即可访问 IQ 表中的大部分数据，并可体验到所期望的 IQ 查询性能。

## 内存行级版本控制术语

---

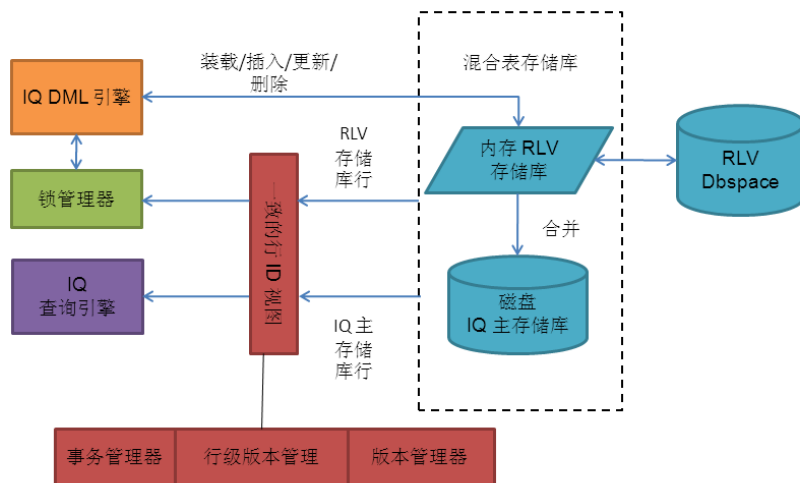
在描述内存行级版本控制时，特定术语的定义非常有用。

- **数据定义语言 (DDL)** - 指的是创建或修改表模式的 SQL 命令，例如 **CREATE TABLE**、**ALTER TABLE**、**DROP TABLE**。
- **数据修改语言 (DML)** - 指的是创建或修改表中数据的 SQL 命令，例如 **INSERT**、**LOAD**、**UPDATE**、**DELETE** 和 **TRUNCATE**。
- **多版本并发控制 (MVCC)** - 一种并发控制机制，可提供稳定的只读版本，以便写入程序不会阻塞同一个表的读取程序。
- **行级版本控制 (RLV)** - 一种 MVCC 版本控制方法，针对写入事务对表行进行逻辑上的版本控制，并允许对同一表的不同行执行并发写入操作。每次写入程序提交事务时，服务器都会创建所更新行的新版本，从而导致创建另一个不同的行级版本。启用 RLV 的表是允许行级版本控制的表。
- **RLV 存储库** - 优化写入的内存存储库，与现有的优化读取的磁盘 IQ 主存储库共同实现内存行级版本控制。

- **RLV 存储库合并** - 将 RLV 存储库中每个表的内存数据定期异步传输到 IQ 主存储库。该合并自动执行，但也可以手动触发。仅合并已提交的事务。
- **快照** - 建立对象稳定版本的方法，在事务开始时确定。
- **表级版本控制 (TLV)** - 一种 MVCC 版本控制方法，针对写入事务对整个表进行逻辑上的版本控制，但不允许对同一表执行并发写入操作。每次写入程序提交事务时，服务器都会创建整个表的新版本，从而导致创建另一个不同的表级版本。TLV 表是未启用行级版本控制的表。

## 内存行级版本控制体系结构

RLV 存储库和 IQ 主存储库一同提供了混合的表存储空间，这对于含有并发事务的表实现了行级快照隔离。服务器在查询和处理数据时会跟踪数据位置，但从逻辑上说，这些数据位于同一个（混合）存储库中。



- **版本管理器** - 版本管理器通过表级版本以确保行 ID 视图前后一致。
- **事务管理器** - 仅将 RLV 事务写入 RLV 存储库中。表的单个 TLV 写入事务将一直阻塞对该表的所有 RLV 写入操作，直到 TLV 事务终止。
- **事务日志** - RLV 存储库事务日志（持久性日志）会跟踪所有存储于内存中的新的、已修改的数据并保证这些数据的持久性。



## 内存行级版本控制限制

内存行级版本控制功能在聚合首选项、表类型和数据类型等方面存在一些限制。

要素	限制
服务器类型	仅支持单服务器配置：不能在 Multiplex 配置中为行级版本控制启用表。
表类型	RLV 存储库仅支持 SAP Sybase IQ 基表：不支持目录表、临时表以及全局临时表。
数据类型	<p>启用 RLV 的表不支持 LONG BINARY (LOB) 和 LONG VARCHAR (CLOB) 数据类型。</p> <ul style="list-style-type: none"> <li>不能将 LOB 列添加到启用 RLV 的表。</li> <li>不能对含有 LOB 列的 TLV 表启用 RLV 存储库。</li> </ul>
索引类型	启用 RLV 的表不支持 TEXT 索引和 WORD 索引。
约束类型	<p>启用 RLV 表或此类表与 TLV 表的合并表不支持外键约束。不支持参照完整性。</p> <ul style="list-style-type: none"> <li>不能将外键约束添加到启用 RLV 的表。</li> <li>不能对已定义外键约束的 TLV 表启用 RLV 存储库。</li> <li>对于 TLV 表，如果其它表含有引用该表的外键约束，则不能对该表启用 RLV 存储库。</li> <li>修改 TLV 表时，不能将外键引用添加到 RLV 表。</li> </ul>
SQL 语句	<ul style="list-style-type: none"> <li>启用 RLV 的表不支持 LOCK TABLE 语句。</li> <li>如果其中一个 CREATE INDEX 语句指定了一个启用 RLV 的表，则 BEGIN PARALLEL IQ...END PARALLEL IQ 语句会导致出错。</li> <li>如果 PARTITION 或 SUBPARTITION 子句指定了一个启用 RLV 的表，则 TRUNCATE 语句会导致出错。</li> <li>如果将 MAX_QUERY_PARALLELISM 数据库选项设置为 1，则 LOAD TABLE 语句的 SKIP 子句将无效。</li> </ul>
数据库选项	如果查询指定了一个启用 RLV 的表，则 REVERT_TO_V15_OPTIMIZER='ON' 会导致出错。

### 另请参见

- 行 ID 稳定性 (第 14 页)

## 行级版本控制 (RLV) 存储库

---

行级版本控制允许多个事务同时修改同一个表的不同行。优化写入的内存 RLV 存储库在实现对 IQ 基表的并发读写访问方面补充了 SAP Sybase IQ 主存储库。

通过 IQ 服务器可以基于连接来配置对 IQ 基表的单写入程序或多写入程序并发访问。多个连接可以修改同一个表，只要这些连接正在添加或修改不同的行就可以。只有启用了 RLV 存储库的表适合进行多写入程序并发访问。

RLV 存储库提供了以下优点：

- 低延迟写入，并最大程度降低索引和压缩开销，从而实现高效的内存读/写操作
- 通过行级版本控制和最低限度的锁争用实现高效的并发更新
- 提高以内存读取操作为主的列扫描的性能

RLV 存储库可记录 IQ 表中的增量 DML 并自动将这些更改合并到 IQ 主存储库中。合并的 RLV 存储库和主存储库对写入进行了优化以实现高效的 DML 和低延迟数据访问，同时仍对大部分数据保证了读取优化的磁盘查询性能。

传统的表级版本控制 (TLV) 表仅驻留在 IQ 主存储库中，启用了行级版本控制 (RLV) 的表则同时驻留在 IQ 主存储库和 RLV 存储库中。

## RLV 存储库合并

RLV 存储库是一个独立的自我管理型内存存储库，将自动合并到 IQ 主存储库（定期合并或在达到阈值时合并）。

RLV 合并过程针对每个表将各行从存储在 RLV 存储库中的已提交行级事务移至 IQ 主存储库。合并的数据在 IQ 主存储库中成为同一表的新表级版本的一部分。

阻塞合并将等待表的所有写入事务完成，然后阻塞所有写入操作，直到完成合并。在非阻塞合并情况下，允许在激活合并时对表继续执行现有的写入操作和新写入操作。

可通过以下三种方式之一触发合并过程：

1. 通过使用 `sp_iqmergerlvtable` 存储过程手动合并。例如：

```
call sp_iqmergerlvstore('BLOCKING', 'my_table', 'DBA')
```

，它对启用 RLV 的表 "DBA.my\_table" 运行一个阻塞合并。

2. 在对启用 RLV 的表执行 DDL 或表级 DML 语句时自动合并。例如：

```
alter table DBA.my_table add c2 int
```

，它对启用 RLV 的表 "DBA.my\_table" 强制执行阻塞合并，然后将新列 "c2" 添加到此表。

3. 在达到资源阈值时自动合并（例如，给定表的 RLV 存储库中的最大行数或 RLV 内存存储库所允许的最大内存量）。

RLV 合并失败情况并不具有代表性。唯一合法的 RLV 合并失败可能是由于 IQ 主存储库的 `dbspace` 不足或发生存储错误（硬件或权限问题）而导致。如果 RLV 合并失

败，将回退事务。届时将撤消对 IQ 主存储库 TLV 版本的所有更改。服务器通过事务执行 DML 时将使用相同过程。

### 另请参见

- 将 RLV 存储库合并到 IQ 主存储库（第 33 页）

### RLV 合并阶段

RLV 合并分为五个阶段执行。

- **开始** - 服务器执行自动提交，开始一个新事务并准备合并。
- **删除** - 服务器应用 IQ 主存储库中记录的所有删除事务。从表的 RLV 存储部分删除的行不会合并，因为它们已在内存中删除。任何在启用 RLV 前删除的已提交 IQ 主存储库行仅由 RLV 存储库记录为已删除。该合并将针对要创建的新 IQ 主存储表级版本应用删除项。
- **更新** - 服务器应用 IQ 主存储库中记录的所有更新事务。从 IQ 主存储表更新的行会将新值存储在 RLV 存储库中。更新后的新值将在合并期间应用于 IQ 主存储库。
- **插入** - 服务器将新 RLV 存储库的插入行传输到 IQ 主存储表中。
- **结束** - 服务器完成合并并提交新的 TLV 版本（或回退）。

合并将被记录在数据库 IQMSG 文件和 SYSIQMERGERLVHISTORY 系统视图中。

### RLV 存储库分区

所有 IQ 基表（无论是未分区表、范围分区表、散列分区表还是范围散列分区表）都可以启用行级版本控制，但内存中的 RLV 存储部分不进行分区。

实际分区发生在 RLV 存储库与（分区）IQ 主存储库合并期间。此时，表行被安置在相应的范围分区 dbspace 中。每个分区表在 RLV 存储库中都有一个对应的伪列，其中包含了分区信息。查询引擎使用这些伪列来支持分区语义。

### 表碎片和子碎片

表碎片是存储数据的位置。碎片还可进一步细分为子碎片。一个表碎片中可能包含一个或多个子碎片。

数据存储在子碎片中的数据块中。有两种类型的数据块：

- **可变长度的数据块** - 用于存储可变长度的数据类型，如 varchar。可变长度数据块的大小为 512KB。
- **固定长度的数据块** - 用于存储固定长度的数据类型，如 int 和 big int。固定长度的数据块的大小为 16MB。

每个子碎片具有一个独立的存储数据结构，各个子碎片之间只共享字典。活动子碎片中可以包含多个数据块，但在向各列中插入数据时，每次只能有一个数据块处于活动状态。

始终使用活动数据块向表中插入数据。如果活动数据块已满，系统将在子碎片中自动创建一个新的活动数据块。如果在并行插入期间出现争用，将会创建一些新的子碎片，每个子碎片中都会包含一个活动数据块。

内存分配由插入操作触发，插入操作需要创建新的子碎片或者在子碎片中创建附加数据块。将子碎片插入到活动数据块中则不需要分配附加内存。直至需要附加子碎片或数据块时，才进行附加内存分配。这可能会导致内存分配在批量装载期间出现周期性内存峰值，而不是连续、平缓上升。

## RLV 存储库持久性和耐久性

---

RLV 存储库中的事务具有最大程度的耐久性。在发生系统崩溃的情况下，也能够保证恢复已提交的事务。

RLV 存储库中的数据不会写入到磁盘，但会一直保留在内存中，直到被合并到 IQ 主存储库。

支持 RLV dbspace 的 I/O 系统的性能是决定 RLV 存储库的提取和提交性能的主要因素。持续的写入吞吐量对于大容量数据提取十分重要。写入延迟对于提交性能十分重要。在同时将数据提取到多个启用 RLV 的表中时，随机写入性能十分重要。

直到将整个事务（包括提交状态）写入稳定的存储库，才会确认应用程序的提交请求。

每个启用 RLV 的表都有自己的逻辑持久性日志。这些日志的存放空间仅来自于 RLV dbspace。日志空间在事务执行期间被消耗并通过合并得以释放。合并结束时，表的日志将截断到合并开始时最早时期的未决事务。跨越多次合并的长时间运行的未决事务有可能会阻止截断日志空间。您可通过 `sp_iqtablesiz` 存储过程来监控用于表的 RLV 日志空间量。

由于在恢复过程开始时未提交数据不可能存在于数据库中，因此恢复是一个逻辑上的面向事务的过程，而不是实体上的面向页面的过程。RLV 恢复涉及的是插入已提交数据，而不是实际修改页面。

### 另请参见

- 将 RLV 存储库合并到 IQ 主存储库（第 33 页）
- 管理 RLV 存储库内存（第 57 页）

## RLV 存储库备份/恢复

---

备份并恢复 RLV 存储库是常规备份过程的一部分。

RLV dbspace 是一个持久性 dbspace。它的内部结构与其它受支持的 dbspace 相同。因此，它会完全集成到 SAP Sybase IQ dbspace 管理系统中。

多种系统过程都可以报告 RLV 存储库持久性 dbspace 中的元数据。此外，它被完全集成到现有 SAP Sybase IQ 备份和恢复体系结构中。在备份恢复中，dbspace 是常规备份的一部分并进行恢复。在常规恢复中，首先恢复目录存储库和 IQ 主存储库，然

后服务器将恢复内存部分（启用 RLV 的表）。RLV 恢复将针对已在内存中提交数据的启用 RLV 的表来执行。

## RLV Dbspace

RLV 存储库需要一个 RLV dbspace 来实现持久性日志记录。

RLV dbspace 用于存放启用 RLV 的表的持久性日志。

除非已配置 RLV dbspace，否则不能对任何表启用 RLV 存储库。

### 另请参见

- 配置 RLV Dbspace（第 21 页）

### RLV Dbspace 限制

每个数据库仅有一个 RLV dbspace，但可具有多个 dbfile。

当数据库中存在启用 RLV 的表时，不可以删除 RLV dbspace 文件或将其设置为只读。要删除 RLV dbspace 文件，必须首先删除所有启用 RLV 的表或禁用 RLV。

与其它 dbspace 不同，RLV dbspace 不允许条带化。

## RLV 存储库持久性日志

RLV 存储库的异步预写持久性日志独立于数据库持久性日志。它会跟踪存储在 RLV 存储库中的所有新的已修改数据并使这些数据成为持久性数据。

RLV 持久性日志中包含 RLV 存储库内容的基于磁盘的副本。它存储为压缩格式，以平衡磁盘利用率和运行时恢复性能。该日志针对每个表进行编排，并专门存储在 RLV dbspace 中。它使用高效的异步 I/O 将表修改开销降到最低，并通过高效的并行处理实现重启时的快速恢复。该日志用于在服务器清理或异常关闭后再重启时恢复内存 RLV 存储库。

### 日志空间使用情况

由 `sp_iqtablesiz` 存储过程报告 RLV 日志空间使用情况。

尤其是 `sp_iqtablesiz` 的两列与 RLV 日志相关。

列名	描述
RlvLogPages	用于为此表存储 RLV 日志的 IQ 页数
RlvLogKBytes	用于为此表存储 RLV 日志的字节数 (KB)

### 另请参见

- `sp_iqtablesiz` 过程（第 142 页）

## 数据和事务管理

---

SAP Sybase IQ 中的所有表均存储在 IQ 主存储库中。此磁盘存储库启用了表级版本控制 (TLV)。如果某表启用了行级版本控制 (RLV)，磁盘版本保持稳定不变，但激活内存 RLV 存储库。执行表事务时，服务器将管理在内存中提交和存储的多个行级版本。随着时间的推移，RLV 存储库中的这些数据将与 IQ 主存储库合并。

### 数据修改语言 (DML)

RLV 存储库中每行的磁盘 I/O 明显小于 IQ 主存储库中的磁盘 I/O，这样便可以使用较小粒度的 DML。

行级快照隔离允许使用并发 DML，但表级快照隔离不允许。

通过执行插入事务（包括批量装载），新行被添加到 RLV 存储库。数据只通过合并操作插入到 IQ 主存储库。对于插入事务从不会有锁定，因为 RLV 存储库是“仅追加”存储库。插入的优势是利用了完全批量并行装载引擎。

如果执行删除事务，则存在锁定。从启用 RLV 的表中删除某行时，提交期间不会将该数据从 RLV 存储库中移除。所有提交的已删除行仍会保留在内存中，直到发生合并为止，但对查询操作是不可见的。

更新事务就是删除事务加上插入事务。

将 RLV 存储库与 IQ 主存储库合并不会释放未决事务所占用的内存，这些内存会一直占用到事务关闭。

#### 另请参见

- 表级 DML 锁定的注意事项（第 13 页）

#### DML 最佳做法

通过最佳做法将改善批量装载和大型更新方面的性能。

针对批量装载：

- 通过 RLV 存储库执行规模较小或中等的并发装载。
- 不要通过 RLV 存储库过多执行大型装载，因为超过 RLV 内存上限的装载可能会失败。

即使一次规模相当大的装载没有超出 RLV 内存上限（并且未失败），但在装载期间可能会对其它已提交的 RLV 数据触发合并。这可能会影响装载性能。

针对大型更新：

- 使用启用了表级版本控制的连接对启用 RLV 的表执行大型更新。
- 不要通过 RLV 存储库执行规模非常大的更新，因为更新操作会将磁盘上的行复制到 RLV 内存存储库以进行修改。

## 约束

要使 IQ 主存储库和 RLV 存储库的行为保持一致，需要在装载、插入和更新语句的执行期间检查各约束。

系统支持主键和唯一索引，但在修改启用 RLV 的表期间需要额外的内存和处理。要在启用 RLV 的表的修改期间优化性能，需要最大程度地减少这些索引类型。启用 RLV 的表不支持参照完整性和外键约束。

如果检测到违反约束，会导致 DML 命令失败。装载命令会提供一个 IGNORE CONSTRAINT 子句，通过该子句可以在装载命令中止前报告指定数目的约束违规并忽略这些违规。

## 表约束

RLV 存储库装载与 IQ 主存储库装载使用相同的表约束评估机制，并且在表约束评估方面具有相同的性能特性。

表约束允许针对每个插入的行进行表达式求值。例如：

```
col1 >= col2, col1 NOT NULL
```

这些表达式可能只涉及要装载的行中的列。

## 另请参见

- CREATE TABLE 语句 (第 173 页)
- ALTER TABLE 语句 (第 154 页)

## 唯一约束和主键约束

RLV 存储库装载、插入以及更新以特定方式支持主键约束和唯一约束。

由于 RLV 存储库不维护次级索引，所以强制实施需要单独的唯一值检查过程。此外，由于 RLV 表写入未采用独占表锁，因此在当前连接执行其操作期间，其它连接可以插入、更新或删除新的行值并随后提交。

尽管在事务快照之后提交的 RLV 修改对于当前连接不可见，但在可提交任何新值之前，会在唯一值和主键检查期间考虑由于这些修改而添加或删除的任何值，以避免约束违规。

唯一值和主键值检查有两个阶段。

1. 将对照 RLV 存储库中的值检查新值的唯一性。当值插入到 RLV 存储库时，将对照 RLV 存储库中的已提交值和未提交值检查每个插入值的唯一性（采用“首个更新优先”方法）。在唯一性检查中不会考虑未提交的删除项，除非它们是当前事务的一部分。
2. 系统对照 IQ 主存储库值检查新值。在将值插入到 RLV 存储库之前，将对照 IQ 主存储库索引检查每个值的唯一性（排除对 RLV 存储库中的事务可见的所有已删除和已更新的 IQ 主存储库行）。

### 参照完整性约束

RLV 存储库不支持参照完整性约束。

在 IQ 主存储库中，参照约束的强制执行机制将通过在事务持续期间对外表采用意图写锁来检查当前是否没有修改参照完整性约束所涉及的任何外表。这种参照并发检查由全局 IQ 内存目录处理。但是 RLV 存储表不支持在参照完整性强制执行中采用全局意图写锁。

启用 RLV 的表中的参照完整性关系还需要从访问表级版本控制外表的 RLV 事务获取全局表锁。

出于这些原因不支持参照完整性约束。

## 事务管理

事务管理器支持对 RLV 存储库的事务访问。事务管理器通过版本管理器来提供表级版本控制和行级版本控制支持。

事务管理器还适用于 RLV 合并以实现阻塞合并和非阻塞合并。在合并提交或回退前，阻塞合并不允许事务访问启用 RLV 的表。非阻塞合并允许事务在合并开始前和结束后都存在，因此数据库对这些事务呈现的状态仍是一致的。

在提交启用 RLV 的表之后，数据将一直驻留在 RLV dbspace 中，直到发生合并。（RLV dbspace 是内存中数据的磁盘表示形式）。合并不会从未决事务（即，尚未结束的事务）中释放内存。

在 TLV 连接访问启用 RLV 的表时，表级读写访问将强制执行 RLV 存储库的合并。TLV 事务将持有一个表级写入锁，该锁会在事务的持续时间里阻塞 RLV 和其它 TLV 连接对表执行写入操作。从其它连接执行的读取操作不受影响。这种情况的一个例子是夜间装载。应用程序在夜间执行大型装载，此时无人访问该系统。与装载到 RLV 存储库并在之后手动触发合并相比，批量装载到 IQ 主存储库的效率更高一些。

### 另请参见

- 将 RLV 存储库合并到 IQ 主存储库（第 33 页）

### 事务

事务将访问在其快照中可见的行。

事务可以是简单事务、多表事务或复合事务：

- 简单事务仅涉及一个启用 RLV 的表和 RLV 存储库。日志记录完全以单个日志流形式存在。
- 多表事务涉及多个启用 RLV 的表并在多个日志流中含有记录。
- 复合事务涉及传统 IQ 表（启用表级版本控制）、SAP Sybase SQL Anywhere® 表（在 IQ 目录存储库中）以及启用 RLV 的表，这些表共存于同一事务中。服务器先提交 TLV 表，随后提交 SQL Anywhere 表，最后提交启用 RLV 的表。



## 锁管理

行锁及其前提条件意图写锁可确保行级上并发事务之间的一致性。例如，事务可锁定特定的行以防止另一个事务对其进行更改。事务将意图写锁放置在表自身和想要修改的表行上，以防止竞争的行级快照版本控制事务和表级快照版本控制事务之间发生冲突。

启用 **RLV** 的表对已更新行和已删除行使用行锁；当通过具有行级快照版本控制的读写事务进行访问时，则使用意图写锁。而当通过具有表级快照版本控制的读写事务进行访问时，则使用 **IQ** 写表锁。

对启用 **RLV** 的表进行 **DDL** 更改时需要一个独占表级锁；写入连接在该表中有一个独占锁。对启用 **RLV** 的表进行 **DML** 更改时，首先删除意图写锁以阻塞表级版本控制事务锁定该表，然后删除行级锁以防止其它行级版本控制事务对这些行执行写入操作。

- **行锁** - 表行写入锁允许持有者向锁定行的任意列写入内容。一次只能存在此锁的一个持有者。意图写锁是前提条件，您必须先持有意图写锁，这样锁管理器才能为您授予行锁。
- **意图写锁** - 表意图写锁指示您以后要向表行中写入内容。一个意图写锁可由多个请求者持有。

意图写锁与表写入锁和表独占锁相冲突。此冲突会阻止表级快照版本控制事务向表中写入内容或执行 **DDL** 操作，一直到锁管理器释放表的所有意图写锁。

### 另请参见

- 模式锁（第 11 页）
- 行锁（第 12 页）
- 意图写锁（第 12 页）
- 行级 **DDL** 锁定的注意事项（第 14 页）
- 表级 **DML** 锁定的注意事项（第 13 页）
- 监控锁和死锁（第 41 页）

### 模式锁

**IQ** 主存储库中的表锁在表的所有行上放置锁，而模式锁在表的模式上放置锁。

模式锁确保使用表的事务不受由其它连接发起的模式更改的影响。例如，模式锁可防止 **ALTER TABLE** 语句在表正被另一连接上打开的游标读取时从该表中删除列。读取程序和写入程序均使用模式锁。

可以修改模式锁，以实现独占访问。仅当不存在任何其它模式锁时，才能授予独占访问权限。也就是说，表中不存在任何其它读取程序和写入程序。所有 **DDL** 语句都将在允许执行前获取独占模式锁。任何时候只有一个连接可获取表的独占模式锁 - 锁定该表模式的所有其它尝试都将阻塞，或失败并产生错误。

### 另请参见

- 行锁 (第 12 页)
- 意图写锁 (第 12 页)
- 表级 DML 锁定的注意事项 (第 13 页)
- 行级 DDL 锁定的注意事项 (第 14 页)

### 行锁

行锁是表行写入锁，它允许持有该锁的事务向锁定行的任意列写入内容。一次只能存在此锁的一个持有者。*意图写锁*是前提条件，事务必须先持有意图写锁，这样锁管理器才能为其授予行锁。

表行写入锁允许持有该锁的事务向锁定行的任意列写入内容。请求事务必须先持有意图写锁，然后才能授予此锁。行写入锁是独占锁，任何时候，只有一个事务能够持有某一行的写入锁。一个事务获取了写入锁后，其它事务要求锁定该行的请求都将被拒绝。

行锁只在行删除期间存在。RLV 存储库是仅附加存储库，意味着每个写入操作都会向存储库中附加一个新行。**INSERT** 语句向存储库中附加一个新行，就像执行 **UPDATE** 语句一样。RLV 存储库将 **UPDATE** 视为 **DELETE** 加上 **INSERT**。在删除某一行之前，数据库将在 **DELETE** 或 **UPDATE** 语句的上下文中删除行级锁。

### 另请参见

- 模式锁 (第 11 页)
- 意图写锁 (第 12 页)
- 表级 DML 锁定的注意事项 (第 13 页)
- 行级 DDL 锁定的注意事项 (第 14 页)
- 教程：监控行级锁 (第 44 页)

### 意图写锁

意图写锁是表写入锁，它授予事务在以后向表行中写入内容的权限。一个意图写锁可由多个请求连接持有。

如果内存中存在表的启用 RLV 部分，将始终存在意图写锁。使用 **sp\_iqlocks** 存储过程可查看意图写锁的详细信息。

意图写锁与表写入锁和表独占锁相冲突。此冲突会阻止表级快照版本控制事务向表中写入内容或执行 DDL 操作，一直到锁管理器释放表的所有意图写锁。在表级快照版本控制事务和行级快照版本控制事务连接同时向表中写入内容的情况下，意图写锁将提供同步。考虑以下情形：

连接	操作
行级快照版本控制事务 A	<ul style="list-style-type: none"> <li>• 执行向 <code>table_1</code> 的多个行中写入内容的查询。</li> <li>• 锁管理器为 <code>table_1</code> 创建一个意图写锁。</li> <li>• 锁管理器为行级 DML 更新创建多个本地意图写锁。锁管理器创建行级锁。</li> </ul>
表级快照版本控制事务 B	试图向 <code>table_1</code> 中写入内容。事务 B 被意图写锁阻塞。
行级快照版本控制事务 A	提交事务 A。将 RLV 存储库中的表更改合并到 IQ 主存储库中。意图写锁被释放。
表级快照版本控制事务 B	继续向 <code>table_1</code> 中写入内容。

### 另请参见

- 模式锁（第 11 页）
- 行锁（第 12 页）
- 表级 DML 锁定的注意事项（第 13 页）
- 行级 DDL 锁定的注意事项（第 14 页）
- 教程：监控意图写锁（第 41 页）

### 表级 DML 锁定的注意事项

如果已为 RLV 存储库启用表，则仍可针对该表发出表级快照版本控制 DML 语句。DML 引擎识别表级和行级版本控制事务，并相应地对锁进行管理。

当事务针对启用 RLV 的表发出表级快照版本控制 DML 语句时：

- 内存数据将合并到表的 IQ 主存储部分
- 意图写锁释放
- 表级 DML 语句继续执行

完成该事务后，下次连接可在行级快照版本控制事务中发出 DML 语句，从而导致在内存中重新创建表的 RLV 部分。在发出表级快照版本控制 DML 语句的当前事务完成之前，行级快照版本控制事务将被阻塞或失败。

---

**注意：** 如果已为 RLV 存储库启用表，则无法使用 `LOCK TABLE` 语句。

---

### 另请参见

- 模式锁（第 11 页）
- 行锁（第 12 页）
- 意图写锁（第 12 页）
- 行级 DDL 锁定的注意事项（第 14 页）
- `LOCK TABLE` 语句（第 217 页）

### 行级 DDL 锁定的注意事项

数据定义语言 (DDL) 对启用 RLV 的表进行的更改 (如 **CREATE INDEX**、**DROP INDEX** 和 **ALTER TABLE ADD**、**ALTER** 或 **DROP**) 需要一个独占表级锁。对于 DDL 事件, 启用 RLV 的表具有与 IQ 主存储表相同的锁定行为: 写入连接在该表中有一个独占锁。当 **BLOCKING** 设置为 ON 时, 将阻塞针对该表的所有竞争 DML 和 DDL 事务, 直至提交 DDL 更改。当 **BLOCKING** 设置为 OFF 时, 竞争事务将立即停止锁定请求。

### 另请参见

- 模式锁 (第 11 页)
- 行锁 (第 12 页)
- 意图写锁 (第 12 页)
- 表级 DML 锁定的注意事项 (第 13 页)

## 版本管理

涉及到启用 RLV 的表的事务将创建行级事务快照版本。通过行级快照, 事务可提交表行版本而不是整个表版本。

行级版本控制允许对表进行并发 DML 更改。(对某一行进行更新可能不会阻塞另一个连接对其它行的更新)。

- 对启用 RLV 的表的行级操作将创建一个行级版本。
- 对启用 RLV 的表的表级操作 (如 DDL 和表级 DML) 将创建一个表级版本。
- 未涉及启用 RLV 的表的事务将创建表级版本。

---

**注意:** 必须先启用行级事务快照版本控制, 然后才能将数据写入 RLV 存储库。

---

未决事务将持有该事务可访问的版本。如果某事务长时间运行, 则服务器将一直保留与 RLV 和 TLV 版本相关联的内存和磁盘空间, 直到该事务终止。即使在执行合并后, 直到有快照引用这些版本的事务终止, 才会释放 RLV 版本的内存。

### 另请参见

- 配置快照版本控制 (第 28 页)
- 指定快照版本控制 (第 29 页)
- 限制快照版本控制 (第 30 页)

### 行 ID 稳定性

启用 RLV 的表中的行的行 ID 可能会改变。

在完全驻留在 IQ 主存储库中的 TLV 表中, 当插入某行时, 行 ID 在该行的整个使用期限内保持稳定不变。一旦该行被删除, 就可以将行 ID 指配到新的插入行中。

在启用 RLV 的表中, 会将一个临时 RLV 行 ID 分配给所插入行。

被分配给 RLV 存储库中的行的临时行 ID 只在可访问它的每个事务的持续期间内保证稳定不变。只有在该行被提交并合并到 IQ 主存储库之后，它的行 ID 才会成为永久性 ID。

## 查询和 RLV 存储库

---

对启用 RLV 的表执行的查询同时从磁盘上的 IQ 主存储库和内存中的 RLV 存储库扫描数据。

某些查询优化由于缺少专用索引而不适用于 RLV 存储库的查询：

- IQ 索引 (HG、LF) 不涉及 RLV 存储库。
- 谓词评估的并行性有限。

通常来讲，这会产生很小的性能影响，并且与 RLV 存储库和 IQ 主存储库中的数据比例相关。

查询的 RLV 部分依赖于内存列扫描。在某些分区散列连接情况下，会根据需要创建内存索引以增强查询性能。

### 另请参见

- 行级版本控制 (RLV) 存储库 (第 4 页)
- 行 ID 稳定性 (第 14 页)

## 行级版本控制对查询的影响

某些特定情况将产生性能影响，包括含有多个不变谓词、索引以及行 ID 的查询。

- **谓词的执行顺序** - 当查询含有多个不变谓词时，在 RLV 存储库中对数据执行谓词的顺序取决于谓词的用处。RLV 存储库中的谓词执行顺序可能与 IQ 主存储库中的执行顺序不同。
- **缺少索引** - 与 IQ 主存储库不同，RLV 存储库不具备使用现有的最佳索引来执行谓词的能力。因此，在查询计划方面可能会有差异，具体取决于是否仅针对 IQ 主存储库中的数据执行查询还是针对 RLV 存储库中的相同数据执行查询。此外，有一些谓词需要特定索引提供特殊支持，在对启用 RLV 的表执行这些谓词时可能会导致出错。例如，`contains` 谓词需要 `TEXT` 或 `WORD` 索引的支持，但这两个索引在启用 RLV 的表中都不受支持。为避免由于 RLV 端缺少索引而导致性能降低，SAP Sybase IQ 可能会在检测到这些查询时创建一个即席散列索引。该查询计划将指明存储库正在使用散列索引。
- **无 DQP 支持** - 只有 Simplex 配置支持对启用 RLV 的表的查询。

---

**注意：** 查询时不要使用特定行 ID。如果使用特定行 ID (为实现连接) 并从另一个表中选择一个行 ID，则该行 ID 可能不会保持一致。

---

另请参见

- RLV 表的 QP 输出详细信息 (第 16 页)

## RLV 表的 QP 输出详细信息

通过查询计划输出详细信息的“优化注释”、“条件 1 RLV 索引”和“输出 1 RLV 索引”字段可了解在使用行级版本控制时查询的工作机制。

在叶节点的查询计划输出详细信息中:

- **优化注释** - 指明是否为 RLV 存储库中的数据创建了按需而定的散列索引。
- **条件 1 RLV 索引** - 描述将哪个索引用于谓词执行。
- **输出 1 RLV 索引** - 列出列中出现的 RLV 索引。

### *在执行中使用 Flat FP 索引的查询*

```
SELECT * from test_char  
WHERE c1>1;
```

如果运行了此查询,则在查询计划输出详细信息中,“条件 1 RLV 索引”字段将显示 FP,表示查询在谓词执行中使用了 Flat FP 索引。“输出 1 RLV 索引”字段也会显示 FP,表示 c1 中仅存在 FP 索引。

### *执行期间创建散列索引的查询*

```
SELECT * from R1KD100  
WHERE R1KD100.c1 in (SELECT R100D100.c1 FROM R100D100 WHERE  
R1KD100.c1  
= R100D100.c1)
```

如果运行了此查询,则在查询计划输出详细信息中,“优化注释”字段将指明已为 RLV 数据创建了散列索引。“输出 1 RLV 索引”字段将显示 FP, Hash,表示 c1 中存在两个索引。

### *在执行中使用先前创建的散列索引的查询*

```
SELECT * from R100D100  
WHERE c1 = 1;
```

如果在上一个查询之后运行了此查询,并且散列索引仍存在于 R100D100.c1 中,则在查询计划输出详细信息中,“条件 1 RLV 索引”字段将显示 Hash。(每当列中存在散列索引时,在对该列执行谓词期间将始终优先选择该索引,而不是 Flat FP 索引)。“输出 1 RLV 索引”字段将显示 FP, Hash,表示 c1 中存在两个索引。

### *执行期间创建分区散列索引的查询*

```
SELECT * from hash1, hash2  
WHERE hash1.c1 = hash2.c1;  
// hash1 and hash2 are hash partitioned on c1
```

假定表 hash1 和表 hash2 是均匀分区表(两者都在列 c1 中进行了散列分区),并且连接条件位于列 c1 中。如果运行了此查询,则 IQ 查询优化程序将在两个表中都创建

分区索引。在每个叶节点的查询计划输出详细信息中，“优化注释”字段将指明已为 RLV 存储库数据创建了分区索引。

**另请参见**

- 行级版本控制对查询的影响（第 15 页）





# 配置内存行级版本控制

行级版本控制允许多个事务同时修改同一个表的不同行。要利用 RLV 存储库，首先必须配置几个设置步骤。

## 1. 配置前提条件

在配置内存行级版本控制之前，除了 SAP Sybase IQ 的基本配置外，您还需要 RAM 以用于 RLV 存储库，以及一个高性能磁盘以用于 RLV dbspace。

## 2. 了解 RLV 内存初始要求和使用情况

每个 RLV 表都需要一个内存存储库来保存其 RLV 数据。存储库的内存要求在表（该表已启用 RLV）的生命周期内发生更改。

## 3. 配置 RLV 内存

除了现有 IQ 高速缓存内存外，还需配置 RLV 内存。主机必须拥有足够的内存来同时满足 RLV 内存和 IQ 主缓冲区高速缓存的需要，这两者是独立的内存池。

## 4. 配置 RLV Dbspace

RLV dbspace 中保存着启用了内存 RLV 存储库的表的持久性日志。

## 5. 针对表配置 RLV 存储库

启用了 RLV 存储库的表具有两部分：一部分驻留在 IQ 主存储库中，另一部分驻留在专用的内存 RLV 列存储库中。

## 6. 配置快照版本控制

快照版本控制用于描述数据库服务器在表中使用的版本控制访问类型：表级快照版本控制或行级快照版本控制。行级版本控制允许对启用 RLV 的表的并发写入程序访问和行级锁定。

## 7. 教程：在表中使用时级版本控制

在本教程中，将创建启用 RLV 的 dbspace 和表，插入数据，然后将 RLV 存储库合并到 IQ 主存储库中。

## 配置前提条件

---

在配置内存行级版本控制之前，除了 SAP Sybase IQ 的基本配置外，您还需要 RAM 以用于 RLV 存储库，以及一个高性能磁盘以用于 RLV dbspace。

平衡内存大小和合并频率。尽管在预期事务频繁的情况下较小的内存分配可能已足够，但要考虑到较小的内存大小可能会影响 RLV 存储库合并频率，从而影响整体性能。

- 对于批量装载，每个表使用的内存至少是单事务装载最大数据大小的两倍。
- 对于连续的/OLTP 装载，该大小取决于传入数据和事务速率。

对于 RLV 日志 dbspace，磁盘要求如下：

- 至少是 RLV 内存大小的两倍。
- 高随机访问写入性能、企业级 SSD 以及连接了 HBA 的磁盘阵列。

## 了解 RLV 内存初始要求和使用情况

每个 RLV 表都需要一个内存存储库来保存其 RLV 数据。存储库的内存要求在表（该表已启用 RLV）的生命周期内发生更改。

存储库的初始 RLV 内存要求（即第一次使用存储库时需要的内存）如下：

$\#COLS\_FIXED = \# \text{ 固定大小列} + 1$ （伪列）

$\#COLS\_VAR = \# \text{ 可变大小列}$

基本 RLV 存储库 (BRS) =  $\#COLS\_FIXED * \text{default\_fixed\_size} + \#COLS\_VAR * \text{default\_variable\_size}$

其中：

	数据类型	缺省大小
default_fixed_size	固定	16MB
default_variable_size	可变	512KB

缺省大小值与所有支持的 RLV 数据类型（bit 除外）的列有关。各种版本的存储库均可视为表碎片。每个表碎片包含各个列的一个列碎片以及一个额外的用于内部管理的列碎片。每个列碎片细分为子碎片，以便于处理大量数据以及实现并行操作。

基本 RLV 存储大小 (BRS) 基于只有初始版本且每个列碎片只有一个子碎片的表。

由于初始 RLV 内存要求是该表中列数的函数，因此在宽（多列）表中插入几行时的 RLV 内存总使用量可以与在窄（少列）表中插入很多行时的 RLV 内存净使用量相同。

以下情况下，存储库的 RLV 内存使用量增加：

原因	描述
添加了新的列子碎片。	当列子碎片已满时，将使用与 <i>default_fixed_size</i> 或 <i>default_variable_size</i> 值相等的内存创建一个新的子碎片。
添加了多个列的子碎片。	为实现并行操作而创建。每个新的子碎片使用与 <i>default_fixed_size</i> 或 <i>default_variable_size</i> 值相等的内存。
创建多种版本的存储库（表碎片）。	处理多个事务时需要。每个新版本的内存使用量至少等于 BRS 的初始内存要求。

多个列的子碎片是某个特定表版本的子项。当不再需要某一表版本时，也不再需要其子列的子碎片。删除不需要的表版本可以减少 RLV 内存使用量。如果新的表版本最初不需要多个列的子碎片，则直到需要时才会创建这些子碎片。

## 配置 RLV 内存

---

除了现有 IQ 高速缓存内存外，还需配置 RLV 内存。主机必须拥有足够的内存来同时满足 RLV 内存和 IQ 主缓冲区高速缓存的需要，这两者是独立的内存池。

RLV 存储库的最大大小（针对所有表）通过一个服务器选项指定，可以使用以下选项进行设置：

- 指定 <max MB> 参数的服务器启动开关 **-iqlvmem**。
- 指定 `rlv_memory_mb`, <max MB> 参数的 **sa\_server\_option** 系统过程

建议大小为 2GB 或者 2GB 以上。如果未指定服务器开关，则缺省值为 2GB。

使用其中一种方法进行定义时，如果此值超过系统虚拟内存限制的 2/3，服务器将生成错误。

### 另请参见

- 管理 RLV 存储库内存（第 57 页）
- `-iqlvmem start_iq` 服务器选项（第 150 页）
- `sa_server_option` 系统过程（第 81 页）

## 配置 RLV Dbspace

---

RLV dbspace 中保存着启用了内存 RLV 存储库的表的持久性日志。

必须为关键操作期间数据结构使用的内存预留一部分 RLV dbspace。可使用 **RV\_RESERVED\_DBSPACE\_MB** 数据库选项设置这部分的大小。

### 另请参见

- 针对表配置 RLV 存储库（第 26 页）
- RLV Dbspace（第 7 页）
- `RV_RESERVED_DBSPACE_MB` 选项（第 76 页）

## 创建 RLV Dbspace

---

要将 RLV 存储库用于表，首先需要创建 RLV dbspace 并至少在其中添加一个 dbfile。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。

### 过程

使用语句 **CREATE DBSPACE *dbspace name* IQ RLV STORE:**

```
CREATE DBSPACE d1 using file f1 '/dev/raw/raw1/f1.iq' size 1000 IQ
RLV STORE
```

RLV dbspace 的磁盘条带化始终为 OFF。不支持在 STRIPING ON|OFF 的情况下或使用 STRIPESIZEKB 选项创建 RLV dbspace。

另请参见

- CREATE DBSPACE 语句 (第 169 页)

## RLV 存储库允许的 ALTER DBSPACE 语法

可使用 ALTER DBSPACE 语句配置 RLV dbspace。该语句用法可能与其它 dbspace 不同。

RLV Dbspace 状态	ALTER 类型	RLV Dbspace 是否允许
联机	ALTER DBSPACE OFFLINE	是 (如果 dbspace 是只读模式)
	ALTER DBSPACE ONLINE	否
	ALTER DBSPACE READONLY	是 (如果 dbspace 是读写模式并且不存在启用 RLV 的对象)
	ALTER DBSPACE READWRITE	是 (如果 dbspace 是只读模式)
	ALTER STRIPING 或 STRIPESIZEKB	否
	RENAME DBSPACE	是
	ADD FILE	是
	DROP FILE	是 (如果文件未在使用中)
	ALTER FILE READONLY	否
	ALTER FILE READWRITE	否
	ALTER FILE SIZE	是 (如果 dbspace 是读写模式)
	ALTER FILE RENAME LOGICAL NAME	是
	ALTER FILE RENAME PATH	否
	脱机	ALTER DBSPACE OFFLINE
ALTER DBSPACE ONLINE		是
ALTER DBSPACE READONLY		否
ALTER DBSPACE READWRITE		否
ALTER STRIPING 或 STRIPESIZEKB		否

RLV Dbspace 状态	ALTER 类型	RLV Dbspace 是否允许
	RENAME DBSPACE	是
	ADD FILE	否
	DROP FILE	是 (按照定义, 文件为空)
	ALTER FILE READONLY	否
	ALTER FILE READWRITE	否
	ALTER FILE SIZE	否
	ALTER FILE RENAME LOGICAL NAME	是
	ALTER FILE RENAME PATH	是
只读	ALTER DBSPACE OFFLINE	是
	ALTER DBSPACE ONLINE	是
	ALTER DBSPACE READONLY	否
	ALTER DBSPACE READWRITE	是
	ALTER STRIPING 或 STRIPESIZEKB	否
	RENAME DBSPACE	是
	ADD FILE	是
	DROP FILE	是
	ALTER FILE READONLY	否
	ALTER FILE READWRITE	否
	ALTER FILE SIZE	否
	ALTER FILE RENAME LOGICAL NAME	是
	ALTER FILE RENAME PATH	否
读写	ALTER DBSPACE OFFLINE	否
	ALTER DBSPACE ONLINE	否
	ALTER DBSPACE READONLY	是 (如果不存在启用 RLV 的对象)
	ALTER DBSPACE READWRITE	否
	ALTER STRIPING 或 STRIPESIZEKB	否

RLV Dbspace 状态	ALTER 类型	RLV Dbspace 是否允许
	RENAME DBSPACE	是 (如果不存在启用 RLV 的对象)
	ADD FILE	是 (如果不存在启用 RLV 的对象)
	DROP FILE	是 (如果文件未在使用中)
	ALTER FILE READONLY	否
	ALTER FILE READWRITE	否
	ALTER FILE SIZE	是 (但在减小文件大小时, 截断空间必须为空)
	ALTER FILE RENAME LOGICAL NAME	是
	ALTER FILE RENAME PATH	否

### 另请参见

- ALTER DBSPACE 语句 (第 150 页)

### 将 RLV 存储 Dbspace 变更为只读

可使用 **ALTER DBSPACE** 语句将 RLV 存储库设置为只读模式。但按照定义, RLV 存储库为读写存储库。因此, 请仅在必要时才将 **dbspace** 设置为只读 (例如, 在删除 **dbspace** 的情况下)。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 数据库中存在一个 RLV 存储 **dbspace**。

### 过程

使用语句 **ALTER DBSPACE *dbspacename* READONLY** 将 **dbspace** 变更为只读模式。

```
ALTER DBSPACE d1 READONLY
```

### 另请参见

- RLV 存储库允许的 ALTER DBSPACE 语法 (第 22 页)
- ALTER DBSPACE 语句 (第 150 页)
- 删除 RLV Dbspace (第 25 页)

## 删除 RLV Dbspace

除非 RLV dbspace 为空，否则无法删除。删除 RLV dbspace 意味着您将再也无法创建启用 RLV 的表或将现有表修改为启用 RLV 存储库。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 数据库中存在一个 RLV dbspace。
- RLV dbspace 是只读的。

### 过程

与 IQ 主存储 dbspace 不同，您不能重新安置驻留在 RLV dbspace 中的对象（事务日志）以将其清空。要清空 RLV dbspace，必须确保存储库中不存在启用 RLV 的表。

**注意：**删除 RLV dbspace 意味着您将不再可以创建启用 RLV 的表或将现有表修改为启用 RLV 的表。

1. 检查是否有启用 RLV 的表（例如，**SYSIQTAB**）。
  - a) 如果存在启用 RLV 的表，则禁用 RLV 存储库或删除这些表。要显示启用 RLV 的表的列表，请运行以下查询：

```
SELECT b.table_id, b.table_name
FROM SYSIQTAB a,SYSTABLE b
WHERE a.table_id = b.table_id and is_rlv='T'
```

2. 使用语句 **ALTER DBSPACE <dbspacename> READONLY** 将 dbspace 变更为只读模式。

```
ALTER DBSPACE d1 READONLY
```

3. 使用 **DROP DBSPACE <dbspacename>** 命令删除 dbspace。

```
DROP DBSPACE d1
```

### 另请参见

- RLV 存储库允许的 ALTER DBSPACE 语法（第 22 页）

## 在 RLV Dbspace 中添加文件

您可能希望在 RLV dbspace 中添加一个文件以获取额外容量来存储 RLV 事务日志。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 数据库中存在一个 RLV dbspace，且其处于联机状态。
- 在 dbspace 采用读写模式时，不存在任何启用 RLV 的对象。

### 过程

使用语句 **ALTER DBSPACE** *dbspace name* **ADD FILE** *filename*:

```
ALTER DBSPACE d1 ADD FILE 'rlv2.iq'
```

由于内存 RLV 存储库的性质，不能将文件指定为 READONLY。

### 另请参见

- RLV 存储库允许的 ALTER DBSPACE 语法 (第 22 页)

## 从 RLV Dbspace 中删除文件

您可以从 RLV dbspace 中删除文件，前提是該文件不是 RLV dbspace 中唯一一个文件而且該文件未被使用。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 数据库中存在一个 RLV dbspace。
- RLV dbspace 是只读的，或者在 dbspace 采用读写模式时文件未在使用中。

### 过程

使用语句 **ALTER DBSPACE** *<dbspace name>* **DROP FILE** *<dbspace filename>*:

```
ALTER DBSPACE d1 DROP FILE rlv2
```

### 另请参见

- RLV 存储库允许的 ALTER DBSPACE 语法 (第 22 页)

## 针对表配置 RLV 存储库

启用了 RLV 存储库的表具有两部分：一部分驻留在 IQ 主存储库中，另一部分驻留在专用的内存 RLV 列存储库中。

启用了行级版本控制存储库的表与其它具有表级版本控制存储库的表可以共存于同一数据库中。

不支持全局临时表和局部临时表的行级版本控制。

### 另请参见

- 配置 RLV Dbspace (第 21 页)
- 配置快照版本控制 (第 28 页)



## 使用 RLV 存储库设置创建新表

创建新的基表时，可专门启用或禁用 RLV 存储库。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- RLV dbspace 至少有一个 dbfile。
- 自有表 - 针对创建表的 dbspace 需要具备 CREATE 特权。还需要具备以下特权之一：
  - CREATE TABLE 系统特权。
  - CREATE ANY OBJECT 系统特权。
- 由任何用户拥有的表 - 针对创建表的 dbspace 需要具备 CREATE 特权。还需要具备以下特权之一：
  - CREATE ANY TABLE 系统特权。
  - CREATE ANY OBJECT 系统特权。

### 过程

要创建新的启用 RLV 的表，请执行：

```
CREATE TABLE [ owner.] table-name
  {ENABLE | DISABLE} RLV STORE
  [ IN dbspace-name ]
```

- *table-name* - 要启用 RLV 存储库的表的名称。
- *dbspace-name* - 有效的 IQ 主 dbspace。

**注意：** 如果省略 ENABLE | DISABLE RLV STORE 子句，表的 RLV 存储库设置在缺省情况下将采用全局数据库选项 **BASE\_TABLES\_IN\_RLV\_STORE** 的值。

### 另请参见

- BASE\_TABLES\_IN\_RLV\_STORE 选项（第 71 页）
- CREATE TABLE 语句（第 173 页）

## 针对现有表启用或禁用 RLV 设置

您可以更改现有基表以便启用或禁用 RLV 存储库。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- RLV dbspace 至少有一个 dbfile。
- 需要具备 ALTER ANY OBJECT 或 ALTER ANY TABLE 系统特权。

### 过程

要对现有表启用或禁用 RLV 存储库设置，请执行：

```
ALTER TABLE [ owner. ] table-name  
{ ENABLE | DISABLE } RLV STORE
```

- *table-name* - 要启用或禁用 RLV 存储库的表的名称。

### 另请参见

- ALTER TABLE 语句 (第 154 页)

## 为表配置缺省存储库

如果 **CREATE TABLE** 语句的 {ENABLE | DISABLE} RLV STORE 子句不存在，将使用 **BASE\_TABLES\_IN\_RLV\_STORE** 选项设置来确定 RLV 存储库。这样，您无需通过修改脚本来启用 RLV 存储库便可允许现有 **CREATE TABLE** 语句运行。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- RLV dbspace 至少有一个 dbfile。

### 过程

在数据库级别启用该选项。

```
Set option PUBLIC.BASE_TABLES_IN_RLV_STORE = 'ON'
```

**注意：** 使用 **CREATE TABLE** 语句对单个表进行的设置会覆盖此选项。缺省设置为 OFF。如果设置为 ON，执行不含 ENABLE | DISABLE RLV 子句的 **CREATE TABLE** 语句将创建启用 RLV 的表。

### 另请参见

- BASE\_TABLES\_IN\_RLV\_STORE 选项 (第 71 页)

## 配置快照版本控制

快照版本控制用于描述数据库服务器在表中使用的版本控制访问类型：表级快照版本控制或行级快照版本控制。行级版本控制允许对启用 RLV 的表的并发写入程序访问和行级锁定。

表级版本控制为缺省配置，并能提供基于整个表的版本控制。

行级版本控制针对固定的表版本提供行级的版本控制。版本控制发生在事务启动时。为已启动事务定义了快照版本后，直至该事务完成后才能对其进行更改。

### 另请参见

- 针对表配置 RLV 存储库 (第 26 页)

- 教程：在表中使用行级版本控制（第 31 页）

## 行级快照版本控制

行级快照版本控制仅适用于为内存 RLV 存储库启用的表。行级快照版本控制允许多个写入程序对表进行并发 DML 更改，但永远也不要同时对相同的行进行并发 DML 更改。

行级快照版本控制使用行锁在行级锁定表。行锁将为某个表行提供写入锁，意味着事务被阻塞或失败，具体情况取决于 **BLOCKING** 和 **BLOCKING\_TIMEOUT** 选项的设置。如果 **BLOCKING** 设置为 ON，则事务阻塞。如果 **BLOCKING** 设置为 OFF，事务立即失败，并抛出 ALREADY LOCKED SQL 异常。

事务阻塞允许行级快照版本控制同时写入到同一个表的不同行中。根据 **BLOCKING** 和 **BLOCKING\_TIMEOUT** 选项设置，行锁争用将导致出错，或者，如果在指定超时期限内释放锁，则会尝试重新获取锁。当为表级版本控制配置的事务尝试向含有由行级版本控制事务锁定的行的表中写入内容时，表级事务或者失败，显示一条错误，或者阻塞，并且如果锁在指定超时期限内被释放则尝试重新获取锁。

DDL 更改为表（**CREATE**、**DROP** 和 **ALTER**），但在表级锁定表。

### 另请参见

- 管理 RLV 存储库中的阻塞（第 51 页）
- **BLOCKING** 选项（第 71 页）
- **BLOCKING\_TIMEOUT** 选项（第 72 页）

### 指定快照版本控制

使用 **SNAPSHOT\_VERSIONING** 选项将快照版本控制类型设置为 Row-level 或 Table-Level。您可以在数据库 (PUBLIC) 级别、连接级别 (TEMPORARY) 或者用户级别设置该选项。要使用内存 RLV 存储库，请为事务启用行级快照版本控制。要同时更新同一个表的不同行，每个事务或连接还要必须启用行级快照版本控制

### 前提条件

- 如果设置为行级，RLV 存储 dbspace 存在，且包含至少一个 dbfile。
- 如果设置为行级，表为启用 RLV 的表。
- 必须具备 SET ANY PUBLIC OPTION 系统特权才能为 PUBLIC 或者其他用户或角色设置此选项。

### 过程

为某个事务设置快照版本控制属性后，该属性将一直保持不变，直到该事务提交。

1. 确定 **SET OPTION** 命令的范围以将选项设置为数据库范围的选项、连接级别的选项或用户级别的选项：

- **SET OPTION public.SNAPSHOT\_VERSIONING...**
- **SET TEMPORARY OPTION SNAPSHOT\_VERSIONING...**
- **SET OPTION *username*.SNAPSHOT\_VERSIONING...**

2. 指定快照版本控制的类型。

级别	选项
<b>Row-level</b>	行级快照版本控制。内存 RLV 存储库需要此选项。行级快照版本控制允许多个写入程序对表进行并发 DML 更改，但永远也不要同时对相同的行进行并发 DML 更改。
<b>Table-level</b>	典型（向后兼容）SAP Sybase IQ 版本控制行为。在表级拍摄快照。多个写入程序不能对表进行并发 DML 更改。

```
SET TEMPORARY OPTION Snapshot_Versioning = 'Row-level';

CREATE TABLE rv_locks(c1 int, c2 int, c3 int);

INSERT INTO rv_locks VALUES (1,1,1);
INSERT INTO rv_locks VALUES (2,2,2);
INSERT INTO rv_locks VALUES (3,3,3);
INSERT INTO rv_locks VALUES (4,4,4);
COMMIT;
```

另请参见

- SNAPSHOT\_VERSIONING 选项（第 77 页）

### 限制快照版本控制

使用 **ALLOW\_SNAPSHOT\_VERSIONING** 数据库选项将数据库中允许的版本控制限制为 Table-level、Row-level 或 any（无限制）。只能在数据库 (PUBLIC) 级别设置 **ALLOW\_SNAPSHOT\_VERSIONING**。不能在连接级别或用户级别设置该选项。

前提条件

- 如果设置为行级，则 SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 需要 SET ANY SYSTEM OPTION 系统特权。

过程

1. 使用 **SET OPTION** 命令在 PUBLIC 级别设置该选项。不允许使用 **SET TEMPORARY OPTION...** 和 **SET OPTION <username>...**。
2. 将数据库中允许的版本控制类型限制为“仅表级”或“仅行级”。

限制	选项
仅表级	set PUBLIC option ALLOW_SNAPSHOT_VERSIONING = 'Table-level'

限制	选项
仅行级	<code>set PUBLIC option ALLOW_SNAPSHOT_VERSIONING = 'Row-level'</code>

将该选项设置为“表级”，阻止 RLV 对数据库中的任何表进行访问，并有效地关闭 RLV 存储库。

要从数据库中删除版本控制限制，需要将该选项设为“任何”。

限制	选项
无限制	<code>set PUBLIC option ALLOW_SNAPSHOT_VERSIONING = 'any'</code>

另请参见

- SNAPSHOT\_VERSIONING 选项（第 77 页）

## 教程：在表中使用行级版本控制

在本教程中，将创建启用 RLV 的 dbspace 和表，插入数据，然后将 RLV 存储库合并到 IQ 主存储库中。

前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。

过程

1. 使用 db 文件 file1 创建一个 RLV dbspace dbsp1。

```
CREATE DBSPACE dbsp1
USING FILE file1 'file1.iqrlv'
SIZE 1000
IQ RLV STORE
```

2. 在 dbspace 中创建一个启用 RLV 的表，或者 RLV 启用一个现有表。  
启用新表：

```
CREATE TABLE test1
(rowid INT NOT NULL,
 col1 char(25) NOT NULL,
 col2 char(25) NULL,
 col3 varchar(25) NOT NULL,
 col4 varchar(25) NULL)
ENABLE RLV STORE
```

启用现有表：

```
ALTER TABLE test1
ENABLE RLV STORE
```

3. 检查该表是否启用了 RLV。

```
SELECT is_rlv FROM sysiqtab  
WHERE table_id = (SELECT table_id FROM systab WHERE table_name =  
'test1')
```

4. 将快照版本控制选项设置为“行级”。

```
SET TEMPORARY OPTION snapshot_versioning = 'Row-level'
```

5. 向启用 RLV 的表中插入数据前，检查 RLV 内存使用情况。

```
SELECT TOTAL FROM sp_iqrlvmemory('test1', 'DBA')
```

6. 将数据插入表中。

```
INSERT INTO test1 VALUES (1, 'char25', NULL, 'varchar25', NULL)  
INSERT INTO test1 VALUES (2, 'char25', NULL, 'varchar25', NULL)  
INSERT INTO test1 VALUES (3, 'char25', NULL, 'varchar25', NULL)
```

7. 向启用 RLV 的表中插入数据后，检查 RLV 内存使用情况。

```
SELECT TOTAL FROM sp_iqrlvmemory('test1', 'DBA')
```

8. 手动将数据合并到主阻塞合并中。

```
sp_iqmergerlvstore 'BLOCKING', 'test1', 'DBA'
```

9. 将数据合并到主阻塞后，检查 RLV 内存使用情况。

```
SELECT TOTAL FROM sp_iqrlvmemory('test1', 'DBA')
```

### 另请参见

- 配置快照版本控制 (第 28 页)

## 将 RLV 存储库合并到 IQ 主存储库

随着时间的推移或触发阈值时，将通过一个异步数据传输过程（RLV 存储库合并）将内存中提交的数据合并到 IQ 主存储库中。

可通过以下两种方法之一将启用 RLV 的表数据合并到 IQ 主存储库中：

1. 非阻塞（后台）合并：在非阻塞合并进行过程中，各事务通常仍可以对 RLV 存储库执行读写操作。由于合并过程会占用系统资源，因此可能会影响到运行时操作。非阻塞合并会短暂锁定对正合并的表的写入访问。这可能导致一个短时间的暂停，但不会导致事务失败。
  - 服务器会根据需要启动非阻塞合并。服务器合并评测程序会按照可配置的时间间隔来定期执行合并。该合并也可以由自动合并阈值触发。
  - 您还可以手动执行非阻塞合并。但是，不建议这么做。
2. 阻塞（前台）合并：发生合并时，要合并到 IQ 主存储库的数据的所在表将被锁定。RLV 合并将作为一个事务运行，从而创建一个新版本的表。这个表版本的可见性遵循常规的表版本控制规则。
  - 某些事件会触发阻塞合并自动执行。
  - 如果需要执行手动合并，则大多数情况下您将运行阻塞合并。

### 另请参见

- RLV 存储库持久性和耐久性（第 6 页）
- 管理 RLV 存储库内存（第 57 页）
- RLV 存储库合并（第 4 页）

## 前台自动合并

---

IQ 服务器将在必要时自动执行阻塞（前台）合并。

某些操作需要先执行一个自动阻塞合并才能开始：

- 以表级快照隔离（而非行级快照隔离）访问表
- RLV DML 正在接近 RLV 内存限值
- 使用 DDL 命令，例如 **CREATE** 或 **ALTER**

在执行合并时，需要即时自动阻塞合并的命令可能会经历一个暂停，然后才会继续运行。

### 另请参见

- 设置合并触发阈值（第 34 页）
- 运行手动合并（第 35 页）

将 RLV 存储库合并到 IQ 主存储库

- 查看合并历史记录（第 36 页）
- IQMSG 文件中记录的合并阶段（第 37 页）
- 合并后的表碎片（第 38 页）
- 教程：在表中使用行级版本控制（第 38 页）

## 设置合并触发阈值

---

IQ 服务器会定期针对每个启用 RLV 的表评测可调整的合并阈值集，并根据需要自动执行后台（非阻塞）合并。阈值设置是可以更改的，但在更改之前应加以谨慎考虑，因为这样可能会影响到性能。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 已配置 RLV 存储库。

### 过程

发生合并时，由于资源被占用，所以可能会影响到服务器的性能。

#### 1. (可选) 更改 IQ 服务器合并评测时间间隔。

合并评测程序将对照配置的阈值检查每个启用行级版本控制 (RLV) 的表的合并参数，以确定是否应在非阻塞模式下（后台）将 RLV 表合并到 IQ 主存储库。您可更改合并评测程序的激活时间间隔。如果合并评测程序已激活，或时间间隔已结束但合并仍在运行，则合并评测程序将等到下个时间间隔再重新启动。

数据库选项	描述	缺省值
RV_AUTO_MERGE_EVAL_INTERVAL	合并评测的时间间隔	15 分钟

#### 2. (可选) 更改表阈值。

如果超出此表阈值，则对特定表触发合并。

表阈值	描述	缺省值
RV_MERGE_TABLE_NUMROWS	提交的 RLV 行数	1000 万
RV_MERGE_TABLE_MEMPERCENT	RLV 总内存大小百分比	100/RLV 表数

#### 3. (可选) 更改节点阈值。

如果超出节点阈值，则合并条件评测程序将确定要对哪些表进行合并。如果必须合并多个表才能满足该节点阈值，则对要合并的每个表触发并行合并。



节点阈值	描述	缺省值
RV_MERGE_NODE_MEMSIZE	RLV 总内存大小	可配置大小的 75%

### 另请参见

- 前台自动合并 (第 33 页)
- 运行手动合并 (第 35 页)
- 查看合并历史记录 (第 36 页)
- IQMSG 文件中记录的合并阶段 (第 37 页)
- 合并后的表碎片 (第 38 页)
- 教程：在表中使用行级版本控制 (第 38 页)
- 管理 RLV 存储库内存 (第 57 页)
- RV\_AUTO\_MERGE\_EVAL\_INTERVAL 选项 (第 73 页)
- RV\_MERGE\_TABLE\_NUMROWS 选项 (第 76 页)
- RV\_MERGE\_TABLE\_MEMPERCENT 选项 (第 75 页)
- RV\_MERGE\_NODE\_MEMSIZE 选项 (第 75 页)

## 运行手动合并

RLV 存储库采取自我管理模式，根据需要执行自动合并。但在少数情况下，您可能希望手动触发合并。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 已配置 RLV 存储库。

### 过程

在以下情况下您可能要考虑执行手动合并：

- 您准备执行表级装载（确保自动合并不会影响到 DDL 或装载命令性能）您将在这种情况下运行阻塞合并。
- 在对表执行会影响大量数据的 DML 操作（如批量装载）之前需要释放内存（以确保自动合并不会与 DML 命令同时运行）。您将在这种情况下运行阻塞合并。
- 关闭之前要缩短启动时间（否则，将会执行十分耗时的 RLV 恢复）。
- 为自动合并期设置的时间过长并且系统资源正在接近阈值限制。在这种情况下您将运行非阻塞合并，但还需修改时间和阈值以便日后不需要对其进行密切监控。

要手动运行 RLV 合并，请使用 SQL 存储过程：

```
sp_iqmergerlvstore [ [ merge_type ],
                   [ table_name ], [ table_owner ] ]
```

## 将 RLV 存储库合并到 IQ 主存储库

- 如果未指定表名称，则会将 RLV 存储库中的所有活动数据（来自所有启用 RLV 的表）合并到 IQ 主存储库中。
- 合并类型可以是 **BLOCKING** | **NON-BLOCKING**。
- 执行合并后，存储过程将自动提交合并事务。

### 另请参见

- 前台自动合并（第 33 页）
- 设置合并触发阈值（第 34 页）
- 查看合并历史记录（第 36 页）
- IQMSG 文件中记录的合并阶段（第 37 页）
- 合并后的表碎片（第 38 页）
- 教程：在表中使用时级版本控制（第 38 页）
- sp\_iqmergerlvstore 过程（第 107 页）

## 查看合并历史记录

---

查看对特定表执行的一系列合并，包括合并日期、合并类型和合并统计信息。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 已配置 RLV 存储库。

### 过程

1. 使用 **SYSIQLVMERGEHISTORY** 视图查看合并的发生时间、合并的数据以及合并原因。
2. 查阅 **IQMSG** 文件以查看记入日志的合并故障。

### 另请参见

- 前台自动合并（第 33 页）
- 设置合并触发阈值（第 34 页）
- 运行手动合并（第 35 页）
- IQMSG 文件中记录的合并阶段（第 37 页）
- 合并后的表碎片（第 38 页）
- 教程：在表中使用时级版本控制（第 38 页）
- **SYSIQLVMERGEHISTORY** 系统视图（第 226 页）

## IQMSG 文件中记录的合并阶段

服务器在数据库 IQMSG 文件中记录 RLV 合并活动。

合并的五个阶段通过一个带有 **Mrg** 后缀的数据行进行记录。

日志后缀	合并阶段
Mrg B <table_id> <merge_type> [merge_host]	开始
Mrg D	删除
Mrg U	更新
Mrg I	插入
Mrg E	结束

对于合并开始阶段, <merge\_type> 为以下之一:

- **D** - 自动 DDL 阻塞合并
- **M** - 自动 DML 阻塞合并
- **N** - 非阻塞合并

[merge\_host] 是用于运行合并的内部连接的可选连接句柄。这发生在您执行 DDL 合并时。运行 DDL 的连接将使用另一内部服务器连接来运行合并。合并完成后, 调用 DDL 连接将重新恢复运行。

下面是合并的 IQMSG 条目示例:

```
I. 01/20 17:25:27. 0000000022 Txn 179 0 179
I. 01/20 17:25:27. 0000000021 Mrg B 775 D 0000000017
I. 01/20 17:26:28. 0000000021 Mrg D
I. 01/20 17:26:29. 0000000021 Mrg U
I. 01/20 17:26:34. 0000000021 Mrg I
I. 01/20 17:26:39. 0000000021 Mrg E
I. 01/20 17:26:39. 0000000021 Cmt 188
```

每行的前缀是一个时间戳以及对请求进行日志记录的连接 ID。在本例中, 连接 21 是运行合并的内部连接。在第一行中, Txn 179 0 179 是使用 ID 179 指示开始事务的标准消息。第三行表明已针对表 ID 775 开始合并, 该合并是 DDL 阻塞合并以及启动该合并的连接是 17。最后一行表明的是连接 21 提交以及提交 ID 是 188。

### 另请参见

- 前台自动合并 (第 33 页)
- 设置合并触发阈值 (第 34 页)
- 运行手动合并 (第 35 页)
- 查看合并历史记录 (第 36 页)

将 RLV 存储库合并到 IQ 主存储库

- 合并后的表碎片 (第 38 页)
- 教程: 在表中使用时级版本控制 (第 38 页)

## 合并后的表碎片

---

在非阻塞合并中, 将从 RLV 存储库最近提交的数据写入到 IQ 主存储库以创建启用 RLV 的表的新表级版本。这个新的表级版本合并了先前的表级版本以及 RLV 存储库的更改 (来自已提交事务的内存更改)。未提交的事务可能引用预合并 RLV 存储库中的快照版本。在事务终止前, 这些碎片会一直保留在内存中。

合并操作本身对 RLV 存储库具有影响:

- 开始合并时, 将创建一个新的 RLV 存储库实例。
- 从这时起, 所有数据更改都转移到新实例。
- RLV 存储库原始实例的已提交更改将合并到 IQ 主存储库中。
- 然后合并结束。

激活的合并操作使用两个 RLV 存储库无交集实例。原始 RLV 存储库实例包含在合并开始前所有提交的更改; 新的 RLV 存储库包含合并开始后做出的所有更改。由于原始实例中存在未决事务 (事务已开始, 但在合并前未提交), 该原始实例将一直保留到所有事务都提交为止。

对于阻塞合并, 情形更为简单一些。在预合并的 RLV 存储库中不存在引用快照的未提交事务, 在合并运行期间也未发生任何数据更改。因此, 阻塞合并完成后, 仅存在一个单一的、空的 RLV 表碎片。

### 另请参见

- 前台自动合并 (第 33 页)
- 设置合并触发阈值 (第 34 页)
- 运行手动合并 (第 35 页)
- 查看合并历史记录 (第 36 页)
- IQMSG 文件中记录的合并阶段 (第 37 页)
- 教程: 在表中使用时级版本控制 (第 38 页)

## 教程: 在表中使用时级版本控制

---

在本教程中, 将创建启用 RLV 的 dbspace 和表, 插入数据, 然后将 RLV 存储库合并到 IQ 主存储库中。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。

## 过程

1. 使用 db 文件 file1 创建一个 RLV dbspace dbsp1。

```
CREATE DBSPACE dbsp1
USING FILE file1 'file1.iqrlv'
SIZE 1000
IQ RLV STORE
```

2. 在 dbspace 中创建一个启用 RLV 的表，或者 RLV 启用一个现有表。  
启用新表：

```
CREATE TABLE test1
(rowid INT NOT NULL,
 col1 char(25) NOT NULL,
 col2 char(25) NULL,
 col3 varchar(25) NOT NULL,
 col4 varchar(25) NULL)
ENABLE RLV STORE
```

启用现有表：

```
ALTER TABLE test1
ENABLE RLV STORE
```

3. 检查该表是否启用了 RLV。

```
SELECT is_rlv FROM sysiqtab
WHERE table_id = (SELECT table_id FROM systab WHERE table_name =
'test1')
```

4. 将快照版本控制选项设置为“行级”。

```
SET TEMPORARY OPTION snapshot_versioning = 'Row-level'
```

5. 向启用 RLV 的表中插入数据前，检查 RLV 内存使用情况。

```
SELECT TOTAL FROM sp_iqrlvmemory('test1', 'DBA')
```

6. 将数据插入表中。

```
INSERT INTO test1 VALUES (1, 'char25', NULL, 'varchar25', NULL)
INSERT INTO test1 VALUES (2, 'char25', NULL, 'varchar25', NULL)
INSERT INTO test1 VALUES (3, 'char25', NULL, 'varchar25', NULL)
```

7. 向启用 RLV 的表中插入数据后，检查 RLV 内存使用情况。

```
SELECT TOTAL FROM sp_iqrlvmemory('test1', 'DBA')
```

8. 手动将数据合并到主阻塞合并中。

```
sp_iqmergerlvstore 'BLOCKING', 'test1', 'DBA'
```

9. 将数据合并到主阻塞后，检查 RLV 内存使用情况。

```
SELECT TOTAL FROM sp_iqrlvmemory('test1', 'DBA')
```

## 另请参见

- 前台自动合并 (第 33 页)
- 设置合并触发阈值 (第 34 页)
- 运行手动合并 (第 35 页)
- 查看合并历史记录 (第 36 页)

将 RLV 存储库合并到 IQ 主存储库

- IQMSG 文件中记录的合并阶段 (第 37 页)
- 合并后的表碎片 (第 38 页)

# 监控锁和死锁

使用 `sp_iqlocks` 存储过程显示数据库中有关行锁、意图写锁和死锁的详细信息。

另请参见

- `sp_iqlocks` 过程 (第 104 页)

## 教程：监控意图写锁

---

在本教程中，您将创建启用 RLV 的表，执行事务，并使用 `sp_iqlocks` 存储过程报告数据库中的模式级锁和意图写锁。然后使用 `sp_iqconnection` 和 `sa_conn_info` 存储过程查看控制着意图写锁的内部连接。

前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 已配置 RLV 存储库。

过程

**提示：**可以使用 SAP Control Center 对锁进行监控。有关详细信息，请参见 SCC 中针对 SAP Sybase IQ 的 SAP Control Center 联机帮助，或访问 <http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680>。

---

1. 创建启用 RLV 的表 `rv_locks` 和 `rv_locks2`，并配置表级快照版本控制。

```
SET TEMPORARY OPTION SNAPSHOT_VERSIONING = 'Table-level';

CREATE TABLE rv_locks(c1 INT, c2 INT, c3 INT);

INSERT INTO rv_locks VALUES (1,1,1);
INSERT INTO rv_locks VALUES (2,2,2);
INSERT INTO rv_locks VALUES (3,3,3);
INSERT INTO rv_locks VALUES (4,4,4);
COMMIT;

CREATE TABLE rv_locks2(c1 int, c2 int, c3 int);

INSERT INTO rv_locks2 VALUES (1,1,1);
INSERT INTO rv_locks2 VALUES (2,2,2);
INSERT INTO rv_locks2 VALUES (3,3,3);
INSERT INTO rv_locks2 VALUES (4,4,4);
COMMIT;
```

```
ALTER TABLE rv_locks ENABLE RLV STORE;
ALTER TABLE rv_locks2 ENABLE RLV STORE;
```

2. 启用连接阻塞并设置阻塞超时阈值:

```
SET TEMPORARY OPTION BLOCKING = 'ON';
SET TEMPORARY OPTION BLOCKING_TIMEOUT = '0';
```

3. 使用 **sp\_iqlocks** 存储过程查看当前的数据库锁集。此时，未返回任何锁。

```
sp_iqlocks
```

启用 **RLV** 的表缺少意图写锁，表明该表的内存 **RLV** 部分仍有待创建。

4. 将事务的快照版本控制属性设置为行级。

```
SET TEMPORARY OPTION SNAPSHOT_VERSIONING = 'row-level';
```

5. 写入表中。

```
INSERT INTO rv_locks VALUES (5,5,5);
```

对启用 **RLV** 的表进行写入或查询将会在要求时在内存中创建该表的启用 **RLV** 的部分。

6. 重新执行 **sp\_iqlocks**。

```
sp_iqlocks
```

这次，该过程返回意图写锁。

```
conn_name,conn_id,user_id,table_type,creator,table_name,index_id,
lock_class,lock_duration,lock_type,row_identifer,row_range
'SQL_DBC_13cd6038',
3,'DBA','BASE','DBA','rv_locks','','Schema','Transaction','Shared',
,
'RLV_CONN_T775',
1000000407','','BASE','DBA','rv_locks','','Table','Transaction','Int
ent',,
```

连接 ID 100000407 在 **rv\_locks** 表中具有意图写锁。该锁类型设置为 Intent，表示意图写锁。

**注意：** 连接 ID 号 (100000407) 很大，因为它表示服务器本身的内部连接。此内部连接用于管理启用 **RLV** 的表中的锁。

ConnectionID 3 在该表中具有模式锁。该锁类型设置为 Shared，表示共享模式锁。

7. 返回到执行插入的未提交事务，并将其提交：

```
Commit
```

在提交过程中，数据库会释放该事务持有的锁。对于本教程，此操作仅释放共享模式锁。启用 **RLV** 的表现现在存在于内存中，其中含有已提交的数据。因此，此时仅存的锁是由该表启用 **RLV** 的部分所持有的意图写锁。

8. 重新执行 **sp\_iqlocks**：

```
sp_iqlocks
```

模式锁已不存在，但意图写锁仍然存在：



```
conn_name,conn_id,user_id,table_type,creator,table_name,index_id,
lock_class,lock_duration,lock_type,row_identifiers,row_range
'RVL_CONN T775',
100000407,,'BASE','DBA','rv_locks2',,'Table','Transaction','In
tent',,
```

---

**注意：**自从上次执行 `sp_iqlocks` 后，`conn_id 100000407` 行一直没有发生更改。

---

**9. 执行 `sp_iqconnection` 查看连接详细信息：**

```
sp_iqconnection
```

您会看到：

```
ConnHandle,Name,Userid,LastReqTime,ReqType,IQCmdType,LastIQCmdTim
e,IQCursors,LowestIQCursorState,IQthreads,TxnID,ConnCreateTime,Te
mpTableSpaceKB,TempWorkSpaceKB,IQconnID,satoiq_count,iqtosa_count
,CommLink,NodeAddr,LastIdle,MPXServerName,LSName,INCConnName,INCC
onnSuspended
1,'SQL_DBC_13de5fd8','DBA','2012-08-08
08:49:25.629','PREFETCH','NONE',2012-08-08 08:49:25.0,0,'NONE',
0,0,2012-08-08 08:49:24.0,0,0,70,40,2,'local','','0','','N'
3,'SQL_DBC_13cd6038','DBA','2012-08-08
09:25:32.920','OPEN','IQUILITYOPENCURSOR',2012-08-08
09:25:32.0,0,'NONE',0,1008,2012-08-08
08:50:04.0,0,92,187,413,'local','','8789','','N'
100000407,'INT: RLVLockConn','','','unknown (0)','NONE',
0001-01-01 00:00:00.0,0,'NONE',0,0,2012-08-08
09:00:40.0,0,0,410,2,0,'NA','NA',0','','N'
```

第三行 (ConnHandle 100000407) 提供关于启用 RLV 的表在控制意图写锁时所用的内部连接 (RLVLockConn) 的信息。

---

**注意：**ConnHandle 100000407 与 `sp_iqlocks` 输出中的 `conn_id 100000407` 相对应。它也与 `sp_iqtransaction` 输出中的 ConnHandle 100000407 相对应。

---

**10. 执行 `sa_conn_info` 查看更多的连接详细信息。 `sa_conn_info` 类似于 `sp_iqconnection`。**

```
sa_conn_info
```

您会看到：

```
Number,Name,Userid,DBNumber,LastReqTime,ReqType,CommLink,NodeAddr
,ClientPort,ServerPort,BlockedOn,LockRowID,LockIndexID,LockTable,
UncommitOps,ParentConnection
100000407,sa_'INT: RLVLockConn','','0','','unknown (0)','NA','NA',
0,0,0,0,,'0,
3,'SQL_DBC_13cd6038','DBA',0,'2012-08-08
09:30:43.799','FETCH','local','','0,0,0,0,,'0,
1,'SQL_DBC_13de5fd8','DBA',0,'2012-08-08
08:49:25.629','PREFETCH','local','','0,0,0,0,,'0,
```

---

**注意：**在第一行中，数字 100000407 与 `sp_iqconnection` 输出中的 ConnHandle 100000407 和 `sp_iqlocks` 输出中的 `conn_id 100000407` 相对应。

---

Userid "INT:RLVLockConn" 表示内部连接。该连接由启用 RLV 的表用于控制意图写锁。

---

### 另请参见

- 管理 RLV 存储库中的阻塞 (第 51 页)
- 行级快照版本控制 (第 29 页)
- sa\_conn\_info 系统过程 (第 78 页)
- sp\_iqconnection 过程 (第 94 页)
- sp\_iqlocks 过程 (第 104 页)

## 教程：监控行级锁

---

在本教程中，将创建启用 RLV 的表、提交事务并删除已提交的行，从而显示行锁定和行范围锁定。**sp\_iqlocks** 已存储行级锁的相关过程报告。

### 前提条件

SAP Sybase IQ 服务器具有一个 Simplex 数据库。

### 过程

**提示：** 可以使用 SAP Control Center 对锁进行监控。请参见 SAP Control Center 在线帮助。

---

1. 创建启用 RLV 的表 rv\_locks 和 rv\_locks2，并配置表级快照版本控制。

```
SET TEMPORARY OPTION SNAPSHOT_VERSIONING = 'Table-level';

CREATE TABLE rv_locks(c1 INT, c2 INT, c3 INT);

INSERT INTO rv_locks VALUES (1,1,1);
INSERT INTO rv_locks VALUES (2,2,2);
INSERT INTO rv_locks VALUES (3,3,3);
INSERT INTO rv_locks VALUES (4,4,4);
COMMIT;

CREATE TABLE rv_locks2(c1 int, c2 int, c3 int);

INSERT INTO rv_locks2 VALUES (1,1,1);
INSERT INTO rv_locks2 VALUES (2,2,2);
INSERT INTO rv_locks2 VALUES (3,3,3);
INSERT INTO rv_locks2 VALUES (4,4,4);
COMMIT;

ALTER TABLE rv_locks ENABLE RLV STORE;
ALTER TABLE rv_locks2 ENABLE RLV STORE;
```

2. 启用连接阻塞并设置阻塞超时阈值。

```
set temporary option blocking = 'On';
set temporary option blocking_timeout = '0';
```

### 3. 写入表中。

```
insert into rv_locks values (5,5,5);
```

对启用 RLV 的表进行写入或查询将会在要求时在内存中创建该表的启用 RLV 的部分。

### 4. 执行 `sp_iqlocks`。

```
sp_iqlocks
```

随即显示意图写锁。

```
conn_name,conn_id,user_id,table_type,creator,table_name,index_id,
lock_class,lock_duration,lock_type,row_identifier,row_range
'SQL_DBC_13cd6038',
3,'DBA','BASE','DBA','rv_locks','','Schema','Transaction','Shared',
,
'RLV_CONN_T775',
1000000407','','BASE','DBA','rv_locks','','Table','Transaction','Int
ent',,
```

连接 ID 100000407 在 `rv_locks` 表中具有意图写锁。该锁类型设置为 `Intent`，表示意图写锁。

**注意：** 连接 ID 号 (100000407) 很大，因为它表示服务器本身的内部连接。此内部连接用于管理启用 RLV 的表中的锁。

`ConnectionID 3` 在该表中具有模式锁。该锁类型设置为 `Shared`，表示共享模式锁。共享模式锁会阻止其它事务针对该表执行 DML 操作。

### 5. 提交事务。

```
Commit
```

在提交过程中，数据库会释放该事务持有的锁。在本例中，此操作仅释放共享模式锁。启用 RLV 的表现在存在于内存中，其中含有已提交的数据。因此，此时仅存的锁是由该表启用 RLV 的部分所持有的意图写锁。

### 6. 删除先前提提交的行。

```
delete from rv_locks where c1 = 5;
```

删除该行前，数据库将删除行级锁。

### 7. 再次执行 `sp_iqlocks`。

```
sp_iqlocks
```

将显示三个锁：一个共享锁、一个行锁和一个意图写锁。

```
conn_name,conn_id,user_id,table_type,creator,table_name,index_id,
lock_class,lock_duration,lock_type,row_identifier,row_range
'SQL_DBC_13cd6038',
3,'DBA','BASE','DBA','rv_locks2','','Schema','Transaction','Shared'
,
'SQL_DBC_13cd6038',
3,'DBA','BASE','DBA','rv_locks2','','Row','Transaction','Row',
```

```
281474976710656,1
'RVL_CONN_T775',
1000000407,','BASE','DBA','rv_locks2','','Table','Transaction','In
tent',,
```

第 1 行显示锁类型为 Shared，表示共享模式锁。此锁由 **DELETE** 语句的 DML 持有。共享模式锁会阻止其它事务针对该表执行 DDL 操作。

第 2 行显示锁类型为 Row，表示行级锁：

- **row\_identifier** - 281474976710656 为锁的启动行的行标识符。
- **row\_range** - 1 表示锁定单个行。

#### 8. 删除多行以说明行范围锁定。

a) 回退当前正在执行删除操作的事务，其中 c1=5。

```
Rollback
```

b) 删除所有行，其中 c1>0。

```
Delete from rv_locks2 where c1 > 0;
```

#### 9. 再次执行 **sp\_iqlocks**。

```
sp_iqlocks
```

将显示四个锁：一个共享锁、两个行锁（其中一个含 **row\_range** 值）和一个意图写锁：

```
conn_name,conn_id,user_id,table_type,creator,table_name,index_id,
lock_class,lock_duration,lock_type,row_identifier,row_range
'SQL_DBC_13cd6038',
3,'DBA','BASE','DBA','rv_locks2','','Schema','Transaction','Shared'
''
'SQL_DBC_13cd6038',
3,'DBA','BASE','DBA','rv_locks2','','Row','Transaction','Row',1,4
'SQL_DBC_13cd6038',
3,'DBA','BASE','DBA','rv_locks2','','Row','Transaction','Row',
281474976710656,1
'RVL_CONN_T775',
1000000407,','BASE','DBA','rv_locks2','','Table','Transaction','In
tent',,
```

**注意：** 第二个输出行表示表级版本的锁定行，而第三个输出行表示行级版本的锁定行。

#### 另请参见

- 管理 RLV 存储库中的阻塞（第 51 页）
- 行级快照版本控制（第 29 页）
- 行锁（第 12 页）
- **sp\_iqlocks** 过程（第 104 页）

## 教程：监控死锁

---

在本教程中，将向 **RLV** 存储库添加死锁、出于报告目的记录死锁，以及使用 **sa\_report\_deadlocks** 报告死锁相关信息。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 已配置 RLV 存储库。

### 过程

为了创建死锁，本教程将在两个事务之间创建循环：

1. 事务 A 具有锁 A，将要请求锁 B。事务 B 具有锁 B，将尝试请求锁 A。
2. 事务 A 将阻塞对锁 B 的请求，直至事务 B 释放该请求后才将其释放。
3. 同时，事务 B 将请求锁 A，直到事务 A 释放该请求后才将其释放。

由于两个事务正在等待的资源当前都是由对方持有，所以永远不会发生释放。这就是典型的死锁情形。数据库服务器将阻止用户语句包含死锁情形，并针对引入死锁的语句自动回退事务。

---

**提示：** 可以使用 **SAP Control Center** 对锁进行监控。请参见 **SAP Control Center** 在线帮助。

---

1. 创建启用 RLV 的表 **rv\_locks** 和 **rv\_locks2**，并配置表级快照版本控制。

```
SET TEMPORARY OPTION SNAPSHOT_VERSIONING = 'Table-level';

CREATE TABLE rv_locks(c1 INT, c2 INT, c3 INT);

INSERT INTO rv_locks VALUES (1,1,1);
INSERT INTO rv_locks VALUES (2,2,2);
INSERT INTO rv_locks VALUES (3,3,3);
INSERT INTO rv_locks VALUES (4,4,4);
COMMIT;

CREATE TABLE rv_locks2(c1 int, c2 int, c3 int);

INSERT INTO rv_locks2 VALUES (1,1,1);
INSERT INTO rv_locks2 VALUES (2,2,2);
INSERT INTO rv_locks2 VALUES (3,3,3);
INSERT INTO rv_locks2 VALUES (4,4,4);
COMMIT;

ALTER TABLE rv_locks ENABLE RLV STORE;
ALTER TABLE rv_locks2 ENABLE RLV STORE;
```

2. 使用 **sp\_iqlocks** 确保 **rv\_locks** 和 **rv\_locks2** 表中不存在行锁。

3. 启用连接阻塞并设置阻塞超时阈值。

```
set temporary option blocking = 'On';
set temporary option blocking_timeout = '0';
```

4. 每次连接删除一行。

- 连接 A:

```
delete from rv_locks where c1 = 1
```

- 连接 B:

```
delete from rv_locks2 where c1 = 1
```

这两个 DML 操作将开始两个独立的事务。

5. 执行 **sp\_iqlocks**。

```
sp_iqlocks
```

6. 查看这两个事务持有的锁。

```
Conn_name,conn_id,user_id,table_type,creator,table_name,index_id,
lock_class,lock_duration,lock_type,row_identifer,row_range
'SQL_DBC_13de5fd8',
1,'DBA','BASE','DBA','rv_locks2','','Schema','Transaction','Shared'
,,
'SQL_DBC_13de5fd8',
1,'DBA','BASE','DBA','rv_locks2','','Row','Transaction','Row',1,1
'SQL_DBC_13cd6038',
3,'DBA','BASE','DBA','rv_locks','','Schema','Transaction','Shared',
/
'SQL_DBC_13cd6038',
3,'DBA','BASE','DBA','rv_locks','','Row','Transaction','Row',1,1
'RVL_CONN_T775',
100000407','','BASE','DBA','rv_locks2','','Table','Transaction','In
tent',,,
'RVL_CONN_T774',
1000006141','','BASE','DBA','rv_locks','','Table','Transaction','Int
ent',,,
```

在输出行中，请注意 **conn\_id** 1 和 3。

7. 连接 A 将删除连接 B 已锁定的同一行:

```
delete from rv_locks2 where c1 = 1
```

此连接阻塞，因为连接 B 已拥有该行的锁。

8. 连接 B 试图删除连接 A 已锁定的同一行:

```
delete from rv_locks where c1 = 1
```

该连接发生死锁。数据库服务器识别出死锁，不允许该连接继续进行操作。数据库服务器取消此删除语句，回退事务、释放锁，并向连接 B 的应用程序发出此错误消息: SQL error, state = 40001 “Deadlock Detected”。

9. 回退连接 A 删除操作。

- 连接 A:

```
rollback
```

## 10. 启用死锁记录。

- 连接 B:

```
set option public.log_deadlocks = 'on';
```

为连接 B 设置记录，因为连接 B 将引起死锁。为连接 A 设置 **log\_deadlocks** 选项，将不会记录任何死锁。

## 11. 重复死锁情形。

- 连接 A:

```
delete from rv_locks where c1 = 1
```

- 连接 A:

```
delete from rv_locks2 where c1 = 1
```

- 连接 B:

```
delete from rv_locks2 where c1 = 1
```

- 连接 B:

```
delete from rv_locks where c1 = 1
```

连接 B 接收到 SQL 错误后，其事务将像以前一样回退。但是，因为已启用了死锁记录，系统会记录死锁事件。

## 12. 执行 sa\_report\_deadlocks。

```
sa_report_deadlocks
```

## 13. 查看记录的死锁事件。

```
snapshotId,snapshotAt,waiter,who,what,object_id,record_id,owner,i
s_victim,rollback_operation_count,iq_rid,iq_txn_id
1,2012-08-08 12:24:04.339,3,'DBA',delete from rv_locks2 where c1 =
1,1,775,1,false,0,1,13184
1,2012-08-08 12:24:04.339,1,'DBA',delete from rv_locks where c1 =
1,0,774,3,true,0,1,13160
```

is\_victim 列用于指示选择哪个事务作为回退候选事务。

## 在 Interactive SQL 中创建死锁报告事件

创建一个表和一个系统事件，用于获取死锁相关信息。

### 前提条件

SAP Sybase IQ 服务器具有一个 Simplex 数据库。

### 过程

1. 创建一个表以存储从 **sa\_report\_deadlocks** 系统过程返回的数据。

```
CREATE TABLE DeadlockDetails (
  deadlockId INT PRIMARY KEY DEFAULT AUTOINCREMENT,
  snapshotId BIGINT,
```

```

snapshotAt TIMESTAMP,
waiter INTEGER,
who VARCHAR(128),
what LONG VARCHAR,
object_id UNSIGNED BIGINT,
record_id BIGINT,
owner INTEGER,
is_victim BIT,
rollback_operation_count UNSIGNED INTEGER );

```

2. 创建一个事件，用于在发生死锁时发送电子邮件通知。

```

CREATE EVENT DeadlockNotification
TYPE Deadlock
HANDLER
BEGIN
    INSERT INTO DeadlockDetails WITH AUTO NAME
    SELECT snapshotId, snapshotAt, waiter, who, what, object_id,
    record_id,
        owner, is_victim, rollback_operation_count
    FROM sa_report_deadlocks ();
    COMMIT;
    CALL xp_startmail ( mail_user = 'John Smith',
                        mail_password = 'mypwd' );
    CALL xp_sendmail( recipient = 'DBAdmin',
                    subject = 'Deadlock details added to the
    DeadlockDetails table.' );
    CALL xp_stopmail ( );
END;

```

此事件会将 **sa\_report\_deadlocks** 系统过程的结果复制到表中，并通知管理员有关死锁的情况。

3. 将 **log\_deadlocks** 选项设置为 on。

```
SET OPTION PUBLIC.log_deadlocks = 'On';
```

4. 允许记录最近执行的语句。

```
CALL sa_server_option( 'RememberLastStatement', 'YES' );
```



## 管理 RLV 存储库中的阻塞

RLV 存储库与 IQ 主存储库使用相同的事务阻塞机制。假定将 **BLOCKING** 选项设置为 ON。如果事务 1 持有的写入锁 A 与事务 2 试图获取的写入锁 B 冲突，则事务 2 必须等到写入锁 A 释放，或达到 **BLOCKING\_TIMEOUT** 阈值。缺省情况下，**BLOCKING** 为 OFF。

尽管事务阻塞将最大程度减少锁争用，但可能会导致死锁。

### 另请参见

- `sp_iqconnection` 过程 (第 94 页)
- `sp_iqtransaction` 过程 (第 143 页)
- `sa_conn_info` 系统过程 (第 78 页)

## 启用连接阻塞

---

如果某一事务尝试获取的锁与其它事务中的现有锁相冲突，则启用连接阻塞可以强制该事务进行等待：一直等到所有冲突锁均已被释放，或等到达到 **BLOCKING\_TIMEOUT** 阈值。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。

### 过程

将 **BLOCKING** 数据库选项设置为 ON。

```
set temporary option blocking = 'On';
```

---

**注意：** 可以在连接级别或 PUBLIC 级别上设置该阻塞选项。

---

### 另请参见

- 禁用连接阻塞 (第 52 页)
- 设置阻塞超时阈值 (第 52 页)
- 事务阻塞死锁 (第 53 页)
- 教程：监控阻塞 (第 54 页)
- **BLOCKING** 选项 (第 71 页)

## 禁用连接阻塞

---

如果某一事务尝试获取的锁与其它事务中的现有锁相冲突，则禁用连接阻塞可以强制该事务回退事务并显示错误：

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。

### 过程

将 **BLOCKING** 数据库选项设置为 OFF。

```
set temporary option blocking = 'Off';
```

---

**注意：** 可以在连接级别或 **PUBLIC** 级别上设置该阻塞选项。

---

### 另请参见

- 启用连接阻塞（第 51 页）
- 设置阻塞超时阈值（第 52 页）
- 事务阻塞死锁（第 53 页）
- 教程：监控阻塞（第 54 页）
- **BLOCKING** 选项（第 71 页）
- **BLOCKING\_TIMEOUT** 选项（第 72 页）

## 设置阻塞超时阈值

---

可使用该阈值设置事务为获得锁而等待的时间长度（以毫秒为单位）。如果尝试获得锁的事务与其它事务中的现有锁相冲突，则该事务会一直等待，直至达到 **BLOCKING\_TIMEOUT** 选项阈值。如果冲突仍然存在，则该事务会回退并显示错误。

### 前提条件

SAP Sybase IQ 服务器具有一个 Simplex 数据库。

### 过程

---

**注意：** 缺省值 0 表示受阻塞的事务必须进行无限期等待，直至所有冲突事务释放其中的锁。

---

将 **BLOCKING\_TIMEOUT** 数据库选项值设置成您希望事务等待冲突事务释放锁的毫秒数。

```
set temporary option blocking_timeout = '400';
```

---

**注意：**可以在连接级别或 PUBLIC 级别上设置该阻塞选项。

---

### 另请参见

- 启用连接阻塞 (第 51 页)
- 禁用连接阻塞 (第 52 页)
- 事务阻塞死锁 (第 53 页)
- 教程：监控阻塞 (第 54 页)

## 事务阻塞死锁

---

事务阻塞可能会导致死锁。出现死锁情况时，一组事务将进入一种特殊状态，在该状态下这些事务都不能继续执行。

有两个原因会导致死锁：

- **循环阻塞冲突** – 事务 A 被事务 B 阻塞，而事务 B 又被事务 A 阻塞。等待并不能解决这个问题，必须取消其中一个事务，使另一个事务能够继续执行。如果多个（大于二）事务出现循环阻塞，也会导致这种情况。

为消除事务死锁，数据库服务器会从陷入死锁的连接中选择一个连接，回退对该连接上活动事务的更改并返回错误。数据库服务器使用内部启发式算法选择要回退的连接，该算法优先选择剩余阻塞等待时间最少的连接（由 **BLOCKING\_TIMEOUT** 选项确定）。如果所有连接均设置为始终等待，那么促使服务器检测到死锁的连接会被选作牺牲品连接。

- **所有工作线程被阻塞** – 事务被阻塞时，不会释放其工作线程。例如，为数据库服务器配置三个工作线程。事务 A、B 和 C 被当前未执行请求的事务 D 阻塞。由于没有可用的工作线程，导致出现死锁情况。这种情况称为线程死锁。

假定数据库服务器有  $n$  个工作线程。当  $n-1$  个工作线程被阻塞，并且最后一个工作线程也即将被阻塞时，就会发生线程死锁。数据库服务器的内核不会允许这最后一个工作线程被阻塞，因为这样会导致所有工作线程都被阻塞，数据库服务器将因此而停止响应。数据库服务器会转而结束即将阻塞最后一个工作线程的任务，回退对该连接上的活动事务的更改，并返回错误。

如果因数据库大小或阻塞问题而出现很多长时间运行的请求，则具有数十个或数百个连接的数据库服务器可能会出现线程死锁。此时，您可能需要增大 **start\_iq** 实用程序的 **-gn** 服务器选项的值。

要查看 SAP Control Center 中的锁和死锁，请参见 SAP Control Center 联机帮助。

### 另请参见

- 启用连接阻塞 (第 51 页)
- 禁用连接阻塞 (第 52 页)
- 设置阻塞超时阈值 (第 52 页)
- 教程：监控阻塞 (第 54 页)

- LOG\_DEADLOCKS 选项 (第 73 页)
- sa\_report\_deadlocks 系统过程 (第 80 页)
- sa\_conn\_info 系统过程 (第 78 页)
- sp\_iqconnection 过程 (第 94 页)
- sp\_iqtransaction 过程 (第 143 页)

## 教程：监控阻塞

---

在本教程中，将创建启用 RLV 的表、执行事务，并使用 **sp\_iqtransaction** 存储过程报告数据库中所有事务的连接阻塞和阻塞超时的相关信息。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 已配置 RLV 存储库。

### 过程

1. 创建启用 RLV 的表 `rv_locks` 和 `rv_locks2`，并配置表级快照版本控制。

```
SET TEMPORARY OPTION SNAPSHOT_VERSIONING = 'Table-level';

CREATE TABLE rv_locks(c1 INT, c2 INT, c3 INT);

INSERT INTO rv_locks VALUES (1,1,1);
INSERT INTO rv_locks VALUES (2,2,2);
INSERT INTO rv_locks VALUES (3,3,3);
INSERT INTO rv_locks VALUES (4,4,4);
COMMIT;

CREATE TABLE rv_locks2(c1 int, c2 int, c3 int);

INSERT INTO rv_locks2 VALUES (1,1,1);
INSERT INTO rv_locks2 VALUES (2,2,2);
INSERT INTO rv_locks2 VALUES (3,3,3);
INSERT INTO rv_locks2 VALUES (4,4,4);
COMMIT;

ALTER TABLE rv_locks ENABLE RLV STORE;
ALTER TABLE rv_locks2 ENABLE RLV STORE;
```

2. 将事务的快照版本控制属性设置为行级。

```
set temporary option Snapshot_Versioning = 'Row-level';
```

3. 启用连接阻塞并设置阻塞超时阈值。

```
set temporary option blocking = 'On';
set temporary option blocking_timeout = '0';
```

4. 写入表中。

```
insert into rv_locks values (5,5,5);
```

对启用 RLV 的表进行写入或查询将会在要求时在内存中创建该表的启用 RLV 的部分。

## 5. 执行 `sp_iqtransaction`，查看数据库中所有事务的相关信息。

```
sp_iqtransaction
```

显示的事务信息中每行代表一个不同的事务：

```
Name,Userid,TxnID,CmtID,VersionID,State,ConnHandle,IQConnID,MainTableKBCr,MainTableKBDr,TempTableKBCr,TempTableKBDr,TempWorkspaceKB,TxnCreateTime,CursorCount,SpCount,SpNumber,MPXServerName,GlobalTxnID,VersioningType,Blocking,BlockingTimeout
'SQL_DBC_13cd6038','DBA',1008,0,0,'ACTIVE',
3,92,0,0,0,0,0,'2012-08-08 09:00:39.511',0,4,36,,0,'Row-level','True',0
```

Blocking 值为 True，表示已使用 **BLOCKING** 数据库选项启用了连接阻塞。连接阻塞表示当检测到锁争用时，事务会等待（或阻塞）冲突锁释放后再次发出锁定请求。

Blocking\_Timeout 值为 0（缺省值），表示事务将无限期地等待冲突锁释放。

### 另请参见

- 启用连接阻塞（第 51 页）
- 禁用连接阻塞（第 52 页）
- 设置阻塞超时阈值（第 52 页）
- 事务阻塞死锁（第 53 页）



## 管理 RLV 存储库内存

您可以配置要用于 RLV 存储库的内存量，该内存量与 RLV 存储库可保存的数据量直接相关。当内存消耗量达到所设阈值时，RLV 存储库将自动与 IQ 主存储库合并。

当 RLV 存储库内存大小接近其限值时，自动合并会将已提交行从 RLV 存储库移至 IQ 主存储库，从而为新的 RLV 存储库行释放内存。

如果合并操作不能释放足够的内存，并且在 RLV 存储库中有来自其它事务的未提交行在等待处理，则当前操作将一直阻塞，直到其它操作提交并发生自动合并为止。如果通过合并不能释放更多的内存，则回退当前操作。

### 另请参见

- 将 RLV 存储库合并到 IQ 主存储库（第 33 页）
- RLV 存储库持久性和耐久性（第 6 页）

## 配置 RLV 存储库内存大小

---

您可以针对每个服务器逐一配置 RLV 存储库的最大内存大小。

### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。

### 过程

请仔细选择 RLV 存储库的大小以避免超出主机的物理内存。因此，请考虑以下因素：

- 主机的可用物理内存
- IQ 主存储库以及临时缓冲区高速缓存的大小
- IQ 大内存池的大小
- 主机上运行的其它应用程序的内存需求

选择的值过小会导致额外的合并。如果自动合并的频率过高，这可能会导致 DML 失败。

1. 使用 `-iqrlvmem boot` 参数来指定 RLV 存储库的最大大小（以 Mb 为单位）。

```
-iqrlvmem 8192
```

这将 RLV 大小指定为 8192 Mb。

---

**注意：** 如果 DML 导致 RLV 存储库大小超出配置的内存限值，则它将立即触发一个自动合并。但是通常情况下，RV\_MERGE\_NODE\_MEMSIZE 节点阈值会在达到此限值之前触发自动合并。

---

2. (可选) 在运行时使用 **sa\_server\_option** 更改 RLV 存储库的大小。

```
sa_server_option 'RLV_memory_mb', 16384
```

这将 RLV 存储库大小重新配置为 16384 Mb。

---

**注意：** 此大小是一个“软”限制。RLV 存储库有可能会临时超出配置的内存大小。RLV 存储库内存通常通过合并释放。但是合并本身需要内存，因此在合并过程中可能会临时超出内存限制。此外，即使在合并后，未决的事务也会保留在 RLV 存储库中（这样便会占用内存）。

---

#### 另请参见

- RV\_MERGE\_NODE\_MEMSIZE 选项 (第 75 页)
- -iqlvmem start\_iq 服务器选项 (第 150 页)
- sa\_server\_option 系统过程 (第 81 页)
- 设置合并触发阈值 (第 34 页)
- 监控 RLV 内存使用情况 (第 58 页)

## 监控 RLV 内存使用情况

---

监控系统范围内 RLV 内存使用情况和/或每个表的内存使用情况。

#### 前提条件

- SAP Sybase IQ 服务器具有一个 Simplex 数据库。
- 已配置 RLV 存储库。

#### 过程

1. 可使用 **sp\_iqstatus** 存储过程来监控系统范围的 RLV 内存使用情况。

行名	描述
RLV 内存限制	由 <b>sp_iqlvmemory</b> 存储过程或 <b>sa_server_option RLV_memory_mb</b> 指定的内存限制 (以 MB 为单位)
已使用的 RLV 内存	已使用的 RLV 存储库内存量 (以 MB 为单位)。 <b>注意：</b> 在 RLV 存储库和主存储库合并期间使用的内存量可以合法超出内存限制。



2. 针对特定所有者和表名称使用 `sp_iqrlvmemory` 存储过程监控每个表的 RLV 内存使用情况。

该过程将针对每个消耗 RLV 内存的表都输出一行数据。

输出列	描述
table_id	此行所代表的表的 ID
fragments	此表的存储碎片数
total	此表所用的 RLV 总内存量 (以 MB 为单位)
data	存储表数据所用的 RLV 内存量 (以 MB 为单位)
dictionary	存储此表的字典所用的 RLV 内存量 (以 MB 为单位)
bitmap	存储表级位图所用的 RLV 内存量 (以 MB 为单位)

**注意：** 针对特定版本的数据 (例如版本位图和按需而定索引) 不会影响 RLV 内存限制, 并且不会在 `sp_iqrlvmemory` 中报告。`sp_iqrlvmemory` 会接受所有者名称和表名称这样的可选参数, 这些参数将输出限定为一行。

```
sp_iqrlvmemory 'RLV_table','DBA'
```

#### 另请参见

- `sp_iqrlvmemory` 过程 (第 108 页)
- 配置 RLV 存储库内存大小 (第 57 页)



## 附录：RLV 存储库疑难解答

疑难解答附录中列出了一系列可能遇到的问题，包括有关进一步诊断或解决方案的信息。

### RLV 存储库内存不足

---

问题：您收到错误消息“RLV 存储库内存不足”。

#### 解释 A

长时间运行的活动事务可能正在占用内存中的 RLV 存储碎片。要解决此问题，请尝试以下步骤：

- 运行 `sp_iqrlvmemory` 存储过程以确定是否存在旧的 RLV 存储碎片。
- 使用 `sp_iqtransaction` 存储过程查找早期的活动事务并终止它们。

#### 解释 B

如果启用行级版本控制的事务在启用行级版本控制的表中添加或修改过多数据，则最终将用尽存储库内存。要解决此问题，请尝试以下步骤：

- 减小更新大小。
- 减小装载大小。
- 提高提交频率（以减少每个事务修改的数据量）。

#### 解释 C

自动合并与 RLV 事务数不同步。要解决此问题，请尝试以下步骤：

- 查看合并历史记录表 `SYSIQRLVMERGEHISTORY`。
- 使用 `RV_AUTO_MERGE_EVAL_INTERVAL` 选项调整自动合并期。
- 使用 `RV_MERGE_TABLE_NUMROWS`、`RV_MERGE_TABLE_MEMPERCENT` 和 `RV_MERGE_NODE_MEMSIZE` 选项调整自动合并阈值。

#### 解释 D

最大 RLV 内存配置值过低。要解决此问题，请尝试以下步骤：

- 使用服务器启动选项 `-iqrlvmem` 增加最大 RLV 内存。

---

**注意：** 增加最大 RLV 内存可能需要调整 IQ 高速缓存或添加 RAM。它还需要重新启动服务器，因为参数不是动态的。

---

#### 另请参见

- `sp_iqrlvmemory` 过程（第 108 页）

- sp\_iqtransaction 过程 (第 143 页)
- SYSIQRLVMERGEHISTORY 系统视图 (第 226 页)
- RV\_AUTO\_MERGE\_EVAL\_INTERVAL 选项 (第 73 页)
- RV\_MERGE\_NODE\_MEMSIZE 选项 (第 75 页)
- RV\_MERGE\_TABLE\_MEMPERCENT 选项 (第 75 页)
- RV\_MERGE\_TABLE\_NUMROWS 选项 (第 76 页)
- -iqrlvmem start\_iq 服务器选项 (第 150 页)

---

## 无法转换成 Multiplex

---

问题：您收到错误消息“无法将数据库转换成 Multiplex。存在 RLV dbspace”。

### *解释*

您正尝试转换为 Multiplex 的 Simplex 数据库中存在一个行级版本控制 (RLV) dbspace。在当前版本的 SAP Sybase IQ 中，Multiplex 不支持行级版本控制。

---

## 无法在 Multiplex 中创建 RLV Dbspace

---

问题：您收到错误消息“不允许在 Multiplex 中创建 RLV dbspace”。

### *解释*

在当前版本的 SAP Sybase IQ 中，Multiplex 不支持行级版本控制 (RLV)。

---

## RLV Dbspace 已存在

---

问题：您收到错误消息“RLV dbspace 已存在。无法创建多个 RLV dbspace”。

### *解释 A*

数据库中已存在行级版本控制 (RLV) dbspace。SAP Sybase IQ 对于每个 Simplex 数据库仅支持一个 RLV dbspace。

### *解释 B*

您在尝试增大 RLV dbspace 的大小时可能遇到此错误。如果遇到此错误，请在现有 RLV dbspace 中添加一个新 dbfile，而不是创建另一个 RLV dbspace。

## 无法将 RLV Dbspace 设置为只读

---

问题：您收到错误消息“因为 RLV dbspace N 包含启用 RLV 的表，所以无法将其设置为只读”。

### 解释

由于行级版本控制 (RLV) dbspace 在实时事务中的作用，本来预期将其设置为读写模式。但也可能将 dbspace 设置为只读模式（只要不存在启用 RLV 的表即可）。

- 将 RLV dbspace 设置为只读之前，删除或禁用启用 RLV 的表。

### 另请参见

- 配置 RLV Dbspace（第 21 页）

## 无法在 RLV Dbspace 中创建表

---

问题：您收到错误消息“无法在 RLV dbspace 中创建表 N”。

### 解释

在事务执行期间，每个启用 RLV 的表都会自动使用 RLV dbspace。但是，SAP Sybase IQ 不支持在 RLV dbspace 中创建表。

## 无法为表启用 RLV 存储库

---

问题：您收到错误消息，指明无法为表启用行级版本控制 (RLV)。

### 解释 A

除非已配置 RLV 存储 dbspace，否则无法对表启用 RLV 存储库。

### 解释 B

对于符合以下条件的表不能启用 RLV 存储库：

- 含有 LONG BINARY (LOB) 或 LONG VARCHAR (CLOB) 数据类型。
- 含有 WORD (WD) 或 TEXT 索引。
- 具有外键约束。

### 解释 C

不能对临时表或全局临时表启用 RLV 存储库。

## 无法在启用 RLV 的表中使用外键

---

问题：您收到错误消息“不支持在 RLV 表中使用外键”。

### 解释

在当前版本的 SAP Sybase IQ 中，不支持在启用行级版本控制 (RLV) 的表中使用外键。您不能在启用行级版本控制的表中创建外键。此外，您不能在含有外键的表中启用行级版本控制。

## 无法在启用 RLV 的表中使用索引类型

---

问题：您收到错误消息“索引类型 "TEXT" 无法和已启用 RLV 的表一起使用”或“索引类型 "WD" 无法和已启用 RLV 的表一起使用”。

### 解释

在当前版本的 SAP Sybase IQ 中，启用行级版本控制 (RLV) 的表不支持使用文本索引和字索引。您不能在启用了行级版本控制的表中创建这些索引。此外，您不能对含有文本索引或字索引的表启用行级版本控制。

## 进行表级修改前需要合并

---

问题：您收到错误消息“进行表级修改前，表 N 需要 RLV 存储库合并”。

### 解释 A

某个表级 (TLV) 读写操作正尝试在 **BLOCKING** 选项设为 **OFF** 的情况下修改启用了行级版本控制 (RLV) 的表。这个 TLV 事务需要合并 RLV 存储库，但由于 **BLOCKING** 已设置为 **OFF**，因此它无法等到自动合并完成。要解决此问题，请尝试以下步骤：

- 将 **BLOCKING** 选项设置为 **ON**，并确保此事务中的第一个语句是表级读写模式。
- 在执行表级 DML 事务之前，使用 `sp_iqmergerlvstore` 存储过程执行手动合并。

### 解释 B

某个表级 (TLV) 读写操作正尝试修改启用了行级版本控制 (RLV) 的表。这个 TLV 事务需要合并 RLV 存储库。但由于该 TLV 事务已为此表建立了快照版本，因此无法自动执行合并。要解决此问题，请尝试以下步骤：

- 确保该事务中的第一个语句是表级读写模式。
- 在执行表级 DML 事务之前，使用 `sp_iqmergerlvstore` 存储过程执行手动合并。

### 解释 C

某个 TLV 事务正尝试修改多个启用 RLV 的表。要解决此问题，请尝试以下步骤：

- 拆分该事务，让每个单个事务各自修改一个 RLV 表。

#### 另请参见

- BLOCKING 选项 (第 71 页)
- sp\_iqmergerlvstore 过程 (第 107 页)

## 无法执行 RLV 存储库合并

---

问题：您收到错误消息，指明无法将 RLV 存储库与 IQ 主存储库进行合并。

#### 解释 A

您尝试执行 RLV 存储库合并的表未启用 RLV。只可以合并启用 RLV 的表。

#### 解释 B

合并期间出错。请查阅数据库 IQMSG 文件了解可能发生的异常情况或错误代码，其中可能描述了问题以及如何执行 DML 操作以处理导致此问题的数据项。

例如，如果将某行从 RLV 存储库插入到 IQ 主存储库时出现错误，并且 IQMSG 文件在行 X 或值 Y 中含有一个异常，则可能发生：

- 找到源行
- 保存该行
- 删除该行。

在这种情况下，合并随后应成功执行。然后可单独处理导出的行。

#### 另请参见

- sp\_iqtransaction 过程 (第 143 页)

## RLV 存储库合并已在进行中

---

问题：您收到错误消息“表 N 的 RLV 存储库合并已在进行中”。

#### 解释

此表已经有一个 RLV 存储库合并正在进行中。RLV 表每次仅允许执行一个 RLV 合并。

## 在当前事务中无法打开请求的对象进行写入

---

问题：您收到一条错误消息，指示在当前事务中系统无法打开请求的对象进行写入，且另一用户已将行锁定。

### 解释

运行手动合并时，多个连接会尝试更新表中的同一行。表级版本控制事务存在，且拥有对表的写锁定。

## 事务似乎挂起

---

问题：事务似乎挂起。

### 解释

可能有多个连接正在表级写入锁上等待。如果在触发 RLV 存储库和 IQ 主存储库的前台（阻塞）合并时事务正在进行中，则这些事务看起来似乎被挂起。要进一步解决此问题，请从另一个连接运行相应的存储过程以查看是否有正被持有的锁或是否有合并正在执行中。

## 失败的 RLV 恢复

---

您在恢复期间遇到问题，例如：从磁盘读取页面时出现校验和错误；页头/页尾的序号不匹配；从磁盘读取页面时发生 OS 异常。

恢复分为四个高级阶段执行：

1. 初始化（扫描 `SYSIQRVLOG` 表并装载日志标识块。如果存在日志页面，则将表添加到恢复列表）。
2. 提交日志分析。
3. 表日志分析。
4. 重新进行属于已提交事务的操作。

在阶段 1、3 或 4 发生的恢复错误将导致 IQ 服务器关闭。阶段 2 中发生的错误通过延长阶段 3 来处理。

建议

1. 使用两个服务器启动开关限制访问：
  - 使用 `-gd DBA` 以便只有具备 `SERVER OPERATOR` 系统特权的用户才能在正在运行的服务器上启动和停止数据库。
  - 使用 `-gm 1` 可允许单个连接外加一个超限 DBA 连接，这样 DBA 便可在紧急情况下进行连接并断开其它连接。



2. 将 `-iqrrec_bypass = 1` 设置为绕过所有 RLV 恢复。此选项用于紧急情况下的修复，例如删除有问题的 RLV 表。按照目前的实施情况，这将禁止进一步的日志记录，但在代码中不会有其它将阻止常规 RLV 操作的校验。正因为如此，如果允许非 DBA 用户进行常规访问，则此模式有可能不稳定。
3. 手动建立数据库的一致性并在不一致时加以纠正。
4. 执行以下语句可删除表的 RLV 部分。这可能导致数据库不一致，但允许后续恢复。

```
delete * from table where rowid() > 2^28
```

---

**注意：** 表的 RLV 部分中的所有数据都将丢失。

---

5. 通过常规恢复重新启动。



## 附录：SQL 参考

本文中提及的 SQL 语句、数据库选项、函数、系统过程、系统表以及视图的参考资料。

### 数据库选项

---

数据库选项可以自定义并修改数据库行为。

#### **AGGREGATION\_PREFERENCE** 选项

控制用于处理集合的算法的选择。

*允许值*

-6 到 6

*默认值*

0

*范围*

可在数据库 (PUBLIC) 或用户级别设置选项。在数据库级别进行设置时，值将变为任何新用户的缺省值，但不会对现有用户产生任何影响。在用户级别进行设置时，仅替换该用户的 PUBLIC 值。为自身设置选项无需任何系统特权。在数据库或用户级别为任何其他用户设置选项都需要系统特权。

必须具有 SET ANY PUBLIC OPTION 系统特权才能设置此选项。可针对个别连接或 PUBLIC 角色进行临时设置。设置立即生效。

*注释*

对于查询中的集合 (**GROUP BY**、**DISTINCT**、**SET** 函数)，SAP Sybase IQ 优化程序可以选择使用多种算法对该集合进行相应处理。**AGGREGATION\_PREFERENCE** 使您可以覆盖优化程序在选择算法时的开销决策。该选项不会覆盖用于确定算法在查询引擎内是否合法的内部规则。

此选项通常用于内部测试，以及优化程序处理不好的手动调优查询。只有经验丰富的 DBA 才能使用。如果需要设置 **AGGREGATION\_PREFERENCE**，请通知 SAP Sybase 技术支持部门，因为设置此选项可能意味着相应地更改优化程序。

值	操作
0	让优化程序选择
1	选用已进行排序的集合

值	操作
2	选用采用 IQ 索引的集合
3	选用具有散列的集合
4	首选已进行非重复/分组排序的集合
5	如果分组列包含散列分区表的所有分区键，则首选已进行排序的集合。
6	如果分组列包含散列分区表的所有分区键，则首选具有散列的集合。
-1	不选用已进行排序的集合
-2	不选用采用 IQ 索引的集合
-3	不选用具有散列的集合
-4n	不选用已进行非重复/分组排序的集合
-5	如果分组列包含散列分区表的所有分区键，则不选用已进行排序的集合。
-6	如果分组列包含散列分区表的所有分区键，则不选用具有散列的集合。

## ALLOW\_SNAPSHOT\_VERSIONING 选项

适用于数据库中的所有基表（而不是仅适用于启用 RLV 的表）。将所有基表的表版本控制限制为表级或行级快照版本控制。此选项不适用于 IQ 目录存储库。

### 允许值

any、table-level、row-level

- **any** - 对快照版本控制没有限制。
- **row-level** - 仅允许行级快照版本控制。任何试图使用表级版本控制来修改表的事务都将失败，并且系统会提示非法的快照隔离错误。
- **table-level** - 仅允许表级快照版本控制。任何试图使用行级版本控制来修改表的事务都将失败，并且系统会提示非法的快照隔离错误。

### 缺省值

any

### 范围

只能在数据库 (PUBLIC) 级别设置选项。

必须具有 SET ANY SYSTEM OPTION 系统特权才能设置此选项。设置立即生效。

## **BASE TABLES IN RLV STORE 选项**

在 RLV 存储库中注册新建表，从而启用行级版本控制。启用 RLV 的表适合进行多写入程序并发访问。可以使用 **CREATE\_TABLE** 语句在表级别覆盖此设置。

*允许值*

ON、OFF

*缺省值*

OFF

*范围*

只能在数据库 (PUBLIC) 级别设置选项。

必须具有 **SET ANY SYSTEM OPTION** 系统特权才能设置此选项。设置立即生效。

*注释*

设置为 ON 时，将在 RLV 存储库中注册新建表。启用 RLV 的表将针对实时更新进行优化。

**CREATE\_TABLE** 语句的 { ENABLE | DISABLE } RLV STORE 子句始终覆盖 **BASE\_TABLES\_IN\_RLV\_STORE** 选项。

一旦启用了 **Base Tables in RLV STORE option**，所有新建的 IQ 基表都将自动启用 RLV。启用此选项不会影响到现有的 IQ 基表。

## **BLOCKING 选项**

控制响应锁定冲突的行为。**Multiplex** 辅助级节点不支持 **BLOCKING**。

*允许值*

ON、OFF

*默认值*

OFF

*范围*

可在数据库 (PUBLIC) 或用户级别设置选项。在数据库级别进行设置时，值将变为任何新用户的缺省值，但不会对现有用户产生任何影响。在用户级别进行设置时，仅替换该用户的 **PUBLIC** 值。为自身设置选项无需任何系统特权。在数据库或用户级别为任何其他用户设置选项都需要系统特权。

必须具有 **SET ANY PUBLIC OPTION** 系统特权才能设置此选项。可针对个别连接或 **PUBLIC** 角色进行临时设置。设置立即生效。

*注释*

当 **BLOCKING** 设置为 **OFF** 时，如果某个事务尝试执行写入操作但被另一事务的读取锁阻止，则会收到一条错误。

将 **BLOCKING** 选项设置为 **ON** 后，如果事务试图获取锁定，而该锁定与另一事务包含的现有锁定冲突，则该事务要等待所有冲突的锁定都解除或达到 **blocking\_timeout**。如果该锁定在 **blocking\_timeout** 毫秒的时间内没有被释放，则会向正在等待的事务返回错误。

## **BLOCKING\_TIMEOUT** 选项

控制事务为获取锁而需要等待的时长。**Multiplex** 辅助节点不支持 **BLOCKING\_TIMEOUT**。

*允许值*

整数（以毫秒为单位）。

*缺省值*

0

*范围*

可在数据库 (**PUBLIC**) 或用户级别设置选项。在数据库级别进行设置时，值将变为任何新用户的缺省值，但不会对现有用户产生任何影响。在用户级别进行设置时，仅替换该用户的 **PUBLIC** 值。为自身设置选项无需任何系统特权。在数据库或用户级别为任何其他用户设置选项都需要系统特权。

必须具有 **SET ANY PUBLIC OPTION** 系统特权才能设置此选项。可针对个别连接或 **PUBLIC** 角色进行临时设置。设置立即生效。

*注释*

当 **blocking** 选项设置为 **on** 时，如果事务试图获取与现有锁冲突的锁，则该事务需等待指定的毫秒数以便释放相应的冲突锁。如果在 **blocking\_timeout** 毫秒的时间内，未释放该锁，则会向正在等待的事务返回一条错误。

将此选项设置为 0 会强制所有试图获取锁的事务等待所有冲突的事务释放各自的锁。

## **ENABLE\_ASYNC\_IO** 选项

DBA 使用此选项启用或禁用 **RLV** 持久性日志使用的异步 **IO**。

*允许值*

**TRUE**、**FALSE**

如果要更改值，需要关闭数据库然后重新打开，或重新启动服务器。

*缺省值*

TRUE

*范围*

可在数据库 (PUBLIC) 或用户级别设置选项。在数据库级别进行设置时，值将变为任何新用户的缺省值，但不会对现有用户产生任何影响。在用户级别进行设置时，仅替换该用户的 PUBLIC 值。为自身设置选项无需任何系统特权。在数据库或用户级别为任何其他用户设置选项都需要系统特权。

必须具有 SET ANY PUBLIC OPTION 系统特权才能设置此选项。可针对个别连接或 PUBLIC 角色进行临时设置。如果允许，可针对其他任意用户或角色设置，或通过角色为所有用户设置。设置立即生效。

## **LOG\_DEADLOCKS 选项**

控制是打开还是关闭死锁报告功能。

*允许值*

On、Off

*缺省值*

Off

*范围*

只能在数据库 (PUBLIC) 级别设置选项。

必须具有 SET ANY SYSTEM OPTION 系统特权才能设置此选项。设置立即生效。

*注释*

当此选项设置为 On 时，数据库服务器在内部缓冲区中记录有关死锁的信息。缓冲区大小是固定的，即 10000 字节。您可以使用 sa\_report\_deadlocks 存储过程来查看死锁信息。当此选项设置为 Off 时，缓冲区的内容会被保留。

发生死锁后，仅报告死锁中涉及的那些连接的信息。报告连接的顺序取决于连接等待行的情况。对于线程死锁，将报告有关所有连接的信息。

死锁报告功能打开时，您还可以使用 Deadlock 系统事件在发生死锁时执行操作。

## **RV\_AUTO\_MERGE\_EVAL\_INTERVAL 选项**

此选项用于配置一个评测期，系统根据此评测期来确定应在何时对行级版本控制 (RLV) 存储库和 IQ 主存储库执行自动合并。

*允许值*

1 - MAX\_UINT (分钟)

*缺省值*

15 (分钟)

*范围*

只能在数据库 (PUBLIC) 级别设置选项。

必须具有 SET ANY SYSTEM OPTION 系统特权才能设置此选项。设置立即生效。

*注释*

此选项用于配置合并评测程序两次激活间的等待期 (以分钟为单位)。合并评测程序将对照配置的阈值检查每个启用行级版本控制 (RLV) 的表的合并参数, 以确定是否应在非阻塞模式下 (后台) 将 RLV 表合并到 IQ 主存储库。

如果在评测程序激活期间时间间隔结束, 或有合并已在进行中, 则重置时间间隔。

任何新的时间间隔值将在下次激活合并评测程序时使用。

## **RV\_MAX\_ACTIVE\_SUBFRAGMENT\_COUNT 选项**

该值能最大化地利用计算机上的内核数量。

*允许值*

$\geq 0$

*缺省值*

0

---

**注意:** 不建议使用缺省值以外的任何其它值, 否则可能对 CPU 利用率和批量装载的可伸缩性造成负面影响。

---

*范围*

只能在数据库 (PUBLIC) 级别设置选项。

必须具有 SET ANY SYSTEM OPTION 系统特权才能设置此选项。该选项在首次对表执行写入操作时生效。如果该选项的值在第一次写入操作后发生变更, 重新启动服务器后新值才会对表生效。

*注释*

如果该值设置为缺省值外的其它值, 系统将使用指定的值或计算机中内核的总数量 (取两者中的较小值)。



## RV\_MERGE\_NODE\_MEMSIZE 选项

根据合并阈值（包括 **RV\_MERGE\_NODE\_MEMSIZE**）对行级版本控制 (RLV) 存储库和 IQ 主存储库进行自动合并。如果超出此节点阈值，则将触发合并。

*允许值*

0 - 100（百分比）

*缺省值*

75（百分比）

*范围*

只能在数据库 (PUBLIC) 级别设置选项。

必须具有 **SET ANY SYSTEM OPTION** 系统特权才能设置此选项。设置立即生效。

*注释*

此选项设置的是 RLV 总内存大小百分比，以用作节点的合并阈值。如果 RLV 总内存大小超出此阈值，则合并条件评测程序将确定合并哪些表。如果必须合并多个表才能满足该节点阈值，则对要合并的每个表触发并行合并。

## RV\_MERGE\_TABLE\_MEMPERCENT 选项

根据合并阈值（包括 **RV\_MERGE\_TABLE\_MEMPERCENT**）对行级版本控制 (RLV) 存储库和 IQ 主存储库进行自动合并。如果超出此表阈值，则对特定表触发合并。

*允许值*

0 - 100（百分比）

*缺省值*

0（百分比）

---

**注意：** 如果 **RV\_MERGE\_TABLE\_MEMPERCENT = 0**，则系统针对每个表使用的阈值为  $100\% / N$ ，其中 N 代表已装载的启用 RLV 的表数。

---

*范围*

只能在数据库 (PUBLIC) 级别设置选项。

必须具有 **SET ANY SYSTEM OPTION** 系统特权才能设置此选项。设置立即生效。

*注释*

此选项将设置一个内存占用百分比，以用作启用 RLV 的表的合并阈值。如果占用的内存超出此阈值，则将发生合并。

系统将逐个表地评测是否合并 RLV 存储库和 IQ 主存储库。它将枚举所有加载的 RLV 表，针对每个表来确定是否有必要进行合并。在以下情况下，将认为有必要对单个表进行合并：

1. 该表违反内存阈值 (RV\_MERGE\_TABLE\_MEMPERCENT) 或行阈值 (RV\_MERGE\_TABLE\_NUMROWS)；
2. 系统未检测到有很大比例的 RLV 行未提交，从而无法被合并。

## **RV\_MERGE\_TABLE\_NUMROWS 选项**

根据合并阈值（包括 **RV\_MERGE\_TABLE\_NUMROWS**）对行级版本控制 (RLV) 存储库和 IQ 主存储库进行自动合并。如果超出此表阈值，则对特定表触发合并。

*允许值*

1000 - 100000000

*缺省值*

10000000

*范围*

只能在数据库 (PUBLIC) 级别设置选项。

必须具有 SET ANY SYSTEM OPTION 系统特权才能设置此选项。设置立即生效。

*注释*

此选项可设置用作启用 RLV 表的合并阈值的行数。如果使用的行数超出此阈值，则将发生合并。

系统将逐个表地评测是否合并 RLV 存储库和 IQ 主存储库。它将枚举所有加载的 RLV 表，针对每个表来确定是否有必要进行合并。在以下情况下，将认为有必要对单个表进行合并：

1. 该表违反内存阈值 (RV\_MERGE\_TABLE\_MEMPERCENT) 或行阈值 (RV\_MERGE\_TABLE\_NUMROWS)；
2. 系统未检测到有很大比例的 RLV 行未提交，从而无法被合并。

## **RV\_RESERVED\_DBSPACE\_MB 选项**

必须为关键操作期间数据结构使用的内存预留一部分 RLV 存储库。

*允许值*

大于或等于 50 的整数（以兆字节为单位）

*缺省值*

50 Mb 和 RLV dbspace 大小值一半中的较小值

*范围*

只能在数据库 (PUBLIC) 级别设置选项。

必须具有 **SET ANY SYSTEM OPTION** 系统特权才能设置此选项。设置立即生效。不需要重新启动服务器即可更改保留空间大小。

#### 描述

此选项允许您控制在 **RLV** 存储库中为在发布保存点、提交和回退操作期间所使用的某些小型但关键的数据结构留出的空间量。

## **SNAPSHOT\_VERSIONING** 选项

仅适用于启用 **RLV** 的表（而不是适用于数据库中的所有基表）。用于管理是使用单写入程序表级版本控制还是使用多写入程序行级版本控制来访问启用 **RLV** 的表。此选项不适用于 **IQ** 目录存储库。

#### 允许值

值	操作
row-level	对启用 <b>RLV</b> 的表启用并发写入程序访问和行级版本控制。 修改表行的第一个事务将建立一个行写入锁，该锁的作用将一直持续到事务结束。 试图修改锁定行的后续事务将失败并提示锁定/未来版本错误消息，或基于 <b>BLOCKING</b> 选项的值发生阻塞，直到锁被释放。
table-level	启用单写入程序访问和表级版本控制。 访问表的第一个事务将建立一个表写入锁，该锁的作用将一直持续到事务结束。 试图对锁定表执行写入操作的后续事务将失败并提示锁定/未来版本错误消息，或基于 <b>BLOCKING</b> 选项的值发生阻塞，直到锁被释放。

#### 缺省值

table-level

#### 范围

可在数据库 (**PUBLIC**) 或用户级别设置选项。在数据库级别进行设置时，值将变为任何新用户的缺省值，但不会对现有用户产生任何影响。在用户级别进行设置时，仅替换该用户的 **PUBLIC** 值。为自身设置选项无需任何系统特权。在数据库或用户级别为任何其他用户设置选项都需要系统特权。

必须具有 **SET ANY PUBLIC OPTION** 系统特权才能设置此选项。设置立即生效。

## 过程

在 SAP Sybase IQ 数据库中使用系统提供的存储过程来检索系统信息。

**sa\_conn\_info** 系统过程

报告连接属性信息。

## 语法

```
sa_conn_info( [ connidparm ] )
```

## 参数

- **connidparm** – 此可选 INTEGER 参数指定连接 ID 号。缺省值为 NULL。

## 结果集

列名	数据类型	说明
Number	INTEGER	返回当前连接的连接 ID (数字)。
Name	VARCHAR(255)	返回当前连接的连接 ID (数字)。 临时连接名在连接名前面附有 INT:。
Userid	VARCHAR(255)	返回连接的用户 ID。
DBNumber	INTEGER	返回数据库的 ID 编号。
LastReqTime	VARCHAR(255)	返回指定连接的最后一个请求开始的时间。此属性可以为内部连接 (如事件) 返回空字符串。
ReqType	VARCHAR(255)	返回上次请求的类型。如果连接已由连接池进行了高速缓存, 则其 ReqType 值为 CONNECT_POOL_CACHE。
CommLink	VARCHAR(255)	返回连接的通信链接。这是 SAP Sybase IQ 所支持的网络协议之一, 或者 local, 如果为相同计算机连接。
NodeAddr	VARCHAR(255)	返回客户端/服务器连接中客户端的地址。
ClientPort	INTEGER	返回客户端的 TCP/IP 端口号; 或者如果连接不是 TCP/IP 连接, 则返回 0。
ServerPort	INTEGER	返回数据库服务器的 TCP/IP 端口号或 0。
BlockedOn	INTEGER	如果当前连接未被阻塞, 则返回零; 如果当前连接被阻塞, 则返回由于锁定冲突而阻塞连接的连接号。
LockRowID	UNSIGNED BIGINT	返回锁定行的标识符。 如果连接并未等待与行相关联的锁 (即它未等待锁或正在等待不与行相关联的锁), 则 LockRowID 为 NULL。

列名	数据类型	说明
LockIndexID	INTEGER	返回锁定索引的标识符。 如果锁与 LockTable 中的表上的所有索引相关联，则 LockIndexID 为 -1。如果连接并未等待与索引相关联的锁（即它未等待锁或正在等待不与索引相关联的锁），则 LockRowID 为 NULL。
LockTables	VARCHAR(255)	如果连接正在等待锁，则返回与锁相关联的表名称。否则，LockTable 返回一个空字符串。
UncommitOps	INTEGER	返回未提交的操作数。
ParentConnection	INTEGER	返回已创建临时连接以执行数据库操作（如执行备份或创建数据库）的连接的连接 ID。对于其它类型的连接，此属性返回 NULL。

### 注释

如果 *connidparm* 小于零，则返回由当前连接的连接属性组成的结果集。如果 *connidparm* 未提供或为 NULL，则为数据库服务器上运行的所有数据库的所有连接返回连接属性。

当发生阻塞时，使用此过程所返回的 **BlockedOn** 值可以检查哪些用户被阻塞以及是谁阻塞的。*sa\_locks* 系统过程可用于显示阻塞的连接所持有的锁。

有关基于任意这些属性的详细信息，可执行与以下内容类似的内容：

```
SELECT *, DB_NAME( DBNumber ),
        CONNECTION_PROPERTY( 'LastStatement', Number )
FROM sa_conn_info( );
```

LockRowID 的值可用于在 *sa\_locks* 过程的输出中查找锁。

LockIndexID 的值可用于在 *sa\_locks* 过程的输出中查找锁。此外，LockIndexID 的值对应于 **ISYSIDX** 系统表的主键，可使用 **SYSIDX** 系统视图查看此系统表。

每个锁都具有关联的表，因此，LockTable 的值可用于明确地确定某连接是否正在等待锁。

### 特权

为当前连接 ID 执行此系统过程不需要任何特权。要为其它连接执行此系统过程，您必须具有 **SERVER OPERATOR**、**MONITOR** 或 **DROP CONNECTION** 系统特权。

### 副作用

无

## 示例

以下示例使用 `sa_conn_info` 系统过程为连接到服务器的所有连接返回一个汇总连接属性的结果集。

```
CALL sa_conn_info( );
```

Number	Name	Userid	DBNumber	...
79	SQL_DBC_10dcf810	DBA	0	...
46	setup	User1	0	...
...	...	...	...	...

以下示例使用 `sa_conn_info` 系统过程返回结果集，其中显示是哪个连接创建了临时连接。

```
SELECT Number, Name, ParentConnection FROM sa_conn_info( );
```

Connection 8 创建了执行 `CREATE DATABASE` 语句的临时连接。

Number	Name	ParentConnection
1000000048	INT: CreateDB	8
9	SQL_DBC_14675af8	(NULL)
8	SQL_DBA_152d5ac0	(NULL)

## sa\_report\_deadlocks 系统过程

从数据库服务器创建的内部缓冲区中检索有关死锁的信息。

### 语法

```
sa_report_deadlocks( )
```

### 特权

您必须具有系统过程的 `EXECUTE` 特权，以及 `MONITOR` 系统特权。

### 注释

列名	数据类型	描述
snapshotId	BIGINT	死锁实例（与特定死锁相关的所有行都具有相同的 ID）。
snapshotAt	TIMESTAMP	发生死锁的时间。
waiter	INT	正等待的连接的连接句柄。
who	VARCHAR(128)	与正等待的连接关联的用户 ID。

列名	数据类型	描述
what	LONG VARCHAR	正等待的连接所执行的命令。 仅当通过在数据库服务器命令行中指定 -zl 选项打开捕获最新准备的 SQL 语句的功能时，此信息才可用。
object_id	UNSIGNED BIGINT	包含该行的表的对象 ID。
record_id	BIGINT	系统表的行 ID。
owner	INT	拥有正等待的锁的连接的连接句柄。
is_victim	BIT	标识回退的事务。
rollback_operation_count	UNSIGNED INT	未提交操作的数量，如果事务回退，这些操作有可能丢失。
iq_rid	UNSIGNED BIGINT	启用 IQ RLV 的表的行 ID。
iq_txn_id	UNSIGNED BIGINT	关联行的事务 ID。

当 `log_deadlocks` 选项设置为 On 时，数据库服务器在内部缓冲区中记录有关死锁的信息。可以使用 `sa_report_deadlocks` 系统过程查看日志中的信息。

*副作用*  
无。

## **sa\_server\_option** 系统过程

当服务器正在运行时替换服务器选项。

*语法*

```
sa_server_option( opt , val )
```

*参数*

- **opt** - 使用此 CHAR(128) 参数指定服务器选项名称。
- **val** - 使用此 CHAR(128) 参数为服务器选项指定新值。

*特权*

您必须具有系统过程的 EXECUTE 特权。您还必须具有以下选项与应用程序分析或请求记录相关，要使用这些选项，您必须具有 MANAGE PROFILING 系统特权：

- ProcedureProfiling
- ProfileFilterConn
- ProfileFilterUser

- RequestFilterConn
- RequestFilterDB
- RequestLogFile
- RequestLogging
- RequestLogMaxSize
- RequestLogNumFiles

## 注释

选项名称	值	其它信息
AutoMultiProgrammingLevel	YES、NO	缺省值为 YES。 设置为 YES 时，数据库服务器会自动调整其进程并发水平，该属性控制可以同时处于活动状态的最大任务数。如果通过将该选项设置为 NO 来选择手动控制进程并发水平，则还可以设置进程并发水平的初始值、最小值和最大值。
AutoMultiProgrammingLevelStatistics	YES、NO	缺省值为 NO。 设置为 YES 时，进程并发水平自动调整的统计信息将显示在数据库服务器消息日志中。
CacheSizingStatistics	YES、NO	缺省值为 NO。 当设置为 YES 时，每当高速缓存大小发生变化时，数据库服务器消息窗口中就会显示高速缓存信息。
CollectStatistics	YES、NO	缺省值为 YES。 当设置为 YES 时，数据库服务器收集性能监控器统计信息。
ConnsDisabled	YES、NO	缺省值为 NO。 当设置为 YES 时，不允许与数据库服务器上的任何数据库建立任何其它连接。
ConnsDisabled-ForDB	YES、NO	缺省值为 NO。 当设置为 YES 时，不允许与当前数据库建立任何其它连接。
ConsoleLogFile	filename	用来记录数据库服务器消息日志信息的文件的名称。指定一个空字符串会停止将信息记录到文件。由于此值为 SQL 字符串，所以路径中的所有反斜线字符均需写两遍。



选项名称	值	其它信息
ConsoleLogMaxSize	file-size (以字节为单位)	用来记录数据库服务器消息日志信息的文件的最大大小 (以字节为单位)。当数据库服务器消息日志文件达到该属性或 <code>-on</code> 服务器选项指定的大小后, 通过附加扩展名 <code>.old</code> 重命名该文件 (如果已经存在具有相同名称的文件, 则将其替换)。然后, 重新启动该数据库服务器消息日志文件。
CurrentMultiProgrammingLevel	整数	缺省值为 20。 设置数据库服务器的进程并发水平。
DatabaseCleaner	ON、OFF	缺省值为 ON。 除非有技术支持部门的建议, 否则不要更改此选项的设置。
DeadlockLogging	ON、OFF、RESET、CLEAR	缺省值为 OFF。 控制死锁记录。同时支持值 <code>deadlock_logging</code> 。支持以下值: <ul style="list-style-type: none"> <li>• <b>ON</b> - 启用死锁记录。</li> <li>• <b>OFF</b> - 禁用死锁记录, 但使死锁数据可供查看。</li> <li>• <b>RESET</b> - 清除记录的死锁数据 (如果存在的话), 然后启用死锁记录。</li> <li>• <b>CLEAR</b> - 清除记录的死锁数据 (如果存在的话), 然后禁用死锁记录。</li> </ul> 启用死锁记录后, 可以使用 <code>sa_report_deadlocks</code> 系统过程从数据库中检索死锁信息。
DebuggingInformation	YES、NO	缺省值为 NO。 显示诊断消息和其它消息以用于疑难解答。消息显示在数据库服务器消息窗口中。
DiskSandbox	ON、OFF	缺省值为 OFF。 为在数据库服务器上启动的所有没有显式磁盘沙箱设置的数据库设置缺省磁盘沙箱设置。使用 <code>sa_server_option</code> 系统过程更改磁盘沙箱设置并不会影响数据库服务器上已运行的数据库。要使用 <code>sa_server_option</code> 系统过程更改磁盘沙箱设置, 必须为 <code>manage_disk_sandbox</code> 安全功能提供安全功能密钥。

选项名称	值	其它信息
DropBadStatistics	YES、NO	缺省值为 YES。 允许通过自动统计信息管理删除返回数据库错误估计的统计信息。
DropUnusedStatistics	YES、NO	缺省值为 YES。 允许通过自动统计信息管理删除数据库中连续 90 天尚未使用的统计信息。
IdleTimeout	整数（以分钟为单位）	缺省值为 240。 断开未在指定分钟内提交请求的 TCP/IP 连接。这可防止不活动的连接无限期地持有锁。
IPAddressMonitorPeriod	整数（以秒为单位）	最小值为 10，缺省值为 0。对于便携式设备，缺省值为 120。 设置检查新 IP 地址的时间（以秒为单位）。
LivenessTimeout	整数（以秒为单位）	缺省值为 120。 定期通过客户端/服务器 TCP/IP 网络发送活动包，用以确认连接的完整性。如果网络服务器运行了一个 LivenessTimeout 周期但没有检测到活动包，则切断通信。
MaxMultiProgrammingLevel	整数	缺省值是 CurrentMultiProgrammingLevel 值的四倍。 设置数据库服务器进程并发水平的最大值。
MessageCategoryLimit	整数	缺省值为 400。 设置可使用 sa_server_messages 系统过程检索的各个严重级和类别的最小消息数。
MinMultiProgrammingLevel	整数	缺省值是 -gtc 服务器选项值和计算机上逻辑 CPU 数量这二者中的最小值。

选项名称	值	其它信息
OptionWatchAction	MESSAGE、 ERROR	<p>缺省值为 MESSAGE。</p> <p>指定数据库服务器在尝试设置列表中的选项时所采取的操作。如果 OptionWatchAction 设置为 MESSAGE，并且设置了一个由 OptionWatchList 指定的选项，数据库服务器消息窗口中将显示一条消息，指示正在设置的选项位于选项监视列表中。如果 OptionWatchAction 设置为 ERROR，将返回一条错误，指示由于该选项位于选项监视列表中，从而无法设置该选项。</p> <p>可通过执行以下语句来查看此属性的当前设置：</p> <pre>SELECT DB_PROPERTY( 'OptionWatchAction' );</pre>
OptionWatchList	以逗号分隔的数据库选项列表	<p>指定一个以逗号分隔的数据库选项列表，在设置其中的选项时数据库服务器会向您发出通知，或返回错误。字符串的长度不应超过 128 个字节。缺省情况下它是一个空字符串。例如，以下命令将 automatic_timestamp、float_as_double 和 tsq_hex_constant 选项添加到监视选项列表中。</p> <pre>CALL sa_server_option( 'OptionWatchList','automatic_timestamp, float_as_double,tsq_hex_constant' );</pre> <p>可通过执行以下语句来查看此属性的当前设置：</p> <pre>SELECT DB_PROPERTY( 'OptionWatchList' );</pre>
ProcedureProfiling	YES、NO、 RESET、CLEAR	缺省值为 NO。
ProfileFilterConn	connection-id	<p>指示数据库服务器捕获指定连接 ID 的分析信息，而不阻止其它连接使用该数据库。启用连接过滤后，SELECT PROPERTY('ProfileFilterConn') 返回的值是被监控连接的连接 ID。如果未指定 ID 或禁用连接过滤，则返回的值为 -1。</p>

选项名称	值	其它信息
ProcessorAffinity	以逗号分隔的处理器编号和/或范围列表。缺省设置是使用所有处理器或 <code>-gta</code> 选项的设置。	<p>指示数据库服务器在 Windows 或 Linux 上使用哪些逻辑处理器。指定以逗号分隔的处理器编号和/或范围列表。如果省略范围下限，则假定为零。如果省略范围上限，则假定为操作系统已知的最高 CPU。sa_cpu_topology 系统过程返回的 in_use 列中包含数据库服务器的当前处理器关联，in_use 列还指示数据库服务器是否正在使用处理器。另外，也可以查询 ProcessorAffinity 数据库服务器属性的值。</p> <p>在以下情况下数据库服务器可能不会使用所有指定的逻辑处理器：</p> <ul style="list-style-type: none"> <li>• 如果一个或多个指定的逻辑处理器不存在或处于脱机状态。</li> <li>• 如果许可证不允许。</li> </ul> <p>如果指定的处理器 ID 无效，则 sa_server_option 返回一条错误。</p>
ProfileFilterUser	user-id	指示数据库服务器捕获特定用户 ID 的分析信息。
QuittingTime	有效的日期和时间	指示数据库服务器在指定的时间关闭。
RememberLastPlan	YES、NO	<p>缺省值为 NO。</p> <p>指示数据库服务器捕获在连接上执行的最后一个查询的详细文本计划。此设置也由 <code>-zp</code> 服务器选项控制。启用 RememberLastPlan 时，通过查询 LastPlanText 连接属性的值来获取在连接上执行的最后一个查询计划的文本表示：</p> <pre>SELECT CONNECTION_PROPERTY( 'Last-PlanText' );</pre>

选项名称	值	其它信息
RememberLastStatement	YES、NO	<p>缺省值为 NO。</p> <p>指示数据库服务器为在服务器上运行的每个数据库捕获最近预准备 SQL 语句。对于存储过程调用，仅显示最外面的过程调用，不显示过程中的语句。当启用 RememberLastStatement 时，可以通过查询 LastStatement 连接属性的值为连接获取 LastStatement 的当前值：</p> <pre>SELECT CONNECTION_PROPERTY( 'Last-Statement' );</pre> <p>如果启用客户端语句高速缓存并且重用了某个高速缓存的语句，则此属性返回空字符串。当启用 RememberLastStatement 时，下面的语句为指定连接返回最新准备的语句：</p> <pre>SELECT CONNECTION_PROPERTY( 'Last-Statement', connection-id );</pre> <p>sa_conn_activity 系统过程为所有连接都返回此相同的信息。</p> <p><b>注意：</b> 当指定 -zl 或启用 RememberLastStatement 服务器设置时，任何用户都可以调用 sa_conn_activity 系统过程或获取 LastStatement 连接属性的值，以找到为任何其他用户最新准备的 SQL 语句。应慎用此选项，并在不需要时将其关闭。</p>
RequestFilterConn	connection-id、-1	<p>对请求记录信息进行过滤以便只记录特定连接的信息。在对具有多个活动连接或多个数据库的数据库服务器进行监控时，此过滤功能可减小请求日志文件的大小。可通过执行以下语句来获得连接 ID：</p> <pre>CALL sa_conn_info( );</pre> <p>获得连接 ID 后，即可执行以下语句来记录特定连接：</p> <pre>CALL sa_server_option( 'RequestFilterConn', connection-id );</pre> <p>在显式地重置过滤或关闭数据库服务器之前，过滤将一直有效。要重置过滤，请使用以下语句：</p> <pre>CALL sa_server_option( 'RequestFilterConn', -1 );</pre>

选项名称	值	其它信息
RequestFilterDB	database-id、-1	<p>对请求记录信息进行过滤以便只记录特定数据库的信息。在对具有多个数据库的服务器进行监控时，此过滤功能可减小请求日志文件的大小。在连接到所需的数据库后，可通过执行以下语句来获得数据库 ID:</p> <pre>SELECT CONNECTION_PROPERTY( 'DBNumber' );</pre> <p>要仅记录特定数据库的信息，请执行以下语句:</p> <pre>CALL sa_server_option( 'RequestFilterDB', database-id );</pre> <p>在显式地重置过滤或关闭数据库服务器之前，过滤将一直有效。要重置过滤，请使用以下语句:</p> <pre>CALL sa_server_option( 'RequestFilterDB', -1 );</pre>
RequestLogFile	filename	<p>用来记录请求信息的文件名。指定一个空字符串会停止将请求信息记录到请求日志文件。如果启用请求记录，但请求日志文件未指定或设置为空字符串，则服务器会将请求记录到数据库服务器消息窗口中。由于此值为 SQL 字符串，所以路径中的所有反斜线字符均需写两遍。</p> <p>启用客户端语句高速缓存时，如果要使用 <b>tracetime.pl</b> Perl 脚本对请求日志进行分析，应将 <b>max_client_statements_cached</b> 选项设置为 0，以便在捕获请求日志时禁用客户端语句缓存。</p>

选项名称	值	其它信息
RequestLogging	SQL、 HOSTVARS、 PLAN、 PROCEDURES、 TRIGGERS、 OTHER、 BLOCKS、 REPLACE、 ALL、YES、 NONE、NO	<p>缺省值为 NONE。</p> <p>此调用启用对发送到数据库服务器的各 SQL 语句的记录，以便结合数据库服务器选项 <code>-zr</code> 和 <code>-zo</code> 一起用来进行疑难解答。值可以是以下项的组合，这些项由加号 (+) 或逗号分隔：</p> <ul style="list-style-type: none"> <li>• <b>PLAN</b> - 启用执行计划的记录（简单形式）。如果启用过程 (PROCEDURES) 的记录，还将记录过程的执行计划。</li> <li>• <b>HOSTVARS</b> - 启用主机变量值的记录。如果指定 HOSTVARS，还将记录针对 SQL 列出的信息。</li> <li>• <b>PROCEDURES</b> - 启用在过程中执行的语句的记录。</li> <li>• <b>TRIGGERS</b> - 启用在触发器中执行的语句的记录。</li> <li>• <b>OTHER</b> - 启用 SQL 不包含的其它请求类型（如 FETCH 和 PREFETCH）的记录。然而，如果您指定 OTHER 但未指定 SQL，则其等效于指定 SQL+OTHER。包括 OTHER 可导致日志文件迅速增大，并可能对服务器性能造成负面影响。</li> <li>• <b>BLOCKS</b> - 启用显示何时在另一个连接上阻塞和解除阻塞某个连接的详细信息的记录。</li> <li>• <b>REPLACE</b> - 在记录开始时，使用同名的新（空）请求日志来替换现有的请求日志。否则，将打开现有的请求日志，并将新条目附加到文件的结尾。</li> <li>• <b>ALL</b> - 记录所有支持的信息。此值等效于指定 SQL+PLAN+HOSTVARS+PROCEDURES+TRIGGERS+OTHER+BLOCKS。此设置可导致日志文件迅速增大，会对服务器的性能产生负面影响。</li> <li>• <b>NO 或 NONE</b> - 禁用记录请求日志。</li> </ul> <p>可通过执行以下语句来查看此属性的当前设置：</p> <pre>SELECT PROPERTY( 'RequestLogging' );</pre>

选项名称	值	其它信息
RequestLogMaxSize	file-size (以字节为单位)	用来记录请求记录信息的文件的最大大小 (以字节为单位)。如果指定为 0, 则对请求记录文件而言没有最大大小, 且该文件也决不会被重命名。这是缺省值。当请求日志文件达到 sa_server_option 系统过程或 -zs 服务器选项指定的大小后, 会用附加的扩展名 .old 对该文件进行重命名 (如果已经存在这样一个文件, 则替换具有相同名称的现有文件)。然后, 重新启动请求日志文件。
RequestLogNum-Files	整数	要保留的请求日志文件副本的数目。如果请求记录在较长一段时间内处于启用状态, 则请求日志文件可能会变得很大。-zn 选项可用于指定要保留的请求日志文件副本的数量。
RequestTiming	YES、NO	<p>缺省值为 NO。</p> <p>指示数据库服务器维护每个新连接的计时信息。缺省情况下, 此功能关闭。此功能打开时, 数据库服务器会为所有新连接维护累计计时器, 该计时器表明连接在服务器的各个状态下所花费的具体时间。此更改仅对新连接有效, 并且一直持续到每个连接结束。您可以使用 sa_performance_diagnostics 系统过程以获得此计时信息的汇总, 或者通过检查以下连接属性来检索各个值:</p> <ul style="list-style-type: none"> <li>• ReqCountUnscheduled</li> <li>• ReqTimeUnscheduled</li> <li>• ReqCountActive</li> <li>• ReqTimeActive</li> <li>• ReqCountBlockIO</li> <li>• ReqTimeBlockIO</li> <li>• ReqCountBlockLock</li> <li>• ReqTimeBlockLock</li> <li>• ReqCountBlockContention</li> <li>• ReqTimeBlockContention</li> </ul> <p>当 RequestTiming 服务器属性设置为 on 时, 为每个请求维护附加计数器会产生少量开销。</p>



选项名称	值	其它信息
rlv_auto_merge	ON、OFF	<p>缺省值为 ON。</p> <p>针对启用行级版本控制的表，启用或禁用 RLV 存储库与 IQ 主存储库的自动合并。</p> <p>如果 rlv_auto_merge 为 OFF，则 RLV 和 IQ 主存储库不会自动合并。这表示您将负责手动合并数据，这样在达到 rlv_memory_mb 上限之前，RLV 存储库能够与 IQ 主存储库保持同步。</p>
rlv_memory_mb	<p>最小值为 1 MB。</p> <p>最大值为 2048。</p> <p>任何其它值均会将内存量设置为 2048 MB。</p>	<p>指定为行级版本控制保留的最大内存量（RLV 存储库，以 MB 为单位）。缺省值为 2048 MB。如果此值超过系统虚拟内存限制的 2/3，服务器将生成错误。</p>
SecureFeatures	feature-list	<p>用于管理已经运行的数据库服务器的安全功能。feature-list 是以逗号分隔的功能名称或功能集的列表。通过向列表中添加功能可以限制功能的可用性。要从安全功能列表中删除项目，请在安全功能名称前指定一个减号 (-)。</p> <p>要调用 sa_server_option('SecureFeatures',...)，必须在相应连接上启用 ManageFeatures 安全功能。通过 -sf 密钥（系统安全功能密钥）启用 ManageFeatures 以及所有其它功能。因此，如果已使用系统安全功能密钥，则更改 SecureFeatures 的设置将不会对相关连接产生任何影响。但是，如果使用了其它密钥（例如，通过 create_secure_feature_key 系统过程创建的密钥），则更改可能会立即影响到相关连接，具体取决于该密钥中包含的其它功能。</p> <p>对于数据库服务器，为了允许或禁止对功能进行访问而所做的更改都会立即生效。执行 sa_server_option 系统过程的连接是否会受到影响，取决于该连接所使用的安全功能密钥以及此密钥是否允许该连接访问指定的功能。</p> <p>例如，可使用以下语法来保护两个功能的安全：</p> <pre>CALL sa_server_option('SecureFeatures', 'CONSOLE_LOG,WEBCLIENT_LOG');</pre> <p>执行上述语句后，安全功能列表将根据更改内容进行设置。</p>

选项名称	值	其它信息
StatisticsCleaner	ON、OFF	缺省值为 ON。 统计信息清理程序通过对表执行扫描修复提供错误估计的统计信息。缺省情况下，统计信息清理程序在后台运行，且对性能影响很小。 关闭统计信息清理程序不会禁用统计信息调控器，但关闭统计信息清理程序时，只有在运行查询时才能创建或修复统计信息。
WebClientLogFile	filename	Web 服务客户端日志文件的名称。每次使用 -zoc 服务器选项或 WebClientLogFile 属性设置或重置文件名时，Web 服务客户端日志文件都将被截断。由于此值为字符串，所以路径中的所有反斜线字符均需写两遍。
WebClientLogging	ON、OFF	缺省值为 OFF。 此选项可启用或禁用对 Web 服务客户端的记录。记录的信息包括 HTTP 请求和响应数据。指定 ON 将开始向 Web 服务客户端日志文件记录信息，而指定 OFF 可停止向该文件记录信息。

*副作用*  
无。

*示例*

以下语句的作用是，每当高速缓存大小发生变化，数据库服务器消息窗口中便会显示高速缓存信息：

```
CALL sa_server_option( 'CacheSizingStatistics', 'YES' );
```

以下语句禁止与当前数据库建立新连接：

```
CALL sa_server_option( 'ConnsDisabledForDB', 'YES' );
```

以下语句启用所有 SQL 语句、过程调用、计划、阻塞和取消阻塞事件的记录，并启动一个新请求日志：

```
CALL sa_server_option( 'RequestLogging', 'SQL+PROCEDURES+BLOCKS+PLAN+REPLACE' );
```

## sp\_iqcolumn 过程

显示有关数据库中的列的信息。

*语法 1*

```
sp_iqcolumn ( [ table_name ], [ table_owner ], [ table_loc ] )
```

## 语法 2

```
sp_iqcolumn [ table_name='table_name' ], [ table_owner='tableowner' ],
[table_loc='table_loc' ]
```

## 参数

- **语法 1** - 如果您指定 *table\_owner* 而不指定 *table\_name*，则必须用 NULL 代替 *table\_name*。例如，sp\_iqcolumn NULL, DBA。
- **语法 2** - 可按任意顺序指定参数。将 '*table\_name*' 和 '*table\_owner*' 用单引号引起来。

## 特权

您必须具有系统过程的 EXECUTE 特权。

## 注释

显示有关数据库中的列的信息。如果指定 *table\_name* 参数，则仅返回具有该名称的表的列。如果指定 *table\_owner* 参数，则仅返回该用户拥有的表。如果同时指定 *table\_name* 和 *table\_owner* 参数，则选择唯一表中的列（如果该表存在）。如果指定 *table\_loc*，则仅返回用该段类型定义的表。如果不指定任何参数，则返回数据库中所有表的所有列。**sp\_iqcolumn** 不返回系统表的列信息。

列名	描述
table_name	表的名称。
table_owner	表的所有者。
column_name	列的名称。
domain_name	数据类型。
width	具有精度和标度的数值数据类型的精度或不具有标度的数值数据类型的存储宽度；字符数据类型的宽度。
scale	数值数据类型的标度。
nulls	如果列可以包含 NULL 则为 "Y"，如果列不能包含 NULL 则为 "N"。
default	如果列是标识/自动增量列，则为 'Identity/Autoincrement'，如果不是，则为空。
cardinality	按索引进行的非重复计数（如果已知）。
location	TEMP = IQ 临时存储库，MAIN = IQ 主存储库，SYSTEM = 目录存储库。
isPartitioned	如果列属于分区表并有一个或多个分区，而列分区的 dbspace 与表分区的 dbspace 不同，则为 "Y"，如果列的表没有分区或列的每个分区驻留在与表分区相同的 dbspace 中，则为 "N"。

列名	描述
remarks	使用 <b>COMMENT</b> 语句添加的用户注释。
check	检查约束表达式。

### **sp\_iqcolumn** 过程示例

使用示例作为 **sp\_iqcolumn** 用法的参考。

以下两种语法变体都会返回 Departments 表中的所有列：

```
sp_iqcolumn Departments
```

```
call sp_iqcolumn (table_name='Departments')
```

table_name	table_owner	column_name	domain_name	width	scale
Departments	GROUPO	DepartmentID	integer	4	0
N	(NULL)				
Departments	GROUPO	DepartmentName	char	40	0
N	(NULL)				
Departments	GROUPO	DepartmentHead	integer	4	0
Y	(NULL)				
cardinality	location	isPartitioned	remarks	check	
5	Main	N	(NULL)	(NULL)	
0	Main	N	(NULL)	(NULL)	
0	Main	N	(NULL)	(NULL)	

以下语法变体会返回表所有者 DBA 拥有的所有表中的所有列。

```
sp_iqcolumn table_owner='DBA'
```

### **sp\_iqconnection** 过程

显示有关连接和版本的信息，包括哪些用户正在使用临时 **dbspace**、哪些用户正在使版本保持活动状态、连接在 **SAP Sybase IQ** 内部执行哪些操作、连接状态、数据库版本状态，等等。

*语法*

```
sp_iqconnection [ connhandle ]
```

*适用于*

Simplex 和 Multiplex。

*特权*

您必须具有系统过程的 **EXECUTE** 特权。必须具有以下一种系统特权：

- **DROP CONNECTION**
- **MONITOR**
- **SERVER OPERATOR**

**注释**

*connhandle* 相当于 Number 连接属性，是连接的 ID 编号。**connection\_property** 系统函数返回连接 ID：

```
SELECT connection_property ( 'Number' )
```

当使用有效的 *connhandle* 的输入参数进行调用时，**sp\_iqconnection** 仅对该连接返回一行。

**sp\_iqconnection** 为每个活动连接都返回一行。列 ConnHandle、Name、Userid、LastReqTime、ReqType、CommLink、NodeAddr 和 LastIdle 分别是连接属性 Number、Name、Userid、LastReqTime、ReqType、CommLink、NodeAddr 和 LastIdle，并返回与系统函数 **sa\_conn\_info** 相同的值。其它列返回来自 SAP Sybase IQ 引擎的 SAP Sybase IQ 端的连接数据。各行按 ConnCreateTime 排序。

MPXServerName 列存储与节点间通信 (INC) 有关的信息，如下所示：

运行服务器	MPXServerName 列内容
Simplex 服务器	NULL (所有连接均为本地/用户连接)
Multiplex 协调器	<ul style="list-style-type: none"> <li>NULL 用于本地/用户连接。</li> <li>包含每个 INC 连接 (根据需要或专用活动连接) 的辅助节点服务器名称 (连接源)。</li> </ul>
Multiplex 辅助服务器	<ul style="list-style-type: none"> <li>NULL 用于本地/用户连接。</li> <li>包含协调器服务器名的值 (连接源)。</li> </ul>

在 Java 应用程序中，请在 "RemotePWD" 字段中指定来自 TDS 客户端的特定于 SAP Sybase IQ 的连接属性。以下示例显示如何指定 IQ 特定连接参数，其中的 **myconnection** 为 IQ 连接名：

```
p.put ("RemotePWD", ", , CON=myconnection");
```

列名	描述
ConnHandle	连接的 ID 号。
名称	服务器的名称。
Userid	连接的用户 ID。
LastReqTime	指定连接的上次请求开始的时间。
ReqType	表示上次请求类型的字符串。

列名	描述
IQCmdType	在 SAP Sybase IQ 端执行的当前命令（如果有）。该命令类型反映在引擎的实现级别定义的命令。这些命令由事务命令、用于处理 IQ 存储库中的数据的 DDL 和 DML 命令、内部 IQ 游标命令和特殊控制命令（如 <b>OPEN</b> 和 <b>CLOSE</b> 、 <b>BACKUP DATABASE</b> 、 <b>RESTORE DATABASE</b> ）以及其它命令组成。
LastIQCmdTime	此连接上最后一个 IQ 命令在 SAP Sybase IQ 引擎 IQ 端上的启动或完成时间。
IQCursors	此连接上在 IQ 存储库中打开的游标数。
LowestIQCursorState	IQ 游标状态（如果有）。如果连接上存在多个游标，则显示的状态是所有游标中位置最靠下游标的状态，即距离完成时间最远的那个游标的状态。游标状态反映 SAP Sybase IQ 内部实现的详细信息，并可能在未来发生更改。对于此版本，游标状态包括：NONE、INITIALIZED、PARSED、DESCRIBED、COSTED、PREPARED、EXECUTED、FETCHING、END_OF_DATA、CLOSED 和 COMPLETED。就像名称所暗示的那样，游标状态在操作结束时发生更改。例如，PREPARED 状态指示游标正在执行。
IQthreads	当前分配给连接的 SAP Sybase IQ 线程数。某些线程可能已分配，但仍处于空闲状态。此列可以帮助您确定哪些连接使用了最多的资源。
TxnID	连接上当前事务的事务 ID。该事务 ID 与 .iqmsg 文件中 BeginTxn、CmtTxn 和 PostCmtTxn 消息显示的事务 ID 以及打开数据库时所记录的 Txn ID Seq 相同。
ConnCreateTime	连接的创建时间。
TempTableSpaceKB	此连接在处理 IQ 临时表中存储的数据时所用的 IQ 临时存储空间的字节数 (KB)。
TempWorkSpaceKB	此连接在处理诸如排序、散列和临时位图时所用的 IQ 临时存储空间的字节数 (KB)。由位图或由属于 SAP Sybase IQ 临时表索引一部分的其它对象所使用的空间将反映在 TempTableSpaceKB 中。
IQConnID	作为 .iqmsg 文件中所有消息的一部分显示的十位连接 ID。它是一个单调递增整数，在整个服务器会话内唯一。
satoiq_count	内部计数器，用于显示从 SAP Sybase IQ 引擎的 SQL Anywhere 端到 IQ 端的相交数。这在确定连接活动时有可能会用到。结果集将在行缓冲区中返回，但不会每行都增加一次 satoiq_count 或 iqtosa_count。
iqtosa_count	内部计数器，用于显示从 SAP Sybase IQ 引擎的 IQ 端到 SQL Anywhere 端的相交数。这在确定连接活动时有可能会用到。
CommLink	连接的通信链接。这是 SAP Sybase IQ 所支持的网络协议之一，如果为相同计算机连接，则为 local。

列名	描述
NodeAddr	客户端/服务器连接中客户端的节点。
LastIdle	请求间隔时间数。
MPXServerName	如果是 INC 连接，则 varchar(128) 值包含发起 INC 连接的 Multiplex 服务器的名称。如果不是 INC 连接，则为 NULL。
LSName	连接的逻辑服务器名。如果逻辑服务器上下文未知或不适用，则为 NULL。
INCConnName	某个用户连接的基础 INC 连接的名称。此列的数据类型为 varchar(255)。如果 <b>sp_iqconnection</b> 显示某个已挂起用户连接的 INC 连接名，则该用户连接将有一个同样挂起的关联 INC 连接。
INCConnSuspended	此列中的 "Y" 值表示某个用户连接的基础 INC 连接处于挂起状态。"N" 值则表示该连接未挂起。

*示例***sp\_iqconnection**

```

ConnHandle      Name      Userid      LastReqTime      ReqType
=====
1  'SQL_DBC_100525210'  'DBA'      '2011-03-28 09:29:24.466'  'OPEN'

      IQCmdType      LastIQCmdTime      IQCursors      LowestIQCursorState
=====
'IQUTILITYOPENCURSOR'  2011-03-28 09:29:24.0      0      'NONE'

IQthreads      TxnID      ConnCreateTime      TempTableSpaceKB      TempWorkSpaceKB
=====
0  3352568  2011-03-28 09:29:20.0      0      0

IQconnID      satoiq_count      iqtosa_count      CommLink      NodeAdd      LastIdle      MPXServerName
=====
34      43      2  'local'      ''      244  (NULL)

LSName      INCConnName      INCConnSuspended
=====
Finance_LS  'IQ_MPX_SERVER_P54'      'Y'

```

**sp\_iqdbsize 过程**

显示当前数据库的大小。

*语法*

```
sp_iqdbsize ( [ main ] )
```

*适用于*

Simplex 和 Multiplex。

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 ALTER DATABASE 系统特权。

*注释*

返回数据库的总大小。此外，还返回当压缩数据库（在磁盘上）时在内存中保留数据库所需的页数和 IQ 页数。

如果在 Multiplex 数据库上运行，缺省参数为返回共享 IQ 存储库大小的 main。

如果在启用 RLV 的表中不存在行的情况下运行，则 Physical Blocks、RLVLogBlocks 和 RLVLogKBytes 列将包含非零条目，而其它列包含零条目。这表示没有行级版本控制的表。

列名	描述
Database	数据库文件的路径名称。
Physical Blocks	数据库总大小（以块为单位）。 IQ 数据库由一个或多个 dbspace 组成。每个 dbspace 都有固定大小，该大小最初以 MB 为单位指定。使用此 IQ 页大小和该 IQ 页大小对应的块大小将此 MB 数量转换为块数。Physical Blocks 列反映每个 SAP Sybase IQ dbspace 大小的累计总数，以块为单位表示。
KBytes	数据库总大小（以千字节为单位）。此值是数据库的总块数（前一 sp_iqdbsize 列中的 Physical Blocks）乘以块大小的结果。块大小取决于 IQ 页大小。
Pages	在内存中表示表中存储的所有数据以及这些对象的元数据所需的总 IQ 页数。此值始终大于或等于 Compressed Pages（下一 sp_iqdbsize 列）的值。
Compressed Pages	将表中的数据以及这些对象的元数据存储存储在磁盘上所需的总 IQ 页数。此值始终小于或等于 Pages（前一 sp_iqdbsize 列）的值，因为将 IQ 页从内存写入磁盘时 SAP Sybase IQ 会压缩页。sp_iqdbsize Compressed Pages 列表示压缩的页数。
NBlocks	用于存储表中数据的块总大小。此值始终小于或等于 sp_iqdbsize Physical Blocks 值。
Catalog Blocks	用于存储表中元数据的块总大小。
RLVLogBlocks	用于存储 RLV 存储库日志信息的块数。
RLVLogKBytes	RLV 日志的总大小 (KB)。

*示例*

显示数据库 iqdemo 的大小信息：

```
sp_iqdbsize
```



```

Database

PhysicalBlocks KBytes Pages CompressedPages NBlocks CatalogBlocks
RLVLogBlocks RLVLogKBytes
=====
/ system1/sybase/IQ-16_0/demo/iqdemo.db
          1280    522    688                257    1119                18

```

## sp\_iqdbspace 过程

显示每个 SAP Sybase IQ dbspace 的详细信息。

语法

```
sp_iqdbspace [ dbspace-name ]
```

适用于

Simplex 和 Multiplex。

特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY DBSPACE 系统特权。

注释

利用 **sp\_iqdbspace** 中的信息确定是否必须移动数据，以及对于已移动的数据是否已释放旧版本。

列名	描述
DBSpaceName	在 <b>CREATE DBSPACE</b> 语句中指定的 dbspace 的名称。无论是指定 <b>CREATE DATABASE...CASE IGNORE</b> 还是指定 <b>CASE RESPECT</b> ，Dbspace 名称始终不区分大小写。
DBSpaceType	dbspace 的类型 (MAIN、SHARED_TEMP、TEMPORARY、RLV 或 CACHE)。
Writable	T (可写) 或 F (不可写)。
Online	T (联机) 或 F (脱机)。
Usage	所有文件当前使用的 dbspace 占整个 dbspace 的百分比。
TotalSize	dbspace 中所有文件的总大小，以 B (字节)、K (千字节)、M (兆字节)、G (千兆字节)、T (千吉字节) 或 P (千万亿字节) 为单位。
Reserve	dbspace 中可以添加到所有文件的保留空间总大小。
NumFiles	dbspace 中的文件数。
NumRWFiles	dbspace 中的读/写文件数。

列名	描述
Stripingon	F (关闭)。
StripeSize	如果磁盘条带化已开启, 则始终为 1。
BlkTypes	用户数据和内部系统结构占用的空间。
OkToDrop	"Y" 表示可删除 dbspace; 否则为 "N"。

BlkTypes 块类型标识符的值:

标识符	块类型
A	活动版本
B	备份结构
C	检查点日志
D	数据库标识
F	空闲列表
G	全局空闲列表管理器
H	空闲列表的头块
I	索引建议存储
M	Multiplex CM*
O	旧版本
R	RLV 空闲列表管理器
T	表使用
U	索引使用
N	列使用
X	在检查点处删除

\*Multiplex 提交标识块 (实际 128 块) 存在于所有 IQ 数据库中, 即使 Simplex 数据库不使用它也不例外。

示例

显示有关 dbspace 的信息:

```
sp_iqdbspace;
```

**注意：** 以下示例显示 iqdemo 数据库中的对象，以便更好地阐释输出。iqdemo 包括一个名为 iq\_main 的示例用户 dbspace，您自己的数据库中可能不包括此 dbspace。

DBSpaceName	DBSpaceType	Writable
IQ_MAIN	MAIN	T
IQ__SYSTEM_MAIN	MAIN	T
IQ_SYSTEM_TEMP	TEMPORARY	T
myDas	CACHE	T

(继续) Online	Usage	DBSpaceName
T	55	IQ_MAIN
T	21	IQ__SYSTEM_MAIN
T	1	IQ_SYSTEM_TEMP
T	1	myDas

(继续) Reserve	NumFiles	NumRWFiles
200M	1	1
50M	1	1
50M	1	1
0B	5	5

(继续) DBSpaceName	Stripingon	Stripe Size
IQ_MAIN	T	1K
IQ__SYSTEM_MAIN	F	8K
IQ_SYSTEM_TEMP	F	8K
myDas	T	1K

(继续) Blk Types	OkTo Drop
1H, 5169A, 190	N
1H, 7648F, 32D, 128M	N
1H, 64F, 32A	N
5, 192FH	Y

## sp\_iqfile 过程

显示有关 dbspace 中每个 dbfile 的详细信息。

### 语法

```
sp_iqfile [ dbspace-name ]
```

### 适用于

Simplex 和 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY DBSPACE 系统特权。

### 注释

**sp\_iqfile** 显示 dbspace 中每个 dbfile 中的数据的用法、属性和类型。可以使用这些信息确定是否必须移动数据，以及对于已移动的数据是否已释放旧版本。

列名	描述
DBSpaceName	在 <b>CREATE DBSPACE</b> 语句中指定的 dbspace 的名称。无论是指定 <b>CREATE DATABASE...CASE IGNORE</b> 还是指定 <b>CASE RESPECT</b> ，Dbspace 名称始终不区分大小写。
DBFileName	逻辑文件名。
Path	物理文件或原始分区的位置。
SegmentType	dbspace 的类型 (MAIN、TEMPORARY、RLV 或 CACHE)。
RWMode	dbspace 的模式：始终为读写 (RW)。
Online	T (联机) 或 F (脱机)。
Usage	此文件当前使用的 dbspace 占整个 dbspace 的百分比。在 Multiplex 配置中针对辅助节点运行时，此列将显示 NA。

列名	描述
DBFileSize	文件或原始分区的当前大小。对于原始分区来说，此大小值可以小于实际大小。
Reserve	dbspace 中可添加到此文件的保留空间。
StripeSize	如果磁盘条带化已开启，则始终为 1。
BlkTypes	用户数据和内部系统结构占用的空间。
FirstBlk	分配给文件的第一个 IQ 块号。
LastBlk	分配给文件的最后一个 IQ 块号。
OkToDrop	"Y" 表示可删除文件；否则为 "N"。

标识符	块类型
A	活动版本
B	备份结构
C	检查点日志
D	数据库标识
F	空闲列表
G	全局空闲列表管理器
H	空闲列表的标头块
I	索引建议存储
M	<b>Multiplex CM*</b>
O	旧版本
R	<b>RLV 空闲列表管理器</b>
T	表使用
U	索引使用
N	列使用
X	在检查点处删除

\***Multiplex** 提交标识块（实际 128 块）存在于所有 IQ 数据库中，即使 **Simplex** 数据库不使用也不例外。

**示例**

显示有关 `dbspace` 中文件的信息：

```
sp_iqfile;
```

```
sp_iqfile;
DBSpaceName,DBFileName,Path,SegmentType,RWMode,Online,
Usage,DBFileSize,Reserve,StripeSize,BlkTypes,FirstBlk,
LastBlk,OkToDrop

'IQ_SYSTEM_MAIN','IQ_SYSTEM_MAIN','/sun1-c1/users/smith/mpx/m/
mpx_db.iq','MAIN','RW','T','21','
2.92G','0B','1K','1H,76768F,32D,19A,1850,128M,34B,32C'
,1,384000,'N'

'mpx_main1','mpx_main1','/sun1-c1/users/smith/mpx/m/
mpx_main1.iq','MAIN','RW','T','1'
,100M','0B','1K','1H',1045440,1058239,'N'

'IQ_SHARED_TEMP','sharedfile1_bcp','/sun1-c1/users/smith/mpx/m/
f1','SHARED_TEMP','RO','T','0',
'50M','0B','1K','1H',1,6400,'N'

'IQ_SHARED_TEMP','sharedfile2_bcp','/sun1-c1/users/smith/mpx/m/
f2','SHARED_TEMP','RO','T','0',
'50M','0B','1K','1H',1045440,1051839,'N'

'myDAS','ssd_dev_1','/dev/raw/ssd_dev_1','CACHE','RW','T','2',
'20M','0B','1K','1H','64F','1','5120','N'
'myDAS','ssd_dev_2','/dev/raw/ssd_dev_2','CACHE','RW','T','1',
'20M','0B','1K','1H','32F','522208','527327','N'
'myDAS','ssd_dev_3','/dev/raw/ssd_dev_3','CACHE','RW','T','1',
'20M','0B','1K','1H','32F','1044416','1049535','N'
'myDAS','ssd_dev_4','/dev/raw/ssd_dev_4','CACHE','RW','T','1',
'20M','0B','1K','1H','32F','1566624','1571743','N'
'myDAS','ssd_dev_5','/dev/raw/ssd_dev_5','CACHE','RW','T','1',
'20M','0B','1K','1H','32F','2088832','2093951','N'

'IQ_SYSTEM_TEMP','IQ_SYSTEM_TEMP','/sun1-c1/users/smith/mpx/m/
mpx_db.iqtmp','TEMPORARY','RW',
'T','1','2.92G','0B','1K','1H,64F,33A',1,384000,'N'
```

**sp\_iqlocks 过程**

针对 IQ 主存储库和 IQ 目录存储库显示数据库中锁的相关信息。

**语法**

```
sp_iqlocks ([connection,] [[owner.]table_name,] max_locks,]
[sort_order])
```

**参数**

所有参数均可用于限制结果。

参数	数据类型	描述
connection	integer	连接 ID。使用该选项时，该过程仅返回指定连接的锁的相关信息。缺省值为零，将返回所有连接的相关信息。
owner.table_name	char(128)	表名。使用该选项时，该过程仅返回指定表的锁的相关信息。缺省值为 NULL，将返回数据库中所有表的相关信息。如果未指定所有者，系统将假定过程的调用者拥有该表。
max_locks	integer	可返回其信息的锁的最大数目。缺省值为 0，将返回所有锁的信息。
sort_order	char(1)	信息返回的顺序： <ul style="list-style-type: none"> <li>• C 表示按 connection 排序（缺省值）</li> <li>• T 表示按 table_name 排序</li> </ul>

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

### 注释

显示数据库中当前锁的相关信息。根据您所指定的选项，可以将结果限制为显示单个连接或单个表的锁，或显示指定数目的锁。

sp\_iqlocks 显示以下信息，并按 sort\_order 参数中指定的顺序进行排序：

列	数据类型	描述
conn_name	VARCHAR(128)	当前连接的名称。
conn_id	INTEGER	具有锁的连接 ID。
user_id	CHAR(128)	与该连接 ID 相关联的用户。
table_type	CHAR(6)	表的类型。该类型为 BASE（对于表）、GLBTMP（对于全局临时表）或 MVIEW（对于实例化视图）。 仅 IQ 目录存储库中的 SQL Anywhere 表支持实例化视图。
creator	VARCHAR(128)	表的所有者。
table_name	VARCHAR(128)	控制锁的表。
index_id	INTEGER	索引 ID 或 NULL。
lock_class	CHAR(8)	锁类。Schema、Row、Table 或 Position 之一。
lock_duration	CHAR(11)	锁的持续时间。Transaction、Position 或 Connection 之一。

列	数据类型	描述
lock_type	CHAR(9)	锁类型（这取决于锁类）。
row_identifier	UNSIGNED BIGINT	锁的启动行标识符或 NULL。
row_range	BIGINT	锁定的连续行行数。RLV 存储库中的行锁可以是单行也可以是多行。

如果 **sp\_iqlocks** 无法找到拥有表锁的用户的连接 ID 或用户名，将以 0（零）来表示连接 ID 并以 User unavailable 表示用户名。

lock\_type 列中的值取决于 lock\_class 列中的锁分类。可以返回以下值：

锁类	锁类型	注释
模式	<ul style="list-style-type: none"> <li>• <b>已共享</b> - 共享模式锁。</li> <li>• <b>独占</b> - （仅 IQ 目录存储表）独占模式锁。</li> </ul>	对于模式锁，row_identifier 和索引 ID 值均为 NULL。
行	<ul style="list-style-type: none"> <li>• <b>读取</b> - 读锁定。</li> <li>• <b>意图</b> - 意图锁。</li> <li>• <b>ReadPK</b> - 读锁定。</li> <li>• <b>写入</b> - 写锁定。</li> <li>• <b>WriteNoPK</b> - 写锁定。</li> <li>• <b>代理</b> - 代理锁定。</li> </ul>	<p>行的读锁定可以是短期锁定（在隔离级别 1 的扫描），也可以是在更高隔离级别的长期锁定。lock_duration 列指明读锁定是由于游标稳定性而短期锁定 (Position)，还是一直保持到 COMMIT/ROLLBACK 的长期锁定 (Transaction)。行锁定始终保持在 8 字节行标识符作为 row_identifier 列中的 64 位整数值而被报告的特定行上。</p> <p>代理锁定是行锁定的一种特殊情况。代理锁定被保持在代理条目上，这些代理条目是在参照完整性检查延迟时创建的。对于表中所创建的每个代理条目，没有唯一的代理锁定。而是一个代理锁定对应于由给定连接为给定表所创建的一组代理条目。对于与代理锁定关联的表和连接，row_identifier 值是唯一的。</p> <p>如果需要，现在可单独锁定一行中的键值部分和非键值部分。连接可以获得共享（读）访问行的键值部分的读锁定，这样其它连接仍可获取该行的其它非键值列的写锁定。更新某行的非键值列时不会妨碍对引用该行的外行执行插入和删除操作。</p>



锁类	锁类型	注释
表	<ul style="list-style-type: none"> <li>• <b>已共享</b> - 共享表锁。</li> <li>• <b>意图</b> - 意图更新表锁。</li> <li>• <b>独占</b> - (仅 IQ 目录存储表) 独占表锁。</li> </ul>	无
位置	<ul style="list-style-type: none"> <li>• <b>幻像</b> - (仅 IQ 目录存储表) 幻像锁。</li> <li>• <b>插入</b> - 插入锁。</li> </ul>	位置锁通常也被保持在特定行上，且此行的 64 位行标识符出现在结果集的 <code>row_identifier</code> 列中。然而，位置锁还可以被保持在整个扫描上 (索引扫描或顺序扫描)，在此情况下， <code>row_identifier</code> 列为 NULL。

**注意：**独占、幻像或反幻像锁可以放置在 IQ 目录存储表中，但不能放在 IQ 主存储库的 SAP Sybase IQ 表中。除非已在目录存储库中显式撤消表锁，否则永远无法在 SAP Sybase IQ 数据库中看到这些类型的锁。

### 示例

此示例显示 SAP Sybase IQ 数据库中的 `sp_iqlocks` 过程调用及其输出。该过程进行调用时使用的均为缺省选项，因此其输出按连接排序显示所有锁。

```
call sp_iqlocks()
```

```
conn_name      conn_id
user_id table_type creator  table_name
=====
SQL_DBC_13cd6038 3          DBA      BASE      DBA      rv_locks2
SQL_DBC_13cd6038 3          DBA      BASE      DBA      rv_locks2
SQL_DBC_13cd6038 3          DBA      BASE      DBA      rv_locks2
RVL_CONN_T775    1000000407      BASE      DBA      rv_locks2

index_id  lock_class  lock_duration  lock_type  row_identifier
row_range
=====
Schema    Transaction  Shared
Row       Transaction  Row           1          4
Row       Transaction  Row           281474976710656 1
Table     Transaction  Intent
```

## sp\_iqmergerlvstore 过程

触发行级版本 (RLV) 存储库和 IQ 主存储库的合并。

### 语法

```
sp_iqmergerlvstore 'merge_type', 'table_name', [ 'table_owner' ]
```

*特权*

您必须具有系统过程的 EXECUTE 特权。

*注释*

- 合并类型可以是 BLOCKING | NON-BLOCKING。
- 执行合并后，存储过程将自动提交合并事务。

## **sp\_iqrlvmemory 过程**

按表监控 RLV 存储内存使用情况。

*语法*

```
sp_iqrlvmemory ( [table_name [,table_owner ] ])
```

*特权*

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

*注释*

版本特定数据（如版本位图和按需索引）不包含在 RLV 内存使用情况报告中。它们不会影响 RLV 内存限制，因此不在 sp\_iqrlvmemory 中报告。

如果不指定任何参数，将返回有关所有占用内存的 RLV 表的信息。可提供 *table\_name* 和 *table\_owner* 选项，将输出限定为一个表。如果不指定 *table\_owner*，则缺省视为当前用户。

sp\_iqrlvmemory 针对占用 RLV 存储内存的每个表输出一行，输出列为：

列名	描述
table_id	该行所代表的表的 ID。
fragments	此表的存储碎片数。
total	此表使用的 RLV 存储内存总计 (MB)。
data	此表的列碎片所占用的 RLV 存储内存 (MB)。
dictionary	此表的字典所占用的 RLV 存储内存 (MB)。
bitmap	用于存储表级位图的 RLV 存储内存 (MB)。

*示例*

以下示例返回用户 DBA 拥有的表 rlv\_table1 的当前 RLV 内存使用情况。

```
sp_iqrlvmemory 'rlv_table1', 'DBA'
```

输出：

	table_id	fragments	total	data	dictionary	bitmap
1	778	1	1	0	1	1
2	779	1	48	48	0	1
3	785	2	1596	1584	0	12

## sp\_iqspaceinfo 过程

显示当前数据库中每个对象使用的块数和对象所在的 **dbspace** 的名称。

### 语法

```
sp_iqspaceinfo [ 'main
| [table table-name | index index-name] [...] ' ]
```

### 适用于

Simplex 和 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY DBSPACE 系统特权。

### 注释

针对当前数据库显示对象名、每个对象使用的块数和 **dbspace** 的名称。**sp\_iqspaceinfo** 不需要任何参数。

**sp\_iqspaceinfo** 返回的信息对于管理 **dbspace** 很有帮助。

如果在 Multiplex 数据库上运行，缺省参数为返回共享 IQ 存储库大小的 **main**。

如果不提供任何参数，则必须至少拥有一个用户创建对象（如表）来接收结果。

### 示例

该输出来自 iqdemo 数据库中运行的 **sp\_iqspaceinfo** 存储过程。本例中删除了一些表和索引的输出。

Name	NBlocks	dbspace_name
Contacts	19	IQ_SYSTEM_MAIN
SalesOrderItems.DBA.ASIQ_IDX_T205_C5_FP	56	IQ_SYSTEM_MAIN
Contacts.DBA.ASIQ_IDX_T206_C10_FP	55	IQ_SYSTEM_MAIN
Contacts.DBA.ASIQ_IDX_T206_C1_FP	61	IQ_SYSTEM_MAIN
...		
Contacts.DBA.ASIQ_IDX_T206_C9_FP	55	IQ_SYSTEM_MAIN
Contacts.DBA.ASIQ_IDX_T206_I11_HG	19	IQ_SYSTEM_MAIN
Customers	20	IQ_SYSTEM_MAIN
Customers.DBA.ASIQ_IDX_T207_C1_FP	61	IQ_SYSTEM_MAIN
Customers.DBA.ASIQ_IDX_T207_C2_FP	55	IQ_SYSTEM_MAIN
...		

```
Customers.DBA.ASIQ_IDX_T207_I10_HG          19          IQ_SYSTEM_MAIN
...
```

## **sp\_iqspaceused** 过程

显示 IQ 存储库、IQ 临时存储库、RLV 存储库以及 IQ 全局和局部共享临时存储库中可用空间和已用空间的相关信息。

### 语法

```
sp_iqspaceused(out mainKB          unsigned bigint,
               out mainKBUsed      unsigned bigint,
               out tempKB          unsigned bigint,
               out tempKBUsed      unsigned bigint,
               out shTempTotalKB   unsigned bigint,
               out shTempTotalKBUsed unsigned bigint,
               out shTempLocalKB   unsigned bigint,
               out shTempLocalKBUsed unsigned bigint,
               out rlvLogKB        unsigned bigint,
               out rlvLogKBUsed    unsigned bigint)
```

### 适用于

Simplex 和 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- ALTER DATABASE
- MANAGE ANY DBSPACE
- MONITOR

### 注释

**sp\_iqspaceused** 以 unsigned bigint 输出参数形式返回多个值。该系统存储过程可由用户定义的存储过程调用，以确定正在使用的主存储空间、临时存储空间和 RLV 存储空间的大小。

**sp\_iqspaceused** 返回 **sp\_iqstatus** 提供的信息的部分内容，但允许用户以 SQL 变量返回信息以在计算中使用。

如果在 Multiplex 数据库上运行，该过程将应用于在其上运行该过程的服务器。另外还返回 IQ\_SHARED\_TEMP 的已用空间。

列名	描述
mainKB	IQ 主存储空间的总大小 (KB)。
mainKBUsed	数据库使用的 IQ 主存储空间的大小 (KB)。辅助 Multiplex 节点返回 '(Null)'。

列名	描述
tempKB	IQ 临时存储空间的总大小 (KB)。
tempKBUsed	数据库使用的 IQ 临时存储空间的总大小 (KB)。
shTempTotalKB	IQ 全局共享临时存储空间的总大小 (KB)。
shTempLocalKB	IQ 局部共享临时存储空间的总大小 (KB)。
shTempLocalKBUsed	数据库使用的 IQ 局部共享临时存储空间的大小 (KB)。
rlvLogKB	RLV 存储空间的总大小 (KB)。
rlvLogKBUsed	数据库使用的 RLV 存储空间的大小 (KB)。

### 示例

**sp\_iqspaceused** 需要 7 个输出参数。创建一个用于声明 7 个输出参数的用户定义存储过程 **myspace**，然后调用 **sp\_iqspaceused**：

```

create or replace procedure dbo.myspace()
begin
    declare mt unsigned bigint;
    declare mu unsigned bigint;
    declare tt unsigned bigint;
    declare tu unsigned bigint;
    declare gt unsigned bigint;
    declare gu unsigned bigint;
    declare lt unsigned bigint;
    declare lu unsigned bigint;
    declare tt_t unsigned bigint;
    declare mt_t unsigned bigint;
    declare gt_t unsigned bigint;
    declare lt_t unsigned bigint;
    call sp_iqspaceused(mt,mu,tt,tu,gt,gu,lt,lu);
    if (tt = 0) then
        set tt_t = 0;
    else
        set tt_t = tu*100/tt;
    end if;
    if (mt = 0) then
        set mt_t = 0;
    else
        set mt_t = mu*100/mt;
    end if;
    if (gt = 0) then
        set gt_t = 0;
    else
        set gt_t = gu*100/gt;
    end if;
    if (lt = 0) then
        set lt_t = 0;
    else
        set lt_t = lu*100/lt;
    end if;
end

```

```

        end if;
        select cast(mt/1024 as unsigned bigint) as mainMB,
               cast(mu/1024 as unsigned bigint) as mainusedMB, mt_t as
mainPerCent,
               cast(tt/1024 as unsigned bigint) as tempMB,
               cast(tu/1024 as unsigned bigint) as tempusedMB, tt_t as
tempPerCent,
               cast(gt/1024 as unsigned bigint) as shTempTotalKB,
               cast(gu/1024 as unsigned bigint) as shTempTotalKBUsed, gt_t
as globalshTempPerCent,
               cast(lt/1024 as unsigned bigint) as shTempLocalMB,
               cast(lu/1024 as unsigned bigint) as shTempLocalKBUsed, lt_t
as localshTempPerCent;
end
    
```

要显示 **sp\_iqspaceused** 的输出，请执行 **myspace**：

```
myspace
```

## sp\_iqstatistics 过程

返回每项可用统计或指定统计的序列号、名称、说明、值和单位指示符。

### 语法

```
sp_iqstatistics [ stat_name ]
```

### 参数

- **stat\_name** - (可选) VARCHAR 参数指定统计的名称。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MANAGE ANY STATISTICS 系统特权。

### 注释

如果提供 **stat\_name**，**sp\_iqstatistics** 将为给定统计返回一行，如果名称无效，则不返回行。在不指定任何参数的情况下调用时，**sp\_iqstatistics** 将返回所有统计。

### 结果集

列名	数据类型	描述
stat_num	UNSIGNED INTEGER	统计的序列号
stat_name	VARCHAR(255)	统计的名称
stat_desc	VARCHAR(255)	统计的说明
stat_value	LONG VARCHAR	统计的值
stat_unit	VARCHAR(128)	单位指示符

可能返回以下统计：

stat_num	stat_name	stat_desc	stat_unit
0	CpuTotalTime	自上次服务器启动以来 SAP Sybase IQ 服务器使用的总 CPU 时间（以秒为单位）	Second
1	CpuUserTime	自上次服务器启动以来 SAP Sybase IQ 服务器使用的 CPU 用户时间（以秒为单位）	Second
2	CpuSystemTime	自上次服务器启动以来 SAP Sybase IQ 服务器使用的 CPU 系统时间（以秒为单位）	Second
3	ThreadsFree	SAP Sybase IQ 的可用线程数	N/A
4	ThreadsInUse	SAP Sybase IQ 正在使用的线程数	N/A
5	MemoryAllocated	分配的内存（以 MB 为单位）	MB
6	MemoryMaxAllocated	分配的最大内存（以 MB 为单位）	MB
7	MainCacheCurrentSize	主高速缓存的当前大小（以 MB 为单位）	MB
8	MainCacheFinds	主高速缓存查找请求总数	N/A
9	MainCacheHits	主高速缓存总命中次数	N/A
10	MainCachePagesPinned	已占用的主高速缓存页数	Page
11	MainCachePagesPinnedPercentage	已占用的主高速缓存页百分比	%
12	MainCachePagesDirtyPercentage	主高速缓存脏页百分比	%
13	MainCachePagesInUsePercentage	使用中的主高速缓存页百分比	%
14	TempCacheCurrentSize	临时高速缓存的当前大小（以 MB 为单位）	MB
15	TempCacheFinds	临时高速缓存查寻请求总数	N/A
16	TempCacheHits	临时高速缓存总命中次数	N/A
17	TempCachePagesPinned	已占用的临时高速缓存页数	Page

stat_num	stat_name	stat_desc	stat_unit
18	TempCachePagesPinnedPercentage	已占用的临时高速缓存页百分比	%
19	TempCachePagesDirtyPercentage	临时高速缓存脏页百分比	%
20	TempCachePagesInUsePercentage	使用中的临时高速缓存页百分比	%
21	MainStoreDiskReads	从主存储库中读取到的空间大小 (KB)	KB
22	MainStoreDiskWrites	写入主存储库的空间大小 (KB)	KB
23	TempStoreDiskReads	从主存储库中读取到的空间大小 (KB)	KB
24	TempStoreDiskWrites	写入主存储库的空间大小 (KB)	KB
25	ConnectionsTotalConnections	服务器启动后的连接总数	N/A
26	ConnectionsTotalDisconnections	服务器启动后断开连接的总数	N/A
27	ConnectionsActive	活动连接数	N/A
28	OperationsWaiting	等待 SAP Sybase IQ 资源调控器的操作数	N/A
29	OperationsActive	SAP Sybase IQ 资源调控器允许的活动并发操作数	N/A
30	OperationsActiveLoadTableStatements	活动 LOAD TABLE 语句数	N/A

*示例*

显示单个统计的总 CPU 时间：

```
sp_iqstatistics 'CpuTotalTime'
```

显示 MainCache% 的所有统计：

```
SELECT * from sp_iqstatistics() WHERE stat_name LIKE 'MainCache%'
```

stat_num	stat_name	stat_desc	stat_value	stat_unit
7	MainCacheCurrentSize	高速缓存的当前大小 (以 MB 为单位)	64	mb



stat_num	stat_name	stat_desc	stat_value	stat_unit
8	MainCacheFinds	高速缓存 dbspace 查寻请求总数	95303	
9	MainCacheHits	高速缓存 dbspace 总命中次数	95283	
10	MainCachePagesPinned	已占用的高速缓存 dbspace 页数	0	page
11	MainCachePagesPinned-Percentage	已占用的高速缓存 dbspace 页的百分比	0	%
12	MainCachePagesDirty-Percentage	高速缓存 dbspace 脏页百分比	0.39	%
13	MainCachePagesInUse-Percentage	使用中的高速缓存 dbspace 页百分比	4.44	%

## sp\_iqstatus 过程

显示有关当前数据库的各种 SAP Sybase IQ 状态信息。

### 语法

```
sp_iqstatus
```

### 适用于

Simplex 和 Multiplex。

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- ALTER DATABASE
- MANAGE ANY DBSPACE
- MONITOR
- SERVER OPERATOR

### 注释

显示有关当前数据库的状态信息，包括数据库名称、创建日期、页大小、dbspace 的段数、块使用情况、缓冲区使用情况、I/O、备份信息等。

**sp\_iqstatus** 显示主存储库和临时存储库的空间不足状态。如果存储库出现空间不足状态，**sp\_iqstatus** 会在该存储库的空间不足状态显示值中显示 Y。

可以使用 **sp\_iqstatus** 监控行级版本控制 (RLV) 存储库使用的内存。**RLV memory limit** 行显示了 `-iqrlvmem` 服务器选项或 `sa_server_option rlv_memory_mb` 指定的内存限制。RLV memory used 行可显示 RLV 存储库使用的内存量。

可以使用 **sp\_iqstatus** 监控高速缓存 **dbspace** 中直连存储设备使用的内存：

测量	描述
高速缓存 DbSPACE 文件数	数据库中高速缓存 <b>dbspace</b> <b>dbfile</b> 的数量。
高速缓存 DbSPACE 块	标识高速缓存 <b>dbspace</b> 块和相应的存储设备 <b>dbfile</b> 名称。
已用高速缓存 DbSPACE IQ 块	已用 IQ 块数与 IQ 块总数进行比较。也可以按百分比显示使用情况。如果此百分比偏高，则考虑添加额外存储。

**sp\_iqspaceused** 返回与 **sp\_iqstatus** 提供的信息相同的部分内容，但允许用户以 SQL 变量返回信息以在计算中使用。

要显示可通过删除连接回收的空间，请使用 **sp\_iqstatus** 并将返回的两行的结果相加：

```
(DBA)> select * from sp_iqstatus() where name like '%Versions:%'
Execution time: 6.25 seconds
Name                Value
-----
Other Versions: 2 = 1968Mb
Active Txn Versions: 1 = C:2175Mb/D:2850Mb

(First 2 rows)
```

上面的输出样本显示一个活动写入事务已创建 2175MB 的数据，并销毁 2850MB 的数据。事务中使用的且尚未释放的总数据为 4818MB，或 1968MB + 2850MB = 4818MB。

**sp\_iqstatus** 忽略将在下一个检查点释放的块。但是，这些块将以 X 类型出现在 **sp\_iqdbspace** 输出中。

在 **Multiplex** 中，该过程还列出有关共享 IQ 存储库和 IQ 临时存储库的信息。如果 **sp\_iqstatus** 显示 **Multiplex** 服务器上正在使用的主块的百分比较高，请运行 **sp\_iqversionuse** 查明正在使用哪些版本以及释放这些版本可恢复的空间大小。

### 示例

**注意：** 此示例包括一个名为 **iq\_main** 的用户 **dbspace** 示例，您自己的数据库中可能不存在该示例。

以下输出来自 **sp\_iqstatus** 存储过程：

```
Sybase IQ (TM) Copyright (c) 1992-2013 by SAP AG or an SAP affiliate
company. All rights reserved.
Version: 16.0.0.562/130821/P/Mainline/Sun_x64/OS 5.10/64bit/
2013-08-21 06:15:41
Time Now: 2013-08-21 06:27:14.150
Build Time: 2013-08-21 06:15:41
File Format: 23 on 03/18/1999
Server mode: IQ Server
Catalog Format: 2
Stored Procedure Revision: 1
Page Size: 65536/4096blks/16bpp
```

```

Number of Main DB Files: 2
Main Store Out Of Space: N
Number of Cache Dbspace Files: 5
Number of Shared Temp DB Files: 0
Shared Temp Store Out Of Space: N
Number of Local Temp DB Files: 1
Local Temp Store Out Of Space: N
DB Blocks: 1-25600 IQ_SYSTEM_MAIN
DB Blocks: 522208-547807 MainUser
Cache Dbspace Blocks: 1-5120 ssd_dev_1
Cache Dbspace Blocks: 522208-527327 ssd_dev_2
Cache Dbspace Blocks: 1044416-1049535 ssd_dev_3
Cache Dbspace Blocks: 1566624-1571743 ssd_dev_4
Cache Dbspace Blocks: 2088832-2093951 ssd_dev_5
Local Temp Blocks: 1-25600 IQ_SYSTEM_TEMP
Create Time: 2013-08-21 06:27:05.444
Update Time: 2013-08-21 06:27:14.035
Main IQ Buffers: 1588, 100Mb
Temporary IQ Buffers: 1588, 100Mb
Main IQ Blocks Used: 5250 of 38400, 13%=20Mb, Max Block#: 5313
Cache Dbspace IQ Blocks Used: 197 of 25600, 0%=0Mb, Max Block#: 0
Shared Temporary IQ Blocks Used: 0 of 0, 0%=0Mb, Max Block#: 0
Local Temporary IQ Blocks Used: 65 of 12800, 0%=0Mb, Max Block#: 0
Main Reserved Blocks Available: 12800 of 12800, 100%=50Mb
Shared Temporary Reserved Blocks Available: 0 of 0, 0%=0Mb
Local Temporary Reserved Blocks Available: 12800 of 12800, 100%=50Mb
IQ Dynamic Memory: Current: 292mb, Max: 308mb
Main IQ Buffers: Used: 18, Locked: 0
Temporary IQ Buffers: Used: 4, Locked: 0
Main IQ I/O: I: L459/P9 O: C21/D33/P22 D:1 C:100.0
Temporary IQ I/O: I: L320/P0 O: C54/D59/P8 D:50 C:100.0
Other Versions: 0 = 0Mb
Active Txn Versions: 0 = C:0Mb/D:0Mb
Last Full Backup ID: 0
Last Full Backup Time:
Last Backup ID: 0
Last Backup Type: None
Last Backup Time:
DB Updated: 0
Blocks in next ISF Backup: 0 Blocks: =0Mb
Blocks in next ISI Backup: 0 Blocks: =0Mb
IQ large memory space: 2048Mb
IQ large memory flexible percentage: 50
IQ large memory flexible used: 0Mb
IQ large memory inflexible percentage: 90
IQ large memory inflexible used: 0Mb
IQ large memory anti-starvation percentage: 50
DB File Encryption Status: OFF
RLV memory limit: 2048Mb
RLV memory used: 0Mb

```

下面是了解 Main IQ I/O 和 Temporary IQ I/O 输出代码的关键：

- I: 输入
- L: 读取的逻辑页（“查找”）

- P: 读取的物理页
- O: 输出
- C: 创建的页
- D: 脏页
- P: 物理写入
- D: 损坏的页
- C: 压缩率

## sp\_iqsysmon 过程

监控 SAP Sybase IQ 的多个组件，其中包括管理缓冲区高速缓存、内存、线程、锁、I/O 功能和 CPU 利用率。

### *批处理模式语法*

```
sp_iqsysmon start_monitor  
sp_iqsysmon stop_monitor [, 'section(s)' ]  
or  
sp_iqsysmon 'time-period' [, 'section(s)' ]
```

### *文件模式语法*

```
sp_iqsysmon start_monitor, 'filemode' [, 'monitor-options' ]  
sp_iqsysmon stop_monitor
```

### *批处理模式参数*

- **start\_monitor** - 开始监控。
- **stop\_monitor** - 停止监控并显示报告。
- **time-period** - 监控时间段，采用 HH:MM:SS 格式。
- **section(s)** - 要由 **sp\_iqsysmon** 显示的一个或多个部分的缩写。

请参见“注释（第 0 页）”部分了解有关缩写的完整列表的信息。

如果指定多个部分，请使用空格分隔各个部分缩写，并将列表括在单引号或双引号中。缺省情况下显示所有部分。

对于与 IQ 主存储库相关的部分，可通过分别在部分缩写前面添加前缀 'm' 或 't' 来指定主存储库或临时存储库。如果不添加前缀，则会监控这两个存储库。例如，如果指定 'mbufman'，则仅监控 IQ 主存储库缓冲区管理器。如果指定 'mbufman tbufman' 或 'bufman'，则会同时监控主存储库缓冲区管理器和临时存储库缓冲区管理器。

---

**注意：** **sp\_iqsysmon** 当前不支持 SAP Sybase IQ 组件磁盘 I/O 和锁管理器。

---

### *文件模式参数*

- **start\_monitor** - 开始监控。

- **stop\_monitor** - 停止监控并将剩余输出写入日志文件。
- **filemode** - 指定 **sp\_iqsysmon** 在文件模式下运行。在文件模式下，将在监控周期的每个间隔显示统计信息样本。缺省情况下，输出写入到名为 *dbname.connid-iqmon* 的日志文件中。使用 **file\_suffix** 选项可更改输出文件的后缀。有关 **file\_suffix** 选项的说明，请参见 *monitor\_options* 参数。
- **monitor\_options** - *monitor\_options* 字符串可以包含一个或多个选项：
  - **-interval seconds** - 指定报告间隔（秒）。每一间隔后，将监控器示例统计信息输出到日志文件中。如果未指定 **-interval** 选项，则缺省设置为每 60 秒输出一次。最小报告间隔为 2 秒。如果为该选项指定的间隔无效或小于 2 秒，则会将间隔设置为 2 秒。

初次显示时，显示自服务器启动以来的计数器信息。后续显示则显示与先前显示的不同之处。在运行有关性能问题的查询期间或在通常会出现性能问题的那一天中的某个时间，以 60 秒的缺省间隔运行监控器通常可以获得有用的结果。非常短的间隔可能不会提供有意义的结果。间隔应与作业时间成比例；60 秒通常已足够。

- **-file\_suffix suffix** - 创建一个名为 *dbname.connid-suffix* 的监控器输出文件。如果未指定 **-file\_suffix** 选项，则后缀缺省为 *iqmon*。如果指定 **-file\_suffix** 选项，但是未提供后缀或提供空字符串作为后缀，则不使用后缀。
- **-append** 或 **-truncate** - 分别指示 **sp\_iqsysmon** 向现有输出文件附加内容或截断现有输出文件。截断是缺省设置。如果同时指定这两个选项，则在字符串中较晚指定的那个选项有效。
- **-section section(s)** - 指定要写入监控器日志文件的一个或多个部分的缩写。

请参见“注释（第 0 页）”部分了解有关缩写的完整列表的信息。

缺省情况下写入所有部分。文件模式下在部分列表中指定的缩写与批处理模式下使用的缩写相同。如果指定多个部分，则必须以空格分隔各个部分的缩写。

如果指定 **-section** 选项时没有指定任何部分，则不会对任何部分进行监控。无效部分缩写将被忽略，并向 IQ 消息文件中写入一条警告。

### 特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

### 注释

要报告的报告部分或 IQ 组件	要输入的缩写
缓冲区分配	(主) - mbufalloc (临时) - tbufalloc
缓冲区管理器	(主) - mbufman (临时) - tbufman

要报告的报告部分或 IQ 组件	要输入的缩写
缓冲池	(主) - mbufpool (临时) - tbufpool
目录统计信息	目录
CPU 利用率	cpu
空闲列表管理	(主) - mfreelist (临时) - tfreelist
内存管理	memory
预取管理	(主) - mprefetch (临时) - tprefetch
IQ RLV 内存存储库统计信息	rlv
大容量内存分配器 (LMA) 统计信息	lma
服务器上下文统计信息	server
线程管理	threads
事务管理	txn

**sp\_iqsysmon** 存储过程监控 SAP Sybase IQ 的多个组件，其中包括管理缓冲区高速缓存、内存、线程、锁、I/O 功能和 CPU 利用率。

**sp\_iqsysmon** 过程支持两种监控模式：

- **批处理模式** - **sp\_iqsysmon** 收集在监控器启动到停止期间或 *time-period* 参数指定的期间内监控器的统计信息。监控期间结束时，**sp\_iqsysmon** 显示合并统计信息的列表。

批处理模式下的 **sp\_iqsysmon** 与 SAP Adaptive Server® Enterprise 过程 **sp\_sysmon** 相似。

- **文件模式** - **sp\_iqsysmon** 将监控器启动到停止期间的每个时间间隔的示例统计信息写入日志文件。

在文件模式下初次显示时，将显示自服务器启动以来的计数器信息。后续显示则显示与先前显示的不同之处。

文件模式下的 **sp\_iqsysmon** 与 IQ UTILITIES 命令 **START MONITOR** 和 **STOP MONITOR** 接口相似。

*批处理模式语法示例*

示例 1:

在批处理模式下启动监控器，并显示主存储库和临时存储库的所有部分：

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor
```

示例 2：

在批处理模式下启动监控器，并显示主存储库的缓冲区管理器和缓冲池统计信息。

```
sp_iqsysmon start_monitor
sp_iqsysmon stop_monitor 'mbufman mbufpool'
```

示例 3：

在 10 分钟后输出监控信息：

```
sp_iqsysmon '00:10:00'
```

示例 4：

5 分钟之后，仅输出 **sp\_iqsysmon** 报告的内存管理器部分：

```
sp_iqsysmon '00:05:00', memory
```

示例 5：

启动监控器，执行两个过程和一个查询，停止监控器，然后仅输出报告的缓冲区管理器部分：

```
sp_iqsysmon start_monitor
go
execute proc1
go
execute proc2
go
select sum(total_sales) from titles
go
sp_iqsysmon stop_monitor, bufman
go
```

示例 6：

2 分钟后仅输出报告的主缓冲区管理器和主缓冲池部分：

```
sp_iqsysmon '00:02:00', 'mbufman mbufpool'
```

示例 7：

1 小时后仅输出报告的 **RLV** 部分：

```
sp_iqsysmon '01:00:00', 'rlv'
```

示例 8：

5 秒后仅输出报告的 **LMA** 部分：

```
sp_iqsysmon '00:00:05', 'lma'
```

示例 9:

在批处理模式下运行监控器 10 秒钟，并在该时间段结束时显示合并统计信息:

```
sp_iqsysmon '00:00:10', 'mbufpool memory'
```

*文件模式语法示例*

示例 1:

在监控器启动到停止期间，每 2 秒截断信息一次并将其写入日志文件中:

```
sp_iqsysmon start_monitor, 'filemode', '-interval 2'
.
.
.
sp_iqsysmon stop_monitor
```

示例 2:

仅将主缓冲区管理器和内存管理器部分的输出附加到名为 dbname.connid-testmon 的 ASCII 文件。对于数据库 iqdemo，将结果写入文件 iqdemo.2-testmon 中:

```
sp_iqsysmon start_monitor, 'filemode',
'-file_suffix testmon -append -section mbufman memory'
.
.
.
sp_iqsysmon stop_monitor
```

示例 3:

仅输出报告的 RLV 和 LMA 部分:

```
sp_iqsysmon start_monitor, 'filemode', '-section rlv lma'
sp_iqsysmon stop_monitor
```

示例 4:

在文件模式下启动监控器，并将主缓冲池和内存管理器的统计信息写入日志文件（每 5 秒写入一次）:

```
sp_iqsysmon start_monitor, 'filemode', '-interval 5 -section
mbufpool memory'
sp_iqsysmon stop_monitor
```

### **sp\_iqsysmon 过程示例**

sp\_iqsysmon 输出示例。

示例 1:

在 20 分钟后显示缓冲区分配（主存储和临时存储）的输出。

```
sp_iqsysmon '00:20:00', 'mbufalloc tbufalloc'
```

```
=====
Buffer Allocator (Main)"
```



```

=====
STATS-NAME                VALUE
NActiveCommands           2
BufAllocMaxBufs           2275 ( 81.6% )
BufAllocAvailBufs         2115 ( 93.0% )
BufAllocReserved           160 ( 7.0% )
BufAllocAvailPF           750 ( 33.0% )
BufAllocSlots              100
BufAllocNPinUsers         0
BufAllocNPFUsers          2
BufAllocNPostedUsrs       0
BufAllocNUnpostUsrs       0
BufAllocPinQuota          0
BufAllocNPostEst          0
BufAllocNUnPostEst        0
BufAllocMutexLocks        0
BufAllocMutexWaits        0 ( 0.0% )

STATS-NAME                VALUE
NActiveCommands           2
BufAllocMaxBufs           2275 ( 81.6% )
BufAllocAvailBufs         2115 ( 93.0% )
BufAllocReserved           160 ( 7.0% )
BufAllocAvailPF           750 ( 33.0% )
BufAllocSlots              100
BufAllocNPinUsers         0
BufAllocNPFUsers          2
BufAllocNPostedUsrs       0
BufAllocNUnpostUsrs       0
BufAllocPinQuota          0
BufAllocNPostEst          0
BufAllocNUnPostEst        0
BufAllocMutexLocks        0
BufAllocMutexWaits        0 ( 0.0% )

STATS-NAME                TOTAL  UNKNWN  HASH  CSORT  ROW
ROWCOL      FP  GARRAY  LOB   BTREE   BM     BV    STORE  TEST
NumClients  2      0      0     0     0     0     0     2     0
0           0      0     0     0     0     0     0     0     0
PinUserQuota 0      0     0     0     0     0     0     0     0
0           0      0     0     0     0     0     0     0     0
PrefetchUserQuota 160   0     0     0     0     0     0     160   0
0           0      0     0     0     0     0     0     0     0
PinUserRegisters 2      2     0     0     0     0     0     0     0
0           0      0     0     0     0     0     0     0     0
PfUserRegisters 4697  0     0     0     0     0     0     0     382
2621      377   182   0     2     0     0     0     0     0

ClientCountOfPinner 0      1     3     6     10
33       66   100   333   666   1000  3333  6666  10000
Unknown  0      0     0     0     0     0     0     0     0
0       0     0     0     0     0     0     0     0     0
Hash    0      0     0     0     0     0     0     0     0
0       0     0     0     0     0     0     0     0     0

```

附录：SQL 参考

Sort				0	0	0	0	0	0	0	0
0	0	0	0	2	0	0	0	0	0	0	0
Row											
0	0	0	0	0	0	0	0	0	0	0	0
RowColumn											
0	0	0	0	0	0	0	0	0	0	0	0
FP											
0	0	0	0	0	0	0	0	0	0	0	0
Garray											
0	0	0	0	0	0	0	0	0	0	0	0
LOB											
0	0	0	0	0	0	0	0	0	0	0	0
BTree											
0	0	0	0	0	0	0	0	0	0	0	0
BM											
0	0	0	0	0	0	0	0	0	0	0	0
BV											
0	0	0	0	0	0	0	0	0	0	0	0
Store											
0	0	0	0	0	0	0	0	0	0	0	0
Test											
0	0	0	0	0	0	0	0	0	0	0	0
DBCC											
0	0	0	0	0	0	0	0	0	0	0	0
Unknown											
0	0	0	0	0	0	0	0	0	0	0	0
Unknown											
0	0	0	0	0	0	0	0	0	0	0	0
Run											
0	0	0	0	0	0	0	0	0	0	0	0
QCPRun											
0	0	0	0	0	0	0	0	0	0	0	0
TextDoc											
0	0	0	0	0	0	0	0	0	0	0	0
Unknown											
0	0	0	0	0	0	0	0	0	0	0	0
Unknown											
0	0	0	0	0	0	0	0	0	0	0	0
VDO											
0	0	0	0	0	0	0	0	0	0	0	0
Load				Pass							
0	0	0	0	0	2	0	0	0	0	0	0
STATS-NAME (cont'd)					DBCC	BLKMAP	IQUTIL				
NumClients					0	0	0	0	0		
0	0	0	0	0	0	0	0	0	0		
PinUserQuota					0	0	0	0	0		
0	0	0	0	0	0	0	0	0	0		
PrefetchUserQuota					0	0	0	0	0		
0	0	0	0	0	0	0	0	0	0		
PinUserRegisters					0	0	0	0	0		
0	0	0	0	0	0	0	0	0	0		
PfUserRegisters					0	0	0	0	0		
0	0	0	1133	0	0	0	0	0	0		
ClientCountOfPinner					33333	66666	100000	4294967295			

Unknown	0	0	0	0
Hash	0	0	0	0
Sort	0	0	0	0
Row	0	0	0	0
RowColumn	0	0	0	0
FP	0	0	0	0
Garray	0	0	0	0
LOB	0	0	0	0
BTree	0	0	0	0
BM	0	0	0	0
BV	0	0	0	0
Store	0	0	0	0
Test	0	0	0	0
DBCC	0	0	0	0
Unknown	0	0	0	0
Unknown	0	0	0	0
Run	0	0	0	0
QCPRun	0	0	0	0
TextDoc	0	0	0	0
Unknown	0	0	0	0
Unknown	0	0	0	0
VDO	0	0	0	0
Load	0	0	0	0

```
=====
Buffer Allocator (Temporary)
=====
```

STATS-NAME	VALUE
NActiveCommands	2
BufAllocMaxBufs	2275 ( 81.6% )
BufAllocAvailBufs	2263 ( 99.5% )
BufAllocReserved	12 ( 0.5% )
BufAllocAvailPF	908 ( 39.9% )
BufAllocSlots	100
BufAllocNPinUsers	2
BufAllocNPFUsers	2
BufAllocNPostedUsrs	0
BufAllocNUnpostUsrs	0
BufAllocPinQuota	175
BufAllocNPostEst	2
BufAllocNUnPostEst	2
BufAllocMutexLocks	0
BufAllocMutexWaits	0 ( 0.0% )

STATS-NAME	TOTAL	UNKNWN	HASH	CSORT	ROW	
ROWCOL	LOB	BTREE	BM	BV	STORE	TEST
NumClients	4	0	0	4	0	0
0	0	0	0	0	0	0
PinUserQuota	10	0	0	10	0	0
0	0	0	0	0	0	0
PrefetchUserQuota	2	0	0	2	0	0
0	0	0	0	0	0	0
PinUserRegisters	668	0	300	247	0	0
0	0	0	0	0	0	0

附录：SQL 参考

PfUserRegisters	675	0	0	295	0				
0	0	0	0	0	0	0	0	1	0
ClientCountOfPinners	0	1	3	6	10				
33	66	100	333	666	1000	3333	6666	10000	
Unknown	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Hash	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Sort	2	0	1	0	1	0	0	1	0
0	0	0	0	0	0	0	0	0	0
Row	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
RowColumn	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
FP	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Garray	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
LOB	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
BTree	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
BM	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
BV	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Test	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
DBCC	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Unknown	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Unknown	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Run	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
QCPRun	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
TextDoc	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Unknown	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Unknown	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
VDO	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
Load	Pass	2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
STATS-NAME (cont'd)				DBCC	BLKMAP	IQUTIL			
NumClients	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
PinUserQuota	0	0	0	0	0	0	0	0	0

```

0      0      0      0      0
PrefetchUserQuota      0      0      0      0      0
0      0      0      0      0
PinUserRegisters      0      0      0      110      2
0      0      0      0      9
PfUserRegisters      0      0      0      378      0
0      0      1      0      0

ClientCountOfPinners      33333      66666      100000      4294967295
Unknown      0      0      0      0
Hash      0      0      0      0
Sort      0      0      0      0
Row      0      0      0      0
RowColumn      0      0      0      0
FP      0      0      0      0
Garray      0      0      0      0
LOB      0      0      0      0
BTree      0      0      0      0
BM      0      0      0      0
BV      0      0      0      0
Store      0      0      0      0
Test      0      0      0      0
DBCC      0      0      0      0
Unknown      0      0      0      0
Unknown      0      0      0      0
Run      0      0      0      0
QCPRun      0      0      0      0
TextDoc      0      0      0      0
Unknown      0      0      0      0
Unknown      0      0      0      0
VDO      0      0      0      0
Load      0      0      0      0      0
0

```

## 示例 2:

在 20 分钟后显示缓冲区管理器（主存储和临时存储）的输出。

```

sp_iqsysmon '00:20:00', 'mbufman tbufman'

=====
Buffer Manager (Main)
=====

STATS-NAME          TOTAL      NONE  TXTPOS  TXTDOC  CMPACT
BTREEV  BTREEF      BV      VDO  DBEXT  DBID    SORT  STORE  GARRAY
Finds
3307      0  20829      0  0      0      0      0  9046
Hits
3291      0  20829      0  0      0      0      0  9015
Hit%
99.5      0  100      99.9  0      0      0      0  99.7
FalseMiss
63      40      0  1097      0  0      0      0  0
UnOwnRR      48      0      0      0      0      0      0  31

```

附录：SQL 参考

16	0	1	0	0	0	0	0	0	0	0
Cloned				0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Creates			1557	0	0	0	0	0	60	
179	0	256	0	0	0	0	0	58		
Destroys			546	0	0	0	0	0	0	
12	21	0	6	0	0	0	0	0	0	29
Dirtyes			7554	0	0	0	0	0	1578	
585	0	0	0	0	0	0	0	0		
RealDirtyes			2254	0	0	0	0	0	0	
117	180	0	542	0	0	0	0	0	0	58
PrefetchReqs			80	0	0	0	0	0	0	
0	0	0	74	0	0	0	0	0	0	0
PrefetchNotInMem			1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0
PrefetchInMem			1466	0	0	0	0	0	0	0
0	0	0	1466	0	0	0	0	0	0	0
Reads			48	0	0	0	0	0	31	
16	0	1	0	0	0	0	0	0	0	
PReadBlks			114	0	0	0	0	0	0	0
80	32	0	2	0	0	0	0	0	0	0
PReadKB			0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
ReReads			0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Writes			2002	0	0	0	0	0	104	
163	0	538	0	0	0	0	0	29		
PWriteBlks			6506	0	0	0	0	0	210	
326	0	1115	0	0	0	0	0	58		
PWriteKB			0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
GrabbedDirty			0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
ReadRemoteRpc			0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
ReadRemotePhyIO			0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
STATS-NAME (cont'd)			BARRAY	BLKMAP	HASH	CKPT	BM			
TEST	CMID	RIDCA	LOB	LVCRID	FILE	RIDMAP	RVLOG			
Finds			2681	8329	0	0	0	35670		
0	0	0	0	0	0	0	0	0		
Hits			2681	8329	0	0	0	35670		
0	0	0	0	0	0	0	0	0		
Hit%			100	100	0	0	0	100		
0	0	0	0	0	0	0	0	0		
FalseMiss			84	8329	0	0	0	16856		
0	0	0	0	0	0	0	0	0		
UnOwnRR			0	0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
Cloned			0	0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
Creates			108	358	0	0	0	538		
0	0	0	0	0	0	0	0	0		
Destroys			0	126	0	0	0	59		
0	0	0	0	0	0	0	0	0		

Dirtyies	512	235	0	0	0	4644
0	0	0	0	0	0	0
RealDirtyies	128	593	0	0	0	636
0	0	0	0	0	0	0
PrefetchReqs	6	0	0	0	0	0
0	0	0	0	0	0	0
PrefetchNotInMem	0	0	0	0	0	0
0	0	0	0	0	0	0
PrefetchInMem	0	0	0	0	0	0
0	0	0	0	0	0	0
Reads	0	0	0	0	0	0
0	0	0	0	0	0	0
PReadBlks	0	0	0	0	0	0
0	0	0	0	0	0	0
PReadKB	0	0	0	0	0	0
0	0	0	0	0	0	0
ReReads	0	0	0	0	0	0
0	0	0	0	0	0	0
Writes	128	466	0	0	0	574
0	0	0	0	0	0	0
PWriteBlks	239	3728	0	0	0	830
0	0	0	0	0	0	0
PWriteKB	0	0	0	0	0	0
0	0	0	0	0	0	0
GrabbedDirty	0	0	0	0	0	0
0	0	0	0	0	0	0
ReadRemoteRpc	0	0	0	0	0	0
0	0	0	0	0	0	0
ReadRemotePhyIO	0	0	0	0	0	0
0	0	0	0	0	0	0
STATS-NAME	VALUE					
BusyWaits	98					
LRUNumLocks	401784					
LRUNumSpinsWoTO	0	0%				
LRUNumSpinLoops	4315					
LRUNumTimeOuts	4315	-1.10%				
BmapHTNumLocks	0					
BmapHTNumWaits	0	0%				
CacheTeamTimesWoken	182					
CacheTeamNumAsleep	10					
BmapHTMaxEntries	4096					
BmapHTNEntries	27					
BmapHTNInserts	31954					
BmapHTNCollisn	203					
BmapHTNFindis	51419					
BmapHTNHits	19576					
BmapHTNHits1	19550					
BmapHTNHits2	26					
BmapHTNClears	31933					
BmapHTNLChain	1					
BmapHTNRehash	0					
BlockmapMutexsNLocks	0					
BlockmapMutexsNWaits	0					
BlockmapUID	3659					
BlockmapUIDnallocs	3652					

附录：SQL 参考

BlockmapRegEver				31851						
BlockmapRegisters				31844						
BufHTNBuckets				4608						
BufHTNEntries				1208						
BufHTNw2orMore				158						
BufHTMaxBucketSize				19						
BufHTNFoiledOps				0						
IONumLocks				0						
IONumWaits				0			0%			
=====										
Buffer Manager (Temporary)										
=====										
STATS-NAME				TOTAL	NONE	TXTPOS	TXTDOC	CMPACT		
BTREEV	BTREEF	BV	VDO	DBEXT	DBID	SORT	STORE	GARRAY		
Finds			31656	0	0	0	0	0	0	0
0	0	0	0	0	0	1022	0	0	0	0
Hits			31655	0	0	0	0	0	0	0
0	0	0	0	0	0	1022	0	0	0	0
Hit%			100	0	0	0	0	0	0	0
0	0	0	0	0	0	100	0	0	0	0
FalseMiss			23898	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
UnOwnRR			0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Cloned			0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Creates			5682	0	0	0	0	0	0	0
0	0	0	0	0	0	1048	716	0	0	0
Destroys			5670	0	0	0	0	0	0	0
0	0	0	0	0	0	821	17	0	0	0
Dirtyies			6702	0	0	0	0	0	0	0
0	0	0	0	0	0	379	0	0	0	0
RealDirtyies			5692	0	0	0	0	0	0	0
0	0	0	0	0	0	1048	716	0	0	0
PrefetchReqs			1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PrefetchNotInMem			1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PrefetchInMem			446	0	0	0	0	0	0	0
0	0	0	0	0	0	446	0	0	0	0
Reads			2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PReadBlks			4096	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PReadKB			0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
ReReads			2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
Writes			10	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PWriteBlks			80	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
PWriteKB			0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0



GrabbedDirty	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
ReadRemoteRpc	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
ReadRemotePhyIO	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
STATS-NAME (cont'd)				BARRAY	BLKMAP	HASH	CKPT			BM
TEST	CMID	RIDCA	LOB	LVCRID	FILE	RIDMAP	RVLOG			
Finds				0	8569	124	0	21939		
0	0	0	0	0	2	0	0	0		
Hits				0	8569	124	0	21939		
0	0	0	0	0	1	0	0	0		
Hit%				0	100	100	0	100		
0	0	0	0	0	50	0	0	0		
FalseMiss				0	8569	0	0	15328		
0	0	0	0	0	1	0	0	0		
UnOwnRR				0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
Cloned				0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
Creates				0	1440	777	0	1041		
0	0	0	0	0	0	660	0	0		
Destroys				0	1434	777	0	123		
0	0	0	0	0	0	660	0	0		
Dirtyies				0	0	0	0	6323		
0	0	0	0	0	0	0	0	0		
RealDirtyies				0	1440	777	0	1051		
0	0	0	0	0	0	660	0	0		
PrefetchReqs				0	0	0	0	0		
0	0	0	0	0	1	0	0	0		
PrefetchNotInMem				0	0	0	0	0		
0	0	0	0	0	1	0	0	0		
PrefetchInMem				0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
Reads			0	0	0	0	0	0	0	
0	0	0	0	2	0	0	0	0	0	
PReadBlks				0	0	0	0	0		
0	0	0	0	0	4096	0	0	0		
PReadKB				0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
ReReads				0	0	0	0	0		
0	0	0	0	0	2	0	0	0		
Writes				0	0	0	0	10		
0	0	0	0	0	0	0	0	0		
PWriteBlks				0	0	0	0	80		
0	0	0	0	0	0	0	0	0		
PWriteKB				0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
GrabbedDirty				0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
ReadRemoteRpc				0	0	0	0	0		
0	0	0	0	0	0	0	0	0		
ReadRemotePhyIO				0	0	0	0	0		
0	0	0	0	0	0	0	0	0		

## 附录：SQL 参考

STATS-NAME	VALUE	
BusyWaits	0	
LRUNumLocks	136253	
LRUNumSpinsWoTO	0	0%
LRUNumSpinLoops	2780	
LRUNumTimeOuts	2780	-0.02%
BmapHTNumLocks	0	
BmapHTNumWaits	0	0%
CacheTeamTimesWoken	1	
CacheTeamNumAsleep	10	
BmapHTMaxEntries	4096	
BmapHTNEntries	17	
BmapHTNInserts	2334	
BmapHTNCollisn	0	
BmapHTNFinds	183	
BmapHTNHits	0	
BmapHTNHits1	0	
BmapHTNHits2	0	
BmapHTNClears	2327	
BmapHTNLChain	0	
BmapHTNRehash	0	
BlockmapMutexsNLocks	0	
BlockmapMutexsNWaits	0	
BlockmapUID	2380	
BlockmapUIDnallocs	2335	
BlockmapRegEver	2344	
BlockmapRegisters	2334	
BuFHTNBuckets	4608	
BuFHTNEntries	24	
BuFHTNw2orMore	0	
BuFHTMaxBucketSize	3	
BuFHTNFoiledOps	0	
IONumLocks	0	
IONumWaits	0	0%

### 示例 3:

在 20 分钟后显示缓冲池（主存储和临时存储）的输出。

```
sp_iqsysmon '00:20:00', 'mbufpool tbufpool'
```

```
=====
Buffer Pool (Main)
=====
```

STATS-NAME			TOTAL	NONE	TXTPOS	TXTDOC	COMPACT	
BTREEV	BTREEF	BV	VDO	DBEXT	DBID	SORT	STORE	GARRAY
MovedToMRU			68731	0	0	0	0	9094
2767	0	21083	0	0	0	0	303	
MovedToWash			0	0	0	0	0	
0	0	0	0	0	0	0	0	0
RemovedFromLRU			67564	0	0	0	0	0
9020	2597	0	20830	0	0	0	0	274
RemovedFromWash			11457	0	0	0	0	0
1559	356	0	2189	0	0	0	0	68
RemovedInScanMode			0	0	0	0	0	

```

0      0      0      0      0      0      0      0      0      0
MovedToPSList
0      0      0      0      0      0      0      0      0      0
RemovedFromPSList
0      0      0      0      0      0      0      0      0      0

```

```

STATS-NAME (cont'd)          BARRAY  BLKMAP  HASH    CKPT      BM
TEST      CMID  RIDCA    LOB  LVCRID  FILE  RIDMAP  RVLOG
MovedToMRU          2169  8561    0      0  24754
0      0      0      0      0      0      0      0
MovedToWash
0      0      0      0      0      0      0      0
RemovedFromLRU      2065  8330    0      0  24448
0      0      0      0      0      0      0
RemovedFromWash     233   1437    0      0  5615
0      0      0      0      0      0      0
RemovedInScanMode
0      0      0      0      0      0      0      0
MovedToPSList
0      0      0      0      0      0      0      0
RemovedFromPSList
0      0      0      0      0      0      0      0

```

```

STATS-NAME          VALUE
Pages                2787
InUse                1208 ( 43.3% )
Dirty                11 ( 0.4% )
Pinned               19 ( 0.7% )
Flushes              0
FlushedBufferCount  0
GetPageFrame         1605
GetPageFrameFailure  0
GotEmptyFrame        1605
Washed               0
TimesSweepersWoken  0
PriorityWashed        0
NPrioritySweepersWoken 0
washTeamSize         10
WashMaxSize          455 ( 16.3% )
washNBuffers          455 ( 16.3% )
washNDirtyBuffers    0 ( 0.0% )
washSignalThreshold  46 ( 1.7% )
washNActiveSweepers  0
NPriorityWashBuffers  0
NActivePrioritySweepers 0
washIntensity        0
FlushAndEmpties     0
EmptiedBufferCount  0
EmptiedSkippedCount  0
EmptiedWriteCount    0
EmptiedErrorCount    0
nAffinityTotal       0 ( 0.0% )
nAffinityArea        0 ( 0.0% )

```

```

=====
Buffer Pool (Temporary)

```

附录：SQL 参考

```

=====
STATS-NAME
BTREEV  BTREEF      BV      VDO  DBEXT  NONE  TXTPOS  TXTDOC  CMPACT
MovedToMRU      30514      0      0      0      0      1218    696      0
0      0      0      0      0      0      0      0      0
MovedToWash     258      0      0      0      0      0      256      0
0      0      0      0      0      0      0      0      0
RemovedFromLRU  30506      0      0      0      0      1218    694      0
0      0      0      0      0      0      0      0      0
RemovedFromWash 30503      0      0      0      0      1218    694      0
0      0      0      0      0      0      0      0      0
RemovedInScanMode
0      0      0      0      0      0      0      0      0
MovedToPSList
0      0      0      0      0      0      0      0      0
RemovedFromPSList
0      0      0      0      0      0      0      0      0

STATS-NAME (cont'd)
TEST  CMID  RIDCA  BARRAY  BLKMAP  HASH  CKPT  BM
MovedToMRU      0  8575  124  0  19898
0      0      0      0      3      0      0
MovedToWash
0      0      0      0      2      0      0
RemovedFromLRU
0      0      0      0  8569  124  0  19898
RemovedFromWash
0      0      0      0  8569  124  0  19898
RemovedInScanMode
0      0      0      0      0      0      0      0
MovedToPSList
0      0      0      0      0      0      0      0
RemovedFromPSList
0      0      0      0      0      0      0      0

STATS-NAME
VALUE
Pages      2787
InUse      24 ( 0.9% )
Dirty      17 ( 0.6% )
Pinned     4 ( 0.1% )
Flushes    0
FlushedBufferCount
GetPageFrame      5684
GetPageFrameFailure
GotEmptyFrame     5684
Washed            0
TimesSweepersWoken
PriorityWashed     0
NPrioritySweepersWoken
washTeamSize      10
WashMaxSize      455 ( 16.3% )
washNBuffers      20 ( 0.7% )
washNDirtyBuffers  13 ( 0.5% )
washSignalThreshold
washNActiveSweepers

```

```

NPriorityWashBuffers          0
NActivePrioritySweepers      0
washIntensity                0
FlushAndEmpties              0
EmptiedBufferCount           0
EmptiedSkippedCount          0
EmptiedWriteCount            0
EmptiedErrorCount            0
nAffinityTotal                0 ( 0.0% )
nAffinityArea                 0 ( 0.0% )

```

#### 示例 4:

在 20 分钟后显示预取管理器（主存储和临时存储）的输出。

```

sp_iqsysmon '00:20:00', 'mprefetch tprefetch'

=====
Prefetch Manager (Main)
=====

STATS-NAME                VALUE
PFMgrNThreads             10
PFMgrNSubmitted           81
PFMgrNDropped              0
PFMgrNValid                0
PFMgrNRead                 1
PFMgrNReading              0
PFMgrCondVar               Locks 0 Lock-Waits 0 ( 0.0% ) Signals 0
Broadcasts 2 Waits 2

=====
Prefetch Manager (Temporary)
=====

STATS-NAME                VALUE
PFMgrNThreads             10
PFMgrNSubmitted           1
PFMgrNDropped              0
PFMgrNValid                0
PFMgrNRead                 1
PFMgrNReading              0
PFMgrCondVar               Locks 0 Lock-Waits 0 ( 0.0% ) Signals 0
Broadcasts 2 Waits 2

```

#### 示例 5:

在 20 分钟后显示 IQ 存储库空闲列表（主存储和临时存储）的输出。

```

sp_iqsysmon '00:20:00', 'mfreelist tfreelist'

=====
IQ Store (Main) Free List
=====

STATS-NAME                VALUE
FLBitCount                 74036

```

```

FLIsOutOfSpace          NO
FLMutexLocks            0
FLMutexWaits            0 ( 0.0% )

```

```

=====
IQ Store (Temporary) Free List
=====

```

```

STATS-NAME              VALUE
FLBitCount               4784
FLIsOutOfSpace          NO
FLMutexLocks            0
FLMutexWaits            0 ( 0.0% )

```

### 示例 6:

在 20 分钟后显示内存管理器、线程管理器、CPU 利用率以及事务管理器的输出。

```
sp_iqsysmon '00:20:00', 'memory threads cpu txn'
```

```

=====
Memory Manager
=====

STATS-NAME              VALUE
MemAllocated            67599536 ( 66015 KB )
MemAllocatedMax         160044816 ( 156293 KB )
MemAllocatedEver        1009672456 ( 986008 KB )
MemNAllocated           77309
MemNAllocatedEver       914028
MemNTimesLocked         0
MemNTimesWaits         0 ( 0.0 % )

=====
Thread Manager
=====

STATS-NAME              VALUE
ThrNumOfCpus            4
ThreadLimit             99
ThrNumThreads           98 ( 99.0 %)
ThrReserved             15 ( 15.2 %)
ThrNumFree              55 ( 55.6 %)
NumThrUsed              44 ( 44.4 %)
UsedPerActiveCmd        22
ThrNTeamsInUse          5
ThrMaxTeams             7
NumTeamsAlloc           238
TeamThrAlloc            421
SingleThrAlloc          492
ThrMutexLocks           0
ThrMutexWaits           0 ( 0.0 % )

=====
CPU time statistics
=====

```

```

STATS-NAME                                VALUE
Elapsed Seconds                           59.65      ( 25.0 %)
CPU User Seconds                           37.79      ( 15.8 %)
CPU Sys Seconds                            1.89       ( 0.8 %)
CPU Total Seconds                          39.68      ( 16.6 %)

=====
Transaction Manager
=====

STATS-NAME                                VALUE
TxnMgrNPending                             0
TxnMgrNBlocked                             2
TxnMgrNWaiting                             0
TxnMgrPCcondvar                            Locks      0      Lock-Wait 0 ( 0.0 %)
Signals 0 Broadcasts 2 Waits 2
TxnMgrTxnIDseq                             407
TxnMgrtxncblock                            Locks      0      Lock-Wait 0 ( 0.0 %)
TxnMgrVersionID                             0
TxnMgrOAVI                                 0
TxnMgrVersionLock                          Locks      0      Lock-Wait 0 ( 0.0 %)
Signals 0 Broadcasts 0 Waits 0

```

示例 7:

在 20 分钟后显示服务器上下文和目录统计信息的输出。

```

sp_iqsysmon '00:20:00', 'context catalog'

=====
Context Server statistics
=====

STATS-NAME                                VALUE
StCntxNumConns                             1
StCntxNResource                             16
StCntxNOrigResource                         18
StCntxNWaiting                              0
StCntxNWaited                               0
StCntxNAdmitted                             1116
StCntxLock                                  Locks      0 Lock-Waits 0 ( 0.0 %)
StCntxCondVar                               Locks      0 Lock-Waits 0 ( 0.0 %)

=====
Catalog, DB Log, and Repository statistics
=====

STATS-NAME                                VALUE
CatalogLock                                RdLocks 0 RdWaits 0 ( 0.0 %) RdTryFails
0 WrLocks 30037 WrWaits 0 ( 0.0 %) WrTryFail 0
DbLogMLock                                  Locks      0 Lock-Waits 0 ( 0.0 %)
DbLogSLock                                  Locks      0 Lock-Waits 0 ( 0.0 %)
RepositoryNList                              0
RepositoryLock                              Locks      1 SpinsWoTO 0 ( 0.0 %) Spins
0 TimeOuts 0 ( 0.0 %)

```

示例 8:

在 20 分钟后显示 IQ RLV 内存存储库和大容量内存分配器 (LMA) 统计信息的输出。

```
sp_iqsysmon '00:20:00', 'rlv lma'
```

```
=====
IQ In-Memory Store
=====
```

STATS-NAME	VALUE
RLV Memory Limit	2048 MB
RLV Memory Used	0 MB
RLV Chunks Used	0

```
=====
Large Memory Allocator
=====
```

STATS-NAME	VALUE
Large Memory Space	2048 MB
Large Memory Max Fle	512 MB
Large Memory Num Fle	0
Large Memory Flexibl	0.5
Large Memory Flexibl	0 MB
Large Memory Inflexi	0.9
Large Memory Inflexi	0 MB
Large Memory Anti-St	0.5
Large Memory Num Con	0

## sp\_iqtable 过程

显示数据库中的表的相关信息。

语法 1

```
sp_iqtable ( [ table_name ], [table_owner ], [table_type ] )
```

**table\_type :**

```
TEMP
| VIEW
| ALL
| any_other_value
```

语法 2

```
sp_iqtable [table_name='tablename'],
[table_owner='tableowner' ], [table_type='tabletype' ]
```

参数

- **table\_type** -
  - **TEMP** - 全局临时表
  - **VIEW** - 视图



- **ALL** - IQ 表、全局临时表和视图
- **any\_other\_value** - IQ 表

**特权**

您必须具有系统过程的 EXECUTE 特权。

**注释**

就语法 1 而言，如果不指定前两个参数中的任何一个参数，但指定序列中的下一个参数，则必须用 NULL 替换省略的参数。例如，sp\_iqtable NULL,NULL,TEMP 和 sp\_iqtable NULL,dbo,SYSTEM。

---

**注意：**在语法 1 中，*table\_type* 值 ALL 和 VIEW 必须用单引号引起来。

---

就语法 2 而言，可按任意顺序指定参数。将其用单引号引起来。

如果指定一个参数，则将仅返回与该参数匹配的表。如果指定多个参数，将根据指定的所有参数对结果进行过滤。如果不指定任何参数，则将返回数据库中的所有 SAP Sybase IQ 表。无法返回局部临时表的名称。

列名	描述
table_name	表的名称。
table_type	<ul style="list-style-type: none"> <li>• <b>BASE</b> - 基表。</li> <li>• <b>MAT VIEW</b> - 实例化视图。（仅 SA 表）</li> <li>• <b>GBL TEMP</b> - 全局临时表。</li> <li>• <b>PARTITION</b> - 表分区（该表仅供内部使用，SAP Sybase IQ 用户无法使用）。</li> <li>• <b>VIEW</b> - 视图。</li> </ul>
table_owner	表的所有者
server_type	IQ - 在 IQ 存储中创建的对象。 SA - 在 SA 存储中创建的对象。 所有视图都是在 SA 存储中创建的。
location	TEMP - IQ 临时存储。 MAIN - IQ 存储。 SYSTEM - 目录存储。
dbspace_id	标识 dbspace 的编号。
isPartitioned	如果列属于分区表并有一个或多个分区，而列分区的 dbspace 与表分区的 dbspace 不同，则为 "Y"，如果列的表没有分区或列的每个分区驻留在与表分区相同的 dbspace 中，则为 "N"。

列名	描述
remarks	使用 <b>COMMENT</b> 语句添加的用户注释。
table_constraints	对表的约束。
PartitionType	如果进行了分区，则表示分区类型。 <ul style="list-style-type: none"> <li>散列范围</li> <li>范围</li> <li>散列</li> <li>无</li> </ul>
isRLV	表示表是否启用了 RLV。

**sp\_iqtable 过程示例****sp\_iqtable 输出示例。**

以下两种语法变体都返回表 Departments 的相关信息：

```
sp_iqtable ('Departments')
```

```
sp_iqtable table_name='Departments'
```

Table_name	Table_type	Table_owner
Departments	BASE	GROUPO

Server_type	Location	dbspace_id
IQ	主	16387

isPartitioned	注释	table_constraints
N	包含体育用品公司中各部门的名称及部门领导名称	(NULL)

PartitionType	isRLV
无	F

以下两种语法变体都会返回表所有者 GROUPO 拥有的所有表：

```
sp_iqtable NULL, GROUPO
sp_iqtable table_owner='GROUPO'
```

Table_name	Table_type	Table_owner	Server_type	Location
Contacts	BASE	GROUPO	IQ	主

Table_name	Table_type	Table_owner	Server_type	Location
Customers	BASE	GROUPO	IQ	主
Departments	BASE	GROUPO	IQ	主
Employees	BASE	GROUPO	IQ	主
FinancialCodes	BASE	GROUPO	IQ	主
FinancialData	BASE	GROUPO	IQ	主
Products	BASE	GROUPO	IQ	主
SalesOrders	BASE	GROUPO	IQ	主
SalesOrderItems	BASE	GROUPO	IQ	主

dbspace_id	isPartitioned	注释	table_constraints
16387	N	公司希望保留其联系信息的所有人员的姓名、地址和电话号码	(NULL)
16387	N	体育用品公司的客户	(NULL)
16387	N	包含体育用品公司中各部门的名称及部门领导名称	(NULL)
16387	N	包含姓名、薪水、雇用日期和生日等信息	(NULL)
16387	N	体育用品公司的收入和支出类型	(NULL)
16387	N	体育用品公司的收入和支出	(NULL)
16387	N	体育用品公司销售的产品	(NULL)
16387	N	构成销售订单的各项	(NULL)
16387	N	客户向体育用品公司提交的销售订单	(NULL)

PartitionType	isRlvd
无	F
无	F
无	F
无	F
无	F

PartitionType	isRlvd
无	F
无	F
无	F
无	F

## sp\_iqtablesize 过程

返回指定表的大小。

### 语法

```
sp_iqtablesize ( table_owner.table_name )
```

### 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- MANAGE ANY DBSPACE
- ALTER ANY TABLE
- 您拥有该表

### 注释

以 KBytes 和 NBlocks (IQ 块数) 为单位返回表的总大小。此外，还返回将表保存在内存中所需的页数，以及压缩表时 (在磁盘上) 压缩的 IQ 页数。对于此过程，必须指定 *table\_name* 参数。如果您是 *table\_name* 的所有者，则不必指定 *table\_owner* 参数。

列名	描述
Ownername	所有者的名称
Tablename	表名
Columns	表中的列数
KBytes	表的物理大小 (KB)
Pages	将表保存在内存中所需的 IQ 页数
CompressedPages	(在磁盘上) 压缩表时，压缩的 IQ 页数
NBlocks	IQ 块数
RlvLogPages	将 RLV 表日志信息存储在磁盘上所需的 IQ 页数
RlvLogKBytes	RLV 表日志的大小 (KB)。

Pages 是表的总 IQ 页数。页的测量单位是 IQ 页大小。所有内存缓冲区 (IQ 缓冲区高速缓存中的缓冲区) 的大小都相同。

磁盘上的 IQ 页经过压缩。磁盘上的每个 IQ 页使用 1 到 16 个块。如果 IQ 页大小为 128KB，则 IQ 块大小为 8KB。在这种情况下，磁盘上的单个页可能为 8、16、24、32、40、48、56、64、72、80、88、96、104、112、120 或 128KB。

用 KBytes 值除以页大小，即可得到磁盘上页的平均大小。

**注意：** SAP Sybase IQ 始终读写整个页而非块。例如，如果将某页压缩到 88K，则 IQ 会在一个 I/O 中读取并写入 88K。一般页按因数 2 到 3 进行压缩。

NBlocks 是 Kbytes 除以 IQ 块大小得到的数值。

CompressedPages 是压缩的页数。例如，如果 Pages 是 1000，CompressedPages 是 992，这表示压缩了 1000 页中的 992 页。CompressedPages 除以 Pages 的结果通常接近 100%，因为大部分页都会压缩。空白页不会压缩，因为 SAP Sybase IQ 不写入空白页。无论 IQ 页是否写满，这些页都会正常压缩。

示例

```
call sp_iqtablesize ('dba.t1')
```

Ownername	Tablename	Columns
DBA	t1	3

(继续)	Pages	CompressedPages
KBytes		
192	5	4

(继续)	RivLogPages	RivLogBytes
NBlocs		
24	96	12288

## **sp\_iqtransaction** 过程

显示事务和版本的相关信息。

语法

```
sp_iqtransaction
```

适用于

Simplex 和 Multiplex。

特权

您必须具有系统过程的 EXECUTE 特权，以及 MONITOR 系统特权。

**注释**

**sp\_iqtransaction** 将针对 SAP Sybase IQ 事务管理器中的每个事务控制块返回一行信息。列 Name、Userid 和 ConnHandle 分别是连接属性 **Name**、**Userid** 和 **Number**。行按 TxnID 排序。

**sp\_iqtransaction** 输出只包含具有进行中事务的那些连接。要包括所有连接，请使用 **sp\_iqconnection**。

**注意：** 尽管您可以使用 **sp\_iqtransaction** 来标识阻止其他用户向表中写入内容的用户，但最好选择使用 **sp\_iqlocks** 来实现此目的。

列名	描述
名称	应用程序的名称。
Userid	连接的用户 ID。
TxnID	此事务控制块的事务 ID。该事务 ID 在 <b>begin transaction</b> 时分配。它将出现在 .iqmsg 文件的 BeginTxn、CmtTxn 和 PostCmtTxn 消息中，并与打开数据库时所记录的 Txn ID Seq 相同。
CmtID	提交事务时事务管理器分配的 ID。对于活动事务，CmtID 为零。
VersionID	对于 Simplex 和 Multiplex 节点，值 0 表示事务未实现版本化，且尚未分配 VersionID。  对于 Multiplex 协调器，在事务建立表锁后分配 VersionID。Multiplex 辅助服务器自协调器接收 VersionID。VersionID 由 SAP Sybase IQ 内存中目录和 IQ 事务管理器在内部使用，以对 Multiplex 数据库中的所有节点唯一标识数据库版本。
状态	事务控制块的状态。此变量反映 SAP Sybase IQ 内部实现的详细信息，并可能在未来发生更改。当前，事务状态为 NONE、ACTIVE、ROLLING_BACK、ROLLED_BACK、COMMITTING、COMMITTED 和 APPLIED。  NONE、ROLLING_BACK、ROLLED_BACK、COMMITTING 和 APPLIED 是瞬时状态，寿命很短。  ACTIVE 指示事务处于活动状态。  COMMITTED 指示事务已完成并正在等待 APPLIED，此时，将对对于任何事务都不可见的版本执行垃圾回收。  一旦事务状态变为 ROLLED_BACK、COMMITTED 或 APPLIED，除了打开游标所持有的锁，该事务将不再拥有任何其它锁。
ConnHandle	连接的 ID 号。

列名	描述
IQConnID	十位连接 ID，是 .iqmsg 文件中所有消息的一部分。它是一个单调递增整数，在整个服务器会话内唯一。
MainTableKBCr	此事务创建的 IQ 存储空间的大小（以 KB 为单位）。
MainTableKBDr	此事务删除的 IQ 存储空间的大小（以 KB 为单位），但由于该空间在其它数据库版本或此事务的其它保存点仍可见，因此该空间仍位于磁盘存储中。
TempTableKBCr	此事务创建的用于存储 IQ 临时表数据的 IQ 临时存储空间的大小（以 KB 为单位）。
TempTableKBDr	此事务删除的 IQ 临时表空间的大小（以 KB 为单位），但由于该空间对 IQ 游标仍可见，或者归此事务的其它保存点所有，因此该空间仍位于磁盘的 IQ 临时存储库中。
TempWorkSpaceKB	<p>对于 ACTIVE 事务，该事务此时使用的工作空间的快照，如排序、散列和临时位图。该数值因您运行 <b>sp_iqtransaction</b> 的时间而异。例如，查询引擎可能会在临时高速缓存中创建 60MB，但很快释放其中的大部分空间，即使查询处理仍在继续也是如此。如果在查询完成后运行 <b>sp_iqtransaction</b>，该列显示要小得多的数值。当事务不再处于活动状态时，该列为零。</p> <p>对于 ACTIVE 事务，该列与 <b>sp_iqconnection</b> 的 TempWorkSpaceKB 列相同。</p>
TxnCreateTime	事务的开始时间。一旦建立活动连接或者前一个事务执行提交或回退操作，所有 SAP Sybase IQ 事务将立即隐式开始。
CursorCount	引用该事务控制块的打开的 SAP Sybase IQ 游标数。如果事务处于 ACTIVE 状态，则它指示事务中创建的打开的游标数。如果事务处于 COMMITTED 状态，则它指示引用该事务控制块拥有的数据库版本的 HOLD 游标数。
SpCount	事务控制块中存在的保存点结构数。保存点可以隐式创建和释放。因此，此数值不指示事务中用户创建的保存点的数量。
SpNumber	事务的活动保存点数。这是实现详细信息，因此可能不能反映用户创建的保存点。
MPXServerName	指示活动事务是否来自节点间通信 (INC) 连接。如果来自 INC 连接，则值是发起事务的 Multiplex 服务器的名称。如果不是来自 INC 连接，则为 NULL。如果事务不是活动的，则始终为 NULL。
GlobalTxnID	与当前事务关联的全局事务 ID，如果没有则为 0（零）。

列名	描述
VersioningType	事务的快照版本控制类型；表级（缺省值）或行级。行级快照版本控制 (RLV) 仅适用于启用 RLV 的表。一旦启动某个事务，该值便不能进行更改。
Blocking	指示连接阻塞是已启用 (True) 还是已禁用 (False)。可使用 <b>BLOCKING</b> 数据库选项设置连接阻塞。如果为 true，事务阻塞，即事务先等待解除冲突锁，然后尝试重试锁定请求。
BlockingTimeout	指示事务等待清除锁定冲突的时间（毫秒）。可使用 <b>BLOCKING_TIMEOUT</b> 数据库选项设置超时阈值。值 0（缺省值）表示事务无限期等待。

示例

示例 `sp_iqtransaction` 输出：

```

Name      Userid  TxnID  CmtID  VersionID  State  ConnHandle  IQConnID
=====  =====  =====  =====  =====  =====  =====  =====
red2      DBA     10058  10700  10058     Active  419740283  14

MainTableKBCr      MainTableKBDr      TempTableKBCr  TempTableKBDr
=====  =====  =====  =====
0          0          65824          0

TempWorkspaceKB  TxnCreateTime          CursorCount  SpCount
SpNumber
=====
0          2013-03-26 13:17:27.612          1          3          2

MPXServerName  GlobalTxnID  VersioningType  Blocking
BlockingTimeout
=====  =====  =====  =====
0          (NULL)      0          Row-level      True
    
```

**sp\_iqwho** 过程

显示所有当前用户和连接的相关信息，或显示特定用户或连接的相关信息。

语法

```
sp_iqwho [ { connhandle | user-name } [, arg-type ] ]
```



## 参数

- **connhandle** - 表示连接 ID 的整数。如果已指定该参数，则 **sp\_iqwho** 仅返回指定连接的相关信息。如果指定的连接未打开，则输出中不显示任何行。
- **user-name** - 表示用户登录名的 **char(255)** 参数。如果指定此参数，则 **sp\_iqwho** 将仅返回指定用户的相关信息。如果指定的用户尚未打开任何连接，则输出中不显示任何行。如果在数据库中不存在指定的用户名，则 **sp\_iqwho** 会返回错误消息 “User user-name does not exist”
- **arg-type** - 是可选的，并且只有在指定第一个参数以后才能指定该参数。*arg-type* 的值只能是 “user”。如果将 *arg-type* 的值指定为 “user”，则 **sp\_iqwho** 会将第一个参数解释为用户名，即使第一个参数为数字。如果为 *arg-type* 指定 “user” 以外的任何值，则 **sp\_iqwho** 将返回错误 “无效参数”。

请用双引号将 *arg-type* 值引起来。

## 特权

您必须具有系统过程的 EXECUTE 特权。必须具有以下一种系统特权：

- DROP CONNECTION
- MONITOR
- SERVER OPERATOR

## 注释

**sp\_iqwho** 存储过程显示所有当前用户和连接的相关信息，或显示特定用户或连接的相关信息。

列名	描述
ConnHandle	SA 连接句柄
IQConnID	SAP Sybase IQ 特定连接 ID
Userid	打开连接 “ConnHandle” 的用户的姓名
BlockedOn	在其上阻塞特定连接的连接；如果在任何连接上均未进行阻塞则为 0
BlockUserid	阻塞连接的所有者；如果没有阻塞连接则为 NULL
ReqType	通过连接进行的请求的类型；如果未发出任何命令则为 DO_NOTHING
IQCmdType	连接所发出的 SAP Sybase IQ 命令的类型；如果未发出任何命令则为 NONE
QIdle	自上个 SAP Sybase IQ 命令通过连接发出以来所经历的时间（以秒为单位）；如果没有上个 SAP Sybase IQ 命令，则显示自 "01-01-2000" 起所经历的时间

列名	描述
SAIdle	自通过连接发出上个 SA 请求以来所经过的时间（以秒为单位）；如果之前没有 SA 命令，则显示自 2000 年 1 月 1 日起所经过的时间。
IQCursors	连接中的活动游标数；如果没有游标则为 0
IQThreads	连接的线程数。连接打开后至少会立即启动一个线程，因此 IQThreads 的最小值为 1。
TempTableSpaceKB	临时表空间的大小（以 KB 为单位）；如果没有使用临时表空间则为 0
TempWorkSpaceKB	临时工作空间的大小（以 KB 为单位）；如果没有使用临时工作空间则为 0

表 1. sp\_who 和 sp\_iqwho 列的对应关系

sp_who 列	sp_iqwho 列
fid	块所属的系列；已忽略，因为不适用于 SAP Sybase IQ
spid	ConnHandle、IQConnID
status	QIdle、SAIdle
loginame	Userid
origname	用户别名；已忽略，因为不适用于 SAP Sybase IQ
hostname	运行服务器的主机的名称；当前不支持
blk_spid	BlockedOn
dbname	已忽略，因为对于 SAP Sybase IQ，有一个服务器和一个数据库，并且对于每个连接它们都相同
cmd	ReqType、IQCmdType
block_xloid	BlockUserid

如果未指定任何参数，则 **sp\_iqwho** 将显示所有当前活动连接和用户的相关信息。

第一个 **sp\_iqwho** 参数可以指定为连接句柄或用户名。参数 *connhandle* 和 *user-name* 都是排它参数和可选参数。一次只能指定这些参数中的一个。缺省情况下，如果第一个参数是数字，则假定此参数为连接句柄。如果第一个参数不是数字，则假定此参数为用户名。

SAP Sybase IQ 允许数字形式的用户名。*arg-type* 参数指示 **sp\_iqwho** 将第一个参数中的数值解释为用户名。例如：

```
sp_iqwho 1, "user"
```

指定 *arg-type* “user” 时，**sp\_iqwho** 将第一个参数解释为用户名，而不是连接 ID。如果名为 1 的用户存在于数据库中，则 **sp\_iqwho** 将显示由用户 1 打开的连接相关的信息。

语法	输出
<b>sp_iqwho</b>	显示所有活动连接
<b>sp_iqwho 3</b>	显示连接 3 的相关信息
<b>sp_iqwho “DBA”</b>	显示由用户 DBA 打开的连接
<b>sp_iqwho 3, “user”</b>	将 3 解释为用户名，并显示由用户 3 打开的连接。如果用户 3 不存在，则返回错误 “User 3 does not exist”
<b>sp_iqwho non-existing-user</b>	返回错误 “User non-existing-user does not exist”
<b>sp_iqwho 3, “xyz”</b>	返回错误 “Invalid parameter: xyz”

### **sp\_iqwho** 过程示例

使用示例作为 **sp\_iqwho** 用法的参考。

显示所有活动连接：

ConnHandle	IQConnID	Userid	ReqType	IQCmdType	Blocked
On	BlockUserid	IQCursors			
12	118	DBA	CURSOR_OPEN	IQUTILITYOPENCURSOR	0
(NULL)	0				
13	119	shweta	DO_NOTHING	NONE	0
(NULL)	0				
IQThreads	IQIdle	SAIdle	TempTableSpaceKB	TempWorkSpaceKB	
1	1	0	0	0	
1	16238757	470	0	0	

### **sp\_iqwho** 与 Adaptive Server 的兼容性

SAP Sybase IQ **sp\_iqwho** 存储过程将合并由 Adaptive Server **sp\_who** 过程显示的 SAP Sybase IQ 等效列。

一些 Adaptive Server 列会被忽略，因为它们并不适用于 SAP Sybase IQ。

## 服务器启动选项

数据库启动实用程序 **start\_iq** 用于启动 SAP Sybase IQ 网络数据库服务器。这里列出了与行级版本控制相关的参数的开关。

有关所有可用开关的完整描述，请参见《实用程序指南》。

## **-iqlrvmem start iq 服务器选项**

指定 RLV 存储可用的内存量（以 MB 为单位）。

*语法*

**-iqlrvmem** *size*

*缺省值*

2048 (MB)

*注释*

如果指定 0 或无效值，则使用缺省值 (2048 MB)。如果该值超过系统虚拟内存限制的 2/3，则显示错误消息并关闭服务器。

*用法*

**-iqlrvmem** 用于在服务器启动时将行级版本控制保留的内存量通知给服务器。

## **SQL 语句**

---

列出与内存行级版本控制相关的 SQL 语句。

### **ALTER DBSPACE 语句**

更改读/写模式、更改大小或扩展现有数据库空间。

快速链接：

[转至参数](#)（第 151 页）

[转至示例](#)（第 153 页）

[转至用法](#)（第 153 页）

[转至标准](#)（第 154 页）

[转至权限](#)（第 154 页）

**语法**

```
ALTER DBSPACE dbspace-name
  { ADD new-file-spec [, new-file-spec ... ]
  | DROP FILE logical-file-name [, FILE logical-file-name ... ]
  | RENAME TO newname | RENAME 'new-file-pathname'
  | READONLY | READWRITE
  | ONLINE | OFFLINE
  | STRIPING{ ON | OFF }
  | STRIPESIZEKB size-in-KB
ALTER FILE file-name
  { READONLY | [ FORCE ] READWRITE }
```

```

| SIZE file-size [ KB | MB | GB | TB ]
| ADD file-size [ KB | MB | GB | TB | PAGES ] }
RENAME PATH 'new-file-pathname'
RENAME TO newname

new-file-spec - (back to Syntax)
  FILE logical-file-name 'file-path' iq-file-opts

iq-file-opts - (back to new-file-spec)
  [ [ SIZE ] file-size ]
  ...[ KB | MB | GB | TB ] ]
  [ RESERVE reserve-size [ KB | MB | GB | TB ] ]

```

## 参数

(返回顶部) (第 150 页)

- **ADD** - 向指定 `dbspace` 中添加一个或多个文件。必须为每个文件指定 `dbfile` 名称和物理文件路径，且该名称和路径必须唯一。您可以向 `IQ` 主 `dbspace`、`IQ` 共享临时 `dbspace`、`IQ` 临时 `dbspace` 或高速缓存 `dbspace` 添加文件。您可以向只读 `dbspace` 中添加一个文件，但 `dbspace` 仍为只读。您只能以只读模式（`ADD FILE` 的缺省设置）向 `Multiplex` 共享临时 `dbspace` 添加文件。

一个目录 `dbspace` 只能包含一个文件，因此 `ADD FILE` 不可用于目录 `dbspace`。

对于 `RLV dbspace`，仅在 `Simplex` 服务器上使用 `ADD FILE`。不能将文件添加到 `Multiplex RLV dbspace`。

对于高速缓存 `dbspace`，在 `Multiplex` 或 `Simplex` 服务器上使用 `ADD FILE`。

在 `ALTER FILE` 子句中使用 `ADD` 时，以页、千字节 (`KB`)、兆字节 (`MB`)、千兆字节 (`GB`) 或千吉字节 (`TB`) 为单位扩展文件的大小。缺省值为 `MB`。仅当空闲列表（一种分配映射）有足够空间且 `dbspace` 有足够保留的空间时，您才能执行 `ADD` 子句。

- **DROP FILE** - 从 `IQ dbspace` 中删除指定文件。该文件必须为空。您不能删除指定 `dbspace` 中的最后一个文件。如果 `dbspace` 中仅包含一个文件，改为使用 `DROP DBSPACE`。
- **RENAME TO** - 与 `DROP FILE` 子句一起使用时，重命名包含单个文件的 `dbspace` 的路径名。从语义上看，它等同于 `RENAME PATH` 子句。如果 `dbspace` 包含多个文件，则系统会返回错误。不能重命名 `IQ_SYSTEM_MAIN`、`IQ_SYSTEM_MSG`、`IQ_SYSTEM_TEMP`、`IQ_SHARED_TEMP` 或 `SYSTEM`。

与 `ALTER FILE` 子句一起使用时，将指定文件的逻辑名重命名为新名称。新名称在数据库中必须唯一。

- **READONLY** - 与 `DROP` 子句一起使用时，将除 `IQ_SYSTEM_MAIN`、`IQ_SYSTEM_TEMP`、`IQ_SYSTEM_MSG`、`IQ_SHARED_TEMP` 和 `SYSTEM` 外的任何 `dbspace` 更改为只读。禁止对当前已分配至 `dbspace` 的任何对象执行 `DML` 修改。只能用于高速缓存 `dbspace` 和 `IQ` 主存储库中的 `dbspace`。

与 **ALTER FILE** 子句一起使用时，将指定文件更改为只读。文件必须与 IQ 主 **dbspace** 关联。不能将 **IQ\_SHARED\_TEMP** 中的文件更改为 **READONLY** 状态。

- **READWRITE** - 与 **DROP FILE** 子句一起使用时，将 **dbspace** 更改为读写。**dbspace** 必须联机。只能用于高速缓存 **dbspace** 和 IQ 主存储库中的 **dbspace**。

与 **ALTER FILE** 子句一起使用时，将指定的高速缓存 **dbspace**、IQ 主 **dbspace** 或临时存储 **dbfile** 更改为读写。文件必须与高速缓存 **dbspace**、IQ 主 **dbspace** 或临时 **dbspace** 关联。

- **ONLINE** - 将脱机 **dbspace** 及其全部关联文件设置为联机状态。只能用于高速缓存 **dbspace** 和 IQ 主存储库中的 **dbspace**。
- **OFFLINE** - 将联机只读 **dbspace** 及其全部关联文件设置为脱机状态。（如果 **dbspace** 为读写、已脱机或非高速缓存 **dbspace** 或 IQ 主存储库，则系统会返回错误。）只能用于高速缓存 **dbspace** 或 IQ 主存储库中的 **dbspace**。
- **STRIPING** - 按照指定条件更改 **dbspace** 的磁盘条带化。当磁盘条带化设置为 **ON** 时，会以轮转调度方式从 **dbspace** 内的每个文件分配数据。例如，所写的第一个数据库页面将进入第一个文件，所写的第二个页面将进入给定 **dbspace** 内的下一个文件，以此类推。将跳过只读 **dbspace**。
- **STRIPESIZEKB** - 指定磁盘条带化算法移至指定 **dbspace** 的下一条带之前要写入每个文件的字节数 (**KB**)。
- **FORCE READWRITE** - 与 **ALTER FILE** 子句一起使用时，将指定的共享临时存储 **dbfile** 的状态更改为读写，尽管辅助节点可能会出现已知文件状态问题。该文件可能与 IQ 主 **dbspace**、共享临时 **dbspace** 或临时 **dbspace** 相关联，但因为 **IQ\_SYSTEM\_MAIN** 中的新 **dbfile** 和用户主 **dbspace** 创建为读写，所以此子句只会影响共享临时 **dbspace**。
- **SIZE** - 以千字节 (**KB**)、兆字节 (**MB**)、千兆字节 (**GB**) 或千吉字节 (**TB**) 为单位指定文件的新大小。缺省单位为兆字节。仅当空闲列表（一种分配映射）有足够空间且 **dbspace** 有足够保留的空间时，您才能增大 **dbspace**。同样，仅当要截断的部分未被使用时，您才能减小 **dbspace**。
- **RENAME PATH** - 与 **ALTER FILE** 子句一起使用时，重命名与指定文件关联的文件路径名。此子句只是将文件与新文件路径关联起来，而不再关联旧路径。子句并不实际更改操作系统文件名。必须通过操作系统才能更改文件名。**dbspace** 必须脱机才能重命名文件路径。当 **dbspace** 更改为联机状态或当数据库重新启动时，将使用新路径。

您不能重命名 **IQ\_SYSTEM\_MAIN** 中的文件路径，因为如果新路径不可访问，则数据库将无法启动。如果您需要重命名 **IQ\_SYSTEM\_MAIN** 中文件的路径，请将文件设置为只读，清空并删除该文件，然后以新文件路径名再次添加该文件。请将 **dbfile** 的物理文件路径用单引号引起来。

## 示例

(返回顶部) (第 150 页)

- **示例 1** - 将名为 DspHist 的 dbspace 的模式更改为 READONLY:

```
ALTER DBSPACE DspHist READONLY
```

- **示例 2** - 通过添加大小为 500MB 的文件 FileHist3, 向 dbspace DspHist 添加 500MB:

```
ALTER DBSPACE DspHist
ALTER FILE FileHist3 ADD 500MB
```

- **示例 3** - 在 UNIX 系统上, 向 dbspace DspHist 添加两个 500MB 文件:

```
ALTER DBSPACE DspHist ADD
FILE FileHist3 '/History1/data/file3' SIZE 500MB,
FILE FileHist4 '/History1/data/file4' SIZE 500
```

- **示例 4** - 将 dbspace IQ\_SYSTEM\_TEMP 的大小增大 2GB:

```
ALTER DBSPACE IQ_SYSTEM_TEMP ADD 2 GB
```

- **示例 5** - 从 dbspace DspHist 中删除两个文件。两个文件必须为空:

```
ALTER DBSPACE DspHist
DROP FILE FileHist2, FILE FileHist4
```

- **示例 6** - 将 dbspace IQ\_SYSTEM\_MAIN 的大小增大 1000 页。(ADD 子句的缺省单位为页):

```
ALTER DBSPACE IQ_SYSTEM_MAIN ADD 1000
```

- **示例 7** - 将文件添加到高速缓存 dbspace myDAS:

```
ALTER DBSPACE myDAS ADD FILE iqdas2 'sampledb.iqcache' size 1024
```

- **示例 8** - 将 dbfile iqdas2 从高速缓存 dbspace myDAS 移除:

```
ALTER DBSPACE myDAS DROP FILE iqdas2
```

- **示例 9** - 禁用高速缓存 dbspace myDAS:

```
ALTER DBSPACE myDAS OFFLINE
```

- **示例 10** - 将 myDAS 高速缓存 dbspace dbfile iqdas2 设置为只读:

```
ALTER DBSPACE myDAS ALTER FILE iqdas2 READONLY
```

## 用法

(返回顶部) (第 150 页)

**ALTER DBSPACE** 更改读写模式, 更改联机/脱机状态, 更改文件大小, 重命名 dbspace 名称、文件逻辑名或文件路径, 或设置 dbspace 条带化参数。有关现有 dbspace 的详细信息, 请运行 **sp\_iqdbspace** 过程、**sp\_iqdbspaceinfo** 过程、**sp\_iqfile** 过程、**sp\_iqdbspaceobjectinfo** 和 **sp\_iqobjectinfo**。Dspace 和 dbfile 名称始终不区分大小写。

如果数据库为 **CASE RESPECT**，且操作系统支持区分大小写的文件，则物理文件路径区分大小写。否则，文件路径不区分大小写。

对于 **dbspace** 名称和 **dbfile** 名称，要么不用引号引起来，要么用双引号引起来。

在 **Windows** 中，如果您指定一个路径，则任何后跟 **n** 或 **x** 的反斜杠字符 (\) 都必须双写。这样可防止根据 **SQL** 中的字符串规则将这些反斜杠解释为换行符 (\n) 或十六进制数 (\x)。更安全的做法是始终使用两个反斜杠。

副作用：

- 自动提交
- 自动执行检查点操作
- 将模式更改为 **READONLY** 会立即将 **dbspace** 中的内部数据库结构重新分配到某个读/写 **dbspace**。

## **标准**

(返回顶部) (第 150 页)

- **SQL - ISO/ANSI SQL** 语法的服务商扩充。
- **SAP Sybase 数据库产品** - 不受 **Adaptive Server** 支持。

## **权限**

(返回顶部) (第 150 页)

需要 **MANAGE ANY DBSPACE** 系统特权。

## **ALTER TABLE 语句**

修改表的定义。

快速链接：

转至参数 (第 157 页)

转至示例 (第 163 页)

转至用法 (第 165 页)

转至标准 (第 166 页)

转至权限 (第 166 页)

## **语法**

语法 1 - Alter Owner

```
ALTER TABLE table_name ALTER OWNER TO new_owner
[ { PRESERVE | DROP } PERMISSIONS ]
[ { PRESERVE | DROP } FOREIGN KEYS ]
```



## 语法 2

```

ALTER TABLE [ owner.]table-name
  |{ ENABLE | DISABLE } RLV STORE
  { alter-clause, ... }

alter-clause - (back to Syntax 2)
  ADD create-clause
  | ALTER column-name column-alteration
  | ALTER [ CONSTRAINT constraint-name ] CHECK ( condition )
  | DROP drop-object
  | RENAME rename-object
  | move-clause
  | SPLIT PARTITION range-partition-name
    INTO ( range-partition-decl, range-partition-decl )
  | MERGE PARTITION partition-name-1 INTO partition-name-2
  | UNPARTITION
  | PARTITION BY
    range-partitioning-scheme
    | hash-partitioning-scheme
    | composite-partitioning-scheme

create-clause - (back to alter-clause)
  column-name column-definition [ column-constraint ]
  | table-constraint
  | [ PARTITION BY ] range-partitioning-scheme

column definition - (back to create-clause)
  column-name data-type [ NOT NULL | NULL ]
  [ IN dbspace-name ]
  [ DEFAULT default-value | IDENTITY ]

column-constraint - (back to create-clause)
  [ CONSTRAINT constraint-name ]
  { UNIQUE
  | PRIMARY KEY
  | REFERENCES table-name [ (column-name) ] [ actions ]
  | CHECK ( condition )
  | IQ UNIQUE ( integer )
  }

table-constraint - (back to create-clause)
  [ CONSTRAINT constraint-name ]
  { UNIQUE ( column-name [ , ... ] )
  | PRIMARY KEY ( column-name [ , ... ] )
  | foreign-key-constraint
  | CHECK ( condition )
  }

foreign-key-constraint - (back to table-constraint)
  FOREIGN KEY [ role-name ] [ ( column-name [ , ... ] ) ]
  ... REFERENCES table-name [ ( column-name [ , ... ] ) ]
  ... [ actions ]

actions - (back to foreign-key-constraint)
  [ ON { UPDATE | DELETE } { RESTRICT } ]

```

```

column-alteration - (back to alter-clause)
  { column-data-type | alterable-column-attribute } [ alterable-column-attribute ... ]
    | ADD [ constraint-name ] CHECK ( condition )
    | DROP { DEFAULT | CHECK | CONSTRAINT constraint-name }

alterable-column-attribute - (back to column-alteration)
  [ NOT ] NULL
  | DEFAULT default-value
  | [ CONSTRAINT constraint-name ] CHECK { NULL | ( condition )
  }

default-value - (back to alterable-column-attribute)
  CURRENT { DATABASE | DATE | REMOTE USER | TIME | TIMESTAMP | USER |
PUBLISHER )
  | string
  | global variable
  | [ - ] number
  | ( constant-expression )
  | built-in-function ( constant-expression )
  | AUTOINCREMENT
  | NULL
  | TIMESTAMP
  | LAST USER
  | USER

drop-object - (back to alter-clause)
  { column-name
  | CHECK constraint-name
  | CONSTRAINT
  | UNIQUE ( index-columns-list )
  | PRIMARY KEY
  | FOREIGN KEY fkey-name
  | [ PARTITION ] range-partition-name
  }

rename-object - (back to alter-clause)
  new-table-name
  | column-name TO new-column-name
  | CONSTRAINT constraint-name TO new-constraint-name
  | [ PARTITION ] range-partition-name TO new-range-partition-name

move-clause - (back to alter-clause)
  { ALTER column-name
    MOVE
    { PARTITION ( range-partition-name TO new-dbspace-name)
      | TO new-dbspace-name }
    }
  | MOVE PARTITION range-partition-name TO new-dbspace-name
  | MOVE TO new-dbspace-name
  | MOVE TABLE METADATA TO new-dbspace-name
  }

range-partitioning-scheme - (back to alter-clause)

```

```

RANGE ( partition-key )
    ( range-partition-decl [, range-partition-decl ...] )

partition-key - (back to range-partitioning-scheme)
    column-name

range-partition-decl - (back to alter-clause) or (back to range-
partitioning-scheme)
    range-partition-name VALUES <= ( { constant | MAX } ) [ IN dbspace-
name ]

hash-partitioning-scheme - (back to alter-clause) or (back to composite-
partitioning-scheme)
    HASH ( partition-key, ... ] )

composite-partitioning-scheme - (back to alter-clause)
    hash-partitioning-scheme SUBPARTITION range-partitioning-scheme

```

## 参数

(返回顶部) (第 154 页)

- **{ ENABLE | DISABLE } RLV STORE** - 将该表注册到 RLV 存储库以进行实时内存更新。不支持 IQ 临时表，也不能在 Multiplex 环境中运行。此值替代数据库选项 **BASE\_TABLES\_IN\_RLV** 的值。
- **ADD column-definition [ column-constraint ]** - 向表中添加新列。

表必须为空才可指定 NOT NULL。添加 IDENTITY 或 DEFAULT AUTOINCREMENT 列时，表可能包含数据。如果列有缺省 IDENTITY 值，新列的所有行将用连续值来填充。也可添加 FOREIGN 约束作为单列键的列约束。IDENTITY/DEFAULT AUTOINCREMENT 列的值唯一标识表中的每一行。

IDENTITY/DEFAULT AUTOINCREMENT 列存储在插入和更新期间自动生成的顺序编号。DEFAULT AUTOINCREMENT 列也称为 IDENTITY 列。使用 IDENTITY/DEFAULT AUTOINCREMENT 时，列必须是小数位数为 0 的整数数据类型之一或精确的数值类型。有关列约束和 IDENTITY/DEFAULT AUTOINCREMENT 列的详细信息，请参见 CREATE TABLE 语句。

**IQ UNIQUE** 约束 - 定义列的预期基数并确定是将该列装载为 Flat FP 还是 NBit FP。IQ UNIQUE(*n*) 值显式设置为 0 会将该列装载为 Flat FP。不带 IQ UNIQUE 约束的列将隐式装载为 NBit，一直到 FP\_NBIT\_AUTOSIZE\_LIMIT、FP\_NBIT\_LOOKUP\_MB 和 FP\_NBIT\_ROLLOVER\_MAX\_MB 选项定义的限制。

不必使用 *n* 值小于 FP\_NBIT\_AUTOSIZE\_LIMIT 的 IQ UNIQUE。自动大小调整功能会自动将所有中低基数列的大小调整为 NBit。如果想要将列装载为 Flat FP，或者想要在不同值的数量超出 FP\_NBIT\_AUTOSIZE\_LIMIT 时将列装载为 NBit，请使用 IQ UNIQUE。

---

**注意：**

- 当指定高 IQ UNIQUE 值时，请考虑内存的使用情况。如果计算机资源受到限制，应避免带有 `FP_NBIT_ENFORCE_LIMITS='OFF'` (缺省设置) 的装载。在 SAP Sybase IQ 16.0 之前，IQ UNIQUE  $n$  值  $> 16777216$  时会切换到 Flat FP。在 16.0 中，支持对较大的 IQ UNIQUE 值进行标识化，但鉴于基数和列宽，该值可能需要非常多的内存资源。
- BIT、BLOB 和 CLOB 数据类型不支持 NBit 字典压缩。如果 `FP_NBIT_IQ15_COMPATIBILITY='OFF'`，则在包含这些数据类型的 CREATE TABLE 或 ALTER TABLE 语句中指定的非零 IQ UNIQUE 列将返回一个错误。

---

• **ALTER *column-name* *column-alteration*** – 更改列定义：

- **SET DEFAULT *default-value*** – 更改表中现有列的缺省值。您也可以为此任务使用 MODIFY 子句，但 ALTER 与 ISO/ANSI SQL 兼容，而 MODIFY 与其不兼容。修改缺省值不会更改表中的任何现有值。
- **DROP DEFAULT** – 删除表中现有列的缺省值。您也可以为此任务使用 MODIFY 子句，但 ALTER 与 ISO/ANSI SQL 兼容，而 MODIFY 与其不兼容。删除缺省值不会更改表中的任何现有值。
- **ADD** – 在列中添加已命名约束或 CHECK 条件。新约束或条件仅在定义后应用于对表的操作。但不会对表中现有的值进行校验来确认其是否满足新约束或条件。
- **CONSTRAINT *column-constraint-name*** – 可选列约束名用于以后单独修改或删除约束，而不必修改整个列的约束。
- **[ CONSTRAINT *constraint-name* ] CHECK ( *condition* )** – 此子句用于在列上添加 CHECK 约束。
- **SET COMPUTE (*expression*)** – 更改与计算列关联的表达式。当执行语句时，重新计算列中的值，如果新表达式无效，则语句执行失败。
- **DROP COMPUTE** – 将列从计算列更改为非计算列。此语句不更改表中的任何现有值。

• **ADD *table-constraint*** – 向表中添加约束。

也可单列键或多列键添加外键约束作为表约束。如果指定 PRIMARY KEY，表不得具有通过 CREATE TABLE 语句或其它 ALTER TABLE 语句创建的主键。有关表约束的完整说明，请参见 CREATE TABLE 语句。

---

**注意：** 不能通过 MODIFY 修改表约束或列约束。要更改约束，必须通过 DELETE 删除旧约束并通过 ADD 添加新约束。

---

• **DROP *drop-object*** – 删除表对象：

- **DROP *column-name*** – 从表中删除列。如果列包含在任何多列索引、唯一性约束、外键或主键中，则必须删除索引、约束或键后才能删除列。这不会删除引用该列的 CHECK 约束。只有在 IDENTITY\_INSERT 处于关闭状态并且表不

是局部临时表的情况下，才能删除 IDENTITY/DEFAULT AUTOINCREMENT 列。

- **DROP CHECK** - 删除表的所有检查约束。这包括表检查约束和列检查约束。
- **DROP CONSTRAINT *constraint-name*** - 删除表或指定列的已命名约束。
- **DROP UNIQUE ( *column-name, ...* )** - 删除指定列上的唯一约束。同时也删除引用该唯一约束的任何外键（而不是主键）。如果有关联的外键约束，系统会报告错误。在删除主键约束之前，使用 ALTER TABLE 删除所有引用该主键的外键。
- **DROP PRIMARY KEY** - 删除主键。同时也删除引用此表主键的所有外键。如果有关联的外键约束，系统会报告错误。如果主键是未实施的主键，并且存在关联的未实施的外键约束，则 DELETE 会返回错误。
- **DROP FOREIGN KEY *role-name*** - 删除该表的具有给定角色名称的外键约束。系统将保留为该外键约束隐式创建的非唯一 HG 索引。用户可以使用 DROP INDEX 语句显式删除 HG 索引。
- **DROP [ PARTITION ]** - 删除指定分区。分区 P1 中的行将被删除，分区定义也会被删除。不能删除最后一个分区，因为删除最后一个分区会将分区表转变为未分区的表。（要合并分区表，请改用 UNPARTITION 子句。）例如：

```
CREATE TABLE foo (c1 INT, c2 INT)
PARTITION BY RANGE (c1)
(P1 VALUES <= (100) IN dbbsp1,
 P2 VALUES <= (200) IN dbbsp2,
 P3 VALUES <= (MAX) IN dbbsp3
 ) IN dbbsp4);
LOAD TABLE ...
ALTER TABLE DROP PARTITION P1;
```

- **RENAME *rename-object*** - 重命名表中的对象：
  - **RENAME *new-table-name*** - 将表名更改为 *new-table-name*。必须修改任何使用旧表名的应用程序。此外，系统自动分配了旧表名的任何外键都不会更改名称。
  - **RENAME *column-name* TO *new-column-name*** - 将列名更改为 *new-column-name*。必须修改任何使用旧列名的应用程序。
  - **RENAME [ PARTITION ]** - 重命名现有分区。
  - **RENAME *constraint-name* TO *new-constraint-name*** - 将约束名更改为 *new-constraint-name*。必须修改任何使用旧约束名的应用程序。
- **MOVE clause** - 移动表对象。表对象只能驻留在一个 dbspace 内。任何类型的 ALTER MOVE 都会阻止在整个移动期间内对表执行任何修改。

---

**注意：** 无法将对象移动到高速缓存 dbspace。

---

- **MOVE TO** - 当表映射至新的 dbspace 时，移动同一 dbspace 内驻留的所有表对象，其中包括列、索引、唯一约束、主键、外键和元数据。不能在分区表上请求 ALTER Column MOVE TO 子句。

无法将 BIT 数据类型列显式置于 dbspace 中。BIT 数据类型不支持以下内容：

```
ALTER TABLE t2 alter c1_bit MOVE TO iq_main;
```

- **MOVE TABLE METADATA** – 将表的元数据移动到新的 **dbspace**。对于分区表，**MOVE TABLE METADATA** 还会移动各分区之间共享的元数据。
- **MOVE PARTITION** – 将指定分区移动到新的 **dbspace**。
- **PARTITION BY** – 将大表拆分为多个便于管理的小存储对象。各个分区共用父表的相同逻辑属性，但可放置在单独的 **dbspace** 中并分别进行管理。SAP Sybase IQ 支持多个表分区模式：
  - 散列分区
  - 范围分区
  - 复合分区

分区键是包含表分区键的一个或多个列。分区键可包含 **NULL** 和 **DEFAULT** 值，但不能包含：

- **LOB** (**BLOB** 或 **CLOB**) 列
- **BINARY** 或 **VARBINARY** 列
- 长度超过 255 个字节的 **CHAR** 或 **VARCHAR** 列
- **BIT** 列
- **FLOAT/DOUBLE/REAL** 列
- **PARTITION BY RANGE** – 根据分区列中的值范围对行进行分区。范围分区被限定为有 1 个分区键列和最多 1024 个分区。在范围分区模式中，分区键是包含表分区键的列：

```
range-partition-decl:
  partition-name VALUES <= ( {constant-expr | MAX } [ ,
  { constant-expr | MAX } ]... )
  [ IN dbspace-name ]
```

分区名称是存储表行的新分区的名称。在表分区集合中，分区名称必须唯一。分区名称是必需项。

- **VALUE** – 按升序为每个分区指定上限（含上限）。用户必须为每个范围分区指定分区条件，确保将每一行只分配到一个分区。分区列允许 **NULL** 值，将 **NULL** 作为分区键值的行属于第一个表分区。但 **NULL** 不能是界限值。

第一个分区没有下限（**MIN** 值）。分区键第一列中单元格值为 **NULL** 的行属于第一个分区。对于最后一个分区，您可以指定一个上限（含上限）或 **MAX**。如果最后一个分区上限值不是 **MAX**，则加载或插入其分区键值大于最后一个分区上限值的任何行都将生成错误。

- **Max** – 表示无限大的上限，只能为最后一个分区指定。
- **IN** – 在 **partition-decl** 中指定其中应该包含分区行的 **dbspace**。

以下限制影响范围分区表的分区键和界限值：

- 只有所有现有行都属于第一分区，才能对未分区表进行范围分区。

- 分区界限必须是常量，而不是常量表达式。
- 分区界限必须按照分区的创建顺序以升序排列。也就是说，第二个分区上限必须大于第一个分区上限，依此类推。

此外，分区界限值必须与相应的分区键列数据类型兼容。例如，VARCHAR 与 CHAR 兼容。

- 如果某个界限值的数据类型不同于其对应分区键列的数据类型，SAP Sybase IQ 会将界限值的数据类型转换为分区键列的数据类型，但存在以下例外情况：
- 不允许进行显式转换。以下示例尝试将 INT 显式转换为 VARCHAR，并生成错误：

```
CREATE TABLE Employees(emp_name VARCHAR(20))
PARTITION BY RANGE(emp_name)
(p1 VALUES <=(CAST (1 AS VARCHAR(20))),
p2 VALUES <=(CAST (10 AS VARCHAR(20)))
```

- 不允许执行可导致数据丢失的隐式转换。在以下示例中，分区界限与分区键数据类型不兼容。舍入假设可导致数据丢失，并生成错误：

```
CREATE TABLE emp_id (id INT) PARTITION BY RANGE(id) (p1 VALUES
<= (10.5), p2 VALUES <= (100.5))
```

- 在以下示例中，分区界限与分区键数据类型兼容。界限值会直接转换为浮点值。无需进行舍入处理，且支持转换：

```
CREATE TABLE id_emp (id FLOAT)
PARTITION BY RANGE(id) (p1 VALUES <= (10),
p2 VALUES <= (100))
```

- 不允许将非二进制数据类型转换为二进制数据类型。例如，不允许进行以下转换，并会返回错误：

```
CREATE TABLE newemp (name BINARY)
PARTITION BY RANGE(name)
(p1 VALUES <= ("Maarten"),
p2 VALUES <= ("Zymlerman"))
```

- 在范围分区表中，不能将 NULL 用作界限。
- 如果分区键第一列中的单元格值计算结果为 NULL，则对应行将属于第一个分区。SAP Sybase IQ 仅支持一系列分区键，因此分区键中的任何 NULL 都会将对应行分配到第一个分区。

- **PARTITION BY HASH** – 基于内部散列函数处理的分区-键值向各个分区映射数据。散列分区键被限定为最多有 8 个列，其声明的列宽总和小于等于 5300 个字节。对于散列分区，表创建者仅确定分区键列；分区的数量和位置将在内部确定。

在散列分区声明中，分区键是一个或多个列，其合成值将确定存储有各行数据的分区：

```
hash-partitioning-scheme:
HASH ( partition-key [ , partition-key, ... ] )
```

- **限制** –

- 只能对基表进行散列分区。试图对全局临时表或局部临时表进行分区将会引发错误。

- 只能对空的未分区表进行散列分区。
- 不能添加、删除、合并或拆分散列分区。
- 不能为散列分区键添加列或从中删除列。
- **PARTITION BY HASH RANGE** – 按范围对已进行散列分区的表划分子分区。在散列范围分区模式声明中，**SUBPARTITION BY RANGE** 子句将向现有散列范围分区表添加一个新的范围子分区：

```
hash-range-partitioning-scheme:
PARTITION BY HASH ( partition-key [ , partition-key, ... ] )
  [ SUBPARTITION BY RANGE ( range-partition-decl [ , range-
partition-decl ... ] ) ]
```

散列分区指定如何在逻辑上分布和托管数据；范围子分区指定如何在物理上放置数据。新的范围子分区通过散列进行逻辑分区，其散列分区键与现有的散列范围分区表相同。范围子分区键被限定为仅一列。

- **限制** –

- 只能对基表进行散列分区。试图对全局临时表或局部临时表进行分区将会引发错误。
- 只有散列分区表为空表时才能按范围对其进行子分区。
- 不能添加、删除、合并或拆分散列分区。
- 不能为散列分区键添加列或从中删除列。

**注意：**与散列范围分区相同，范围分区和复合分区模式要求有单独授权的 VLDB 管理组件。

- **MERGE PARTITION** – 将 *partition-name-1* 合并到 *partition-name-2*。如果两个分区相邻且数据驻留在同一个 **dbspace** 内，则可合并这两个分区。只能将分区值较低的分区合并到分区值较高的相邻分区内。请注意，服务器不会检查分区所合并到的 **dbspace** 的 **CREATE** 特权。有关如何创建相邻分区的示例，请参见 **CREATE TABLE** 语句示例。
- **RENAME PARTITION** – 重命名现有 **PARTITION**。
- **UNPARTITION** – 从分区表中删除分区。每一列都置于单个 **dbspace** 内。请注意，服务器不会检查所有分区数据将移动到的目标 **dbspace** 的 **CREATE** 特权。**ALTER TABLE UNPARTITION** 会阻塞所有数据库活动。
- **ALTER OWNER** – 更改表的所有者。**ALTER OWNER** 子句不能与 **ALTER TABLE** 语句的任何其它 [alter-clause] 子句配合使用。
  - **[ PRESERVE | DROP ] PERMISSIONS** – 如果不希望新的所有者拥有与旧所有者相同的特权，则可使用 **DROP** 特权子句（缺省）来删除允许用户访问该表的所有显式授予的特权。隐式授予该表所有者的特权将被授予新的所有者，并从旧的所有者删除这些特权。
  - **[ PRESERVE | DROP ] FOREIGN KEYS** – 如果想防止新的所有者访问被引用表中的数据，可使用 **DROP FOREIGN KEYS** 子句（缺省）来删除该表中的所有



外键以及引用该表的所有外键。除非所有引用表都属于新的所有者，否则无法将 **PRESERVE FOREIGN KEYS** 子句与 **DROP PERMISSIONS** 子句配合使用。

在以下情况下，**ALTER TABLE ALTER OWNER** 语句会失败：

- 存在另一个与原始表同名的表，并且该表属于新用户。
- 同时指定了 **PRESERVE FOREIGN KEYS** 和 **PRESERVE PERMISSIONS** 子句，并且有一个外键所属的用户不是引用该表的新表所有者，而且该外键依赖于隐式授予的特权（如授予表所有者的特权）。为了避免语句失败，可将 **SELECT** 特权显式授予引用表的原始所有者，或者删除外键。
- 指定了 **PRESERVE FOREIGN KEYS** 子句，但未指定 **PRESERVE PERMISSIONS** 子句，并且有一个外键所属的用户不是引用该表的新表所有者。为了避免语句失败，可删除外键。
- 指定了 **PRESERVE FOREIGN KEYS** 子句，并且表中包含一个依赖于隐式授予的特权（如授予表所有者的特权）的外键。为了避免语句失败，可将被引用表的 **SELECT** 特权显式授予新所有者，或者删除外键。
- 表包含一个具有缺省值且引用序列的列，而序列生成器的 **USAGE** 特权依赖于隐式授予的特权（如授予序列所有者的特权）。为了避免语句失败，可将序列生成器的 **USAGE** 特权显式授予表的新所有者。
- 存在依赖于原始表的已启用实例化视图。

## 示例

(返回顶部) (第 154 页)

- **示例 1** - 向 **Employees** 表添加一个新列，说明他们在哪个办公室工作：

```
ALTER TABLE Employees
ADD office CHAR(20)
```

- **示例 2** - 从 **Employees** 表中删除 **office** 列：

```
ALTER TABLE Employees
DROP office
```

- **示例 3** - 向 **Customers** 表添加一列，用于为每个客户指派一个销售联系人：

```
ALTER TABLE Customers
ADD SalesContact INTEGER
REFERENCES Employees (EmployeeID)
```

- **示例 4** - 向 **Customers** 表添加一个新列 **CustomerNum**，并指定缺省值 **88**：

```
ALTER TABLE Customers
ADD CustomerNum INTEGER DEFAULT 88
```

- **示例 5** - 将 **c2**、**c4** 和 **c5** 的 **FP** 索引从 **dbspace Dsp3** 移动到 **Dsp6**。c1 的 **FP** 索引保留在 **Dsp1** 中。c3 的 **FP** 索引保留在 **Dsp2** 中。c5 的主键保留在 **Dsp4** 中。**DATE** 索引 **c4\_date** 保留在 **Dsp5** 中。

```
CREATE TABLE foo (
    c1 INT IN Dsp1,
```

```

c2 VARCHAR(20),
c3 CLOB IN Dsp2,
c4 DATE,
c5 BIGINT,
PRIMARY KEY (c5) IN Dsp4) IN Dsp3);

CREATE DATE INDEX c4_date ON foo(c4) IN Dsp5;
ALTER TABLE foo
MOVE TO Dsp6;

```

- **示例 6** – 只将 **FP** 索引 c1 从 dbspace Dsp1 移动到 Dsp7:

```
ALTER TABLE foo ALTER c1 MOVE TO Dsp7
```

- **示例 7** – 使用多个 **ALTER TABLE** 子句对分区进行移动、拆分、重命名和合并。

创建分区表:

```

CREATE TABLE bar (
  c1 INT,
  c2 DATE,
  c3 VARCHAR(10))
PARTITION BY RANGE(c2)
(p1 VALUES <= ('2005-12-31') IN dbsp1,
 p2 VALUES <= ('2006-12-31') IN dbsp2,
 p3 VALUES <= ('2007-12-31') IN dbsp3,
 p4 VALUES <= ('2008-12-31') IN dbsp4);
INSERT INTO bar VALUES(3, '2007-01-01', 'banana nut');
INSERT INTO BAR VALUES(4, '2007-09-09', 'grape jam');
INSERT INTO BAR VALUES(5, '2008-05-05', 'apple cake');

```

将分区 p2 移动到 dbsp5:

```
ALTER TABLE bar MOVE PARTITION p2 TO DBSP5;
```

将分区 p4 拆分为 2 个分区:

```
ALTER TABLE bar SPLIT PARTITION p4 INTO
(p41 VALUES <= ('2008-06-30') IN dbsp4,
 p42 VALUES <= ('2008-12-31') IN dbsp4);
```

下面的 **SPLIT PARTITION** 报告一个错误，因为它需要移动数据。拆分后，并非所有现有行都位于同一分区内。

```
ALTER TABLE bar SPLIT PARTITION p3 INTO
(p31 VALUES <= ('2007-06-30') IN dbsp3,
 p32 VALUES <= ('2007-12-31') IN dbsp3);
```

会报告以下错误:

```
No data move is allowed, cannot split partition p3.
```

下面的 **SPLIT PARTITION** 报告一个错误，因为它更改了分区的边界值:

```
ALTER TABLE bar SPLIT PARTITION p2 INTO
(p21 VALUES <= ('2006-06-30') IN dbsp2,
 p22 VALUES <= ('2006-12-01') IN dbsp2);
```

会报告以下错误:

```
Boundary value for the partition p2 cannot be changed.
```

将分区 p3 合并到 p2。由于不允许将边界值较高的分区合并到边界值较低的分  
区，因此，系统会报告错误。

```
ALTER TABLE bar MERGE PARTITION p3 INTO p2;
```

会报告以下错误：

```
Partition 'p2' is not adjacent to or before partition 'p3'.
```

将分区 p2 合并到 p3：

```
ALTER TABLE bar MERGE PARTITION p2 INTO p3;
```

将分区 p1 重命名为 p1\_new：

```
ALTER TABLE bar RENAME PARTITION p1 TO p1_new;
```

对表 bar 取消分区：

```
ALTER TABLE bar UNPARTITION;
```

对表 bar 进行分区。此命令会报告错误，因为所有行必须位于第一个分区中。

```
ALTER TABLE bar PARTITION BY RANGE (c2)
  (p1 VALUES <= ('2005-12-31') IN dbbsp1,
   p2 VALUES <= ('2006-12-31') IN dbbsp2,
   p3 VALUES <= ('2007-12-31') IN dbbsp3,
   p4 VALUES <= ('2008-12-31') IN dbbsp4);
```

会报告以下错误：

```
All rows must be in the first partition.
```

对表 bar 进行分区：

```
ALTER TABLE bar PARTITION BY RANGE (c2)
  (p1 VALUES <= ('2008-12-31') IN dbbsp1,
   p2 VALUES <= ('2009-12-31') IN dbbsp2,
   p3 VALUES <= ('2010-12-31') IN dbbsp3,
   p4 VALUES <= ('2011-12-31') IN dbbsp4);
```

- **示例 8** - 更改表 tab1，使其不再注册到 RLV 存储库中以进行实时内存更新。

```
ALTER TABLE tab1 DISABLE RLV STORE
```

## 用法

(返回顶部) (第 154 页)

**ALTER TABLE** 语句可更改以前创建的表中的表属性（列定义和约束）。该语法允许使用一组变更子句；但在每个 **ALTER TABLE** 语句中，只能添加、修改或删除一个表约束或列约束。只要 **ALTER TABLE** 语句影响了当前正由其它连接使用的表，就会禁止该语句。**ALTER TABLE** 可能很耗时，并且在处理语句时，服务器不会处理引用相同表的请求。

---

**注意：** 您不能更改本地临时表，但可以在仅有一个连接使用全局临时表时，更改全局临时表。

---

SAP Sybase IQ 强制执行 REFERENCES 和 CHECK 约束。只有在满足以下条件时，才会评估添加到 ALTER TABLE 语句的表和/或列检查约束：变更表操作过程中在所添加的其中一个新列中定义这些约束。有关 CHECK 约束的详细信息，请参见“CREATE TABLE 语句”。

如果在视图定义中使用 SELECT \*，并且变更 SELECT \* 所引用的表，则您必须运行 ALTER VIEW <viewname> RECOMPILE 以确保视图定义正确，并防止在查询该视图时出现意外的结果。

**副作用：**

- 自动提交。ALTER 和 DROP 选项用于关闭当前连接的所有游标。Interactive SQL 数据窗口同时也会被清除。
- ALTER TABLE 操作开始时执行检查点。
- 一旦更改了列或表，引用更改列的任何存储过程、视图或其它项将不再有效。

**标准**

(返回顶部) (第 154 页)

- SQL - ISO/ANSI SQL 语法的服务商扩充。
- SAP Sybase 数据库产品 - 一些子句受 SAP Adaptive Server® Enterprise 的支持。

**权限**

(返回顶部) (第 154 页)

**句法 1**

需要以下特权之一：

- ALTER ANY TABLE 系统特权
- ALTER ANY OBJECT 系统特权
- 表的 ALTER 特权
- 您拥有该表

**句法 2**

语法 1 所需的系统特权因使用的子句而异。

子句	所需特权
Add	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 基础表的 ALTER 特权</li> <li>• 您拥有基础表</li> </ul> <p>UNIQUE、PRIMARY KEY、FOREIGN KEY 或 IQ UNIQUE 列约束 - 需要以上特权，外加基础表的 REFERENCES 特权。</p> <p>FOREIGN KEY 表约束需要以上特权，外加以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY INDEX 系统特权</li> <li>• CREATE ANY OBJECT 系统特权</li> <li>• 基表的 REFERENCES 特权</li> </ul> <p>PARTITION BY RANGE 需要以上特权，外加下列特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY OBJECT 系统特权</li> <li>• 对要创建分区的 dbspace 的 CREATE 特权</li> </ul>
Alter	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> <li>• 您拥有该表。</li> </ul> <p>变更主键或唯一约束还需要表的 REFERENCES 特权。</p>
Drop	<p>删除不含约束的列 - 需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY OBJECT 系统特权</li> <li>• ALTER ANY TABLE 系统特权</li> <li>• 基础表的 ALTER 特权</li> <li>• 您拥有基础表</li> </ul> <p>删除含有约束的列或表需要以上特权以及 REFERENCES 特权（如果使用 ALTER 特权）。</p> <p>删除自身拥有的表分区 - 无需任何权限。</p> <p>删除其他用户拥有的表分区 - 需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> </ul>

子句	所需特权
RENAME	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> <li>• 您拥有该表</li> </ul>
Move	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 系统特权</li> <li>• 基础表的 ALTER 特权</li> <li>• 您拥有基础表</li> </ul> <p>另外，还需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY OBJECT 系统特权</li> <li>• 分区要移动到的目标 dbspace 的 CREATE 特权</li> </ul>
Split Partition	<p>将自身拥有的表分区 - 无需任何权限。</p> <p>将其他用户拥有的表分区 - 需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• SELECT ANY TABLE 系统特权</li> <li>• 表的 SELECT 特权</li> </ul> <p>另外，还需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> </ul>
Merge Partition、Unpartition	<p>自身拥有的表 - 无需任何权限。</p> <p>其他用户拥有的表 - 需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> </ul>

子句	所需特权
Partition By	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY OBJECT 系统特权</li> <li>• 对要创建分区的 dbspace 的 CREATE 特权</li> </ul> <p>另外，还需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> <li>• 表的 ALTER 特权</li> <li>• 您拥有该表</li> </ul>
或 disable RLV store	<p>需要以下特权之一：</p> <ul style="list-style-type: none"> <li>• ALTER ANY TABLE 系统特权</li> <li>• ALTER ANY OBJECT 系统特权</li> </ul>

#### 另请参见

- CREATE TABLE 语句 (第 173 页)

## CREATE DBSPACE 语句

为 IQ 主存储库、高速缓存 dbspace、目录存储库或 RLV 存储库创建一个新的 dbspace 及相关 dbfile。

快速链接：

[转至参数](#) (第 170 页)

[转至示例](#) (第 171 页)

[转至用法](#) (第 172 页)

[转至标准](#) (第 172 页)

[转至权限](#) (第 173 页)

### 语法

语法 1 - 仅用于目录存储 dbspace (SQL Anywhere (SA) dbspace)。

```
CREATE DBSPACE dbspace-name AS file-path CATALOG STORE
```

语法 2 - 用于 IQ 主存储库 dbspace。

```
CREATE DBSPACE dbspace-name USING file-specification  
[ IQ STORE ] iq-dbspace-opts
```

语法 3 - 用于 RLV dbspace。

```
CREATE DBSPACE dbspace-name USING file-specification
IQ RLV STORE
```

语法 4 - 用于高速缓存 `dbspace`。

```
CREATE DBSPACE dbspace-name USING FILE file-specification
IQ CACHE STORE
```

**file-specification** - (back to Syntax 2) or (back to Syntax 3)  
 { **single-path-spec** | **new-file-spec** [, ...] }

**single-path-spec** - (back to file-specification)  
 '*file-path*' | **iq-file-opts**

**new-file-spec** - (back to file-specification)  
**FILE** *logical-file-name* | '*file-path*' **iq-file-opts**

**iq-file-opts** - (back to new-file-spec)  
 [ [ **SIZE** ] *file-size* ]  
 ...[ **KB** | **MB** | **GB** | **TB** ] ]  
 [ **RESERVE** *size*  
 ...[ **KB** | **MB** | **GB** | **TB** ] ]

**iq-dbspace-opts** - (back to Syntax 2)  
 [ **STRIPING** ] { **ON** | **OFF** } ] ...[ **STRIPESIZEKB** *sizeKB* ]

## 参数

(返回顶部) (第 169 页)

- **new-file-spec** - 为 IQ 主存储库创建一个 `dbspace`。可以为 IQ 主存储库指定一个或多个 `dbfile`。必须为每个文件指定 `dbfile` 名称和物理文件路径，且该名称和路径必须唯一。
- **RESERVE** - 以千字节 (KB)、兆字节 (MB)、千兆字节 (GB) 或千吉字节 (TB) 为单位指定要保留的空间大小，从而可在将来增加 `dbspace` 的大小。`size` 参数可为大于 0 的任何数字；缺省单位为兆字节。创建 `dbspace` `dbfile` 后不能更改保留大小。

指定 **RESERVE** 后，数据库将更多空间用于内部（空闲列表）结构。如果保留大小过大，则内部结构所需的空间可能会大于指定大小，这会导致错误。

- **dbspace-name** 和 **dbfile-name** - `dbspace` 和 `dbfile` 的内部名称。一个数据库最多可具有 (32KB - 1) 个 `dbspace`，其中包括在创建数据库时创建的初始 `dbspace`。但是，您的操作系统可能会限制每个数据库的 `dbfile` 数。

**注意：** 您必须获得 `IQ_VLDBMGMT` 选项的使用许可才能创建多个用户 `dbspace`。

- **file-path** - `dbfile` 的实际操作系统文件名，有时需要在前面添加路径。在数据库的目录存储库所在的目录中创建不包含显式目录的 `file-path`。任何相对目录都是相对于该目录存储库而言。



- **SIZE** – 指定您在 *file-path* 中指定的操作系统文件的大小（从 0 至 4 千兆字节）。缺省值取决于存储类型和块大小。对于 IQ 主存储库，缺省字节数等于块大小乘以 1000。无法为目录存储库指定 SIZE 子句。值为 0 的 SIZE 将创建具有最小大小的 dbspace，对 IQ 主存储库来说，该大小为 8MB。

对于原始分区来说，不要显式指定 SIZE。SAP Sybase IQ 会将此参数自动设置为最大原始分区大小，如果尝试指定另一大小，将返回一条错误。

- **STRIPESIZEKB** – 指定磁盘条带化算法移至指定 dbspace 的下一条带之前要写入每个文件的字节数 (KB)。

如果不指定条带化大小或带区大小，则会应用选项 DEFAULT\_DISK\_STRIPING 和 DEFAULT\_KB\_PER\_STRIPE 的缺省值。

## 示例

(返回顶部) (第 169 页)

- **示例 1** – 为 UNIX 系统上包含两个 dbfile 的 IQ 主存储库创建一个名为 DspHist 的 dbspace。每个 dbfile 大小为 1GB 并可扩大 500MB：

```
CREATE DBSPACE DspHist USING FILE
FileHist1 '/History1/data/file1'
SIZE 1000 RESERVE 500,
FILE FileHist2 '/History1/data/file2'
SIZE 1000 RESERVE 500;
```

- **示例 2** – 创建名为 DspCat2 的另一个目录 dbspace：

```
CREATE DBSPACE DspCat2 AS
'catalog_file2'
CATALOG STORE;
```

- **示例 3** – 为 IQ 存储库创建名为 EmpStore1 的 IQ 主 dbspace（三种替代语法示例）：

```
CREATE DBSPACE EmpStore1
USING FILE EmpStore1
'EmpStore1.IQ' SIZE 8 MB IQ STORE;
```

```
CREATE DBSPACE EmpStore1
USING FILE EmpStore1
'EmpStore1.IQ' 8 IQ STORE;
```

```
CREATE DBSPACE EmpStore1
USING FILE EmpStore1
'EmpStore1.IQ' 8;
```

- **示例 4** – 创建名为 d1 的 RLV 存储 dbspace：

```
CREATE DBSPACE d1
USING FILE f1
'f1.iq' SIZE 1000 IQ RLV STORE;
```

- **示例 5** – 创建含 200 GB dbfile 的名为 myDAS 的高速缓存 dbspace：

```
CREATE DBSPACE myDAS
USING FILE iqdas1
'iqdas1.iq' SIZE 200 GB IQ CACHE STORE
```

## 用法

(返回顶部) (第 169 页)

**CREATE DBSPACE** 为 IQ 主存储库、高速缓存 **dbspace**、目录存储库或 **RLV** 存储库创建一个新的 **dbspace**。您添加的 **dbspace** 可以位于与初始 **dbspace** 不同的磁盘设备上，允许创建大于一个物理设备的存储库。

语法 1 为目录存储库创建一个 **dbspace**，在该存储库中 **dbspace** 和 **dbfile** 具有相同的逻辑名称。该目录存储库中的每个 **dbspace** 都具有单一文件。

**Dbpace** 和 **dbfile** 名称始终不区分大小写。如果数据库为 **CASE RESPECT**，则物理文件路径具有操作系统的区分大小写特性，如果数据库为 **CASE IGNORE**，则物理文件路径不区分大小写。

无法为 IQ 临时存储库创建 **dbspace**。当创建一个新数据库或升级一个在 SAP Sybase IQ 15.3 之前版本中创建的数据库时，系统将创建一个临时 **dbspace IQ\_SYSTEM\_TEMP**。可以使用 **ALTER DBSPACE ADD FILE** 语法将附加文件添加到 **IQ\_SYSTEM\_TEMP dbspace**。

---

**注意：** 创建包含最小文件的 **RLV dbspace** 是 **RLV** 存储的前提条件。在 **Simplex** 服务器上启用 **RLV** 存储前，请先检查是否存在 **RLV dbpace**。

---

可在 **Simplex** 或 **Multiplex** 节点上只创建一个高速缓存 **dbspace**。试图创建另一个高速缓存 **dbspace** 会导致错误。

可使用以下任一方式创建唯一路径：

- 为每个文件指定一个不同的扩展名（如 **mydb.iq**）
- 指定一个不同的文件名（如 **mydb2.iq**）
- 指定一个不同的路径名（如 **/iqfiles/main/iq**）或不同的原始分区

---

**警告！** 在 **UNIX** 平台上，要保持数据库的一致性，请指定链接到不同文件的文件名。**SAP Sybase IQ** 无法检测到链接文件指向的目标。即使命令中的文件名不同，也应确保它们不会指向同一个操作系统文件。

---

副作用：

- 自动提交
- 自动执行检查点操作。

## 标准

(返回顶部) (第 169 页)

- **SQL - ISO/ANSI SQL** 语法的服务商扩充。

- SAP Sybase 数据库产品 - 不受 Adaptive Server 支持。

## 权限

(返回顶部) (第 169 页)

需要 MANAGE ANY DBSPACE 系统特权。

## CREATE TABLE 语句

在数据库中或远程服务器上创建一个新表。

快速链接:

转至参数 (第 175 页)

转至示例 (第 182 页)

转至用法 (第 185 页)

转至标准 (第 186 页)

转至权限 (第 187 页)

## 语法

```

CREATE [ { GLOBAL | LOCAL } TEMPORARY ] TABLE
  [ IF NOT EXISTS ] [ owner. ] table-name
  ... ( column-definition [ column-constraint ] ...
  [ , column-definition [ column-constraint ] ... ]
  [ , table-constraint ] ... )
  | { ENABLE | DISABLE } RLV STORE

  ... [ IN dbspace-name ]
  ... [ ON COMMIT { DELETE | PRESERVE } ROWS ]
  [ AT location-string ]
  [ PARTITION BY
    range-partitioning-scheme
    | hash-partitioning-scheme
    | composite-partitioning-scheme ]

column-definition - (back to Syntax)
  column-name data-type
  [ [ NOT ] NULL ]
  [ DEFAULT default-value | IDENTITY ]
  [ PARTITION | SUBPARTITION ( partition-name IN dbspace-name
  [ , ... ] ) ]

default-value - (back to column-definition)
  special-value
  | string
  | global variable
  | [ - ] number
  | ( constant-expression )
  | built-in-function( constant-expression )

```

```

| AUTOINCREMENT
| CURRENT DATABASE
| CURRENT REMOTE USER
| NULL
| TIMESTAMP
| LAST USER

special-value - (back to default value)
CURRENT
{ DATE
| TIME
| TIMESTAMP
| USER
| PUBLISHER }
| USER

column-constraint - (back to Syntax)
[ CONSTRAINT constraint-name ] {
  { UNIQUE
    | PRIMARY KEY
    | REFERENCES table-name [ ( column-name ) ] [ action ]
  }
  [ IN dbspace-name ]
  | CHECK ( condition )
  | IQ UNIQUE ( integer )
}

table-constraint - (back to Syntax)
[ CONSTRAINT constraint-name ]
{ { UNIQUE ( column-name [ , column-name ] ... )
  | PRIMARY KEY ( column-name [ , column-name ] ... )
}
[ IN dbspace-name ]
| foreign-key-constraint
| CHECK ( condition )
| IQ UNIQUE ( integer )
}

foreign-key-constraint - (back to table-constraint)
FOREIGN KEY [ role-name ] [ ( column-name [ , column-name ] ... ) ]
...REFERENCES table-name [ ( column-name [ , column-name ] ... ) ]
...[ actions ] [ IN dbspace-name ]

actions - (back to foreign-key-constraint)
[ ON { UPDATE | DELETE } RESTRICT ]

location-string - (back to Syntax) or (back to composite-partitioning-
scheme)
{ remote-server-name. [ db-name ].[ owner ].object-name
| remote-server-name; [ db-name ]; [ owner ];object-name }

range-partitioning-scheme - (back to Syntax)
RANGE ( partition-key ) ( range-partition-decl [, range-partition-decl ... ] )

partition-key - (back to range-partitioning-scheme) or (back to hash-

```

```
partitioning-scheme)
    column-name

range-partition-decl - (back to range-partitioning-scheme)
    VALUES <= ( {constant-expr
        | MAX } [ , { constant-expr
        | MAX } ]... )
    [ IN dbspace-name ]

hash-partitioning-scheme - (back to Syntax) or (back to composite-
partitioning-scheme)
    HASH ( partition-key [ , partition-key, ... ] )

composite-partitioning-scheme - (back to Syntax)
    hash-partitioning-scheme SUBPARTITION range-partitioning-scheme
```

## 参数

(返回顶部) (第 173 页)

- **IN** - 在 `column-definition`、`column-constraint`、`table-constraint`、`foreign-key` 以及 `partition-decl` 子句中用于指定将创建对象的 `dbspace`。如果省略 `IN` 子句，SAP Sybase IQ 将在已分配表的 `dbspace` 中创建对象。

指定 `SYSTEM` 和该子句，以将永久表或临时表放在目录存储库中。指定 `IQ_SYSTEM_TEMP` 以将临时用户对象（表、分区或表索引）存储在 `IQ_SYSTEM_TEMP` 中，或者在 `TEMP_DATA_IN_SHARED_TEMP` 选项设置为 'ON' 且 `IQ_SHARED_TEMP` `dbspace` 包含 `RW` 文件时，存储在 `IQ_SHARED_TEMP` 中。（无法同时指定 `IN` 子句和 `IQ_SHARED_TEMP`。）`IN` 子句的所有其它用法均被忽略。缺省情况下，所有永久表均放置在主 `IQ` 存储库中，所有临时表放置在临时 `IQ` 存储库中。全局临时表和局部临时表决不能出现在 `IQ` 存储库中。

不支持下列语法：

```
CREATE LOCAL TEMPORARY TABLE tbl1(c1 int) IN IQ_SHARED_TEMP
```

无法将 `BIT` 数据类型列显式置于 `dbspace` 中。`BIT` 数据类型不支持以下内容：

```
CREATE TABLE t1(c1_bit bit IN iq_main);
```

- **ON COMMIT** - 仅允许用于临时表。缺省情况下，临时表的行将在 `COMMIT`（提交）时被删除。
- **AT** - 创建映射到 `location-string` 子句指定的远程位置的代理表。代理表名不能超过 30 个字符。`AT` 子句支持分号 (;) 作为分隔符。如果分号出现在 `location-string` 子句中的任何位置，则分号将用作字段分隔符。如果没有分号，则使用句号作为字段分隔符。这样一来，便可在数据库和所有者字段中使用文件名和扩展名。

分号字段分隔符主要用于当前不支持的服务器类；但是，您也可以在句点用作字段分隔符的情况下使用它们。例如，此语句将表 `proxy_a` 映射到远程服务器 `myasa` 上的 `SQL Anywhere` 数据库 `mydb`：

```
CREATE TABLE proxy_al
AT 'myasa;mydb;;al'
```

忽略远程表的外键定义。引用远程表的局部表的外键定义也会被忽略。如果远程服务器支持主键，则主键定义会发送到此服务器。

在 **Simplex** 环境中，不能在同一节点上创建引用远程表的代理表。在 **Multiplex** 环境中，不能创建引用在 **Multiplex** 中定义的远程表的代理表。

- **IF NOT EXISTS** – 如果指定的对象已存在，则不进行任何更改，也不会返回错误。
- **{ ENABLE | DISABLE } RLV STORE** – 将该表注册到 RLV 存储库以进行实时内存更新。IQ 临时表对此不支持。此值替代数据库选项 **BASE\_TABLES\_IN\_RLV** 的值。将此值设置为 **ENABLE** 需要 **CREATE TABLE** 系统特权以及 RLV 存储 **dbspace** 的 **CREATE** 权限。
- **column-definition** – 定义表列。所允许的数据类型在《参考：构件块、表和过程》> “SQL 数据类型” 中进行了介绍。同一表中的两列不能同名。最多可以创建 45,000 列；但如果一个表中的列多于 10,000，则可能会导致性能下降。
  - **[ NOT ] NULL** – 包括或排除 NULL 值。如果指定 **NOT NULL**，或者列具有 **UNIQUE** 或 **PRIMARY KEY** 约束，则该列不能含有任何 NULL 值。每个表中允许 NULL 的列数存在限制，最多为大约  $8 * (\text{database-page-size} - 30)$ 。
  - **DEFAULT default-value** – 通过 **CREATE TABLE** (和 **ALTER TABLE**) 语句中的 **DEFAULT** 关键字指定缺省列值。**DEFAULT** 值用作未指定列值的任何 **INSERT** (或 **LOAD**) 语句中列的值。
  - **DEFAULT AUTOINCREMENT** – **DEFAULT AUTOINCREMENT** 列的值唯一地标识表中的每一行。此类型的列也称为 **IDENTITY** 列，可与 **Adaptive Server** 兼容。**IDENTITY/DEFAULT AUTOINCREMENT** 列存储在插入和更新期间自动生成的顺序编号。使用 **IDENTITY** 或 **DEFAULT AUTOINCREMENT** 时，列必须是小数位数为 0 的整数数据类型之一或精确的数字类型。列值也可以是 **NULL**。必须使用所有者名称来限定指定的表名。

**ON** 可向表中插入数据。如果没有指定 **IDENTITY/DEFAULT AUTOINCREMENT** 列的值，则生成一个比列中的任何其它值都大的唯一值。如果 **INSERT** 指定了列值，则使用该值；如果指定的值不大于列的当前最大值，该值将用作后续插入的起点。

删除行不会递减 **IDENTITY/AUTOINCREMENT** 计数器的值。由于删除行而产生的间隙只能由使用插入时的显式赋值填充。数据库选项 **IDENTITY\_INSERT** 必须设置为表名，才能向 **IDENTITY/AUTOINCREMENT** 列执行插入操作。

例如，以下示例创建一个带有 **IDENTITY** 列的表，并向其显式添加一些数据：

```
CREATE TABLE mytable(c1 INT IDENTITY);
SET TEMPORARY OPTION IDENTITY_INSERT = "DBA".mytable;
INSERT INTO mytable VALUES(5);
```

显式插入小于该列最大值的行号后，后面没有显式赋值的行仍自动递增为比上一个最大值大 1 的值。

通过检查 @@identity 全局变量，可以找到最近一次插入的列值。

- **IDENTITY** - 使用 AUTOINCREMENT 缺省值的替代方法，与 Transact-SQL® 兼容。在 SAP Sybase IQ 中，可使用 IDENTITY 或 DEFAULT AUTOINCREMENT 子句来创建标识列。
- **table-constraint** - 帮助确保数据库中数据的完整性。共有四种类型的完整性约束：
  - **UNIQUE** - 标识唯一识别表中各行的一列或多列。表中任何两行的值在所有指定的列中不能相同。表可以有多个唯一约束。
  - **PRIMARY KEY** - 与 UNIQUE 约束相同，但表只能有一个主键约束。不能为同一列同时指定 PRIMARY KEY 约束和 UNIQUE 约束。主键通常标识行的最佳标识符。例如，客户号可能是 customer 表的主键。
  - **FOREIGN KEY** - 将某组列的值局限于与另一个表的主键或唯一约束中的值匹配。例如，外键约束可用于确保 invoice 表中的客户号与 customer 表中的客户号相对应。

不能在局部临时表上创建外键约束。全局临时表必须用 ON COMMIT PRESERVE ROWS 创建。

- **CHECK** - 允许对任意条件进行校验。例如，检查约束可用于确保名为 Gender 的列只包含值 male 或 female。表中的任何行都不能违反约束。如果 INSERT 或 UPDATE 语句会导致行违反约束，则不允许执行相应操作并且撤消语句的作用。列检查约束中以符号 '@' 开头的列标识符是实际列名的占位符。以下格式的语句：

```
CREATE TABLE t1(c1 INTEGER CHECK (@foo < 5))
```

与下面的语句完全相同：

```
CREATE TABLE t1(c1 INTEGER CHECK (c1 < 5))
```

表检查约束中以符号 '@' 开头的列标识符不是占位符。

如果一条语句会引起数据库的更改并且这些更改会使数据库违反完整性约束，则该语句不会得到有效执行，并且系统会报告一条错误。（有效是指系统检测到错误之前该语句所做的任何更改都被撤消。）

SAP Sybase IQ 通过为该列创建 HG 索引来强制执行单列 UNIQUE 约束。

---

**注意：** 不能将含有 BIT 数据类型的列定义为 UNIQUE 或 PRIMARY KEY 约束。另外，BIT 数据类型的列缺省设置为不允许 NULL 值；您可以通过显式定义该列允许 NULL 值来更改这种情况。

---

- **column-constraint** - 限制列可以容纳的值。列约束和表约束有助于确保数据库中数据的完整性。如果语句会导致违反约束，则该语句的执行不会完成。该语句在

检测到错误前所做的任何更改都被撤消并报告错误。列约束是相应表约束的缩写。例如，下列语句是等效的：

```
CREATE TABLE Products (
    product_num integer UNIQUE
)
CREATE TABLE Products (
    product_num integer,
    UNIQUE ( product_num )
)
```

通常使用列约束，除非约束引用了表中的多个列。此类情况下，必须使用表约束。

- **IQ UNIQUE** - 定义列的预期基数并确定是作为 Flat FP 还是 NBit FP 来装载该列。IQ UNIQUE(n) 值显式设置为 0 会作为 Flat FP 装载该列。不带 IQ UNIQUE 约束的列将隐式装载为 NBit，一直到 FP\_NBIT\_AUTOSIZE\_LIMIT、FP\_NBIT\_LOOKUP\_MB 和 FP\_NBIT\_ROLLOVER\_MAX\_MB 选项定义的限制：
  - FP\_NBIT\_AUTOSIZE\_LIMIT 限制装载为 NBit 的不同值的数量
  - FP\_NBIT\_LOOKUP\_MB 为 NBit 字典总大小设置阈值
  - FP\_NBIT\_ROLLOVER\_MAX\_MB 为从 NBit 至 Flat FP 的隐式 NBit 切换设置字典大小
  - FP\_NBIT\_ENFORCE\_LIMITS 强制执行 NBit 字典大小限制。此选项缺省设置为 OFF

不必使用 n 值小于 FP\_NBIT\_AUTOSIZE\_LIMIT 的 IQ UNIQUE。自动大小调整功能会自动将所有中低基数列的大小调整为 NBit。如果想要将列装载为 Flat FP，或者想要在不同值的数量超出 FP\_NBIT\_AUTOSIZE\_LIMIT 时将列装载为 NBit，请使用 IQ UNIQUE。

---

#### 注意：

- 当指定高 IQ UNIQUE 值时，请考虑内存的使用情况。如果计算机资源受到限制，应避免带有 FP\_NBIT\_ENFORCE\_LIMITS='OFF'（缺省设置）的装载。在 SAP Sybase IQ 16.0 之前，IQ UNIQUE n 值 > 16777216 时会切换到 Flat FP。在 16.0 中，支持对较大的 IQ UNIQUE 值进行标识化，但鉴于基数和列宽，该值可能需要非常多的内存资源。
  - BIT、BLOB 和 CLOB 数据类型不支持 NBit 字典压缩。如果 FP\_NBIT\_IQ15\_COMPATIBILITY='OFF'，则在包含这些数据类型的 CREATE TABLE 或 ALTER TABLE 语句中指定的非零 IQ UNIQUE 列将返回一个错误。
- 
- **column-constraint** 和 **table-constraint** 子句 - 列约束和表约束有助于确保数据库中数据的完整性。
    - **PRIMARY KEY** 或 **PRIMARY KEY ( column-name, ... )** - 表的主键由列出的列组成，指定的列都不可含有任何 NULL 值。SAP Sybase IQ 确保表中的每一行都有唯一的主键值。表只能有一个 PRIMARY KEY。



采用第二种格式 (**PRIMARY KEY** 后面跟有列的列表) 时, 创建的主键所包括的列是按其定义的顺序排序的, 而不是按列出的顺序排序。

将列指定为 **PRIMARY KEY**、**FOREIGN KEY** 或 **UNIQUE** 时, **SAP Sybase IQ** 会自动为其创建 **High\_Group** 索引。对于多列主键, 该索引位于主键上, 而不是各个列上。为了获得最佳性能, 应单独用 **HG** 或 **LF** 索引对名列创建索引。

- **REFERENCES primary-table-name [(primary-column-name)]** - 将列定义为主键的外键, 或主表的唯一约束。通常, 外键针对的是主键而不是唯一约束。如果指定主列名, 则它必须与主表中的某一列相匹配, 而该列受唯一约束或主键约束的制约, 而且该约束必须仅包含这一列。否则, 外键会引用第二个表的主键。主键与外键必须具有相同的数据类型、精度、小数位数和符号。仅会为单列外键创建非唯一单列 **HG** 索引。对于多列外键, **SAP Sybase IQ** 将创建非唯一复合 **HG** 索引。唯一或非唯一 **HG** 索引的多列组合键的最大宽度为 **1KB**。

临时表不能有引用基表的外键, 而基表不能有引用临时表的外键。局部临时表不能有外键, 也不能被外键引用。

- **FOREIGN KEY [role-name] [(...)] REFERENCES primary-table-name [(...)]** - 定义对另一个表中主键或唯一约束的外键引用。通常, 外键针对的是主键而不是唯一约束。(此处所述的另一个表称为主表。)

如果未指定主表列名, 主表列即是表的主键中的列。如果未指定外键列名, 则外键列的列名与主表的列名相同。如果指定外键列名, 则还必须指定主键列名, 列名按照列表中的位置成对出现。

如果主表与外键表不一样, 则表明唯一约束或主键约束已在被引用键上定义。被引用键和外键必须具有相同的列数, 以及相同的数据类型、符号、精度和小数位数。

行的外键值必须作为主表中某一行的候选键值出现, 除非外键的允许 **null** 值的列中有一列或多列包含 **null** 值。

所有非显式定义的外键列在创建时会被自动赋予与主表中相应列相同的数据类型。这些自动创建的列不能属于外表的主键。因此, 必须显式创建同时在主键和外键中使用的列。

*role-name* 是外键的名称。*role-name* 的主要作用是区分同一表的两个外键。如果未指定 *role-name*, 则按如下方式分配角色名:

1. 如果没有与表名同名的 *role-name* 外键, 则将表名指派为 *role-name*。
2. 如果表名已被使用, 则 *role-name* 为表名加上表的唯一 3 位零填充数字。

参照完整性操作可定义为维护数据库中的外键关系而采取的操作。更改主键值或从数据库表中删除主键值时, 其它表中可能有一些相应的外键值需要以某种方式修改。可以指定 **ON DELETE** 子句, 后跟 **RESTRICT** 子句。

- **RESTRICT** - 如果在数据库的其它地方存在相应的外键时您试图更新或删除主键值, 则将生成错误。如果试图更新外键, 以便通过候选键创建不匹配的新值, 则将生成错误。这是缺省操作, 除非您指定 **LOAD** 可以拒绝违反参照完整性的行。这样便会在语句级别强制实施参照完整性。

如果使用 **CHECK ON COMMIT** 时未指定任何操作，那么 **RESTRICT** 将被视为 **DELETE** 的一项操作。SAP Sybase IQ 不支持 **CHECK ON COMMIT**。

全局临时表不能有引用基表的外键，而基表不能有引用全局临时表的外键。局部临时表不能有外键，也不能被外键引用。

- **CHECK (条件)** – 不允许行违反条件。如果 **INSERT** 语句会导致行违反条件，则不允许进行该操作并且撤销语句的作用。

只有条件为 **FALSE** 时才会拒绝更改；特别是，当条件为 **UNKNOWN** 时，允许进行更改。SAP Sybase IQ 不会强制遵守 **CHECK** 条件。

---

**注意：** 如果可能，不要在 SAP Sybase IQ 中定义参照完整性外键-主键关系，除非您确定其中没有孤立外键。

---

- **远程表** – 忽略远程表的外键定义。引用远程表的局部表的外键定义也会被忽略。如果远程服务器支持主键，则主键定义会发送到此服务器。
- **PARTITION BY** – 将大表拆分为多个便于管理的小存储对象。各个分区共用父表的相同逻辑属性，但可放置在单独的 **dbspace** 中并分别进行管理。SAP Sybase IQ 支持多个表分区模式：
  - 散列分区
  - 范围分区
  - 复合分区

分区键是包含表分区键的一个或多个列。分区键可包含 **NULL** 和 **DEFAULT** 值，但不能包含：

- **LOB (BLOB 或 CLOB)** 列
- **BINARY** 或 **VARBINARY** 列
- 长度超过 255 个字节的 **CHAR** 或 **VARCHAR** 列
- **BIT** 列
- **FLOAT/DOUBLE/REAL** 列
- **PARTITION BY RANGE** – 根据分区列中的值范围对行进行分区。范围分区被限定为有 1 个分区键列和最多 1024 个分区。在范围分区模式中，分区键是包含表分区键的列：

```
range-partition-decl:  
    partition-name VALUES <= ( {constant-expr | MAX } [ ,  
{ constant-expr | MAX } ]... )  
    [ IN dbspace-name ]
```

分区名称是存储表行的新分区的名称。在表分区集合中，分区名称必须唯一。分区名称是必需项。

- **VALUE** – 按升序为每个分区指定上限（含上限）。用户必须为每个范围分区指定分区条件，确保将每一行只分配到一个分区。分区列允许 **NULL** 值，将 **NULL** 作为分区键值的行属于第一个表分区。但 **NULL** 不能是界限值。

第一个分区没有下限（MIN 值）。分区键第一列中单元格值为 NULL 的行属于第一个分区。对于最后一个分区，您可以指定一个上限（含上限）或 MAX。如果最后一个分区上限值不是 MAX，则加载或插入其分区键值大于最后一个分区上限值的任何行都将生成错误。

- **Max** - 表示无限大的上限，只能为最后一个分区指定。
- **IN** - 在 `partition-decl` 中指定其中应该包含分区行的 `dbspace`。

以下限制影响范围分区表的分区键和界限值：

- 分区界限必须是常量，而不是常量表达式。
- 分区界限必须按照分区的创建顺序以升序排列。也就是说，第二个分区上限必须大于第一个分区上限，依此类推。

此外，分区界限值必须与相应的分区键列数据类型兼容。例如，`VARCHAR` 与 `CHAR` 兼容。

- 如果某个界限值的数据类型不同于其对应分区键列的数据类型，**SAP Sybase IQ** 会将界限值的数据类型转换为分区键列的数据类型，但存在以下例外情况：
- 不允许进行显式转换。以下示例尝试将 `INT` 显式转换为 `VARCHAR`，并生成错误：

```
CREATE TABLE Employees(emp_name VARCHAR(20))
PARTITION BY RANGE(emp_name)
(p1 VALUES <=(CAST (1 AS VARCHAR(20))),
p2 VALUES <= (CAST (10 AS VARCHAR(20)))
```

- 不允许执行可导致数据丢失的隐式转换。在以下示例中，分区界限与分区键类型不兼容。舍入假设可导致数据丢失，并生成错误：

```
CREATE TABLE emp_id (id INT) PARTITION BY RANGE(id) (p1 VALUES
<= (10.5), p2 VALUES <= (100.5))
```

- 在以下示例中，分区界限与分区键数据类型兼容。界限值会直接转换为浮点值。无需进行舍入处理，且支持转换：

```
CREATE TABLE id_emp (id FLOAT)
PARTITION BY RANGE(id) (p1 VALUES <= (10),
p2 VALUES <= (100))
```

- 不允许将非二进制数据类型转换为二进制数据类型。例如，不允许进行以下转换，并会返回错误：

```
CREATE TABLE newemp (name BINARY)
PARTITION BY RANGE(name)
(p1 VALUES <= ("Maarten"),
p2 VALUES <= ("Zymlerman"))
```

- 在范围分区表中，不能将 `NULL` 用作界限。
- 如果分区键第一列中的单元格值计算结果为 `NULL`，则对应行将属于第一个分区。**SAP Sybase IQ** 仅支持一列分区键，因此分区键中的任何 `NULL` 都会将对应行分配到第一个分区。
- **PARTITION BY HASH** - 基于内部散列函数处理的分区-键值向各个分区映射数据。散列分区键被限定为最多有 8 个列，其声明的列宽总和小于等于 5300 个字节。对于散列分区，表创建者仅确定分区键列；分区的数量和位置将在内部确定。

在散列分区声明中，分区键是一个或多个列，其合成值将确定存储有各行数据的分区：

```
hash-partitioning-scheme:
HASH ( partition-key [ , partition-key, ... ] )
```

- **限制 -**

- 只能对基表进行散列分区。试图对全局临时表或局部临时表进行分区将会引发错误。
- 不能添加、删除、合并或拆分散列分区。
- 不能为散列分区键添加列或从中删除列。
- **PARTITION BY HASH RANGE** - 按范围对已进行散列分区的表划分子分区。在散列范围分区模式声明中，**SUBPARTITION BY RANGE** 子句将向现有散列范围分区表添加一个新的范围子分区：

```
hash-range-partitioning-scheme:
PARTITION BY HASH ( partition-key [ , partition-key, ... ] )
  [ SUBPARTITION BY RANGE ( range-partition-decl [ , range-
partition-decl ... ] ) ]
```

散列分区指定如何在逻辑上分布和托管数据；范围子分区指定如何在物理上放置数据。新的范围子分区通过散列进行逻辑分区，其散列分区键与现有的散列范围分区表相同。范围子分区键被限定为仅一列。

- **限制 -**

- 只能对基表进行散列分区。试图对全局临时表或局部临时表进行分区将会引发错误。
- 不能添加、删除、合并或拆分散列分区。
- 不能为散列分区键添加列或从中删除列。

---

**注意：** 与散列范围分区相同，范围分区和复合分区模式要求有单独授权的 VLDB 管理组件。

---

## 示例

(返回顶部) (第 173 页)

- **示例 1** - 创建一个名为 SalesOrders2 的表，其中包含 5 列。FinancialCode、OrderDate 和 ID 列的数据页位于 dbspace Dsp3 中。整数列 CustomerID 的数据页位于 dbspace Dsp1 中。CLOB 列 History 的数据页位于 dbspace Dsp2 中。主键 (ID 的 HG) 的数据页位于 dbspace Dsp4 中：

```
CREATE TABLE SalesOrders2 (
  FinancialCode CHAR(2),
  CustomerID int IN Dsp1,
  History CLOB IN Dsp2,
  OrderDate TIMESTAMP,
  ID BIGINT,
```

```
PRIMARY KEY(ID) IN Dsp4
) IN Dsp3
```

- **示例 2** - 创建一个名为 `fin_code2` 的表，其中包含 4 列。`code`、`type` 和 `id` 列的数据页位于数据库选项 `DEFAULT_DBSpace` 的值所确定的缺省 `dbspace` 中。`CLOB` 列 `description` 的数据页位于 `dbspace Dsp2` 中。外键 `fk1` (`c1` 的 `HG`) 中的数据页位于 `dbspace Dsp4` 中：

```
CREATE TABLE fin_code2 (
  code INT,
  type CHAR(10),
  description CLOB IN Dsp2,
  id BIGINT,
  FOREIGN KEY fk1(id) REFERENCES SalesOrders(ID) IN Dsp4
)
```

- **示例 3** - 创建表 `t1`，其中分区 `p1` 与 `p2` 相邻，分区 `p2` 与 `p3` 相邻：

```
CREATE TABLE t1 (c1 INT, c2 INT)
PARTITION BY RANGE(c1)
(p1 VALUES <= (0), p2 VALUES <= (10), p3 VALUES <= (100))
```

- **示例 4** - 创建包含 6 列和 3 个分区的 `RANGE` 分区表 `bar`，用于基于日期向各个分区映射数据：

```
CREATE TABLE bar (
  c1 INT IQ UNIQUE(65500),
  c2 VARCHAR(20),
  c3 CLOB PARTITION (P1 IN Dsp11, P2 IN Dsp12,
    P3 IN Dsp13),
  c4 DATE,
  c5 BIGINT,
  c6 VARCHAR(500) PARTITION (P1 IN Dsp21,
    P2 IN Dsp22),
  PRIMARY KEY (c5) IN Dsp2) IN Dsp1
PARTITION BY RANGE (c4)
(P1 VALUES <= ('2006/03/31') IN Dsp31,
 P2 VALUES <= ('2006/06/30') IN Dsp32,
 P3 VALUES <= ('2006/09/30') IN Dsp33
) ;
```

每个分区的数据页分配：

分区	DbSpace	列
P1	Dsp31	c1、c2、c4、c5
P1	Dsp11	c3
P1	Dsp21	c6
P2	Dsp32	c1、c2、c4、c5
P2	Dsp12	c3

分区	DbSPACE	列
P2	Dsp22	c6
P3	Dsp33	c1、c2、c4、c5、c6
P3	Dsp13	c3
P1、P2、P3	Dsp1	c1 的查找存储和其它共享数据
P1、P2、P3	Dsp2	主键 (c5 的 HG)

- **示例 5** - 创建包含 PRIMARY KEY (列 c1) 和 HASH PARTITION KEY (列 c4 和 c3) 的 HASH 分区表 (table tbl42)。

```
CREATE TABLE tbl42 (
  c1 BIGINT NOT NULL,
  c2 CHAR(2) IQ UNIQUE(50),
  c3 DATE IQ UNIQUE(36524),
  c4 VARCHAR(200),
  PRIMARY KEY (c1)
)
PARTITION BY HASH ( c4, c3 )
```

- **示例 6** - 创建含有 PRIMARY KEY (列 c1)、散列分区键 (列 c4 和 c2) 和范围子分区键 (列 c3) 的散列范围分区表。

```
CREATE TABLE tbl42 (
  c1 BIGINT NOT NULL,
  c2 CHAR(2) IQ UNIQUE(50),
  c3 DATE,
  c4 VARCHAR(200),
  PRIMARY KEY (c1)) IN Dsp1

PARTITION BY HASH ( c4, c2 )
SUBPARTITION BY RANGE ( c3 )
( P1 VALUES <= (2011/03/31) IN Dsp31,
  P2 VALUES <= (2011/06/30) IN Dsp32,
  P3 VALUES <= (2011/09/30) IN Dsp33) ;
```

- **示例 7** - 为图书馆数据库创建用于保存借出图书信息的表：

```
CREATE TABLE borrowed_book (
  date_borrowed DATE NOT NULL,
  date_returned DATE,
  book CHAR(20)
  REFERENCES library_books (isbn),
  CHECK( date_returned >= date_borrowed )
)
```

- **示例 8** - 在远程服务器 SERVER\_A 上创建表 t1，并创建映射到该远程表的名为 t1 的代理表：

```
CREATE TABLE t1
( a INT,
```

```
b CHAR(10))
AT 'SERVER_A.dbl.joe.t1'
```

- **示例 9** - 创建表 `tab1`，其中包含以特殊常量 `LAST USER` 为缺省值的列 `c1`：

```
CREATE TABLE tab1(c1 CHAR(20) DEFAULT LAST USER)
```

- **示例 10** - 创建包含列 `c1` 的局部临时表 `tab1`：

```
CREATE LOCAL TEMPORARY TABLE tab1(c1 int) IN IQ_SYSTEM_TEMP
```

在以下情况下，本例将在 `IQ_SYSTEM_TEMP` dbspace 中创建 `tab1`：

- `DQP_ENABLED` 逻辑服务器策略选项设置为 `ON`，但在 `IQ_SHARED_TEMP` 中没有读写文件
- `DQP_ENABLED` 选项为 `OFF`，`TEMP_DATA_IN_SHARED_TEMP` 逻辑服务器策略选项为 `ON`，但在 `IQ_SHARED_TEMP` 中没有读写文件
- `DQP_ENABLED` 选项和 `TEMP_DATA_IN_SHARED_TEMP` 选项均设置为 `OFF`

在以下情况下，本例将在 `IQ_SHARED_TEMP` dbspace 中创建相同的表 `tab1`：

- `DQP_ENABLED` 设置为 `ON`，且在 `IQ_SHARED_TEMP` 中具有读写文件
  - `DQP_ENABLED` 设置为 `OFF`，`TEMP_DATA_IN_SHARED_TEMP` 设置为 `ON`，且在 `IQ_SHARED_TEMP` 中具有读写文件
- **示例 11** - 创建表 `tab1`，允许在内存 `RLV` 存储库中使用行级版本控制和实时存储。

```
CREATE TABLE tab1 ( c1 INT, c2 CHAR(25) ) ENABLE RLV STORE
```

## 用法

(返回顶部) (第 173 页)

通过指定所有者名称，可为其他用户创建表。如果未指定 `GLOBAL TEMPORARY` 或 `LOCAL TEMPORARY`，该表被称为基表。否则，该表为临时表。

与基表一样，所创建的全局临时表也存在于数据库中，并一直保留在数据库中，直到通过 `DROP TABLE` 语句将其显式删除。临时表中的行仅对插入这些行的连接可见。来自相同或不同应用程序的多个连接可同时使用同一个临时表，但每个连接只能看到它自己的那些行。给定的连接会继承当该连接首次引用全局临时表时该表的模式。连接结束时删除临时表中的行。

创建局部临时表时，请不要指定所有者。如果在创建临时表时指定所有者（例如，`CREATE TABLE dbo.#temp(coll int)`），就会将其错误地创建成基表。

如果连接中存在某一局部临时表，则尝试创建同名的基表或全局临时表将失败，因为 `owner.table` 无法唯一地标识新表。

但是，当存在基表或全局临时表时，可创建同名的局部临时表。引用表名时将访问局部临时表，因为将首先解析局部临时表。

例如，请考虑以下序列：

```
CREATE TABLE t1 (c1 int);
INSERT t1 VALUES (9);

CREATE LOCAL TEMPORARY TABLE t1 (c1 int);
INSERT t1 VALUES (8);

SELECT * FROM t1;
```

返回的结果为 8。对 t1 的任何引用都将引用局部临时表 t1，一直到局部临时表被连接删除。

在过程中，如果要创建一个在过程完成后仍然保留的表，可使用 **CREATE LOCAL TEMPORARY TABLE** 语句，而不是 **DECLARE LOCAL TEMPORARY TABLE** 语句。使用 **CREATE LOCAL TEMPORARY TABLE** 语句创建的局部临时表会一直保留到它被显式删除或连接终止。

利用使用 **CREATE LOCAL TEMPORARY TABLE** 的 **IF** 语句创建的局部临时表，在 **IF** 语句完成后也会继续保留。

SAP Sybase IQ 不支持将 **CREATE TABLE ENCRYPTED** 子句用于对 SAP Sybase IQ 表进行表级别加密。但 SAP Sybase IQ 数据库中的 SQL Anywhere 表支持 **CREATE TABLE ENCRYPTED** 子句。

#### Side Effects

- 自动提交

## 标准

(返回顶部) (第 173 页)

- **SQL - ISO/ANSI SQL 语法的服务商扩充。**  
以下是服务商扩充：
  - { **IN | ON** } *dbspace-name* 子句
  - **ON COMMIT** 子句
  - 部分缺省值
- SAP Sybase 数据库产品 - 受 Adaptive Server 支持，但存在一些差异。
  - **临时表** - 通过在 **CREATE TABLE** 语句中的表名前加上井号 (#)，可创建临时表。这些临时表是 SAP Sybase IQ 声明的临时表，只能在当前连接中可用。有关声明的临时表的信息，请参见 **DECLARE LOCAL TEMPORARY TABLE** 语句。
  - **实际放置** - SAP Sybase IQ 中表的实际放置方式不同于 Adaptive Server 中表的实际放置方式。Adaptive Server 支持的 **ON segment-name** 子句在 SAP Sybase IQ 中也受支持，但 *segment-name* 指的是 IQ *dbspace*。
  - **约束** - SAP Sybase IQ 不支持已命名的约束或已命名的缺省值，但确实支持允许将约束和缺省值定义封装在数据类型定义中的用户定义数据类型。它还支持在 **CREATE TABLE** 语句中使用显式缺省值和 **CHECK** 条件。



- **NULL** - (缺省值) 缺省情况下，Adaptive Server 中的列缺省设置为 NOT NULL，而在 SAP Sybase IQ 中，缺省设置为 NULL，从而允许 NULL 值。可使用 ALLOW\_NULLS\_BY\_DEFAULT 选项控制此设置。请参见 ALLOW\_NULLS\_BY\_DEFAULT 选项 [TSQL]。要使您的数据定义语句成为可移植的语句，应显式指定 NULL 或 NOT NULL。

## 权限

(返回顶部) (第 173 页)

表类型	所需特权
IQ 主存储库中的基表	<p>自有表 - 针对创建表的 dbspace 需要具备 CREATE 特权。还需要具备以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE TABLE 系统特权。</li> <li>• CREATE ANY OBJECT 系统特权。</li> </ul> <p>由任何用户拥有的表 - 需要对创建表的 dbspace 具有 CREATE 特权。还需要具备以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY TABLE 系统特权。</li> <li>• CREATE ANY OBJECT 系统特权。</li> </ul>
全局临时表	<p>由自身拥有的表 - 需要具备以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE TABLE 系统特权。</li> <li>• CREATE ANY OBJECT 系统特权。</li> </ul> <p>由任何用户拥有的表 - 需要具备以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY TABLE 系统特权。</li> <li>• CREATE ANY OBJECT 系统特权。</li> </ul>
代理表	<p>由自身拥有的表 - 需要具备以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE PROXY TABLE 系统特权。</li> <li>• CREATE ANY TABLE 系统特权。</li> <li>• CREATE ANY OBJECT 系统特权。</li> </ul> <p>由任何用户拥有的表 - 需要具备以下特权之一：</p> <ul style="list-style-type: none"> <li>• CREATE ANY TABLE 系统特权。</li> <li>• CREATE ANY OBJECT 系统特权。</li> </ul>

## 另请参见

- CREATE DBSPACE 语句 (第 169 页)

## DELETE 语句

删除指定表中所有满足搜索条件的行。如果没有指定 **WHERE** 子句，则删除指定表中的所有行。

快速链接：

转至参数 (第 188 页)

转至示例 (第 189 页)

转至用法 (第 189 页)

转至标准 (第 190 页)

转至权限 (第 190 页)

### 语法

#### DELETE

```
[ FROM ] [ owner. ] table-name [[AS correlation-name]
...[ FROM table-expression ]
[ WHERE search-condition ]]
```

#### table-expression

```
table-spec
| table-expression join-type table-spec [ ON condition ]
| table-expression, ...
```

### 参数

(返回顶部) (第 188 页)

- **FROM 子句** – 指明要从中删除行的表。**DELETE** 语句中可选的第二个 **FROM** 子句根据与其它表的连接确定要从特定表中删除的行。如果存在第二个 **FROM** 子句，则 **WHERE** 子句将限定此 **FROM** 子句的行。将根据第一个 **FROM** 子句中给定的表名从对应表中删除行。

---

**注意：** 不能在连接虚拟表上使用 **DELETE** 语句。如果尝试在连接虚拟表中进行删除操作，则会报告错误。

---

当 **FROM** 子句并不都使用相关名时，**DELETE** 语句中的表名具有潜在的不明确性。请看下面的示例：

```
DELETE
FROM table_1
FROM table_1 AS alias_1, table_2 AS alias_2
WHERE ...
```

`table_1` 在第一个 **FROM** 子句中未用相关名进行标识，而在第二个 **FROM** 子句中则使用了相关名进行标识。在第二个 **FROM** 子句中使用 `table_1` 的相关名可确保语句中仅存在一个 `table_1` 实例。这是个特例，一般的规则是，如果一个

表在同一语句中被标识两次，一次使用了相关名，另一次未用相关名，则认为这两次标识的是该表的两个实例。

现在，请看下面的示例：

```
DELETE
FROM table_1
FROM table_1 AS alias_1, table_1 AS alias_2
WHERE ...
```

第二个 **FROM** 子句中有两个 `table_1` 实例。由于无法确定应用于标识第一个 **FROM** 子句的实例，相关名的一般规则是不使用第二个子句中的 `table_1` 这两个实例标识第一个 **FROM** 子句中的 `table_1`：语句中有三个 `table_1` 实例。

- **WHERE 子句** - 如果已指定，则只删除满足搜索条件的行。如果未指定 **WHERE** 子句，则删除所有行。

## 示例

(返回顶部) (第 188 页)

- **示例 1** - 将 ID 为 105 的雇员从数据库中删除：

```
DELETE
FROM Employees
WHERE EmployeeID = 105
```

- **示例 2** - 从 `FinancialData` 表中删除 1993 年之前的所有数据：

```
DELETE
FROM FinancialData
WHERE Year < 1993
```

- **示例 3** - 从 `Contacts` 表中删除所有 `Customers` 表中已存在的名称：

```
DELETE
FROM Contacts
FROM Contacts, Customers
WHERE Contacts.Surname = Customers.Surname
AND Contacts.GivenName = Customers.GivenName
```

## 用法

(返回顶部) (第 188 页)

如果定义视图的 **SELECT** 语句的 **FROM** 子句中只有一个表，并且该语句不包含 **GROUP BY** 子句、集合函数或不涉及 **UNION** 操作，则可以对视图使用 **DELETE** 语句。

如果正在从中删除行的表名称同时用于两个 **FROM** 子句，则可将子句视为引用该表，条件是下列其中一项为真：

- 两个表引用均未通过指定用户 ID 进行限定
- 两个表引用均已通过指定用户 ID 进行限定

- 两个表引用均通过相关名指定

如果服务器无法确定表引用是否相同，则会显示错误。这可避免用户删除非预期行后产生意外语义。

## 标准

(返回顶部) (第 188 页)

- SQL - ISO/ANSI SQL 语法的服务商扩充。
- SAP Sybase 数据库产品 - 受 Adaptive Server 支持，包括服务商扩充。

## 权限

(返回顶部) (第 188 页)

需要针对表的 DELETE 特权。

## DROP 语句

从数据库中删除对象。

快速链接：

[转至参数 \(第 190 页\)](#)

[转至示例 \(第 191 页\)](#)

[转至用法 \(第 192 页\)](#)

[转至标准 \(第 192 页\)](#)

[转至权限 \(第 192 页\)](#)

## 语法

### **DROP**

```
{ DBSPACE dbspace-name
| { DATATYPE [ IF EXISTS ]
| DOMAIN } datatype-name
| EVENT [ IF EXISTS ] event-name
| INDEX [ IF EXISTS ] [ [ owner ]. table-name . ] index-name
| MESSAGE message-number
| TABLE [ IF EXISTS ] [ owner . ] table-name
| VIEW [ IF EXISTS ] [ owner . ] view-name
| MATERIALIZED VIEW [ IF EXISTS ] [ owner . ] view-name
| PROCEDURE [ IF EXISTS ] [ owner . ] procedure-name
| FUNCTION [ IF EXISTS ] [ owner . ] function-name }
```

## 参数

(返回顶部) (第 190 页)

- **IF EXISTS** - 如果您不希望在 **DROP** 语句试图删除不存在的数据库对象时返回错误，请使用此子句。
- **INDEX** - 删除任何显式创建的索引。仅当不存在唯一约束或外键约束或者不存在关联的主键时，此语句才删除隐式创建的索引。

**DROP INDEX** 如果存在关联的未实施外键，则删除非唯一 **HG** 索引失败。

---

**警告！** 请勿删除由 **DBO** 用户拥有的视图。删除此类视图或将它们转化为表可能导致出现问题。

---

只要 **DROP INDEX** 语句影响了当前正由其它连接使用的表，就会禁止该语句。

- **TABLE** - 如果主表具有与 **DROP TABLE** 相关联的外键约束（包括未实施的外键约束），则会禁止该语句

如果相应表包含 **IDENTITY** 列且 **IDENTITY\_INSERT** 设置为该表，也会禁止 **DROP TABLE**。要删除该表，必须清除 **IDENTITY\_INSERT**（即将 **IDENTITY\_INSERT** 设置为空字符串 ''），或者将其设置为其它表名。

外键可以具有非唯一的单列或多列 **HG** 索引。主键可以具有唯一的单列或多列 **HG** 索引。无法删除为现有外键、主键和唯一约束隐式创建的 **HG** 索引。

四个初始 **dbspace** 为 **SYSTEM**、**IQ\_SYSTEM\_MAIN**、**IQ\_SYSTEM\_TEMP** 和 **IQ\_SYSTEM\_MSG**。您不能删除这些初始 **dbspace**，但您可以从可能包含多个 **dbspace** 的 **IQ** 主存储库或目录存储库中删除 **dbspace**，前提是至少一个 **dbspace** 仍处于读写模式。

必须先将 **dbspace** 中的表删除，然后才能删除 **dbspace**。如果该 **dbspace** 仍包含用户数据，则会返回错误；该 **dbspace** 删除后，将自动重新分配其它结构。只有使 **dbspace** 处于只读模式后，才能删除它。

---

**注意：** **dbspace** 被命令使用后，在任何时刻都可能包含数据，因而会禁止对它执行 **DROP DBSPACE**。

---

只要 **DROP TABLE** 语句影响了当前正由其它连接使用的表，就会禁止该语句。

- **PROCEDURE** - 其它连接正在使用相应过程时，禁止执行 **DROP PROCEDURE**。
- **DATATYPE** - 如果表中用到相应数据类型，则会禁止执行 **DROP DATATYPE**。要删除数据类型，必须更改以某种用户定义数据类型定义的所有列的数据类型。建议使用 **DROP DOMAIN** 而不是 **DROP DATATYPE**，因为 **DROP DOMAIN** 是 ANSI/ISO SQL3 草案中使用的语法。

## 示例

(返回顶部) (第 190 页)

- **示例 1** - 从数据库中删除 **Departments** 表：

```
DROP TABLE Departments
```

- **示例 2** - 从数据库中删除 emp\_dept 视图:

```
DROP VIEW emp_dept
```

- **示例 3** - 从您连接的 Simplex 或 Multiplex 节点删除 myDAS 主高速缓存:

```
DROP DBSPACE myDAS
```

## 用法

(返回顶部) (第 190 页)

**DROP** 会移除指定数据库结构的定义。如果该结构是 **dbspace**，则必须先删除或重新分配该 **dbspace** 中包含任何数据的所有表，再删除该 **dbspace**；将自动重新分配其它结构。如果该结构是表，则在删除过程中将自动删除该表中的所有数据。另外，**DROP TABLE** 语句还会删除该表的所有索引和键。

只要 **DROP DBSPACE** 语句影响了当前正由其它连接使用的表，就会禁止该语句。其它连接正在使用相应过程时，禁止执行 **DROP PROCEDURE**。

副作用

- 自动提交。清除 **dbisql** 中的数据窗口。**DROP TABLE** 和 **DROP INDEX** 用于关闭当前连接的所有游标。
- 局部临时表是个特例，删除它时不执行提交。

## 标准

(返回顶部) (第 190 页)

- SQL - 符合 ISO/ANSI SQL 标准。
- SAP Sybase 数据库产品 - 受 Adaptive Server 支持。

## 权限

(返回顶部) (第 190 页)

**DBSPACE** 子句 - 需要 **DROP ANY OBJECT** 系统特权并且用户必须是数据库的唯一连接。

**DOMAIN** 子句 - 需要以下特权之一:

- **DROP DATATYPE** 系统特权。
- **DROP ANY OBJECT** 系统特权。
- 您拥有该对象。

**FUNCTION** 子句 - 需要以下特权之一:

- **DROP ANY PROCEDURE** 系统特权。
- **DROP ANY OBJECT** 系统特权。

- 您拥有该函数。

**INDEX** 子句 - 需要以下特权之一：

- DROP ANY INDEX 系统特权。
- DROP ANY OBJECT 系统特权。
- 对要建立索引的基础表的 REFERENCES 特权。
- 您拥有要建立索引的基础表。

DBA 或具有适当特权的用户可以在不使用完全限定名的情况下删除其他用户拥有的表的索引。所有其他用户必须提供完全限定的索引名才能删除 DBA 拥有的基表的索引。

**MATERIALIZED VIEW** 子句 - 需要以下特权之一：

- DROP ANY MATERIALIZED VIEW 系统特权。
- DROP ANY OBJECT 系统特权。
- 您拥有实例化视图。

**PROCEDURE** 子句 - 需要以下特权之一：

- DROP ANY PROCEDURE 系统特权。
- DROP ANY OBJECT 系统特权。
- 您拥有该过程。

**TABLES** 子句 - 需要以下特权之一：

- DROP ANY TABLE 系统特权。
- DROP ANY OBJECT 系统特权。
- 您拥有该表。

除非引用全局临时表的所有用户都已经断开连接，否则不能删除相应的全局临时表。

**VIEW** 子句 - 需要以下特权之一：

- DROP ANY VIEW 系统特权。
- DROP ANY OBJECT 系统特权。
- 您拥有该视图。

所有其它子句 - 需要以下特权之一：

- DROP ANY OBJECT 系统特权。
- 您拥有该对象。

## **INSERT 语句**

将当前数据库中其它位置的单行或选择的一组行插入表中。此命令还可将另一数据库中选择的一组行插入表中。

快速链接：

转至参数 (第 195 页)

转至示例 (第 196 页)

转至用法 (第 197 页)

转至标准 (第 200 页)

转至权限 (第 200 页)

## 语法

### Syntax 1

```
INSERT [ INTO ] [ owner.]table-name [ ( column-name [, ...] ) ]
    ... VALUES ( [ expression | DEFAULT, ... ] )
or
INSERT [ INTO ] [ owner.]table-name DEFAULT VALUES
```

### Syntax 2

```
INSERT [ INTO ] [ owner.]table-name [ ( column-name [, ...] ) ]
    ... insert-load-options insert-select-load-options
    ... select-statement
```

### Syntax 3

```
INSERT [ INTO ] [ owner.]table-name [ ( column-name [, ...] ) ]
    ... insert-select-load-options insert-select-load-options
    LOCATION 'servername.dbname'
    [ location-options ]
    ... { { select-statement } | 'select statement' }
```

**insert-load-options** - (back to Syntax 2) or (back to Syntax 3)

```
[ LIMIT number-of-rows ]
[ NOTIFY number-of-rows ]
[ SKIP number-of-rows ]
```

**insert-select-load-options** - (back to Syntax 2) or (back to Syntax 3)

```
[ WORD SKIP number ]
[ IGNORE CONSTRAINT constraint-type [, ...] ]
[ MESSAGE LOG 'string' ROW LOG 'string' [ ONLY LOG logwhat
[, ...] ] ]
[ LOG DELIMITED BY 'string' ]
```

**constraint-type** - (back to insert-select-load-options)

```
{ CHECK integer
| UNIQUE integer
| NULL integer
| FOREIGN KEY integer
| DATA VALUE integer
} ALL integer
}
```

**logwhat** - (back to insert-select-load-options)

```
{ CHECK
| ALL
```



```

| NULL
| UNIQUE
| DATA VALUE
| FOREIGN KEY
| WORD
|
}

```

## 参数

(返回顶部) (第 193 页)

- **insert-load-options** - 用于约束装载的选项：
  - **LIMIT** - 指定从查询插入表中的最大行数。缺省值为 0，表示无限制。最大数量为 2GB - 1。
  - **NOTIFY** - 指定每次在表中成功插入指定行数时将发送消息通知您。缺省值为每插入 100,000 行通知一次。
  - **SKIP** - 定义此插入要在输入表开头跳过的行数。缺省值为 0。
- **WORD SKIP** - 遇到长度超过创建字索引时所指定限制的数据时，允许装载继续进行。*number* 参数指定忽略错误的次数。将此选项设置为 0 表示没有限制。  
 如果因字超过最大允许大小而未能装载某一行，则会向 `.iqmsg` 文件写入一条警告。可以选择将 **WORD** 大小违规记录到 **MESSAGE LOG** 文件中。如果未指定此选项，则操作将在第一次遇到长度超过指定限制的字时回退。
- **IGNORE CONSTRAINT** - 确定装载引擎是否忽略装载期间发生的 **CHECK**、**UNIQUE**、**NULL**、**DATA VALUE** 和 **FOREIGN KEY** 完整性约束违规，以及在启动回退之前忽略的最大违规次数。  
 如果 *limit* 为零，则将要忽略的 **CHECK** 约束违规次数将无限制。如果未指定 **CHECK**，则首次发生任何 **CHECK** 约束违规时都将导致装载回退。如果 *limit* 为非零，则发生 *limit* + 1 次 **CHECK** 约束违规后将导致装载回退。
- **MESSAGE LOG** - 指定装载引擎记录完整性约束违规的文件名。表示装载开始和完成的时间戳记录在 **MESSAGE LOG** 和 **ROW LOG** 文件中。必须指定 **MESSAGE LOG** 和 **ROW LOG**，否则将不记录与完整性违规有关的信息。  
 对于 **ONLY LOG** 子句中指定的所有完整性约束类型违规，或在指定关键字 **WORD** 情况下的所有字索引长度违规，将记录相关违规信息。如果未指定 **ONLY LOG** 子句，则不会记录有关完整性约束违规的信息。而是只记录表示装载开始和完成的时间戳。
- **LOG DELIMITED BY** - 指定 **ROW LOG** 文件中各数据值之间的分隔符。缺省分隔符为逗号。
- **ENCRYPTED PASSWORD** - 指定当连接到远程服务器时使用 **Open Client Library** 缺省口令加密。如果指定此参数但远程服务器不支持 **Open Client Library** 缺省口令加密，则会报告错误，用以指示使用了无效的用户 **ID** 或口令。

要使 SAP Sybase IQ 服务器接受具有加密口令的 jConnect 连接，请将 jConnect ENCRYPT\_PASSWORD 连接属性设置为 true。

- **PACKETSIZE** - 指定 TDS 包大小（以字节为单位）。在大多数平台上，TDS 包大小的缺省值为 512 字节。如果未指定包大小或者将其指定为零，则使用相应平台的缺省包大小值。

packet-size 的值必须是 512 的倍数，或者等于缺省网络包大小，或者介于缺省网络包大小和最大网络包大小之间。最大网络包大小和缺省网络包大小为 512 的倍数，且在 512 - 524288 字节范围内。最大网络包大小始终大于或等于缺省网络包大小。

- **QUOTED\_IDENTIFIER** - 在远程服务器上设置 QUOTED\_IDENTIFIER 选项。缺省设置为 'OFF'。仅当 SELECT 语句中的任意标识符用双引号引起时（如下例中使用 "c1"），才能将 QUOTED\_IDENTIFIER 设置为 'ON'：

```
INSERT INTO foo
LOCATION 'ase.database'
QUOTED_IDENTIFIER ON {select "c1" from xxx};
```

- **ISOLATION LEVEL** - 指定远程服务器连接的隔离级别：

隔离级别	特性
READ UNCOMMITTED	<ul style="list-style-type: none"> <li>• 隔离级别 0</li> <li>• 允许读取有写锁定或无写锁定的行</li> <li>• 未应用读锁定</li> <li>• 无法确保并发事务将不会修改行或回退对行所做的更改</li> </ul>
READ COMMITTED	<ul style="list-style-type: none"> <li>• 隔离级别 1</li> <li>• 只允许读取没有写锁定的行</li> <li>• 仅为读取当前行获取并保持读锁定，但当游标离开该行时释放读锁定</li> <li>• 无法确保数据在事务执行过程中不发生更改</li> </ul>
SERIALIZABLE	<ul style="list-style-type: none"> <li>• 隔离级别 3</li> <li>• 只允许读取结果中没有写锁定的行</li> <li>• 打开游标时获取读锁定，并一直保持到事务结束为止</li> </ul>

**注意：** 有关 insert-select-load-options 和 location-options 以及 constraint-type 和 logwhat 参数的详细信息，请参见“LOAD TABLE 语句”。

**示例**

(返回顶部) (第 193 页)

- **示例 1** - 将 Eastern Sales 部门添加到数据库：

```
INSERT INTO Departments
  (DepartmentID, DepartmentName, DepartmentHeadID)
VALUES (600, 'Eastern Sales', 501)
```

- **示例 2** – 使用部门主管姓名和部门名称填充表 dept\_head:

```
INSERT INTO dept_head (name, dept)
  NOTIFY 20
  SELECT Surname || ' ' || GivenName
  AS name,
  dept_name
FROM Employees JOIN Departments
  ON EmployeeID= DepartmentHeadID
```

- **示例 3** – 从远程服务器 detroit 上的 SAP Sybase IQ 数据库 iqdet 中将 lineitem 表的 l\_shipdate 和 l\_orderkey 列中的数据插入到当前数据库中的 lineitem 表的相应列中:

```
INSERT INTO lineitem
  (l_shipdate, l_orderkey)
  LOCATION 'detroit.iqdet'
  PACKETSIZE 512
  ' SELECT l_shipdate, l_orderkey
FROM lineitem '
```

- **示例 4** – INSERT 语句允许使用值列表，从而允许一次插入多行。

```
INSERT into t1 values( 10, 20, 30 ), ( 11, 21, 31 ), ( 12, 22,
32 )
```

## 用法

(返回顶部) (第 193 页)

语法 1 允许插入包含指定表达式值的单行。如果未指定列名的列表，则值将以创建时所使用的顺序（与用 **SELECT \*** 检索的顺序相同）插入到表列中。行插入到表中的任意位置。（在关系数据库中，表是不排序的）。

语法 2 允许用户使用完全通用的 **SELECT** 语句的结果对表执行大量插入。除非 **SELECT** 语句包含 **ORDER BY** 子句，否则插入将以任意顺序进行。选择列表中的列将按顺序与列列表中指定的列匹配，或者按这些列的创建顺序匹配。

---

**注意：** **NUMBER(\*)** 函数可在通过 **INSERT** 语句的语法 2 生成主键时使用。

---

语法 3 **INSERT...LOCATION** 是语法 2 的变体，允许从 Adaptive Server 或 SAP Sybase IQ 数据库插入数据。在 **LOCATION** 子句中指定的 *servername.dbname* 用于标识 **FROM** 子句的表中的远程服务器和数据库。要使用语法 3，则所连接的 Adaptive Server 或 SAP Sybase IQ 远程服务器必须存在于本地计算机的 SAP Sybase Open Client interfaces 或 sql.ini 文件中。

在使用语法 3 的查询中，最多可以插入 2147483647 行。

**SELECT** 语句可以使用大括号或直单引号进行分隔。

---

**注意：**在 ODBC 标准中，大括号表示转义序列的开头和结尾，在 ODBC 或 SAP Control Center 的上下文中可能会导致产生错误。解决方案是使用单引号括起 **SELECT** 语句。

---

本地 SAP Sybase IQ 服务器连接至您在 **LOCATION** 子句中指定的服务器和数据库。将返回远程表的查询结果，本地服务器会将结果插入当前数据库中。如果没有在 **LOCATION** 子句中指定服务器名称，SAP Sybase IQ 将忽略您所指定的任何数据库名称，因为本地服务器中的当前数据库是唯一的选择。

当 SAP Sybase IQ 连接到远程服务器时，**INSERT...LOCATION** 将对当前连接的用户 ID 使用远程登录，但前提是已使用 **CREATE EXTERNLOGIN** 创建远程登录，并且已使用 **CREATE SERVER** 语句定义了远程服务器。如果未定义远程服务器，或者未为当前连接的用户 ID 创建远程登录，则 SAP Sybase IQ 将使用当前连接的用户 ID 和口令连接。

---

**注意：**如果依赖于当前连接的用户 ID 和口令且用户已更改口令，则必须停止并重新启动服务器，然后新口令才能在远程服务器上生效。对缺省用户 ID 的口令进行更改不会影响使用 **CREATE EXTERNLOGIN** 创建的远程登录。

---

使用 **CREATE EXTERNLOGIN** 语句创建远程登录并使用 **CREATE SERVER** 语句定义远程服务器，可以针对 **INSERT...LOCATION** 设置外部登录名和口令，从而使任何用户都可以在任意上下文中使用登录名和口令。这样可以避免因无法获得登录名或口令而可能产生的错误。这是连接到远程服务器的理想方式。

例如，用户 russid 连接到 SAP Sybase IQ 数据库并执行以下语句：

```
INSERT local_SQL_Types LOCATION 'ase1.ase1db'
{SELECT int_col FROM SQL_Types};
```

在 ase1 服务器上，存在口令为 sybase 的用户 ID ase1user。ase1user 是表 SQL\_Types 的所有者。在 IQ 服务器上，远程服务器的定义如下：

```
CREATE SERVER ase1 CLASS 'ASEJDBC'
USING 'system1:4100';
```

在 IQ 服务器上，外部登录的定义如下：

```
CREATE EXTERNLOGIN russid TO ase1 REMOTE LOGIN ase1user IDENTIFIED BY
sybase;
```

对于用户 russid，**INSERT...LOCATION** 使用用户 ID ase1user 和口令 sybase 连接到远程服务器 ase1。

使用 **ENCRYPTED PASSWORD** 参数指定当连接到远程服务器时使用 Open Client Library 缺省口令加密。如果指定 **ENCRYPTED PASSWORD** 但远程服务器不支持 Open Client Library 缺省口令加密，则会报告错误，用以指示使用了无效的用户 ID 或口令。

当用作远程服务器时，SAP Sybase IQ 支持 TDS 口令加密。SAP Sybase IQ 服务器接受由客户端发送的使用加密口令的连接。有关针对口令加密设置的连接属性的信息，请参见 Open Server 15.5 “软件开发工具包 15.5” > “Open Client Client-Library/C 参考

手册” > “Client-Library 主题” > “安全功能” > “Adaptive Server 安全功能” > “安全握手：加密口令”。

---

**注意：** 口令加密要求使用 Open Client 15.0。TDS 口令加密要求使用 Open Client 15.0 ESD #7 或更高版本。

---

当 **INSERT...LOCATION** 在 SAP Sybase IQ 服务器与远程 SAP Sybase IQ 或 Adaptive Server 服务器之间传输数据时，**INSERT...LOCATION TDS PACKETSIZE** 参数的值始终是 512 字节，即使为 **PACKETSIZE** 指定其它值也是如此。

---

**注意：** 如果指定了错误的包大小（例如 933，不是 512 的倍数），则连接尝试将失败，并引发 Open Client **ct\_connect** “连接失败”错误。任何不成功的连接尝试都将返回通用的“连接失败”消息。Adaptive Server 错误日志可能包含有关连接失败原因的更多具体信息。

---

SAP Sybase IQ 不支持 Adaptive Server 数据类型 **TEXT**，但是您可以从长度大于 255 字节的 **IQ CHAR** 或 **VARCHAR** 列和数据类型为 **TEXT** 的 **ASE** 数据库列执行

**INSERT...LOCATION**（语法 3）。如果 SAP Sybase IQ 支持内部转换，则可将 **ASE TEXT** 和 **IMAGE** 列插入到其它 SAP Sybase IQ 数据类型的列中。缺省情况下，如果远程数据列包含 2 GB 以上的数据，SAP Sybase IQ 则会以无提示方式将列值截断为 2 GB。

---

**警告！** SAP Sybase IQ 不支持 Adaptive Server 数据类型 **UNICHAR**、**UNIVARCHAR** 或 **UNITEXT**。在 **ISO\_BINENG** 归类中，从 **UNICHAR** 或 **UNITEXT** 到 **CHAR** 或 **CLOB** 列中的 **INSERT...LOCATION** 命令可以正确执行；如果发生这种情况，则这些列中的数据可能不一致。只有转换失败时，这种情况下才会报告错误。

---

用户必须获得专门许可才能使用非结构化数据分析选项的大对象功能。

---

**注意：** 如果使用 **INSERT...LOCATION** 插入从 **VARBINARY** 列中选择的数据，则在远程数据库上将 **ASE\_BINARY\_DISPLAY** 设置为 **OFF**。

---

**INSERT...LOCATION**（语法 3）不支持在 **SELECT** 语句中使用变量。

可执行在视图中的插入，条件是所提供的定义视图的 **SELECT** 语句在 **FROM** 子句中只有一个表，且不包含 **GROUP BY** 子句、集合函数或涉及 **UNION** 操作。

插入到表中的字符串始终以它们输入时的大小写形式存储，无论数据库是否区分大小写。因此，插入到表中的字符串 "Value" 在数据库中保存时 **V** 始终为大写，其余的字母为小写。**SELECT** 语句以 'Value' 形式返回该字符串。但是，如果数据库不区分大小写，则所有比较都会将 'value' 与 'value'、'VALUE' 等不同大小写形式的字符串视为相同。而且，如果单列主键已经包含 Value 条目，则会拒绝对 value 执行 **INSERT**，因为它会导致主键不唯一。

只要执行 **INSERT...LOCATION** 语句，SAP Sybase IQ 就会装载确定语言、归类序列、字符集和日期/时间格式所需的本地化信息。如果数据库使用平台的非缺省区域设置，则必须在本地客户端设置环境变量以确保 SAP Sybase IQ 装载正确的信息。

如果设置了 LC\_ALL 环境变量，则 SAP Sybase IQ 将该变量的值用作区域设置名称。如果未设置 LC\_ALL，SAP Sybase IQ 将使用 LANG 环境变量的值。如果这两个变量均未设置，SAP Sybase IQ 将使用区域设置文件中的缺省条目。

使用 (DEFAULT)、DEFAULT VALUES 或 VALUES () 子句向各行中插入所有缺省值。假设表 t2 中有 3 列，以下示例在语义上是等效的：

```
INSERT INTO t2 values (DEFAULT, DEFAULT, DEFAULT);
```

```
INSERT INTO t2 DEFAULT VALUES;
```

```
INSERT INTO t2() VALUES ();
```

**INSERT...VALUES** 同样支持多行。以下示例在表 t1 中插入 3 行：

```
CREATE TABLE t1(c1 varchar(30));
INSERT INTO t1 VALUES ('morning'),('afternoon'),
('evening');
```

SAP Sybase IQ 将所有装载/插入视为完全宽度插入。如果未在装载/插入语句中显式指定列，则装载的值会成为列的 DEFAULT 值（已定义一个值）或 NULL（没有为列定义 DEFAULT 值）。

SAP Sybase IQ 支持 **INSERT...VALUES**、**INSERT...SELECT** 和 **INSERT...LOCATION** 的列 DEFAULT 值。如果为列指定了 DEFAULT 值，则此 DEFAULT 值将用作任一未指定列值的 **INSERT**（或 **LOAD**）语句中的列值。

如果存储过程或函数使用 **COMMIT**、**ROLLBACK** 或某些 **ROLLBACK TO SAVEPOINT** 语句，则不允许从该存储过程或函数执行 **INSERT**。

**SELECT...FROM** 的结果可能略微不同于 **INSERT...SELECT...FROM** 的结果，原因是针对插入期间的优化而进行的不精确数据类型（如 DOUBLE 或 NUMERIC）的内部数据转换。如果需要更精确的结果，可行的解决方法是将列声明为具有较高精度的 DOUBLE 或 NUMERIC 数据类型。

### 标准

(返回顶部) (第 193 页)

- SQL - 符合 ISO/ANSI SQL 标准。
- SAP Sybase 数据库产品 - 受 Adaptive Server 支持（不包括 *insert-load-options*）。

### 权限

(返回顶部) (第 193 页)

需要针对表的 **INSERT** 特权。

### 另请参见

- **LOAD TABLE** 语句 (第 201 页)

## LOAD TABLE 语句

将数据从外部文件导入数据库表。

快速链接：

转至参数 (第 202 页)

转至示例 (第 211 页)

转至用法 (第 214 页)

转至标准 (第 217 页)

转至权限 (第 217 页)

### 语法

```
[ INTO ] TABLE [ owner. ] table-name
... ( load-specification [, ...] )
... { FROM | USING [ CLIENT ] FILE }
{ 'filename-string' | filename-variable } [, ...]
... [ CHECK CONSTRAINTS { ON | OFF } ]
... [ DEFAULTS { ON | OFF } ]
... [ QUOTES OFF ]
... ESCAPES OFF
... [ FORMAT { ascii | binary | bcp } ]
... [ DELIMITED BY 'string' ]
... [ STRIP { OFF | RTRIM } ]
... [ WITH CHECKPOINT { ON | OFF } ]
... [ BYTE ORDER { NATIVE | HIGH | LOW } ]
... [ LIMIT number-of-rows ]
... [ NOTIFY number-of-rows ]
... [ ON FILE ERROR { ROLLBACK | FINISH | CONTINUE } ]
... [ PREVIEW { ON | OFF } ]
... [ ROW DELIMITED BY 'delimiter-string' ]
... [ SKIP number-of-rows ]
... [ HEADER SKIP number [ HEADER DELIMITED BY 'string' ] ]
... [ WORD SKIP number ]
... [ ON PARTIAL INPUT ROW { ROLLBACK | CONTINUE } ]
... [ IGNORE CONSTRAINT constraint-type [, ...] ]
... [ MESSAGE LOG 'string' ROW LOG 'string' [ ONLY LOG log-what
[, ...] ]
... [ LOG DELIMITED BY 'string' ]
```

**load-specification** - (back to Syntax)

```
{ column-name [ column-spec
| FILLER ( filler-type ) }
```

**column-spec** - (back to load-specification)

```
{ ASCII ( input-width )
| BINARY [ WITH NULL BYTE ]
| PREFIX { 1 | 2 | 4 }
| 'delimiter-string'
| DATE ( input-date-format )
```

```

| DATETIME ( input-datetime-format )
| ENCRYPTED ( data-type 'key-string' [, 'algorithm-string' ] )
| DEFAULT default-value }
[ NULL ( { BLANKS | ZEROS | 'literal', ...} )

filler-type - (back to load-specification)
{ input-width
| PREFIX { 1 | 2 | 4 }
| 'delimiter-string'
}

constraint-type - (back to Syntax)
{ CHECK integer
| UNIQUE integer
| NULL integer
| FOREIGN KEY integer
| DATA VALUE integer
| ALL integer
}

log-what - (back to Syntax)
{ CHECK
| ALL
| NULL
| UNIQUE
| DATA VALUE
| FOREIGN KEY
| WORD
}

```

## 参数

(返回顶部) (第 201 页)

- **FROM** – 标识要从中装载数据的一个或多个文件。要指定多个文件，请使用逗号分隔每个 filename-string。filename-string 以字符串形式传递到服务器。因此，该字符串遵循的格式要求与其它 SQL 字符串相同。

要在 Windows 系统中指示目录路径，反斜杠字符 \ 必须用两个反斜杠来表示。因此，要将文件 c:\temp\input.dat 中的数据装载到 Employees 表的语句是：

```
LOAD TABLE Employees
FROM 'c:\\temp\\input.dat' ...
```

路径名相对于数据库服务器，而不是客户端应用程序。如果在其它某台计算机的数据库服务器上运行此语句，则目录名是指服务器计算机上的目录，而不是客户端计算机上的目录。装载 Multiplex 数据库时，请在所有文件名中使用绝对（完全限定）路径。请不要使用相对路径名。

由于受资源限制的影响，SAP Sybase IQ 无法保证能够装载所有数据。如果资源分配失败，整个装载事务将被回退。一次读取一个文件，并按 FROM 子句中指定的顺序处理这些文件。任何 SKIP 或 LIMIT 值都只在装载开始时应用，而不是针对每个文件应用。



现已不建议使用 **LOAD TABLE FROM** 子句，但可使用该子句指定服务器上存在的文件。此示例从客户端计算机上的文件 **a.inp** 装载数据。

```
LOAD TABLE t1(c1,c2,filler(30))
USING CLIENT FILE 'c:\\client-data\\a.inp'
QUOTES OFF ESCAPES OFF
IGNORE CONSTRAINT UNIQUE 0, NULL 0
MESSAGE LOG 'c:\\client-data\\m.log'
ROW LOG 'c:\\client-data\\r.log'
ONLY LOG UNIQUE
```

- **USING** - **USING FILE** 从服务器装载一个或多个文件。该子句的作用等同于指定 **FROM filename** 子句。**USING CLIENT FILE** 从客户端批量装载一个或多个文件。客户端上文件的字符集必须与服务器归类相同。**SAP Sybase IQ** 将依次处理文件列表中的文件。每个文件在处理时将锁定为读取模式，然后解锁。客户端批量装载不会产生任何管理开销，例如额外的磁盘空间以及内存或网络监控守护程序需求，但会强制对每个文件进行单线程处理。

批量装载大对象时，**USING CLIENT FILE** 子句同时适用于主文件和辅助文件。

**LOAD TABLE** 语句只能装载 **gzip** 格式的压缩客户端和服务器文件。扩展名为 **".gz"** 或 **".gzip"** 的所有文件都视为压缩文件。装载压缩文件时不支持命名管道或辅助文件。压缩文件和未压缩文件可以在同一 **LOAD TABLE** 语句中指定。装载中的每个压缩文件都由一个线程进行处理。

在客户端装载过程中，将在客户端主机上创建 **IGNORE CONSTRAINT** 日志文件，创建日志文件过程中出现的任何错误都会导致操作回退。

客户端批量装载受使用命令序列协议的 **Interactive SQL** 和 **ODBC/JDBC** 客户端支持。使用 **TDS** 协议的客户端则不支持。为了确保网络数据安全，请使用“传送层安全性”。要控制谁能够使用客户端批量装载，请使用安全功能 **(-sf)** 服务器启动开关、**ALLOW\_READ\_CLIENT\_FILE** 数据库选项和/或 **READCLIENTFILE** 访问控制。

- **CHECK CONSTRAINTS** - 对检查约束进行评估，即，可忽略或记录。**CHECK CONSTRAINTS** 缺省设置为 **ON**。

设置 **CHECK CONSTRAINTS OFF** 将导致 **SAP Sybase IQ** 忽略所有检查约束违规。例如，此设置在重建数据库时十分有用。如果表的检查约束调用尚未创建的用户定义函数，则重建将失败，除非此选项设置为 **OFF**。

此选项与以下选项互斥。如果在同一装载中指定了以下任何选项，将导致错误：

- **IGNORE CONSTRAINT ALL**
- **IGNORE CONSTRAINT CHECK**
- **LOG ALL**
- **LOG CHECK**
- **DEFAULTS** - 使用列缺省值。缺省情况下，此选项为 **ON**。如果 **DEFAULTS** 选项为 **OFF**，则为列列表中未显示的列分配 **NULL**。

DEFAULTS 选项的设置适用于所有列 DEFAULT 值，包括 AUTOINCREMENT。

- **QUOTES** – 表示输入字符串用引号字符括起来。**QUOTES** 是可选参数，且缺省设置为 ON。引号字符是撇号（单引号）或引号（双引号）。字符串中出现的第一个这样的字符将被视为该字符串的引号字符。字符串数据必须以匹配的引号结束。

在 **QUOTES ON** 的情况下，可以在列值中包括列或行分隔符。假定前导引号字符和结尾引号字符不是值的一部分并将其从装载的数据值中排除。

要在使用 **QUOTES ON** 的值中包括引号字符，必须使用两个引号。例如，此行在第三列中包含一个单引号字符值：

```
'123 High Street, Anytown', '(715)398-2354', ' ' , ' ' , ' ' , ' '
```

在打开 **STRIP** 时（缺省情况），首先从值中去除尾随空白，然后再插入这些值。仅去除不带引号的字符串的尾随空白。带引号的字符串保留其尾随空白。仅当设置为 ON 时，才会剪裁前导空白或 TAB 字符。

数据抽取工具提供多个引号处理选项（**TEMP\_EXTRACT\_QUOTES**、**TEMP\_EXTRACT\_QUOTES\_ALL** 和 **TEMP\_EXTRACT\_QUOTE**）。如果您计划通过缺省 ASCII 抽取方式抽取数据以将其装载到 IQ 主存储表中，且字符串字段包含列或行分隔符，请使用 **TEMP\_EXTRACT\_BINARY** 选项进行抽取并对 **LOAD TABLE** 使用 **FORMAT binary** 和 **QUOTES OFF** 选项。

限制：

- **QUOTES ON** 仅适用于列分隔的 ASCII 字段。
- 在 **QUOTES** 设置为 ON 的情况下，列分隔符或行终结符的第一个字符不能是单引号或双引号。
- **QUOTES ON** 将强制对给定文件进行单线程处理。
- 无论 **QUOTES** 选项的设置如何，都不适用于从辅助文件装载二进制大对象 (BLOB) 或字符大对象 (CLOB) 数据。前导或尾随引号将作为 CLOB 数据的一部分来装载。使用 **QUOTES ON** 选项，位于引号之间的两个连续引号将作为两个连续引号进行装载。
- Adaptive Server BCP 不支持 **QUOTES** 选项。拷入或拷出所有字段数据的方式与设置 **QUOTES OFF** 时的情况相同。由于 **QUOTES ON** 是 SAP Sybase IQ **LOAD TABLE** 语句的缺省设置，因此将 ASE 数据从 BCP 输出导入至 SAP Sybase IQ 表时，您必须指定 **QUOTES OFF**。

例外情况：

- 如果 **LOAD TABLE** 在引起来的字段的结尾引号字符之后遇到任何非空白字符，则将报告以下错误并回退装载操作：

```
Non-SPACE text found after ending quote character for
an enclosed field.
SQLSTATE: QTA14      SQLCODE: -1005014L
```

- 在 **QUOTES** 设置为 ON 的情况下，如果将单引号或双引号指定为列分隔符的第一个字符，则将报告错误并且装载操作将失败：

```
Single or double quote mark cannot be the 1st character
of column delimiter or row terminator with QUOTES option
ON.
SQLSTATE: QCA90      SQLCODE: -1013090L
```

- **ESCAPES** - 如果省略输入字段的 *column-spec* 定义，则在 **ESCAPES** 为 **ON** (缺省值) 的情况下，数据库服务器会识别跟在反斜杠字符后的字符并将其解释为特殊字符。可以以组合形式 `\n` 添加换行符，以十六进制 **ASCII** 代码形式添加其它字符，例如，使用 `\x09` 表示制表符。两个连续的反斜线字符 (`\\`) 被解释为单个反斜线。对于 **SAP Sybase IQ**，必须设置 **ESCAPES OFF**。
- **FORMAT** - **SAP Sybase IQ** 支持 **ASCII** 和二进制输入字段。此格式通常由上述 *column-spec* 定义。如果省略列的此定义，则在缺省情况下，**SAP Sybase IQ** 使用此选项定义的格式。假定输入行具有 **ascii** (缺省值) 或 **binary** 字段，每个输入行对应一行，并使用列分隔符来分隔各个值。

**SAP Sybase IQ** 也允许将 **BCP** 字符文件中的数据作为 **LOAD TABLE** 命令的输入。

- 使用 **LOAD TABLE FORMAT BCP** 语句装载到 **SAP Sybase IQ** 表中的 **BCP** 数据文件必须利用 `-c` 选项以跨平台文件格式进行导出 (**BCP OUT**)。
- 对于 **FORMAT BCP**，**LOAD TABLE** 语句的缺省列分隔符是 `<tab>`，而缺省行终结符是 `<newline>`。
- 对于 **FORMAT BCP**，行中的最后一列必须以行终结符终止，而不能以列分隔符终止。如果列分隔符位于行终结符之前，则将列分隔符视为数据的一部分。
- 装载规范中除最后一列以外的所有列中的数据只能以列分隔符进行分隔。对于除最后一列以外的所有列，如果在列分隔符之前先遇到行终结符，则将行终结符视为列数据的一部分。
- 可通过 **DELIMITED BY** 子句指定列分隔符。对于 **FORMAT BCP**，分隔符的长度必须小于或等于 10 个字符。如果分隔符长度大于 10，则返回错误。
- 对于 **FORMAT BCP**，装载规范只能包含列名、**NULL** 和 **ENCRYPTED**。如果在装载规范中指定了任何其它选项，则返回错误。

例如，下列即属于有效的 **LOAD TABLE** 装载规范：

```
LOAD TABLE x( c1, c2 null(blanks), c3 )
FROM 'bcp_file.bcp'
FORMAT BCP
...
```

```
LOAD TABLE x( c1 encrypted(bigint,'KEY-ONE','aes'), c2, c3 )
FROM 'bcp_file.bcp'
FORMAT BCP
...
```

- **DELIMITED BY** - 如果在 *column-spec* 定义中省略列分隔符，则缺省的列分隔符为逗号。通过提供单个 **ASCII** 字符或十六进制字符表示形式可指定其它列分隔符。**DELIMITED BY** 子句为：

```
... DELIMITED BY '\x09' ...
```

要将换行符用作分隔符，您可以指定特殊组合 '\n' 或其 ASCII 值 '\x0a'。在 `column-spec delimiter-string` 中最多可指定四个字符，而在 `DELIMITED BY` 子句中只能指定一个字符。

- **STRIP** – 确定在插入不带引号的值之前是否应去除其尾随空白。**LOAD TABLE** 命令接受以下 **STRIP** 关键字：
  - **STRIP OFF** – 不删除尾随空白。
  - **STRIP RTRIM** – 去除尾随空白。
  - **STRIP ON** – 不建议使用。使用 **STRIP RTRIM**。

在打开 **STRIP** 时（缺省情况），SAP Sybase IQ 首先将从值中去除尾随空白，然后再插入这些值。此设置仅适用于 **VARCHAR** 数据。**STRIP OFF** 会保留尾随空白。

仅去除不带引号的字符串的尾随空白。带引号的字符串保留其尾随空白。如果不需要区分空白，可以使用 **FILLER** 选项作为替代选项，以便更确切地指定要去除的字节数，而不是去除所有尾随空格。对于 SAP Sybase IQ，**STRIP OFF** 会更有效，在处理尾随空白时，它会遵循 ANSI 标准。（**CHAR** 数据始终会进行填补，因此 **STRIP** 选项仅影响 **VARCHAR** 数据。）

**STRIP** 选项仅适用于长度可变的非二进制数据，不适用于 **ASCII** 固定宽度插入。以下列模式为例：

```
CREATE TABLE t( c1 VARCHAR(3) );
LOAD TABLE t( c1 ',' ) ..... STRIP RTRIM    // trailing blanks
trimmed

LOAD TABLE t( c1 ',' ) ..... STRIP OFF     // trailing blanks
not trimmed

LOAD TABLE t( c1 ASCII(3) ) ... STRIP RTRIM // trailing blanks
not trimmed
LOAD TABLE t( c1 ASCII(3) ) ... STRIP OFF   // trailing blanks
trimmed

LOAD TABLE t( c1 BINARY ) ..... STRIP RTRIM // trailing blanks
trimmed
LOAD TABLE t( c1 BINARY ) ..... STRIP OFF   // trailing blanks
trimmed
```

始终剪裁二进制数据中的尾随空白。

- **WITH CHECKPOINT** – 确定 SAP Sybase IQ 是否执行检查点。此选项只适用于在 SAP Sybase IQ 数据库中装载 SQL Anywhere 表。

缺省设置为 **OFF**。如果将此子句设置为 **ON**，则会在成功完成并记录语句之后执行检查点操作。如果服务器在提交连接后及下次检查点之前出现故障，则必须具有用来装载表的数据文件才能成功完成恢复。不过，如果指定了 **WITH CHECKPOINT ON**，并且此后需要恢复，则在恢复时不需要数据文件。

如果数据库损坏，需要使用备份并应用当前的日志文件，则无论此子句的设置情况如何，都需要数据文件。

---

**警告！** 如果将数据库选项 `CONVERSION_ERROR` 设置为 `OFF`，则可能会将错误的数据库装载到表中而收不到任何错误报告。如果未指定 `WITH CHECKPOINT ON`，并且需要恢复数据库，则当恢复过程中 `CONVERSION_ERROR` 设置为 `ON`（缺省值）时，恢复可能失败。建议您不要在 `CONVERSION_ERROR` 设置为 `OFF` 且未指定 `WITH CHECKPOINT ON` 的情况下装载表。

另请参见 `CONVERSION_ERROR` 选项 [TSQL]。

---

- **BYTE ORDER** – 指定读取时的字节顺序。此选项适用于所有二进制输入字段。如果未定义，则此选项将被忽略。**SAP Sybase IQ** 始终以其所在计算机的本机格式读取二进制数据（缺省值为 `NATIVE`）。您还可以指定：
  - **HIGH**，当多字节数量以高位字节优先时（对于 Sun、IBM AIX 和 HP 等 `big endian` 平台而言）。
  - **LOW**，当多字节数量以低位字节优先时（对于 Windows 等 `little endian` 平台而言）。
- **LIMIT** – 指定要插入表中的最大行数。缺省值为 0，表示无限制。最大值为  $2^{31} - 1$  (2147483647) 行。
- **NOTIFY** – 指定每次在表中成功插入指定行数时将发送消息通知您。缺省值为 0，表示不打印任何通知。此选项的值将覆盖 `NOTIFY_MODULUS` 数据库选项的值。
- **ON FILE ERROR** – 指定 **SAP Sybase IQ** 在因输入文件不存在或没有读取该文件的正确权限而无法打开文件时应执行的操作。您可以指定以下其中一个操作：
  - **ROLLBACK** – 中止整个事务（缺省设置）。
  - **FINISH** – 结束已完成的插入操作并终止装载操作。
  - **CONTINUE** – 返回错误，但只是跳过该文件，然后继续执行装载操作。
 只允许有一个 `ON FILE ERROR` 子句。
- **PREVIEW** – 显示目标表中输入的布局，包括每列的起始位置、名称和数据类型。**SAP Sybase IQ** 将在装载过程开始时显示此信息。如果写入日志文件，此信息也会包含在日志中。
- **ROW DELIMITED BY delimiter-string** – 指定最大长度 4 字节的字符串，以指示输入记录的结尾。仅当行中的所有字段为以下任意一种情形时才能使用此选项：
  - 由列终结符分隔
  - 通过 `DATE` 或 `DATETIME` *column-spec* 选项定义数据
  - `ASCII` 固定长度字段

始终包含 `ROW DELIMITED BY` 以确保进行并行装载。从 `LOAD` 规范中删除此子句可能会导致 **SAP Sybase IQ** 以串行方式而非并行方式进行装载。

如果任何输入字段包含二进制数据，则无法使用此选项。使用此选项，行终结符将使任何缺失字段设置为 `NULL`。所有行都必须具有相同的行分隔符，并且必须

区别于所有列分隔符。行分隔符字符串和字段分隔符字符串彼此不能为初始子集。例如，不能指定 "\*" 作为字段分隔符并指定 "\*#" 作为行分隔符，但可以与该行分隔符一起指定 "#" 作为字段分隔符。

如果行缺少分隔符，则 SAP Sybase IQ 会返回错误并回退整个装载事务。唯一一种例外情况是文件的最后一条记录，系统会回退该行并返回警告消息。在 Windows 上，通常由换行符后跟回车符表示行分隔符。对于此选项或 FILLER，您可能需要将此指定为 *delimiter-string*（见以上说明）。

- **SKIP** – 定义此装载要在输入表开头跳过的行数。要跳过的最大行数为  $2^{31} - 1$  (2147483647)。缺省值为 0。SKIP 在读取要跳过的行时以单线程模式运行。
- **HEADER SKIP...HEADER DELIMITED BY** – 指定 **LOAD TABLE** 要在数据文件开头跳过的行数，其中包括标题行。在跳过指定的行数之前，将忽略所有的 **LOAD TABLE** 列规范及其它装载选项。
  - 要跳过的行数大于或等于零。
  - 行由 **HEADER DELIMITED BY** 子句中指定的长度为 1 到 4 个字符的分隔符字符串确定。缺省的 **HEADER DELIMITED BY** 字符串为 '\n' 字符。
  - **HEADER DELIMITED BY** 字符串的最大长度为四个字符。如果字符串长度大于 4 或小于 1，将返回错误。
  - 如果指定了非零 **HEADER SKIP** 值，则会忽略包含 **HEADER DELIMITED BY** 分隔符的所有数据，直到分隔符的出现次数达到 **HEADER SKIP** 子句中指定的次数。
  - 在跳过指定的行数之前，将忽略所有的 **LOAD TABLE** 列规范及其它装载选项。在跳过指定的行数后，**LOAD TABLE** 列规范及其它装载选项将应用于剩余数据。
  - 仅忽略数据开头处的“标题”字节。如果在 **USING** 子句中指定了多个文件，**HEADER SKIP** 将只忽略从第一个文件的第一行开始的数据，直到跳过指定的标题行数，即使这些行位于后面的文件中。**LOAD TABLE** 在开始解析实际数据后便不会寻找标题。
  - 如果 **LOAD TABLE** 在跳过 **HEADER SKIP** 指定的行数之前处理所有输入数据，则不会报告错误。
- **WORD SKIP** – 遇到长度超过创建字索引时所指定限制的数据时，允许装载继续进行。

如果因字超过最大允许大小而未能装载某一行，则会向 .iqmsg 文件写入一条警告。还可以选择将 **WORD** 大小违规记录到 **MESSAGE LOG** 文件中，并将被拒绝的行记录到 **LOAD TABLE** 语句中指定的 **ROW LOG** 文件。

- 如果未指定此选项，则 **LOAD TABLE** 将在第一次遇到长度超过指定限制的字符时报告错误并回退。
- *number* 指定忽略“字不能超过允许的最大字长度”错误的次数。
- 0 (零) 表示没有限制。

- **ON PARTIAL INPUT ROW** - 指定在装载期间遇到部分输入行时执行的操作。您可以指定以下其中一个操作：

- **CONTINUE** 发出警告并继续执行装载操作。这是缺省设置。
- **ROLLBACK** 中止整个装载操作并报告错误。

```
Partial input record skipped at EOF.
SQLSTATE: QDC32      SQLSTATE: -1000232L
```

- **IGNORE CONSTRAINT** - 指定是否忽略装载期间发生的 **CHECK**、**UNIQUE**、**NULL**、**DATA VALUE** 和 **FOREIGN KEY** 完整性约束违规，以及在启动回退之前忽略的最大违规次数。指定每个 *constrainttype* 都具有以下结果：

- **CHECK limit** - 如果 *limit* 指定为零，则将要忽略的 **CHECK** 约束违规次数将无限制。如果未指定 **CHECK**，则首次发生任何 **CHECK** 约束违规时都将导致 **LOAD** 语句回退。如果 *limit* 为非零，则发生 *limit*+1 次 **CHECK** 约束违规后将导致装载回退。
- **UNIQUE limit** - 如果 *limit* 指定为零，则要忽略的 **UNIQUE** 约束违规次数将无限制。如果 *limit* 为非零，则发生 *limit*+1 次 **UNIQUE** 约束违规后将导致装载回退。
- **NULL limit** - 如果 *limit* 指定为零，则要忽略的 **NULL** 约束违规次数将无限制。如果 *limit* 为非零，则发生 *limit*+1 次 **NULL** 约束违规后将导致装载回退。
- **FOREIGN KEY limit** - 如果 *limit* 指定为零，则要忽略的 **FOREIGN KEY** 约束违规次数将无限制。如果 *limit* 为非零，则发生 *limit*+1 次 **FOREIGN KEY** 约束违规后将导致装载回退。
- **DATA VALUE limit** - 如果数据库选项 **CONVERSION\_ERROR** 为 **ON**，则将报告错误并回退语句。如果 *limit* 指定为零，则要忽略的 **DATA VALUE** 约束违规（数据类型转换错误）次数将无限制。如果 *limit* 为非零，则发生 *limit*+1 次 **DATA VALUE** 约束违规后将导致装载回退。
- **ALL limit** - 如果数据库选项为 **CONVERSION\_ERROR = ON**，则将报告错误并回退语句。如果 *limit* 指定为零，则要忽略的所有完整性约束违规的累计总数将无限制。如果 *limit* 为非零，则在忽略的所有 **UNIQUE**、**NULL**、**DATA VALUE** 和 **FOREIGN KEY** 完整性约束违规累计总数超过 *limit* 值时，装载将回退。例如，可指定此 **IGNORE CONSTRAINT** 选项：

```
IGNORE CONSTRAINT NULL 50, UNIQUE 100, ALL 200
```

完整性约束违规的总数不能超过 200，而 **NULL** 和 **UNIQUE** 约束违规的总数分别不能超过 50 和 100。只要超出这些限制中的任何一个，**LOAD TABLE** 语句便会回退。

**注意：** 单个行可具有多个完整性约束违规。每出现一次完整性约束违规便计一次，直至达到该类违规的限定值。

如果要记录忽略的完整性约束违规，可将 **IGNORE CONSTRAINT** 选项限制设置为非零值。记录过多的违规数会影响装载性能

如果在 IGNORE CONSTRAINT 子句中未指定 CHECK、UNIQUE、NULL 或 FOREIGN KEY，则在第一次发生上述其中一种完整性约束违规时，装载将回退。

如果未在 IGNORE CONSTRAINT 子句中指定 DATA VALUE，则当首次发生这种类型的完整性约束违规时，将回退装载，但如果数据库选项为 CONVERSION\_ERROR = OFF 则例外。如果为 CONVERSION\_ERROR = OFF，则对于任何 DATA VALUE 约束违规，都将发出警告并继续装载。

装载完成时，将在 .iqmsg 文件中记录一条与完整性约束违规有关的信息性消息。此消息包含装载时发生的完整性约束违规次数及跳过的行数。

- **MESSAGE LOG** - 指定要在其中记录完整性约束违规和要记录的违规类型信息的文件的名称。表示装载开始和完成的时间戳记录在 MESSAGE LOG 和 ROW LOG 文件中。必须指定 MESSAGE LOG 和 ROW LOG，否则将不记录与完整性违规有关的信息。
  - 如果未指定 ONLY LOG 子句，则不会记录有关完整性约束违规的信息。而是只记录表示装载开始和完成的时间戳。
  - 对于 ONLY LOG 子句中指定的所有完整性约束类型违规，或在指定关键字 WORD 情况下的所有字索引长度违规，将记录相关违规信息。
  - 如果记录约束违规，则每次发生完整性约束违规时，将在 MESSAGE LOG 文件中生成一行信息。  
由于装载是由并行运行的多个线程执行的，MESSAGE LOG 文件中的行数（报告的错误）可能会超出 IGNORE CONSTRAINT 选项限制。可能会有多个线程报告约束违规次数超出指定限值。
  - 如果记录约束违规，则对于某一给定行，将在 ROW LOG 文件中记录一行信息，而不考虑该行发生的完整性约束违规次数。  
MESSAGE LOG 文件中的不同错误数可能与 ROW LOG 文件中的行数不完全匹配。行数差是由于 MESSAGE LOG 的上述并行装载处理造成的。
  - MESSAGE LOG 和 ROW LOG 文件不能为原始分区或命名管道。
  - 如果已存在 MESSAGE LOG 或 ROW LOG 文件，则会将新信息附加到相应文件。
  - 为 MESSAGE LOG 或 ROW LOG 文件指定无效文件名将产生错误。
  - 为 MESSAGE LOG 和 ROW LOG 文件指定相同的文件名将产生错误。

IGNORE CONSTRAINT 和 MESSAGE LOG 选项的各种组合会导致不同的记录操作。

表 2. LOAD TABLE 记录操作

是否已指定 IGNORE CONSTRAINT?	是否已指定 MESSAGE LOG?	操作
是	是	回退之前，将记录所有被忽略的完整性约束违规（包括用户指定的限制）。



是否已指定 IGNORE CONSTRAINT?	是否已指定 MESSAGE LOG?	操作
否	是	回退之前，将记录第一次完整性约束违规。
是	否	不记录任何信息。
否	否	不记录任何信息。第一次发生完整性约束违规即会导致回退。

**提示：** 如果要记录忽略的完整性约束违规，可将 IGNORE CONSTRAINT 选项限制设置为非零值。如果单行出现多次完整性约束违规，则将针对每次违规在 MESSAGE LOG 文件中写入一行。记录过多的违规数会影响装载性能。

- **LOG DELIMITED BY** – 指定 ROW LOG 文件中各数据值之间的分隔符。缺省分隔符为逗号。

将 FORMAT BCP 指定为 **LOAD TABLE** 子句时，SAP Sybase IQ 不再返回错误消息。此外，会验证以下情况，并返回相应的错误消息

- 如果指定的装载格式不是 ASCII、BINARY 或 BCP，SAP Sybase IQ 会返回消息“LOAD 格式仅支持 ASCII、BCP 和 BINARY。”
- 如果 **LOAD TABLE** 列规范包含除列名、NULL 或 ENCRYPTED 以外的任何内容，SAP Sybase IQ 会返回错误消息“LOAD ... FORMAT BCP 的装载说明无效。”
- 如果 FORMAT BCP 装载的列分隔符或行终结符的大小超过 10 个字符，SAP Sybase IQ 会返回消息“分隔符 '%2' 的长度必须为 1 到 %3 个字符。”（其中 %3 等于 10）。  
对于 FORMAT BCP 和 FORMAT ASCII 出现的错误或警告情况，与这些情况相对应的消息对于两种格式而言都是相同的。
- 如果指定的装载缺省值为 AUTOINCREMENT、IDENTITY 或 GLOBAL AUTOINCREMENT，SAP Sybase IQ 会返回错误“缺省值 %2 不能用作 LOAD 缺省值。%1”
- 如果 **LOAD TABLE** 规范不包含需要从指定文件装载的任何列，SAP Sybase IQ 会返回错误“LOAD 语句必须至少包含要从输入文件中装载的 1 列。”并回退 **LOAD TABLE** 语句。
- 如果装载超出了具有 TEXT 索引的文本文档的最大词语数限制，SAP Sybase IQ 会返回错误“文本文档超出最大词语数。每个文档最多支持 4294967295 个词语。”

## 示例

(返回顶部) (第 201 页)

- **示例 1** – 将数据从一个文件装载到 Windows 系统上的 Products 表中。使用制表符作为列分隔符，后跟 Description 和 Color 列：

```
LOAD TABLE Products
( ID ASCII(6),
  FILLER(1),
  Name ASCII(15),
  FILLER(1),
  Description '\x09',
  Size ASCII(2),
  FILLER(1),
  Color '\x09',
  Quantity PREFIX 2,
  UnitPrice PREFIX 2,
  FILLER(2) )
FROM 'C:\\mydata\\source1.dmp'
QUOTES OFF
ESCAPES OFF
BYTE ORDER LOW
NOTIFY 1000
```

- **示例 2** – 从客户端计算机上的文件 a.inp 装载数据：

```
LOAD TABLE t1(c1,c2,filler(30))
USING CLIENT FILE 'c:\\client-data\\a.inp'
QUOTES OFF ESCAPES OFF
IGNORE CONSTRAINT UNIQUE 0, NULL 0
MESSAGE LOG 'c:\\client-data\\m.log'
ROW LOG 'c:\\client-data\\r.log'ONLY LOG UNIQUE
```

- **示例 3** – 将数据从两个文件装载到 UNIX 系统上的 product\_new 表中（该表允许 NULL 值）。制表符为缺省的列分隔符，并且使用换行符作为行分隔符：

```
LOAD TABLE product_new
( id,
  name,
  description,
  size,
  color '\x09' NULL( 'null', 'none', 'na' ),
  quantity PREFIX 2,
  unit_price PREFIX 2 )
FROM '/s1/mydata/source2.dump',
'/s1/mydata/source3.dump'
QUOTES OFF
ESCAPES OFF
FORMAT ascii
DELIMITED BY '\x09'
ON FILE ERROR CONTINUE
ROW DELIMITED BY '\n'
```

- **示例 4** – 忽略 10 字长度违规；出现第 11 个字时，配置新错误并回退装载：

```
load table PTAB1(
  ck1      ',,' null ('NULL') ,
  ck3fk2c2 ',,' null ('NULL') ,
  ck4      ',,' null ('NULL') ,
  ck5      ',,' null ('NULL') ,
```

```

ck6c1      ',' null ('NULL') ,
ck6c2      ',' null ('NULL') ,
rid        ',' null ('NULL') )
FROM 'ri_index_selfRI.inp'
row delimited by '\n'
LIMIT 14 SKIP 10
IGNORE CONSTRAINT UNIQUE 2, FOREIGN KEY 8
word skip 10 quotes off escapes off strip
off

```

- **示例 5** - 使用 **FORMAT BCP** 装载选项将数据从 **BCP** 字符文件 `bcp_file.bcp` 装载到表 `t1`:

```

LOAD TABLE t1 (c1, c2, c3)
FROM 'bcp_file.bcp'
FORMAT BCP
...

```

- **示例 6** - 使用 **DEFAULT** 装载选项将缺省值 `12345` 装载到 `c1`，从 `LoadConst04.dat` 文件向 `c2` 和 `c3` 装载数据:

```

LOAD TABLE t1 (c1 DEFAULT '12345 ', c2, c3, filler(1))
FROM 'LoadConst04.dat'
STRIP OFF
QUOTES OFF
ESCAPES OFF
DELIMITED BY ',' ;

```

- **示例 7** - 使用 **FORMAT BCP** 装载选项从文件 `bcp_file.bcp` 向 `c1` 和 `c2` 装载数据，并将 `c3` 的值设置为 `10`:

```

LOAD TABLE t1 (c1, c2, c3 DEFAULT '10' )
FROM 'bcp_file.bcp'
FORMAT BCP
QUOTES OFF
ESCAPES OFF;

```

- **示例 8** - 以下代码段会忽略数据文件开头处用 `'&&'` 分隔的一个标题行:

```

LOAD TABLE
...HEADER SKIP 1 HEADER DELIMITED by '&&'

```

- **示例 9** - 以下代码段会忽略数据文件开头处用 `'\n'` 分隔的 2 个标题行:

```

LOAD TABLE
...HEADER SKIP 2

```

- **示例 10** - 将文件装载到启用 **RLV** 的表中。

使用 **FORMAT BCP** 装载选项将数据从 **BCP** 字符文件 `bcp_file.bcp` 装载到启用 **RLV** 的表 `rvt1` 中:

```

LOAD TABLE rvt1 (c1, c2, c3)
FROM 'bcp_file.bcp'
FORMAT BCP
...

```

## 用法

(返回顶部) (第 201 页)

**LOAD TABLE** 允许执行从带有 ASCII 或二进制数据的文件到数据库表的高效的大量插入。

使用 **LOAD TABLE** 选项也可以控制违反完整性约束时的装载行为并且记录有关违规的信息。

对临时表可以使用 **LOAD TABLE**，但是该临时表必须是用 **ON COMMIT PRESERVE ROWS** 声明的，否则下一条 **COMMIT** 将删除已装载的行。

**LOAD TABLE** 支持装载大对象 (LOB) 数据。

SAP Sybase IQ 支持从 ASCII 和二进制数据装载，并且支持固定长度格式和可变长度格式。要处理所有这些格式，您必须提供一个 *load-specification* 来通知 SAP Sybase IQ 需要从源文件中的每个“列”或字段获得何种数据。可通过 *column-spec* 定义下列格式：

- 具有固定字节长度的 ASCII。*input-width* 值是表示每条记录中输入字段的固定宽度的整数（单位为字节）。
- 使用一定数量的 **PREFIX** 字节（1、2 或 4）指定输入长度的二进制或非二进制字段。

以下两部分与 **PREFIX** 子句相关：

- 前缀值 - 始终为二进制值。
- 关联数据字节 - 始终为字符格式，永远不采用二进制格式。

如果使用抽取工具卸载数据时将 **TEMP\_EXTRACT\_BINARY** 选项设置为 **ON**，则在装载二进制数据时必须针对每个列使用 **BINARY WITH NULL BYTE** 参数。

- 由分隔符分隔的可变长度字符。可以将终结符指定为十六进制 ASCII 字符。*delimiter-string* 可以是由最多 4 个字符构成的任意字符串，包括可打印字符的任意组合以及表示非打印字符的任意 8 位十六进制 ASCII 代码。例如，指定：
  - '\x09'，以将制表符表示为终结符。
  - '\x00'，表示空终结符（如同 "C" 字符串中一样，没有可见终结符）。
  - '\x0a'，表示作为终结符的换行符。还可以使用特殊字符组合 "\n" 作为换行符。

---

**注意：** 分隔符字符串长度可为 1 到 4 个字符，但在 **DELIMITED BY** 子句中只能指定单个字符。对于 **BCP**，分隔符最多可达 10 个字符。

---

- ASCII 字符形式的 **DATE** 或 **DATETIME** 字符串。必须使用 SAP Sybase IQ 支持的日期数据类型和日期时间数据类型的对应格式之一定义字符串的 *input-date-format* 或 *input-datetime-format*。使用 **DATE** 表示日期值，并使用 **DATETIME** 表示日期时间和时间值。

表 3. 格式化日期和时间

选项	含义
yyyy 或 YYYY yy 或 YY	表示年份数字。缺省值为当前年份。
mm 或 MM	表示月份数字。适当情况下始终在月份数字中使用前导零或空白，例如，"05" 表示 5 月。DATE 值必须包含月份。例如，如果您输入的 DATE 值为 1998，则会收到错误提示。如果输入 "03"，则 SAP Sybase IQ 将应用缺省年份和日期，并将其转换为 '1998-03-01'。
dd 或 DD jjj 或 JJJ	表示日期数字。缺省日期为 01。适当情况下始终在日期数字中使用前导零，例如，'01' 表示第一天。J 或 j 表示一年中的儒略日（1 至 366）。
hh HH	表示小时。小时基于 24 小时制。适当情况下始终在小时中使用前导零或空白，例如，'01' 表示凌晨 1 点。'00' 也是有效值，表示中午 12 点。
nn	表示分钟。适当情况下始终在分钟中使用前导零，例如，'08' 表示 8 分钟。
ss[.sssss]	表示秒和秒的小数部分。
aa	表示 a.m. 或 p.m. 标记。
pp	仅当需要时表示 p.m. 标记。（这与 12.0 之前的 SAP Sybase IQ 版本不兼容；先前 "pp" 与 "aa" 是同义词。）
hh	SAP Sybase IQ 假设分钟和秒钟均为零。例如，如果您输入的 DATETIME 值为 '03'，则 SAP Sybase IQ 将该值转换为 '03:00:00.0000'。
hh:nn 或 hh:mm	SAP Sybase IQ 假设秒钟为零。例如，如果您输入的时间值为 '03:25'，则 SAP Sybase IQ 将该值转换为 '03:25:00.0000'。

表 4. 示例 DATE 和 DATETIME 格式选项

输入数据	格式规范
12/31/98	DATE ('MM/DD/YY')
19981231	DATE ('YYYYMMDD')
123198140150	DATETIME ('MMDDYyhnnss')
14:01:50 12-31-98	DATETIME ('hh:nn:ss MM-DD-YY')
18:27:53	DATETIME ('hh:nn:ss')
12/31/98 02:01:50AM	DATETIME ('MM/DD/YY hh:nn:ssaa')

SAP Sybase IQ 针对常用日期、时间和日期时间格式提供了内置装载优化。如果您要装载的数据与其中一种格式匹配，则使用相应的格式可以显著缩短装载时间。

还可以将日期/时间字段指定为 ASCII 固定宽度字段（如上所述），并使用 FILLER(1) 选项跳过列分隔符。

装载到表列中时，*column-spec* 的 **NULL** 部分指示如何将某些输入值视为 **NULL** 值。这些字符可能包括 **BLANKS**、**ZEROS** 或您定义的任何其它文字列表。指定 **NULL** 值或从源文件读取 **NULL** 值时，目标列必须能够包含 **NULL**。

**ZEROS** 含义如下：如果（并且仅当）输入数据（转换前，如果是 ASCII）均为二进制零（而不为字符零），则此单元设置为 **NULL**。

- 如果输入数据为字符零，则：
  1. **NULL (ZEROS)** 永远不会导致单元变为 **NULL**。
  2. **NULL ('0')** 导致单元变为 **NULL**。
- 如果输入数据为二进制零（清除所有位），则：
  1. **NULL (ZEROS)** 导致单元变为 **NULL**。
  2. **NULL ('0')** 永远不会导致单元变为 **NULL**。

例如，如果 **LOAD** 语句包含 `col1 date('yymmdd') null(zeros)` 并且日期为 000000，则会收到错误消息，提示您 000000 无法转换为 **DATE(4)**。若要使 **LOAD TABLE** 在数据为 000000 时在 `col1` 中插入空值，请将 **NULL** 子句编写为 `null('000000')`，或将数据修改为相同数目的二进制零并使用 **NULL(ZEROS)**。

如果 **VARCHAR** 单元的长度为零并且该单元不为空值，则将获得一个零长度单元。对于所有其它数据类型来说，如果单元长度为零，**SAP Sybase IQ** 将插入 **NULL**。这是 **ANSI** 行为。对于零长度字符数据的非 **ANSI** 处理，请设置 **NON\_ANSI\_NULL\_VARCHAR** 数据库选项。

使用 **DEFAULT** 选项指定装载缺省列值。即使列未在表模式中定义缺省值，您仍可以向该列装载缺省值。在装载时，此功能可提供更大的灵活性。

- **LOAD TABLE DEFAULTS** 选项必须为 **ON** 才能使用 **LOAD TABLE** 语句中指定的缺省值。如果 **DEFAULTS** 选项为 **OFF**，则不会使用指定的装载缺省值，而会在列中插入 **NULL** 值。
- **LOAD TABLE** 命令必须包含至少一个需要从 **LOAD TABLE** 命令所指定的文件中装载的列。否则将报告错误，并且不会执行装载。
- 指定的装载缺省值必须符合受支持的列缺省值和缺省值限制。**LOAD TABLE DEFAULT** 选项不支持 **AUTOINCREMENT**、**IDENTITY** 或 **GLOBAL AUTOINCREMENT** 作为装载缺省值。
- **LOAD TABLE DEFAULT default-value** 必须属于数据库的字符集。
- 对于在 **LOAD TABLE DEFAULT** 子句中指定的装载缺省值，不支持对缺省值进行加密。
- 对于在表中插入的每行，将计算因评估指定装载缺省值而引起的约束冲突的总数。

*load-specification* 的另一重要部分是 **FILLER** 选项。此选项指示您希望跳过源输入文件中的指定字段。例如，在输入文件中的行末尾或整个字段可能会存在不希望添加到表中的字符。与 *column-spec* 定义一样，使用 **FILLER** 可以指定 **ASCII** 固定字节长度、由分隔符分隔的可变长度字符，以及使用 **PREFIX** 字节的二进制字段。

## 标准

(返回顶部) (第 201 页)

- SQL - ISO/ANSI SQL 语法的服务商扩充。
- SAP Sybase 数据库产品 - 不适用。

## 权限

(返回顶部) (第 201 页)

执行 **LOAD TABLE** 语句所需的权限取决于数据库服务器的 **-gl** 命令行选项，如下所示：

- **-gl ALL** - 您必须是表的所有者，对表具有 **ALTER** 或 **LOAD** 权限，或者具有 **ALTER ANY TABLE**、**LOAD ANY TABLE** 或 **ALTER ANY OBJECT** 系统特权。
- **-gl DBA** - 您必须具有 **ALTER ANY TABLE**、**LOAD ANY TABLE** 或 **ALTER ANY OBJECT** 系统特权。
- **-gl NONE** - 不允许执行 **LOAD TABLE** 语句。

有关 **-gl** 命令行选项的详细信息，请参阅《实用程序指南》> “start\_iq 数据库服务器启动实用程序”> “start\_iq 服务器选项”。

**LOAD TABLE** 还需要对表进行写锁定。

## LOCK TABLE 语句

禁止其它并发事务在指定时间内访问或修改表。

快速链接：

转至参数 (第 217 页)

转至示例 (第 218 页)

转至用法 (第 218 页)

转至标准 (第 219 页)

转至权限 (第 219 页)

## 语法

```
LOCK TABLE table-list [ WITH HOLD ]
      IN { SHARE | WRITE | EXCLUSIVE } MODE [ WAIT time ]
```

**table-list**

```
[ owner. ] table-name [ , [ owner. ] table-name, ... ]
```

## 参数

(返回顶部) (第 217 页)

- **table-name** - 必须是基表，而不是视图。**WRITE** 模式仅对 **IQ** 基表有效。**LOCK TABLE** 会锁定表列表中的所有表，或者不锁定任何表。必须为行级版本控制 (RLV) 启用该表。当为 **SQL Anywhere** 表获取锁，或者当获取 **SHARE** 或 **EXCLUSIVE** 锁时，只能指定一个表。标准 **SAP Sybase IQ** 对象限定规则用于分析 *table-name*。
- **WITH HOLD** - 该锁会一直保持到连接结束时为止。如果未指定此子句，则提交或回退当前事务时将释放锁定。不支持在同一语句中使用 **WITH HOLD** 和 **WRITE MODE** 子句，这将返回错误 `SQLCODE=-131, ODBC 3 State="42000"`。
- **SHARE** - 防止其它事务修改表，但允许它们读取访问。在此模式中，只要没有其它的事务锁定要修改的行，就可以间接地或显式地使用 **LOCK TABLE** 更改表中的数据。
- **WRITE** - 防止其它事务修改表的列表。无条件提交连接最外层的事务。事务的快照版本不是由 **LOCK TABLE IN WRITE MODE** 语句建立，而是通过执行由 **SAP Sybase IQ** 处理的下一命令建立。  
当事务提交或回退或者断开连接时，会解除 **WRITE** 模式锁。
- **EXCLUSIVE** - 防止其它事务访问表。在此模式中，任何其它事务都不能对表执行任何种类的查询、更新或任何其它操作。
- **WAIT time** - 指定所有锁类型的最大阻塞时间。当锁定模式为 **WRITE** 时，此子句为必填。给定时间参数后，仅当服务器在指定时间内可用时，其才会锁定指定表。可以使用 `hh:nn:ss:sss` 格式指定时间参数。如果指定了日期部分，服务器将忽略此部分，并将参数转换为时间戳。如果未给定时间参数，服务器将无限期等待，直到 **WRITE** 锁可用或发生中断为止。

## 示例

(返回顶部) (第 217 页)

- **示例 1** - 在 **Customers** 和 **Employees** 表中获取 **WRITE** 锁 (如果在 5 分 3 秒内可用) :

```
LOCK TABLE Customers, Employees IN WRITE MODE WAIT  
'00:05:03'
```

- **示例 2** - 无限期等待，直到在 **Customers** 和 **Employees** 表上的 **WRITE** 锁可用或发生中断为止:

```
LOCK TABLE Customers, Employees IN WRITE MODE WAIT
```

## 用法

(返回顶部) (第 217 页)

在协调器的 **IQ** 主存储库中的表上运行 **LOCK TABLE** 语句，不会影响从辅助服务器上的连接访问这些表。例如：

在协调器连接上，发出命令：



```
LOCK TABLE coord1 WITH HOLD IN EXCLUSIVE MODE
```

协调器上的 **sp\_iqlocks** 会确认表 `coord1` 具有排它 (E) 锁。

在辅助服务器的连接上运行的 **sp\_iqlocks** 的结果不会显示表 `coord1` 上的排它锁。此连接上的用户可查看协调器上表 `coord1` 的更新。

协调器上的其它连接可查看 `coord1` 上的排它锁，并尝试通过返回用户 `DBA` 具有在 `coord1` 中锁定的行的协调器上的其它连接从表 `coord1` 中选择。

不支持在视图上执行 **LOCK TABLE**。尝试锁定视图将获取共享模式锁，而不管命令中指定的模式如何。共享模式锁会阻止其它事务修改表模式。

**Transact-SQL (T-SQL)** 存储过程方言不支持 **LOCK TABLE**。例如，此语句返回 `LOCK` 附近的语法错误：

```
CREATE PROCEDURE tproc ()
AS
BEGIN
COMMIT;
LOCK TABLE t1 IN SHARE MODE
INSERT INTO t1 VALUES (30)
END
```

**Watcom-SQL** 存储过程语言支持 **LOCK TABLE**。缺省命令分隔符为分号 (;)。例如：

```
CREATE PROCEDURE tproc ()
AS
BEGIN
COMMIT;
LOCK TABLE t1 IN SHARE MODE
INSERT INTO t1 VALUES (30)
END
```

## 标准

(返回顶部) (第 217 页)

- SQL - ISO/ANSI SQL 语法的服务商扩充。
- SAP Sybase 数据库产品 - 受 Adaptive Server 支持。Adaptive Server 不支持 **WITH HOLD** 子句。Adaptive Server 提供不受 SQL Anywhere 支持的 **WAIT** 子句。

## 权限

(返回顶部) (第 217 页)

若要以 **SHARE** 模式锁定表，需要 **SELECT** 权限。

若要以 **EXCLUSIVE** 模式锁定表，则必须是表所有者或者具有以下系统特权之一：

- ALTER ANY OBJECT
- INSERT ANY TABLE

- UPDATE ANY TABLE
- DELETE ANY TABLE
- ALTER ANY TABLE
- LOAD ANY TABLE
- TRUNCATE ANY TABLE

## TRUNCATE 语句

删除表或实例化视图中的所有行，但不删除表定义。

快速链接：

[转至参数](#) (第 220 页)

[转至示例](#) (第 220 页)

[转至用法](#) (第 221 页)

[转至标准](#) (第 221 页)

[转至权限](#) (第 221 页)

### 语法

语法 1

```
TRUNCATE
  TABLE [ owner.] table-name
  | MATERIALIZED VIEW owner.] materialized-view-name
```

语法 2

```
TRUNCATE TABLE [ owner .] table
  [ PARTITION partition-name
  | SUBPARTITION subpartition-name ]
```

### 参数

[\(返回顶部\)](#) (第 220 页)

- **PARTITION** - 指定要截断的分区，不影响其它分区中的数据。

---

**注意：** 在 PARTITION 子句中指定启用 RLV 的表会导致错误。

---

- **SUBPARTITION** - 截断按复合分区模式进行分区的表。

---

**注意：** 在 SUBPARTITION 子句中指定启用 RLV 的表会导致错误。

---

### 示例

[\(返回顶部\)](#) (第 220 页)

- **示例 1** - 删除 Sale 表的所有行：

```
TRUNCATE TABLE Sale
```

## 用法

(返回顶部) (第 220 页)

**TRUNCATE** 等效于不带 **WHERE** 子句的 **DELETE** 语句，但不在事务日志中记入各个行的删除情况。执行 **TRUNCATE TABLE** 语句后，表结构及所有索引仍然存在，直至发出 **DROP TABLE** 语句。列定义和约束保持不变，且权限仍然有效。

与数据定义语句一样，**TRUNCATE** 语句作为单个语句记入事务日志。每个删除的行不记入事务日志。

## 标准

(返回顶部) (第 220 页)

- SQL - ISO/ANSI SQL 语法的服务商扩充。
- SAP Sybase 数据库产品 - 受 Adaptive Server 支持。

## 权限

(返回顶部) (第 220 页)

需要以下特权之一：

- **TRUNCATE ANY TABLE** 系统特权。
- **ALTER ANY TABLE** 系统特权。
- **ALTER ANY OBJECT** 系统特权。
- 对表的 **TRUNCATE** 特权。
- 您拥有该对象。

对于临时表和基表，可以执行 **TRUNCATE TABLE**，而其他用户则对这样的表具有读取访问权限。此行为有别于 SQL Anywhere，在其中需要具有独占访问权限才能截断基表。SAP Sybase IQ 表版本控制确保在进行 **TRUNCATE TABLE** 时其他用户具有读取访问权限；但是，这些用户看到的表的版本取决于读取和写入事务的提交时间。

## UPDATE 语句

修改单个表的现有行，或修改仅包含一个表的视图的现有行。

快速链接：

转至参数 (第 222 页)

转至示例 (第 223 页)

转至用法 (第 223 页)

转至标准 (第 225 页)

转至权限 (第 225 页)

## 语法

### UPDATE table-name

```
... SET [column-name = expression, ...
[ FROM table-expression ]
... [ WHERE search-condition ]
... [ ORDER BY expression [ ASC | DESC ] , ...]
```

**table-name** - (back to Syntax)

```
[ owner.]table-name [ [ AS ] correlation-name ]
| [ owner.]view-name [ [ AS ] correlation-name ]
```

**table-expression** - (back to Syntax)

```
table-spec
| table-expression join-type table-spec [ ON condition ]
| table-expression, ...
```

## 参数

(返回顶部) (第 221 页)

- **FROM 子句** - 允许根据连接更新表。如果存在 FROM 子句，则 **table-name** 必须指定要更新的唯一表，并且必须按照名称在 FROM 子句中的显示形式对其加以限定。如果在 FROM 子句中使用相关名，则必须将相同的相关名指定为 **table-name**。

以下语句阐释 UPDATE 语句 (使用 FROM 子句且包含使用相关名的表表达式) 的表中潜在的不明确性：

```
UPDATE table_1
SET column_1 = ...
FROM table_1 AS alias_1, table_2 AS alias_2
WHERE ...
```

FROM 子句中 table\_1 的每个实例都有一个相关名，用于表明 table\_1 本身自连。但是，UPDATE 语句无法指定要更新哪些构成自连接的行。可按照如下方法在 UPDATE 语句中指定相关名来更正此问题：

```
UPDATE table_1
SET column_1 = ...
FROM table_1 AS alias_1, table_1 AS alias_2
WHERE ...
```

如果正在从中更新行的同一表名称用于 FROM 子句，则可将子句视为引用该表，条件是下列其中一项为真：

- 两个表引用均未通过指定用户 ID 进行限定
- 两个表引用均已通过指定用户 ID 进行限定
- 两个表引用均通过相关名指定

如果服务器无法确定表引用是否相同，则会显示 **SQL** 错误。这可避免用户更新非预期行后产生意外语义。

- **WHERE** 子句 - 如果已指定，则只更新满足搜索条件的行。如果未指定 **WHERE** 子句，则更新所有行。

## 示例

(返回顶部) (第 221 页)

- **示例 1** - 将雇员 Philip Chin (雇员 129) 从销售部转移到市场部:

```
UPDATE Employees
SET DepartmentID = 400
WHERE EmployeeID = 129;
```

- **示例 2** - 市场部 (400) 将奖金从每位雇员基本薪水的 4% 提高到 6%:

```
UPDATE Employees
SET bonus = base * 6/100
WHERE DepartmentID =400;
```

- **示例 3** - 每位雇员由于获得部门奖金而增加了收入:

```
UPDATE Employees
SET emp.Salary = emp.Salary + dept.bonus
FROM Employees emp, Departments dept
WHERE emp.DepartmentID = dept.DepartmentID;
```

- **示例 4** - 另一种由于部门奖金而增加每位雇员收入的方法:

```
UPDATE Employees
SET emp.salary = emp.salary + dept.bonus
FROM Employees emp JOIN Departments dept
ON emp.DepartmentID = dept.DepartmentID;
```

## 用法

(返回顶部) (第 221 页)

**UPDATE** 语句中引用的表可以是基表，也可以是临时表。

缺省更新仅对当前用户、用户和当前时间戳以及时戳起作用。

每个指定的列被设置为等号右侧表达式的值。表达式中甚至可以使用 *column-name* - 此时将使用旧值。

**FROM** 子句可以包含多个表及连接条件，并返回由连接条件和/或 **WHERE** 条件指定和过滤的所有表的所有列。

在 **FROM** 子句中使用错误的连接条件会导致不可预料的结果。如果 **FROM** 子句指定一对多连接且 **SET** 子句引用连接“多”侧的某单元，则从选择的第一个值更新此单元。换句话说，如果连接条件导致按照行 **ID** 更新表的多个行，则返回的第一个行将成为更新结果。例如：

```
UPDATE T1
SET T1.c2 = T2.c2
FROM T1 JOIN TO T2
ON T1.c1 = T2.c1
```

如果表 T2 依照 T2.c1 具有多个行，则结果可能如下：

T2.c1	T2.c2	T2.c3
1	4	3
1	8	1
1	6	4
1	5	2

如果不带 **ORDER BY** 子句，则 T1.c2 可能为 4、6、8 或 9。

- 如果带有 **ORDER BY T2.c3**，T1.c2 则更新至 8。
- 如果带有 **ORDER BY T2.c3 DESC**，T1.c2 则更新至 6。

SAP Sybase IQ 拒绝要更新的表在外部连接的空值提供方的任何 **UPDATE** 语句。也就是说：

- 在左外部连接中，连接左侧的表不能缺少有关连接列的任何行。
- 在右外部连接中，连接右侧的表不能缺少有关连接列的任何行。
- 在完全外部连接中，两侧的表均不能缺少有关连接列的任何行。

例如，在下面的语句中，表 T1 在左外部连接的左侧，因此不能缺少任何行：

```
UPDATE T1
SET T1.c2 = T2.c4
FROM T1 LEFT OUTER JOIN T2
ON T1.rowid = T2.rowid
```

通常情况下，行以什么顺序更新并不重要。但是，与 **NUMBER(\*)** 函数一起使用时，排序可用于按指定的顺序在行中添加递增的编号。如果不使用 **NUMBER(\*)** 函数，请避免使用 **ORDER BY** 子句，因为 **UPDATE** 语句在不使用此子句的情况下性能会更好。

在 **UPDATE** 语句中，如果在 **SET** 子句中使用 **NUMBER(\*)** 函数且 **FROM** 子句指定一对多连接，则 **NUMBER(\*)** 将生成增加但由于删除行而不按顺序递增的唯一编号。

当 **FROM** 子句包含多个连接的表时，可以使用 **ORDER BY** 子句控制 **UPDATE** 的结果。

SAP Sybase IQ 忽略搜索的 **UPDATE** 中的 **ORDER BY** 子句，并返回指示相应语法不是有效 ANSI 语法的消息。

每个 **SET** 子句的左侧均必须是基表中的列。

如果定义视图的 **SELECT** 语句不包含 **GROUP BY** 子句、集合函数或不涉及 **UNION** 子句，则可以更新视图。视图应仅包含一个表。

插入到表中的字符串始终以它们输入时的大小写形式存储，无论数据库是否区分大小写。因此，用字符串 'Value' 更新的字符数据类型列在数据库中保存的形式始终是 V

为大写，其它字母均为小写。**SELECT** 语句以 'Value' 形式返回该字符串。但是，如果数据库不区分大小写，则所有比较都会将 'Value' 与 'value'、'VALUE' 等不同大小写形式的字符串视为相同。IQ 服务器能够以大小写字母的任意组合返回结果，因此，不区分大小写的数据库中，得不到区分大小写的结果 (**CASE IGNORE**)。而且，如果单列主键已经包含 'Value' 条目，则会拒绝对 'value' 执行 **INSERT**，因为它会导致主键不唯一。

如果更新违背任何检查约束，则将回退整个语句。

SAP Sybase IQ 支持在 **SET** 子句中使用标量子查询，例如：

```
UPDATE r
SET r.o= (SELECT MAX(t.o)
FROM t ... WHERE t.y = r.y),
r.s= (SELECT SUM(x.s)
FROM x ...
WHERE x.x = r.x)
WHERE r.a = 10
```

SAP Sybase IQ 支持 **UPDATE** 语句中的 **DEFAULT** 列值。如果某列具有 **DEFAULT** 值，则在未显式修改此列的值的任何 **UPDATE** 语句中，该 **DEFAULT** 值将用作此列的值。

有关更新作为其它类型 **DEFAULT** 列的 **IDENTITY/AUTOINCREMENT** 列的详细信息，请参见“**CREATE TABLE** 语句”。

## 标准

(返回顶部) (第 221 页)

- SQL - ISO/ANSI SQL 语法的服务商扩充。
- SAP Sybase 数据库产品 - 除下列例外情形之外，IQ **UPDATE** 语句的语法通常与 Adaptive Server **UPDATE** 语句的语法 1 兼容：SAP Sybase IQ 支持在 **FROM** 子句中使用多个表及连接条件。  
远程表的更新受限于 CIS 支持的 SAP Sybase IQ 语法。

## 权限

(返回顶部) (第 221 页)

需要所修改的列的 **UPDATE** 特权。

## 另请参见

- **CREATE TABLE** 语句 (第 173 页)

## 视图

使用系统视图可查看系统表的内容。

## SYSIQDBSPACE 系统视图

以可读格式显示 ISYSIQDBSPACE 中的组信息。

列名	列类型	描述
dbspace_id	small int	数据库中的每个 dbspace 都有一个指定的唯一编号 (dbspace ID)
last_modified	timestamp	上次修改 dbspace 的读写状态的时间
segment_type	char(8)	段类型: Main、Temp 或 Msg
read_write	char(1)	'T' - 可读写; 'F' - 只读
online	char(1)	'T' - 联机; 'F' - 脱机
create_txn_id	unsigned bigint	创建 dbspace 的事务 ID
alter_txn_id	unsigned bigint	上次修改 read_write 状态的事务 ID
striping_on	char(1)	'T' - 磁盘条带化开; 'F' - 磁盘条带化关
stripe_size_kb	unsigned int	在磁盘条带化算法移动到下一个 dbfile 之前要写入 dbspace 的每个文件的字节数 (KB)
is_rlv_store	char(1)	'T' - dbspace 是一个 RLV 存储 dbspace; 'F' - dbspace 是一个 MIN、SHARED_TEMP 或 TEMPORARY 存储 dbspace。

基础系统表的约束:

主键 (dbspace\_id)

外键 (dbspace\_id) 引用 SYS.ISYSDBSPACE(dbspace\_id)

## SYSIQRLVMERGEHISTORY 系统视图

每次开始合并 RLV 存储库和 IQ 主存储库时，都会为每个启用行级版本控制 (RLV) 的表添加一个日志条目。日志条目在合并完成时更新。

列名	列类型	描述
merge_id	unsigned bigint	唯一的日志条目标识符
table_id	unsigned int	Sys.systable 系统表的外键
start_time	timestamp	合并开始时间
end_time	timestamp	合并结束时间
status	char(9)	STARTED   COMPLETED   FAILED
return_code	tinyint	合并完成后的 SQL 代码



列名	列类型	描述
merge_type	char(9)	合并触发原因：AUTOMATIC   DML   DDL   SHUTDOWN   USER
merge_mode	char(12)	BLOCKING   NON-BLOCKING
merge_detail	varchar(255)	附加信息（如果提供），例如错误信息
rows_inserted	unsigned bigint	由于合并而插入的行数
rows_updated	unsigned bigint	由于合并而更新的行数
rows_deleted	unsigned bigint	由于合并而删除的行数
rows_forwarded	unsigned bigint	合并时未提交的行数

## **SYSIQRVLOG** 系统视图

以可读格式显示 ISYSIQRVLOG 中的组信息。SYSIQRVLOG 视图中的每一行都对应于一个启用 RLV 的表的日志。table\_id 为 0 的行表示服务器范围的提交日志。

列名	列类型	描述
stream_id	unsigned int	日志流标识符。
table_id	unsigned int	指示日志流所属的表。NULL 指示提交日志流。
partition_low	unsigned int	与上次激活该日志时使用的分区映射相对应。
partition_high	int	与上次激活该日志时使用的分区映射相对应。
identity_location	unsigned bigint	日志流标识块的位置。

## **SYSIQTAB** 系统视图

以可读格式显示 ISYSIQTAB 中的组信息。SYSIQTAB 视图中的每一行均描述了一个 IQ 表。

```
ALTER VIEW "SYS"."SYSIQTAB"
as select * from SYS.ISYSIQTAB
```

**注意：**此视图将取代不建议使用的 SYSIQTABLE 系统视图。

列名	列类型	描述
table_id	unsigned int	每个表都有一个属于主键的唯一编号（表号）。
block_map	hs_blockmapidentity	供内部使用。
block_map_size	unsigned int	供内部使用。

列名	列类型	描述
vdo	hs_vdoidentity	供内部使用。
vdo_id_size	unsigned int	供内部使用。
info_location	hs_vdorecid	尚未使用。始终为零。
info_recid_size	unsigned int	尚未使用。始终为零。
info_location_size	unsigned int	尚未使用。始终为零。
commit_txn_id	unsigned bigint	供内部使用。
txn_id	unsigned bigint	供内部使用。
update_time	timestamp	最后一次修改 IQ 表的日期和时间。
is_rlv	char(1)	'T' - RLV 存储在表上启用；'F' - RLV 存储不在表上启用。

基础系统表的约束：

主键 (table\_id)

# 索引

## A

AGGREGATION\_PREFERENCE 选项 69  
 ALLOW\_SNAPSHOT\_VERSIONING  
   表级 30  
   所有 30  
   行级 30  
 ALLOW\_SNAPSHOT\_VERSIONING 选项 70  
 ALTER DBSPACE 语句  
   语法 150  
 ALTER TABLE 语句  
   语法 154  
 ALTER VIEW 语句  
   RECOMPILE 154  
 AUTOINCREMENT 列缺省值 173

## B

BASE\_TABLES\_IN\_RLV 选项 71  
 BLOCKING 选项 51, 52, 71  
 BLOCKING\_TIMEOUT 选项 52, 72  
 版本控制 29  
 备份 6  
 本地意图写 12  
 表  
   GLOBAL TEMPORARY 173  
   创建 173  
   更改 154  
   更改定义 154  
   截断 220  
   临时 173  
   删除 190  
   锁定 217  
   装载 201  
 表大小 3  
 表锁  
   查看 41  
 表约束 173  
 别名  
   在 DELETE 语句中 188  
 并发  
   锁定表 217

## C

Catalog 存储  
 监控 118

CHECK ON COMMIT 子句  
   参照完整性 173  
 CHECK 条件  
   关于 173  
 CREATE DBSPACE 语句  
   语法 169  
 CREATE EXTERNLOGIN 语句  
   INSERT...LOCATION 198  
 CREATE SERVER 语句  
   INSERT...LOCATION 198  
 CREATE TABLE 语句  
   语法 173  
 查询 15, 16  
   行为 15  
 持久性 6  
 从表中删除所有行 220

## D

dbo 用户 ID  
   dbo 用户 ID 拥有的视图 190  
 dbspace  
   删除 190  
   设置脱机 150  
 DDL 14  
 DELETE 语句  
   语法 188  
 DML 8–10  
   提示 8  
 DROP DATATYPE 语句  
   语法 190  
 DROP DBSPACE 语句  
   语法 190  
 DROP DOMAIN 语句  
   语法 190  
 DROP EVENT  
   语法 190  
 DROP FUNCTION 语句  
   语法 190  
 DROP INDEX 语句  
   语法 190  
 DROP MESSAGE  
   语法 190  
 DROP PROCEDURE 语句  
   语法 190

## 索引

DROP TABLE  
  IDENTITY\_INSERT 选项 190  
DROP TABLE 语句  
  语法 190  
DROP VIEW 语句  
  限制 190  
  语法 190  
DROP 语句  
  语法 190

## E

ENABLE\_ASYNC\_IO 选项 72

## F

分区 5  
  删除 154

## G

归类  
  客户端文件批量装载 201  
过程  
  删除 190

## H

HEADER SKIP 选项  
  LOAD TABLE 语句 201  
函数  
  删除 190  
合并 4, 33  
  IQMSG 37  
  对表的影响 38  
  后台 34  
  记录的阶段 37  
  阶段 5  
  历史记录 36  
  前台 33  
  手动 35  
  疑难解答 64, 65  
  阈值 34  
  自动 33, 34  
缓冲区高速缓存  
  使用 sp\_iqsysmon 监控 118  
恢复 6

## I

IDENTITY 列  
  以及 DROP TABLE 190

IDENTITY\_INSERT 选项  
  删除表 190  
INSERT  
  语法 193  
INSERT 语句  
  WORD SKIP 选项 193  
iqrivmem 启动开关 150

## J

加密  
  TDS 口令 193  
监控  
  sp\_iqsysmon 过程 118  
剪裁尾随空白 201  
将 dbspace 设置为联机 150  
局部临时表 3  
聚合首选项 3

## K

客户端文件批量装载  
  错误 201  
  回退 201  
  字符集 201  
空白  
  剪裁尾随 201  
口令  
  TDS 加密 193  
  加密 193  
口令加密  
  TDS 193  
快照版本控制 30  
  行级 29

## L

LOAD TABLE 语句  
  HEADER SKIP 选项 201  
  ON PARTIAL INPUT ROW 选项 201  
  QUOTES 选项 201  
  STRIP 关键字 201  
  USING 关键字 201  
  新语法 201  
  性能 201  
  语法 201  
  语法更改 201  
LOB 3

**LOCK TABLE**

语法 217

## 连接

删除 188

显示有关信息 146

## 联机

dbspace 150

## 列

更改 154

约束 173

## 临时 dbspace

创建 169

## 临时表 173

创建 173

**M**

MPXServerName 列 95

## Multiplex

系统过程 94

## Multiplex 数据库

添加 dbspace 169

命名管道 201

模式锁 11

**N**

内存 57

大小 57

监控器 58

使用 sp\_iqsysmon 监控 118

内存行级 (RLV)

关于 1

耐久性 6

**P**

批量装载 201

**Q**

启用 RLV 的表

外键 64

文本索引 64

字索引 64

全局临时表 3

**R**

REFERENCES 子句 154

RESTRICT 操作 173

REVERT\_TO\_V15\_OPTIMIZER 选项 3

RID 稳定性 3

## RLV

配置 19

配置, 内存 21

配置, 前提条件 19

## RLV dbspace

变更 22

创建 21

配置 19, 21

删除 25

文件, 删除 26

文件, 添加 25

限制 7

只读 24

## RLV Dbspace

表 63

在 Multiplex 中创建 62

只读 63

重复 62

## RLV 表

疑难解答 63

## RLV 存储库 2, 4-6

Multiplex 62

查询 15, 16

持久性日志 7

禁用, 所有基表 28

禁用, 现有表 27

内存不足 61

配置 19, 26

启用, 所有基表 28

启用, 现有表 27

启用, 新表 27

手动合并 35

与 IQ 主存储库合并 4, 5, 33

自动合并使用的阈值 34, 36

RV\_AUTO\_MERGE\_EVAL\_INTERVAL 选项 73

rv\_max\_active\_subfragment\_count 选项 74

RV\_MERGE\_NODE\_MEMSIZE 选项 75

RV\_MERGE\_TABLE\_MEMPERCENT 选项 75

RV\_MERGE\_TABLE\_NUMROWS 选项 76

RV\_RESERVED\_DBSPACE\_MB 选项 76

**S**

sa\_conn\_info 41

sa\_report\_deadlocks 系统过程 49

SELECT \* 154

- SNAPSHOT\_VERSIONING 29
  - SNAPSHOT\_VERSIONING 选项 77
  - sp\_iqcolumn 系统过程 92
  - sp\_iqconnection 41
  - sp\_iqconnection 系统过程 94
  - sp\_iqdbsize 系统过程 97
  - sp\_iqdbspace 系统过程 99
  - sp\_iqfile 系统过程 102
  - sp\_iqlocks 41, 44
  - sp\_iqlocks 系统过程 104
  - sp\_iqmergerlvstore 系统过程 107
  - sp\_iqrlvmemory 系统过程 108
  - sp\_iqspaceinfo 系统过程 109
    - 示例输出 109
  - sp\_iqspaceused 系统过程 110
  - sp\_iqstatistics 系统过程 112
  - sp\_iqstatus 系统过程 115
    - 示例输出 115
  - sp\_iqsysmon 系统过程 118
  - sp\_iqtable 系统过程 138
  - sp\_iqtablesize 系统过程 142
  - sp\_iqtransaction 54
  - sp\_iqtransaction 系统过程 143
  - sp\_iqwho 系统过程 146
  - STRIP
    - LOAD TABLE 关键字 201
  - STRIP 选项 201
  - SYSIQDBSPACE 系统视图 226
  - SYSIQRLVMERGEHISTORY 系统视图 226
  - SYSIQRVLOG 系统视图 227
  - SYSIQTAB 系统视图 227
  - 删除
    - 视图 190
  - 删除分区 154
  - 删除实例化视图中的所有行 220
  - 实例化视图
    - 截断 220
    - 删除 190
  - 事件
    - 删除 190
  - 事务 10
    - 阻塞 51, 53
  - 事务管理
    - 使用 sp\_iqsysmon 监控 118
  - 事务日志
    - TRUNCATE 语句 220
  - 视图
    - 删除 190
    - 视图中的变更表 154
  - 术语 1
  - 数据库
    - 将数据装入 201
  - 数据库空间
    - 创建 169
    - 更改 150
  - 数据库文件
    - 创建 169
    - 更改 150
  - 数据类型
    - 删除用户定义数据类型 190
  - 死锁 51, 53
    - 报告 49
    - 查看 47
  - 碎片 5
  - 索引
    - 删除 190
  - 锁
    - 显示 104
  - 锁定
    - 表 217
- ## T
- TDS
    - 口令加密 193
  - TRUNCATE 语句
    - 语法 220
  - 体系结构 2
  - 脱机
    - dbspace 150
- ## U
- USING
    - LOAD TABLE 关键字 201
  - USING FILE 子句
    - LOAD TABLE 语句 201
- ## W
- WORD SKIP 选项 193
    - INSERT 语句 193
  - WORD 索引 3
  - 外键
    - 完整性约束 173
    - 未命名 173
  - 外键约束 3

- 唯一
  - 约束 173
- 唯一约束 3
- 尾随空白
  - 剪裁 201
- 文件
  - 设置联机 150
  - 设置脱机 150
  - 数据库空间 150, 169

## X

- 系统过程
  - sp\_iqcolumn 92
  - sp\_iqconnection 94
  - sp\_iqdbsize 97
  - sp\_iqfile 102
  - sp\_iqspaceinfo 109
  - sp\_iqspaceused 110
  - sp\_iqstatistics 112
  - sp\_iqstatus 115
  - sp\_iqsysmon 118
  - sp\_iqtable 138
  - sp\_iqtablesize 142
  - sp\_iqtransaction 143
  - sp\_iqwho 146
- 系统视图
  - SYSIQDBSPACE 226
  - SYSIQITAB 227
  - SYSIQLVMERGEHISTORY 226
  - SYSIQRVLOG 227
- 限制
  - REVERT\_TO\_V15\_OPTIMIZER 选项 3
  - RID 稳定性 3
  - WORD 索引 3
  - 表大小 3
  - 局部临时表 3
  - 聚合首选项 3
  - 全局临时表 3
  - 外键约束 3
  - 唯一约束 3
- 相关名
  - 在 DELETE 语句中 188
- 消息
  - 删除 190
- 行级版本控制 71
- 行级快照版本控制 14, 29
  - 配置 28
- 行级锁
  - 查看 44

- 行锁 11, 12
- 性能
  - sp\_iqsysmon 过程 118
  - 监控 118
- 选项
  - AGGREGATION\_PREFERENCE 69
  - ALLOW\_SNAPSHOT\_VERSIONING 70
  - ENABLE\_ASYNC\_IO 72
  - RV\_AUTO\_MERGE\_EVAL\_INTERVAL 73
  - rv\_max\_active\_subfragment\_count 74
  - RV\_MERGE\_NODE\_MEMSIZE 75
  - RV\_MERGE\_TABLE\_NUMROWS 76
  - RV\_RESERVED\_DBSPACE\_MBS 76
  - SNAPSHOT\_VERSIONING 75, 77

## Y

- 疑难解答 61
  - Multiplex 62
  - RLV Dbspace 62, 63
  - 挂起事务 66
  - 合并 64, 65
  - 恢复 66
  - 内存不足 61
  - 启用 RLV 的表 63
  - 外键 64
  - 写入事务 66
  - 字索引 64
- 意图写锁 11, 12
  - 查看 41
- 用户
  - 显示有关信息 146
- 用户定义数据类型
  - 删除 190
- 预取
  - 使用 sp\_iqsysmon 监控 118
- 远程服务器
  - 连接 198
- 远程数据访问 221
- 约束 9, 10

## Z

- 只读
  - 锁定表 217
- 主高速缓存
  - 删除 190
- 子碎片 5

## 索引

### 字符集

客户端文件批量装载 201

### 阻塞 51, 53

查看 54

### 阻塞超时

查看 54