

SYBASE®

性能和调优指南

Sybase IQ

15.2

文档 ID: DC00953-01-1520-01

最后修订日期: 2010 年 4 月

版权所有 © 2010 Sybase, Inc. 保留所有权利。

除非新版本或技术声明中另有说明, 否则本出版物适用于 Sybase 软件及所有后续版本。本文档中的信息如有更改, 恕不另行通知。本出版物中描述的软件按许可协议提供, 其使用或复制必须符合许可条款。

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

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

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

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

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

本书中提到的所有其它公司和产品名均可能是与之相关的相应公司的商标。

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

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

目录

关于本手册	ix	
第 1 章	从数据库表中选择数据	1
	前提条件	1
	查看表信息	2
	查询结果排序	4
	选择列和行	5
	使用搜索条件	6
	比较查询中的日期	7
	WHERE 子句中的复合搜索条件	7
	搜索条件中的模式匹配	8
	通过发音匹配行	9
	对搜索条件使用快捷方式	9
	获取集合数据	10
	使用集合函数获取分组数据	11
	限制组	11
	改善小计算	12
	获取分析数据	15
	消除重复行	17
第 2 章	连接表	19
	使用矢量积连接表	19
	限制连接	20
	表的关联方式	21
	主键标识符键	21
	表关系的外键	21
	连接运算符	22
	使用键连接来连接表	22
	使用自然连接来连接表	23
	使用即席连接与连接索引	24
	连接和数据类型	25
	支持存储或数据库之间的连接	26
	查询远程异构数据库	27
	用子查询替换连接	28

第 3 章	优化查询和删除	31
	实现结构化查询提示	31
	GROUP BY 与 UNION ALL 配合使用时对查询性能的影响	32
	增强 ORDER BY 查询性能	34
	增强了查询内的并行度	35
	改进了子查询性能	35
	使用高速缓存方法	35
	导致 SQL Anywhere 进行处理的条件	36
	计划查询	36
	查询评估选项	37
	查询树	38
	使用查询计划	39
	控制查询处理	40
	设置查询时间限制	40
	设置查询优先级	40
	设置查询优化选项	41
	设置用户提供的条件提示	42
	监控负载	42
	优化删除操作	43
	HG 删除操作	43
	WD 删除操作	45
	TEXT 删除操作	46
第 4 章	管理系统资源	47
	性能术语简介	47
	性能设计	48
	内存使用概述	48
	分页增加可用内存	48
	监控交换的实用程序	49
	服务器内存	49
	管理缓冲区高速缓存	50
	确定缓冲区高速缓存大小	50
	设置缓冲区高速缓存大小	54
	指定页大小	55
	节省内存	56
	为大量用户进行优化	56
	特定于平台的内存选项	58
	获取更多内存的其它方式	61
	进程线程模型	62
	线程不足错误	62
	用于管理线程使用的 Sybase IQ 选项	62

平衡 I/O	63
原始 I/O (在 UNIX 操作系统上)	63
使用磁盘分条	63
内部分条	65
使用多个文件	66
策略文件位置	66
用于插入、删除和同步的工作空间	68
设置保留空间选项	69
用于调优资源使用的选项	69
限制并发查询	69
设置可用 CPU 数	70
限制查询使用的临时 dbspace	70
限制按行返回的查询	70
强制游标为非滚动游标	71
限制游标数	71
限制语句数	71
预取高速缓存页	71
针对典型使用进行优化	72
控制预取行数	72
改进资源使用的其它方式	72
管理 Multiplex 数据库中的磁盘空间	73
查询服务器间的负载平衡	73
限制数据库访问	73
磁盘高速缓存	73
索引提示	74
选择正确索引类型	74
使用连接索引	75
为删除留出足够磁盘空间	75
管理数据库大小和结构	75
管理数据库大小	75
控制索引碎片	76
使 Catalog 文件增长最少	76
非规范化以提高性能	77
非规范化有风险	77
非规范化的缺点	77
非规范化的性能优势	78
决定非规范化	78
使用 UNION ALL 视图以便更快装载	78
优化引用 UNION ALL 视图的查询	80
管理 UNION ALL 视图性能	80

	改进的大型单（事实）表装载	81
	网络性能	81
	改进大型数据传输	82
	隔离重网络负荷用户	82
	将少量数据放在较小包中	83
	将大量数据放在较大包中	83
	在服务器级处理	83
第 5 章	监控和调优性能	85
	查看 Sybase IQ 环境	85
	监控性能统计信息	86
	使用存储过程获取信息	92
	分析数据库过程	93
	监控缓冲区高速缓存	101
	启动缓冲区高速缓存监控器	101
	在监控器运行时检查结果	106
	停止缓冲区高速缓存监控器	106
	检查并保存监控器结果	107
	监控器结果的示例	107
	缓冲区高速缓存结构	112
	避免缓冲区管理器抖动	112
	监控 Windows 系统上的分页	114
	监控 UNIX 系统上的分页	114
	缓冲区高速缓存监控清单	116
	监控 CPU 使用的系统实用程序	119
第 6 章	在 32 位 Windows 系统上调优服务器	121
	一般性能准则	121
	最大化吞吐量	121
	防止内存过度分配	122
	监控物理内存	122
	文件系统	122
	监控性能	123
	监控虚拟地址空间和工作集	123
	监控页面错误	124
	使用 NTFS 高速缓存	125
	对插入和查询进行调优	125
	合理调优的插入操作的特征	125
	对查询进行调优	126
	对备份操作进行调优	126

附录 A	性能统计信息	129
	字段引用	129
	缓冲区管理器	130
	缓冲池	132
	预取管理器	133
	IQ 存储空闲列表	133
	缓冲区分配器	133
	内存管理器	135
	线程管理器	135
	CPU 时间统计信息	135
	事务管理器	136
	上下文服务器统计信息	136
	目录、数据库日志和存储库统计信息	136
	统计信息	137
	页类型	142
	锁统计信息	143
	缓冲区分配器 / 预取客户端类型	144
索引		145

关于本手册

读者

本手册提供面向系统和数据库管理员的性能和调优建议。本文档假定您熟悉关系数据库系统和 Sybase® IQ。请将本指南与文档集中的其它手册结合使用。

相关 Sybase IQ 文档

Sybase IQ 15.2 文档集包括：

- 适用于您的平台的《发行公告》— 包含未能及时写入手册的最新信息。
最新版本的发行公告可能已发布。要了解本产品 CD 发行以后增加的重要产品或文档信息，请使用 Sybase Product Manuals Web 站点。
- 适用于您的平台的《安装和配置指南》— 介绍 Sybase IQ 的安装、升级和一些配置过程。
- 《Sybase IQ 15.2 新增功能摘要》— 总结了当前版本的新增功能和行为更改。
- 《Sybase IQ 中的高级安全性》— 涉及如何在 Sybase IQ 数据存储库中使用用户加密列。需要有单独的许可证才能安装此产品选项。
- 《错误消息》— 列出了由 Sybase 错误代码、SQLCode 和 SQLState 引用的 Sybase IQ 错误消息以及 SQL 预处理器错误和警告。
- 《IMSL 数字库用户指南：C Stat 库第二卷（共二卷）》— 包含 IMSL C Stat Library 时序 C 函数的简要说明。本手册仅适用于 RAP — Trading Edition™ Enterprise 用户。
- 《Sybase IQ 简介》— 包括针对不熟悉 Sybase IQ 或 Sybase Central™ 数据库管理工具的用户的学习。
- 《性能和调优指南》— 介绍有关超大型数据库的查询优化、设计和调优问题。
- 《快速入门》— 讨论如何生成并查询 Sybase IQ 提供的演示数据库以验证 Sybase IQ 软件安装。其中包括有关将演示数据库转换为 Multiplex 数据库的信息。

-
- 《参考手册》 — Sybase IQ 的参考指南：
 - 《参考：构件块、表和过程》 — 介绍 Sybase IQ 支持的 SQL、存储过程、数据类型和系统表。
 - 《参考：语句和选项》 — 介绍 Sybase IQ 支持的 SQL 语句和选项。
 - 《系统管理指南》 — 包括：
 - 《系统管理指南，卷 1》 — 介绍启动、连接、数据库创建、填充和编制索引、版本控制、归类、系统备份和恢复、故障排除和数据库修复。
 - 《系统管理指南，卷 2》 — 介绍如何编写和运行过程和批处理、使用 OLAP 编程、访问远程数据、将 IQ 设置为 Open Server。本手册还讨论调度和事件处理、XML 编程和调试。
 - 《时序指南》 — 介绍用于时序预测和分析的 SQL 函数。需要 RAP — Trading Edition Enterprise™ 才能使用此产品选件。
 - 《Sybase IQ 中的非结构化数据分析》 — 说明如何在 Sybase IQ 数据库中存储和检索非结构化数据。需要有单独的许可证才能安装此产品选件。
 - 《用户定义的函数指南》 — 提供有关用户定义的函数、这些函数的参数以及可能的使用情形的信息。
 - 《使用 Sybase IQ Multiplex》 — 说明如何使用管理跨多个节点的大量查询负载的 Multiplex 功能。
 - 《实用程序指南》 — 提供 Sybase IQ 实用程序参考资料，如可用的语法、参数和选项。

可以在以下位置通过联机方式获得 Sybase IQ 15.2 文档集：位于 <http://sybooks.sybase.com> 上的“产品手册” (Product Manuals)。

相关 SQL Anywhere 文档

由于 Sybase IQ 与 SQL Anywhere® Server (SQL Anywhere 软件包的组件) 共享许多相同组件, 因此 Sybase IQ 支持许多与 SQL Anywhere Server 相同的功能。IQ 文档集会提示您参考相应的 SQL Anywhere 文档。

SQL Anywhere 的文档包括:

- 《SQL Anywhere Server — 数据库管理》介绍如何运行、管理和配置 SQL Anywhere 数据库。它介绍数据库连接、数据库服务器、数据库文件、备份过程、安全性、高可用性和使用 Replication Server® 进行复制以及管理实用程序和选项。
- 《SQL Anywhere Server — 编程》介绍如何使用 C、C++、Java、PHP、Perl、Python 和 .NET 编程语言 (如 Visual Basic 和 Visual C#) 生成和部署数据库应用程序。本手册还介绍诸如 ADO.NET 和 ODBC 之类的各种编程接口。
- 《SQL Anywhere Server — SQL 参考》提供系统过程的参考信息和目录 (系统表和视图)。它还提供 SQL 语言的 SQL Anywhere 实现的说明 (搜索条件、语法、数据类型和函数)。
- 《SQL Anywhere Server — SQL 用法》介绍如何设计和创建数据库; 如何导入、导出和修改数据; 如何检索数据以及如何生成存储过程和触发器。

此外, 您还可以参阅位于 <http://sybooks.sybase.com> 的“产品手册”(Product Manuals) 的 SQL Anywhere 11.0.1 集合中和位于 http://dcx.sybase.com/dcx_home.php 的 DocCommentXchange 站点内的 SQL Anywhere 文档。



从数据库表中选择数据

本章概述了基本查询的构建并建议对产品设计的利用加以改进。在本教程中，您将查看表内容，对查询结果排序，选择列和行以及使用搜索条件优化查询。

有关高级查询性能的建议，请参见第 3 章“优化查询和删除”。

主题	页码
前提条件	1
查看表信息	2
查询结果排序	4
选择列和行	5
使用搜索条件	6
获取集合数据	10
获取分析数据	15
消除重复行	17

前提条件

如果您使用图形前端工具而不是 Interactive SQL 查询数据库，该工具可能允许您查看其生成的 SQL 语法。例如，在 InfoMaker 中，您可以通过选择“表”绘制工具栏上的“SQL 语法”按钮来查看 SQL 语句。

本教程介绍用于从数据库检索信息的 SELECT 语句。由于 SELECT 语句向数据库服务器询问有关数据库的信息，因此这些语句通常称为查询。

注释 SELECT 语句属于通用命令。SELECT 语句在应用程序中可能会变得非常复杂，从大型数据库中检索非常具体的信息。本教程仅使用简单的 SELECT 语句；更加高级的查询将在以后的教程中进行介绍。有关 select 语句的完整语法的详细信息，请参见《参考：语句和选项》第 1 章“SQL 语句”中的 SELECT 语句。

理想情况下，在学完本教程的课程时，您应该在计算机上运行 Sybase IQ 软件。

本教程假定您已启动了 Interactive SQL 并已连接到样本数据库。如果您尚未完成这些操作，请参见《实用程序指南》的 [第 2 章 “使用 Interactive SQL \(dbisql\)”](#)。

查看表信息

在本节中，您将查看 Employees 表中的数据。

您在本教程中所使用的样本数据库即为《Sybase IQ 简介》中所用的虚构公司。该数据库包含有关雇员、部门、销售订单等方面的信息。所有信息分布在若干个表中。

列出表

在《Sybase IQ 简介》中，您已了解到如何通过打开 Sybase Central 中的“Tables”文件夹显示表的列表。您还可以使用系统存储过程 `sp_iqtable` 从 Interactive SQL 列出用户表。系统存储过程是在 Sybase IQ 中作为存储过程实现的系统函数。

在“SQL 语句”窗口中，键入 `sp_iqtable` 以运行具有相同名称的系统存储过程。

有关此系统存储过程及其它系统存储过程的完整详细信息，请参见《参考：构件块、表和过程》中的 [第 7 章 “系统过程”](#)。

使用 SELECT 语句

在这一课中，您将查看数据库中的一个表。所使用的命令将查看名为 Employees 的表中的任何内容。

执行命令：

```
SELECT * FROM Employees
```

星号是该表中所有列的简写形式。

SELECT 语句会检索 Employees 表中的所有行和列，而“Interactive SQL 结果”窗口则列出满足以下条件的行和列：

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
102	501	Whitney	Fran	100	...
105	501	Cobb	Matthew	100	...
129	902	Chin	Philip	200	...
148	1,293	Jordan	Julie	300	...
160	501	Breault	Robert	100	...
184	1,576	Espinoza	Melissa	400	...

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
191	703	Bertrand	Jeannette	500	...
195	902	Dill	Marc	200	...
207	1576	Francis	Jane	400	...

`Employees` 表中包含许多分布在多个列中的行。每列都有一个名称，例如 `Surname` 或 `EmployeeID`。针对公司的每个雇员都有一行，每行在每列中都有一个值。例如，`EmployeeID` 为 102 的雇员为 Fran Whitney，其经理的 `ManagerID` 为 501。

您还会在“Interactive SQL 消息”窗口中看到一些信息。稍后将对这些信息进行说明。

注释 本文中显示查询结果的表可能只包含查询返回的某些数据。具有省略值的列和行表示附加查询结果。

区分大小写

即使实际表名均为小写字母，但显示的表名 `Employees` 会以大写字母 E 开头。在其字符串比较中，可以将 Sybase IQ 数据库创建为区分大小写（缺省值）或不区分大小写，但在它们作为标识符使用时总是不区分大小写。

注释 使用 `CREATE DATABASE` 限定符 `CASE IGNORE` 创建本书中的示例时不区分大小写。缺省值为 `CASE RESPECT`，它可提供更好的性能。

有关创建数据库的信息，请参见《系统管理指南，卷 1》中的第 5 章“使用数据库对象”。

您可以键入 `select` 或 `Select` 代替 `SELECT`。Sybase IQ 允许您以大写形式、小写形式或这两种形式的任意组合键入关键字。在本手册中，SQL 关键字通常使用大写字母。

对 Interactive SQL 环境的操作和 Interactive SQL 的使用特定于操作系统。

有关如何滚动浏览数据以及操作 Interactive SQL 环境的信息，请参见《实用程序指南》中的第 2 章“使用 Interactive SQL (dbisql)”。

查询结果排序

在本节中，您将向 `SELECT` 语句添加 `ORDER BY` 子句，以按字母顺序或数值顺序显示结果。

除非另有要求，否则 Sybase IQ 不会以特定顺序显示表中的行。通常，以一种更有意义的顺序查看表中的行将十分有用。例如，您可能要按字母顺序查看雇员。

按字母顺序列出雇员

以下示例显示以何种方式向 `SELECT` 语句添加 `ORDER BY` 子句可导致以字母顺序检索结果。

```
SELECT * FROM Employees ORDER BY Surname
```

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
1,751	1,576	Ahmed	Alex	400	...
1,013	703	Barker	Joseph	500	...
591	1,576	Barletta	Irene	400	...
191	703	Bertrand	Jeannette	500	...
1,336	1,293	Bigelow	Janet	300	...
1,062	1,576	Blaikie	Barbara	400	...
750	703	Braun	Jane	500	...
160	501	Breault	Robert	100	...
1,191	1,576	Bucceri	Matthew	400	...

注意

子句的顺序十分重要。`ORDER BY` 子句必须在 `FROM` 子句和 `SELECT` 子句之后。

注释 如果省略 `FROM` 子句，或者查询中的所有表都在 `SYSTEM dbspace` 中，则查询将由 SQL Anywhere 而非 Sybase IQ 处理且行为可能不同，特别是在关于语法和语义限制和选项设置的影响方面。有关可能应用到处理的规则，请参见 SQL Anywhere 文档。

如果您的查询不需要 `FROM` 子句，则可以通过添加 `FROM iq_dummy` 子句强制 Sybase IQ 处理查询，在这种情况下，`iq_dummy` 表示在数据库中创建的包含一行和一列的表。

选择列和行

列出每个雇员的姓氏、部门和出生日期

通常，您只关注表中的列。例如，要为雇员制作生日卡，您可能希望看到 Surname、DepartmentID 和 BirthDate 列。

在本节中，您将选择每个雇员的出生日期、姓氏和部门 ID。键入以下语句：

```
SELECT Surname, DepartmentID, BirthDate
FROM Employees
```

Surname	DepartmentID	BirthDate
Whitney	100	1958-06-05
Cobb	100	1960-12-04
Chin	200	1966-10-30
Jordan	300	1951-12-13
Breault	100	1947-05-13

对列进行重新排列

这三个列按照您在 **SELECT** 命令中键入的顺序进行显示。如果您要对列进行重新排列，只要在该命令中更改列名的顺序即可。例如，若要将 BirthDate 列放在左边，请使用以下命令：

```
SELECT BirthDate, Surname, DepartmentID
FROM Employees
```

对行进行排序

您可以对行进行排序，但同时只能查看某些列，如以下代码所示：

```
SELECT BirthDate, Surname, DepartmentID
FROM Employees
ORDER BY Surname
```

星号（位于

```
SELECT * FROM Employees
```

中）是该表中所有列的简写形式。

注释 涉及包含大量空值的列的查询比先前版本中的运行速度快。但是，在将大量空值插入到表的情况下，在该表中插入或更新数据这一过程可能会花费较长时间（同先前版本相比）。

使用搜索条件

在本节中，您将了解比较日期的过程，以及如何使用 WHERE 子句中的复合搜索条件、模式匹配和搜索条件快捷方式。

有时，您不想看到有关 **Employees** 表中所有雇员的信息。向 SELECT 语句添加 WHERE 子句只允许从表中选择某些行。

例如，假定您要查看名字为 John 的雇员。

❖ 列出所有名为 John 的雇员：

- 键入以下语句：

```
SELECT *
FROM Employees
WHERE GivenName = 'John'
```

EmployeeID	ManagerID	Surname	GivenName	DepartmentID	...
318	1,576	Crow	John	400	...
862	501	Sheffield	John	100	...
1,483	1,293	Leticq	John	300	...

撇号和区分大小写

- 需要在名字 'John' 的两边加撇号（单引号）。它们指示 John 是一个字符串。引号（双引号）具有不同的含义。对于列名和其它标识符，可以使用引号使原本无效的字符串有效。
- 样本数据库是不区分大小写的，因此不管您搜索 'JOHN'、'john' 还是 'John'，都将得到相同的结果。

此外，您还可以将所学内容结合起来：

```
SELECT GivenName, Surname, BirthDate
FROM Employees
WHERE GivenName = 'John'
ORDER BY BirthDate
```

注意

- 以何种方式对子句进行排序十分重要。FROM 子句排在第一位，后面是 WHERE 子句，再后面是 ORDER BY 子句。如果以不同的顺序键入子句将出现语法错误。
- 无需将语句拆分成多行。您可以在“SQL 语句”窗口中以任意格式输入语句。如果使用的行数超过屏幕所能容纳的限度，文本将在“SQL 语句”窗口中滚动。

比较查询中的日期

有时，您不知道所查看的值具体为多少，或者您要查看一组值。您可以在 **WHERE** 子句中使用比较来选择一组满足搜索条件的行。

列出 1964 年 3 月 3 日前出生的雇员

以下示例显示了日期不等式搜索条件的用法。键入以下语句：

```
SELECT Surname, BirthDate
FROM Employees
WHERE BirthDate < 'March 3, 1964'
```

Surname	BirthDate
Whitney	1958-06-05
Cobb	1960-12-04
Jordan	1951-12-13
Breault	1947-05-13
Espinoza	1939-12-14
Dill	1963-07-19
Francis	1954-09-12
Shishov	1949-04-22
...	...

Sybase IQ 知道 **BirthDate** 列包含日期，并将 *'March 3, 1964'* 自动转换为日期。

WHERE 子句中的复合搜索条件

到目前为止，您已经了解了作为比较运算符的等于 (=) 和小于 (<)。Sybase IQ 还支持其它比较运算符，如大于 (>)、大于或等于 (>=)、小于或等于 (<=) 以及不等于 (<>)。

可以使用 **AND** 和 **OR** 将这些搜索条件组合起来形成更加复杂的搜索条件。

限定列出的内容

列出 1964 年 3 月 3 日前出生的所有雇员（但名为 **Whitney** 的雇员除外）：

```
SELECT Surname, BirthDate
FROM Employees
WHERE BirthDate < '1964-3-3'
AND Surname <> 'Whitney'
```

Surname	BirthDate
Cobb	1960-12-04
Jordan	1951-12-13
Breault	1947-05-13
Espinoza	1939-12-14
Dill	1963-07-19
Francis	1954-09-12
Shishov	1949-04-22
...	...

搜索条件中的模式匹配

查找内容的另一种有效方法是搜索模式。在 SQL 中，单词 LIKE 用于搜索模式。以下示例可以说明 LIKE 的用法。

列出特定雇员

要列出姓氏以 "br" 开头的所有雇员，请键入以下内容：

```
SELECT Surname, GivenName
FROM Employees
WHERE Surname LIKE 'br%'
```

Surname	GivenName
Breault	Robert
Braun	Jane

搜索条件中的 % 表示任何数目的其它字符都可以接在字母 BR 之后。

限定姓搜索

若要列出姓符合以下条件的所有雇员：以 BR 开头，其后有零个或多个字母，然后有一个 T，T 后有零个或多个字母，请键入：

```
SELECT Surname, GivenName
FROM Employees
WHERE Surname LIKE 'BR%T%'
```

Surname	GivenName
Breault	Robert

第一个 % 符号与字符串 "caul" 匹配，而第二个 % 符号与空字符串（不包含任何字符）匹配。

另一个可以与 LIKE 一起使用的特殊字符是 _（下划线）字符，下划线只与一个字符匹配。

模式 BR_U% 与所有以 BR 开头并以 U 作为第四个字母的名称匹配。在 Braun 中，_ 与字母 A 匹配，% 与 N 匹配。

通过发音匹配行

如果使用 SOUNDEX 函数，则可以通过发音以及拼写匹配行。例如，假设有一个电话留言找姓名发音类似 “Ms. Brown” 的人。公司中哪些雇员的姓名发音与 Brown 类似？

通过发音搜索姓

若要列出姓的发音与 Brown 类似的雇员，请键入以下代码：

```
SELECT Surname, GivenName
FROM Employees
WHERE SOUNDEX( Surname ) = SOUNDEX( 'Brown' )
```

Surname	GivenName
Braun	Jane

Jane Braun 是唯一与搜索条件相符的雇员。

对搜索条件使用快捷方式

使用简写形式
BETWEEN

SQL 具有两个用于键入搜索条件的简写形式。第一个简写形式 BETWEEN 用于查找一系列值。例如，以下两个示例的查询结果相等：

```
SELECT Surname, BirthDate
FROM Employees
WHERE BirthDate BETWEEN '1964-1-1'
AND '1965-3-31'
```

```
SELECT Surname, BirthDate
FROM Employees
WHERE BirthDate >= '1964-1-1'
AND BirthDate <= '1965-3-31'
```

使用简写形式 IN

第二个简写形式 IN 用于查找多个值中的一个值。命令。以下两个示例的查询结果相等。

```
SELECT Surname, EmployeeID
FROM Employees
WHERE Surname IN ('Yeung', 'Bucceri', 'Charlton')
```

```
SELECT Surname, EmployeeID
FROM Employees
WHERE Surname = 'Yeung'
OR Surname = 'Bucceri'
OR Surname = 'Charlton'
```

获取集合数据

本节介绍如何构建用于获得集合信息的查询。下面是集合信息示例：

- 一列中所有值的总和
- 一列中各条目的数量
- 一列中各条目的平均值

集合函数简介

假定您想要知道数据库中有多少个雇员。以下语句检索 `employee` 表中的行数：

```
SELECT count( * )
FROM Employees
```

此查询返回的结果是一个仅包含一列（标题为 `count(*)`）和一行（包含雇员数目）的表。

Count()
75

以下命令比集合查询稍复杂一些：

```
SELECT count( * ),
min( BirthDate ),
max( BirthDate )
FROM Employees
```

此查询的结果集包含三列和一行。这三列中包含雇员数目、年纪最大雇员及年纪最小雇员的出生日期。

Count()	Min(BirthDate)	Max(BirthDate)
75	'1936-01-02'	'1973-01-18'

`COUNT`、`MIN` 和 `MAX` 都称为集合函数。集合函数使用 `SELECT` 语句的 `GROUP BY` 子句汇总数据库中的完整表。总共有七个集合函数：`AVG`、`COUNT`、`MAX`、`MIN`、`STDDEV`、`SUM` 和 `VARIANCE`。所有这些函数都以列名或表达式作为参数。正如您所看到的那样，`COUNT` 还有一个星号作为其参数，用于返回每组中的行数。

使用集合函数获取分组数据

除了提供有关整个表的信息之外，集合函数还可用于行组。

对行组使用集合函数

若要列出每个销售代表负责的订单数，请键入：

```
SELECT SalesRepresentative, count( * )
FROM SalesOrders
GROUP BY SalesRepresentative
```

SalesRepresentative	Count()
129	57
195	50
299	114
467	56
667	54
690	52
856	55
902	47
949	53
1,142	57
1,596	53

此查询的结果由对应于每个 SalesRepresentative ID 号的行组成，其中包含 SalesRepresentative ID 及 SalesOrders 表中该 ID 号的行数。

每当使用 GROUP BY 时，结果表中都有一行对应于位于 GROUP BY 列中的每个不同的值。

限制组

您已经了解如何使用 WHERE 子句限制查询中的行。您可以使用 HAVING 关键字来限制 GROUP BY 子句。

限制 GROUP BY 子句

若要列出其订单数超过 55 的所有销售代表，请键入：

```
SELECT SalesRepresentative, count( * )
FROM SalesOrders
GROUP BY SalesRepresentative
HAVING count( * ) > 55
```

SalesRepresentative	count(*)
129	57
299	114
467	56
1,142	57

注释 GROUP BY 必须始终出现在 HAVING 之前。与此相同，WHERE 必须出现在 GROUP BY 之前。

使用 WHERE 和 GROUP BY

若要列出其订单数超过 55 且 ID 大于 1000 的所有销售代表，请键入：

```
SELECT SalesRepresentative, count( * )
FROM SalesOrders
WHERE SalesRepresentative > 1000
GROUP BY SalesRepresentative
HAVING count( * ) > 55
```

Sybase IQ 查询优化程序将谓词从 HAVING 子句移至 WHERE 子句，这样做可以提高性能。例如，如果您指定：

```
GROUP BY SalesRepresentative
HAVING count( * ) > 55
AND SalesRepresentative > 1000
```

来替换前面示例中的 WHERE 子句，查询优化程序会将谓词移至 WHERE 子句中。

Sybase IQ 使用简单的条件（不涉及 OR 或 IN）执行此优化。为此，在使用 WHERE 子句和 HAVING 子句构建查询时，应尤为谨慎地在 WHERE 子句中列出尽可能多的条件。

改善小计计算

如果您已具有在各个维度间变化的数据（如，日期或地点），则可能需要确定数据在每个维度中是如何变化的。您可以使用 ROLLUP 和 CUBE 运算符为对分组列的引用的列表创建多级别的小计和总计。最详细级别的小计“累计”到总计。例如，如果您要分析销售数据，则可以使用同一查询按年计算总体平均值和平均销售额。

使用 ROLLUP

选择按年份、车型和颜色合计汽车销售额：

```
SELECT Name, Size, Color, Sum(Quantity)
FROM Products
GROUP BY ROLLUP (Name, Size, Color);
```

Name	Size	Color	sum(Products.Quantity)
Baseball Cap	One size fits all	Black	112
Baseball Cap	One size fits all	White	12
Baseball Cap	One size fits all	(NULL)	124
Baseball Cap	(NULL)	(NULL)	124
Shorts	Medium	Black	80
Shorts	Medium	(NULL)	80
Shorts	(NULL)	(NULL)	80
Sweatshirt	Large	Blue	32
Sweatshirt	Large	Green	39
Sweatshirt	Large	(NULL)	71
Sweatshirt	(NULL)	(NULL)	71
Tee Shirt	Medium	Orange	54
Tee Shirt	Medium	(NULL)	54
Tee Shirt	One size fits all	Black	75
Tee Shirt	One size fits all	(NULL)	75
Tee Shirt	Small	White	28
Tee Shirt	Small	(NULL)	28
Tee Shirt	(NULL)	(NULL)	157
Visor	One size fits all	Black	28
Visor	One size fits all	White	36
Visor	One size fits all	(NULL)	64
Visor	(NULL)	(NULL)	64
(NULL)	(NULL)	(NULL)	496

处理此查询时，Sybase IQ 首先按三个指定的分组表达式（年份、车型和颜色）对数据进行分组，然后按除最后一个分组表达式（颜色）之外的其余两个分组表达式对数据进行分组。在第五行中，NULL 表示 color 列的 ROLLUP 值，换句话说，它表示该车型所有颜色汽车的总销售额。343 表示 1990 年所有车型和所有颜色汽车的总销售额，314 表示 1991 年所有车型和所有颜色汽车的总销售额。最后一行表示所有年份的所有车型和所有颜色汽车的总销售额。

ROLLUP 需要分组表达式的有序列表作为参数。在列出包含其它组的组时，请首先列出较大的组（如在 city 之前列出 state）。

您可以对以下集合函数使用 ROLLUP: AVG、COUNT、MAX、MIN、STDDEV、SUM 和 VARIANCE。不过，ROLLUP 不支持 COUNT DISTINCT 和 SUM DISTINCT。

使用 CUBE

以下查询使用 Employees 表中的数据，其中包括省/市/自治区（地理位置）、性别、教育水平和国民收入。如果您要计算省/市/自治区的整个人口统计中的人均收入、性别和教育情况，并根据列 state、gender 及 education 的所有可能组合形式计算平均收入，而且只扫描表 census 中的人口统计数据一次，则可以使用 GROUP BY 子句的 CUBE 扩展。例如，如果要计算所有省/市/自治区的所有女性的平均收入，或者要根据所受教育和地理位置计算人口统计中所有人的平均收入，请使用 CUBE 运算符。

当 CUBE 对某个组进行计算时，CUBE 会将空值置于其组已得到计算的列中。区分每行表示的组的类型以及确定 NULL 是存储在数据库中的 NULL 还是由 CUBE 生成的 NULL 十分困难。如果指定的列已被合并到更高级别的组中，则 GROUPING 函数通过返回 1 来解决此问题。

以下查询说明了如何将 GROUPING 函数与 GROUP BY CUBE 结合使用。

```
SELECT case grouping(state) WHEN 1 THEN 'ALL' ELSE state
      END AS c_state, case grouping(sex) WHEN 1 THEN 'ALL'
      ELSE sex END AS c_gender, case
      grouping(DepartmentId)
      WHEN 1 THEN 'ALL' ELSE cast(DepartmentId as char(4))
      end
      AS c_dept, COUNT(*), CAST(ROUND(AVG(salary),2) AS
      NUMERIC(18,2))AS AVERAGE
FROM employees WHERE state IN ('MA' , 'CA')
GROUP BY CUBE(state, sex, DepartmentId)
ORDER BY 1,2,3;
```

下面显示了此查询的结果。请注意，由 CUBE 生成的指示小计行的 NULL 将按照查询中的指定，在小计行中被替换为 ALL。

c_state	c_gender	c_dept	COUNT()	AVERAGE
ALL	ALL	200	3	52,200.00
ALL	ALL	ALL	3	52,200.00
ALL	F	200	2	58,650.00
ALL	F	ALL	2	58,650.00
ALL	M	200	1	39,300.00
ALL	M	ALL	1	39,300.00
CA	ALL	200	3	52,200.00

c_state	c_gender	c_dept	COUNT()	AVERAGE
CA	ALL	ALL	3	52,200.00
CA	F	200	2	58,650.00
CA	F	ALL	2	58,650.00
CA	M	200	1	39,300.00
CA	M	ALL	1	39,300.00

数据仓库管理员发现 ROLLUP 和 CUBE 对如下所示的操作特别有用：

- 对诸如地理或时间等的层次维度进行小计，例如年份 / 月份 / 日期或国家 / 地区 / 省 / 市 / 自治区 / 城市
- 填充摘要表

如果采用多级分组而不是对每个级别进行单独查询，则 ROLLUP 和 CUBE 允许您使用一个查询来对数据进行计算。

有关 ROLLUP 和 CUBE 运算符的详细信息，请参见《参考：语句和选项》“SQL 语句”中的 [SELECT 语句](#)。

获取分析数据

本节介绍如何构建用于获得分析信息的查询。分析函数有以下两种类型：**rank** 分析函数和逆分布分析函数。**rank** 分析函数排列组中的项、计算分布以及将结果集分成多个组。逆分布分析函数返回第 k 个百分点值，该值可用于帮助为一组数据建立可接受阈值。

rank 分析函数有 RANK、DENSE_RANK、PERCENT_RANK 和 NTILE。逆分布分析函数有 PERCENTILE_CONT 和 PERCENTILE_DISC。

假设您要对雇员的薪水划分等级。在下面的示例中，NTILE 函数根据雇员的薪水将雇员分成四组。其 ntile 排名为 1 的雇员是薪水范围中最靠前的 25%。

```
SELECT Name
       Salary,
       NTILE(4) OVER(ORDER BY salary DESC) as Ranking
FROM empl;
```

Name	Salary	Ranking
Sandy	55,000	1
Peter	48,000	1
Lisa	38,000	1
Scott	29,000	1
Tim	29,000	2
Tom	28,000	2
Mike	28,000	2
Adam	25,000	3
Antonia	22,000	3
Jim	22,000	3
Anna	18,000	4
Jeff	18,000	4
Amy	18,000	4

NTILE 为一个分析函数，可将查询结果分配到指定数量的表元或在这些表元中对查询结果划分等级，并为表元中的每一行指派相应的表元号。您可以将结果集分成十组（十等分）、四组（四等分）及其它分组数。

rank 分析函数需要使用 OVER (ORDER BY) 子句。ORDER BY 子句指定对其执行排序的参数以及每组中行的排序顺序。请注意，此 ORDER BY 子句仅在 OVER 子句中使用，而不是用于 SELECT 的 ORDER BY。

OVER 子句表示函数对查询结果集进行操作。结果集是在对 FROM、WHERE、GROUP BY 和 HAVING 子句求值完成之后返回的行。OVER 子句定义要包括在 rank 分析函数计算中的行数据集。

同样，逆分布分析函数需要使用 WITHIN GROUP (ORDER BY) 子句。ORDER BY 指定对其执行百分点函数的表达式以及每组中行的排序顺序。此 ORDER BY 子句仅在 WITHIN GROUP 子句中使用，而不是用于 SELECT 的 ORDER BY。WITHIN GROUP 子句将查询结果分布到排序数据集中，函数通过此数据集计算结果。

有关分析函数的更多详细信息，请参见《参考：构件块、表和过程》第 4 章“SQL 函数”中的“分析函数”。有关各个分析函数的信息，请参见同一章节中针对各个函数的部分。

消除重复行

来自 `SELECT` 语句的结果表可以包含重复行。您可以使用 `DISTINCT` 关键字来消除重复项。例如，以下命令返回多个重复行：

```
SELECT city, state FROM Employees
```

若要只列出城市和省 / 市 / 自治区的唯一组合，请使用此命令：

```
SELECT DISTINCT city, state FROM Employees
```

注释 `ROLLUP` 和 `CUBE` 运算符不支持 `DISTINCT` 关键字。

本章概述了单表 `SELECT` 语句。有关单表 `SELECT` 语句的详细信息，请参见

- 《系统管理指南，卷 1》中的第 5 章“使用数据库对象”。
- 《参考：构件块、表和过程》中的第 2 章“SQL 语言元素”。
- 《参考：语句和选项》的第 1 章“SQL 语句”的中的“`SELECT` 语句”。

`SELECT` 语句的高级用法将在下一章中进行介绍。

连接表

本章说明如何查看多个表中的信息，并且说明各类连接。您将完成有关连接表的教程任务。

主题	页码
使用矢量积连接表	19
限制连接	20
表的关联方式	21
连接运算符	22
使用即席连接与连接索引	24
连接和数据类型	25
支持存储或数据库之间的连接	26
查询远程异构数据库	27
用子查询替换连接	28

使用矢量积连接表

样本数据库中的其中一个表为 `FinancialData`，此表列出公司的财务数据。每个数据记录均具有 `code` 列，此列用于标识记录对应的部门及记录是支出记录还是收入记录。

可以同时从两个表获取信息，方法是在 `SELECT` 查询的 `FROM` 子句中列出两个表，并用逗号分隔。

示例

下面的 `SELECT` 命令列出 `FinancialCodes` 和 `FinancialData` 表中的所有数据：

```
SELECT *  
FROM FinancialCodes, FinancialData
```

此查询的结果显示在 `Interactive SQL` 数据窗口中，并且将 `FinancialCodes` 表中的每一行与 `FinancialData` 表中的每一行进行匹配。此种连接称作完全矢量积，也称作笛卡儿乘积。每一行都包含 `FinancialCodes` 表的所有列，后跟 `FinancialData` 表的所有列。

矢量积连接是了解连接的一个简单的起点，但其本身并不是很有用。本章中的后续小节将介绍如何构造更富选择性的连接，对于这些连接，均可视为对矢量积表应用一些限制。

限制连接

要使矢量积连接有用，通常需要在结果中仅包括满足某条件的行。此条件称作连接条件，它使用比较运算符（=、=>、<等）将一个表中的某一列与另一个表中的某一列进行比较。这样，就可以从由矢量积结果中消除某些行。

例如，要使上一节中的连接有用，可以在结果的每一行中强制使 **SalesOrders** 表中的 **SalesRepresentative** 与 **Employees** 表中的对应内容相同。这样，每行都包含有关订单和负责该订单的销售代表的信息。

示例 1

为此，在前面的查询中添加 **WHERE** 子句，以显示雇员列表及其课程注册：

```
SELECT *
FROM SalesOrders, Employees
WHERE SalesOrders.SalesRepresentative =
Employees.EmployeeID
```

用表名称作前缀来标识列。尽管在此情况下未作严格要求，但使用表名称前缀可以使语句清晰明了，当两个表中存在具有相同名称的列时，必须使用表名称前缀。在此情况下使用的表名称称作**限定符**。

示例 2

以下查询是修改后的查询，此查询仅提取某些列并对结果进行排序：

```
SELECT Employees.Surname, SalesOrders.id,
SalesOrders.OrderDate
FROM SalesOrders, Employees
WHERE SalesOrders.SalesRepresentative =
Employees.EmployeeID
ORDER BY Employees.Surname
```

如果 **SELECT** 命令中有多个表，则可能需要键入若干限定符名称。可以通过使用相关名来减少键入操作。

相关名

相关名是特定表实例的别名。此别名仅在单个语句内有效。通过在表名称之后立即输入表名称的简写形式并用关键字 **AS** 分隔来创建相关名。在此之后，*必须*将简写形式而非相应的表名称用作限定符：

```
SELECT E.Surname, S.id, S.OrderDate
FROM SalesOrders AS S, Employees AS E
WHERE S.SalesRepresentative = E.EmployeeID
ORDER BY E.Surname
```


在此，分别为 `SalesOrders` 和 `Employees` 表创建了 S 和 E 两个相关名。

注释 仅当不同表的两个列具有相同名称时，才需要使用表名称或相关名解决多义性问题。如果创建了相关名，则必须使用此相关名，而不能使用完整表名称，但如果尚未创建相关名，请使用完整表名称。

表的关联方式

要构造其它类型的连接，必须先了解一个表中的信息是如何与另一个表中的信息关联的。

表的主键标识表中的每一行。表使用外键相互关联。

本节说明如何共同使用主键和外键来构造基于多个表的查询。

主键标识符键

`iqdemo` 数据库中的每个表均具有一个主键。主键是唯一标识表中的行的一个或多个列。例如，雇员编号唯一标识雇员 - `EmployeeID` 是 `Employees` 表的主键。

`SalesOrderItems` 表是由两列构成主键的示例表。订单 ID 本身不会唯一标识 `SalesOrderItems` 表中的行，因为一个订单中包含若干项目。此外，`LineID` 编号也不会唯一标识 `SalesOrderItems` 表中的行。需要同时采用订单 ID 名称和 `LineID` 才能唯一标识 `SalesOrderItems` 表中的行。因此，这两列共同作为此表的主键。

表关系的外键

`iqdemo` 数据库中的若干表引用此数据库中的其它表。例如，在 `SalesOrders` 表中，`SalesRepresentative` 列表示负责某订单的雇员。`SalesOrders` 表中仅保留足以唯一标识雇员的信息。`SalesOrders` 表中的 `SalesRepresentative` 列是 `Employees` 表的一个外键。

外键是包含其它表中的候选键值的一个或多个列。（有关候选键的详细信息，请参见《系统管理指南，卷 1》中的第 1 章“SQL 语句”。）

连接运算符

许多常见的连接建立在两个由键关联的表之间。最常见的连接限制一个表中的外键值与另一个表中的主键值相等。所示示例限制 `SalesOrders` 表中的外键值，使其等于 `Employees` 表中的候选键值。

```
SELECT Surname,  
       EmployeeID,  
       OrderDate  
FROM SalesOrders, Employees  
WHERE SalesOrders.SalesRepresentative =  
       Employees.EmployeeID
```

如下节所述，使用 `KEY JOIN` 可以更简单地表达该查询。

使用键连接来连接表

键连接便于连接由外键关联的表。以下示例返回的结果与具有特定 `WHERE` 子句（使两个雇员 ID 号列相等）的查询的结果相同：

```
SELECT Surname,  
       EmployeeID,  
       OrderDate  
FROM SalesOrders  
KEY JOIN Employees  
  
SELECT Surname,  
       EmployeeID,  
       OrderDate  
FROM SalesOrders, Employees  
WHERE SalesOrders.SalesRepresentative =  
       Employees.EmployeeID
```

连接运算符 (`KEY JOIN`) 只是键入 `WHERE` 子句的快捷方式；这两个查询相同。

在《`Sybase IQ` 简介》中的 `iqdemo` 数据库图示中，外键由表之间的直线表示。只要图示中两个表由直线连接，就可以使用 `KEY JOIN` 运算符。请记住，应用程序必须强制外键才能确保从基于键连接的查询获得预期结果。

连接两个或两个以上的表

可以使用连接运算符连接两个或两个以上的表。下面的查询使用四个表列出每个客户所下订单的总值。它将四个表 `customer`、`SalesOrders`、`SalesOrderItems` 和 `Products` 两两之间的单一外键关系连接起来。

```
SELECT CompanyName,
       CAST( SUM(SalesOrderItems.Quantity *
                Products.UnitPrice) AS INTEGER) AS Value
FROM Customers
KEY JOIN SalesOrders
KEY JOIN SalesOrderItems
KEY JOIN Products
GROUP BY CompanyName
```

CompanyName	Value
The Power Group	5,808
The Birds Loft	4,404
Sampson & Sons	6,660
Hats Etc.	2,736
Howard Co.	5,388
...	...

此查询中使用的 `CAST` 函数转换表达式的数据类型。在此示例中，将作为整数返回的总和转换为值。

使用自然连接来连接表

`NATURAL JOIN` 运算符根据公共列名称连接两个表。换句话说，Sybase IQ 生成一个 `WHERE` 子句，该子句使每个表的公共列相等。

示例

例如，对于以下查询：

```
SELECT Surname,
       DepartmentName
FROM Employees
NATURAL JOIN Departments
```

数据库服务器查看这两个表，并确定它们共同具有的唯一列名称为 `DepartmentID`。将在内部生成下面的 `ON` 短语并使用此短语执行此连接：

```
FROM Employees JOIN Departments
...
ON Employees.DepartmentID = Departments.DepartmentID
```

使用 NATURAL JOIN 时出错

此连接运算符可能会使您不想等同的列相等，从而引起问题。例如，以下查询会生成不想要的结果：

```
SELECT *
FROM SalesOrders
NATURAL JOIN Customers
```

此查询的结果不包含任何行。

数据库服务器在内部生成下面的 ON 短语：

```
FROM SalesOrders JOIN Customers
ON SalesOrders.ID = Customers.ID
```

SalesOrders 表中的 id 列是订单的 ID 号。customer 表中的 id 列是客户的 ID 号。没有一个 ID 号匹配。当然，即使找到了匹配的编号，也没有任何意义。

应谨慎使用连接运算符。请牢记，连接运算符只是让您省却为未受强制的外键或公共列名称键入 WHERE 子句的必要。应注意 WHERE 子句，否则创建的查询可能会生成意外结果。

使用即席连接与连接索引

如果已对查询中引用的连接列定义了连接索引，则 Sybase IQ 通常将使用这些索引来执行连接那些表的查询。（有关定义连接索引的信息，请参见《系统管理指南，卷 1》中的第 6 章“使用 Sybase IQ 索引”。

不使用连接索引的任何连接称作**即席连接**。如果查询引用多个表，并且并非所有这些表都定义了连接索引，则 Sybase IQ 将对具有连接索引的表使用连接索引，同时对其余表使用即席连接。

由于无法为所有可能的连接都创建连接索引，因此即席连接有时可能是必需的。借助 Sybase IQ 中的优化功能，您会发现不使用连接索引时查询具有一样好或更高的性能。

当创建连接索引时，请牢记以下规则：

- 所有连接索引均使用完全外连接创建。使用连接索引的查询可以是内连接、左外连接或右外连接。

完全外连接是左指定表和右指定表中的*所有*行均包括在结果中的连接，同时对不具有匹配值的任何列在对应列中返回 NULL。

- 可在连接谓词 ON 子句中使用的唯一比较运算符为 EQUALS。

- 可以使用 `NATURAL` 关键字而非 `ON` 子句，但只能指定一对表。
- 连接索引列必须具有相同的数据类型、精度和标度。
- 当所涉及的表包含类似行数时，连接索引将对即席连接执行最佳比对。当较大的表和较小的表之间存在很大差异时，连接索引将对即席连接执行次优比对。

连接和数据类型

连接列需要相似的数据类型来获得最佳性能。使用 `Sybase IQ` 可对存在隐式转换的任何数据类型进行即席连接。但是，除非连接列数据类型相同，否则性能可能因数据类型和表大小而有不同程度的下降。例如，可以将 `INT` 连接到 `BIGINT` 列，但此连接会阻止某些特定类型的优化。`Sybase IQ` 索引顾问可以标识不匹配的连接数据类型，这些数据类型在此类情况下可能会影响性能。

对于连接键，如果其数据类型的值范围较小，则其提供的性能越优；如果其数据类型的值范围越大，则其提供的性能越差；具数据类型为整数的连接键将比数据类型为数字或字符的连接键速度更快。

虽然这些数据类型可以提供更好的性能，但选择具有匹配数据类型的键比选择具有“更快”但不匹配的数据类型的键通常会提供更有效的连接。如果数据类型不同，`Sybase IQ` 必须在内部转换其中一种数据类型以使列具有可比性，这样可能会降低性能。

有关隐式数据类型转换的表，请参见《系统管理指南，卷1》中的第7章“将数据移入和移出数据库”。

支持存储或数据库之间的连接

本节阐明当前对存储之间或数据库之间的连接的支持。

连接 Sybase IQ 数据库内的表

支持给定 Sybase IQ 数据库内的任何连接。这意味着可以按任何顺序将 Catalog 存储中的任何系统表或用户表与 IQ 存储中的任何表连接。

连接 Adaptive Server Enterprise 表和 Sybase IQ 表

在下列条件下，支持将 Sybase IQ 表与 Adaptive Server Enterprise 数据库中的表进行连接：

- Sybase IQ 数据库可以是本地数据库，也可以是远程数据库。
- 如果要在 ASE 中将 Sybase IQ 表用作代理表，则表名称不得超过 30 个字符。
- 为了连接本地 Adaptive Server Enterprise 表和远程 Sybase IQ 12 表，ASE 版本必须是 11.9.2 或更高版本，并且必须使用正确的服务器类：
 - 要从 Adaptive Server Enterprise 12.5 或更高版本的前端连接到远程 Sybase IQ 12.5 或更高版本，请使用已在 ASE 12.5 中添加的 ASIQ 服务器类。
 - 要从 Adaptive Server Enterprise 11.9.2 至 12.0 的前端连接到远程 Sybase IQ 12.x（或者 SQL Anywhere 6.x 或更高版本），必须使用 ASAnywhere 服务器类。
- 当连接本地 Sybase IQ 表和任何远程表时，必须在 FROM 子句中首先显示本地表，这意味着本地表是连接中的最外层表。

Sybase IQ 和 Adaptive Server Enterprise 之间的连接依赖于组件集成服务 (CIS)。

有关从 Adaptive Server Enterprise 数据库到 Sybase IQ 的查询的详细信息，请参见 Adaptive Server Enterprise 核心文档集中的《组件集成服务用户指南》。

有关从 Sybase IQ 到其它数据库的查询的详细信息，请参见“[查询远程异构数据库](#)”。

连接 SQL Anywhere 和 Sybase IQ 表

当使用 BLANK PADDING OFF 构建数据库时，CHAR 数据类型在 SQL Anywhere 和 Sybase IQ 之间不兼容。如果要将字符数据用作连接键在 SQL Anywhere 和 Sybase IQ 表之间执行跨数据库连接，请使用 CHAR 数据类型并指定 BLANK PADDING ON。

注释 对于新数据库，Sybase IQ CREATE DATABASE 不再支持 BLANK PADDING OFF。此更改对现有数据库无效。可以使用 BlankPadding 数据库属性测试现有数据库的状态：

```
select db_property ( 'BlankPadding' )
```

Sybase 建议更改受 BLANK PADDING OFF 影响的任何现有列，以确保获得正确的连接结果。重新创建 CHAR 数据类型的连接列，而非 VARCHAR 数据类型的连接列。CHAR 列通常用空白填充。

查询远程异构数据库

本节概述如何使用 Sybase IQ 及组件集成服务 (CIS)。使用 CIS 可以通过 Sybase IQ 查询 Adaptive Server Enterprise 数据库和远程数据库或非关系型数据源。CIS 作为 Sybase IQ 的部分内容安装。

使用 CIS 可以按访问本地表的方式访问远程服务器上的表。CIS 在多个远程异构服务器中的表之间执行连接，并将一个表中的内容传输到支持的远程服务器。

要查询远程数据库或数据源，需要将其表映射到本地代理表。CIS 向客户端应用程序呈现代理表，如同数据是在本地存储的一样。当查询这些表时，CIS 确定实际的服务器存储位置。

❖ 要连接远程数据库，请执行下列操作：

- 1 按照《系统管理指南，卷 2》中的步骤创建代理表。
- 2 将远程表映射到代理表。
- 3 使用代理数据库名称作为每个远程表的限定名称来在 SELECT 语句中引用代理表。例如：

```
SELECT a.c_custkey, b.o_orderkey  
FROM proxy_iqdemo..cust2 a,  
iqdemo..orders b  
WHERE a.c_custkey = b.o_custkey
```

有关详细信息，请参见《系统管理指南，卷 2》中的第 4 章“访问远程数据”和第 5 章“用于进行远程数据访问的服务器类”。

用子查询替换连接

连接返回根据多个表中的数据构造的结果表。还可以使用子查询检索同一结果表。子查询仅为其它选择语句中的 SELECT 语句。这在建立更复杂和信息量更大的查询时是一种有用的工具。

例如，假设您需要一个按时间顺序排列的订单和下达这些订单的公司的列表，需要列出公司名称而不是它们的客户 ID。您可以使用如下连接获得此结果：

使用连接

要列出自 1994 年起每个订单的 order_id、OrderDate 和 CompanyName，请键入：

```
SELECT SalesOrders.ID,  
SalesOrders.OrderDate,  
Customers.CompanyName  
FROM SalesOrders  
KEY JOIN Customers  
WHERE OrderDate > '1994/01/01'  
ORDER BY OrderDate
```

ID	OrderDate	CompanyName
2131	2000-01-02	BoSox Club
2126	2000-01-03	Leisure Time
2065	2000-01-03	Bloomfields
2127	2000-01-06	Creative Customs Inc.
2135	2000-01-06	East Coast Traders
2129	2000-01-07	Hospital Gifts
2132	2000-01-08	The Pep Squad
2136	2000-01-09	Divas Design
2133	2000-01-10	The Road Side Inn
2083	2000-01-13	Pollys Custom Design

使用外连接

教程前几节中的连接的完整名称为**内连接**。

显式指定**外连接**。在本例中，还需要 GROUP BY 子句：

```
SELECT  CompanyName,
        MAX( SalesOrders.ID ),State
FROM Customers
KEY LEFT OUTER JOIN SalesOrders
WHERE State = 'WA'
GROUP BY CompanyName, State
```

CompanyName	MAX(SalesOrders.ID)	State
Its a Hit!	(NULL)	WA
Custom Designs	2547	WA

使用子查询

要列出库存不足的产品的订单项，请键入：

```
SELECT *
FROM SalesOrderItems
WHERE ProductID IN
( SELECT ID
FROM Products
WHERE Quantity < 20 )
ORDER BY ShipDate DESC
```

ID	LineID	ProductID	Quantity	ShipDate
2082	1	401	48	2001-07-09
2053	1	401	60	2001-06-30
2125	2	401	36	2001-06-28
2027	1	401	12	2001-06-17
2062	1	401	36	2001-06-17
2023	1	401	24	2001-06-09
2031	2	401	48	2001-06-02
2073	1	401	24	2001-06-02
2647	1	401	36	2001-05-26
2639	1	401	36	2001-05-19

该语句中的子查询是括在圆括号中的短语：

```
( SELECT ID
  FROM Products
 WHERE Quantity < 20 )
```

通过使用子查询，只需使用一个查询即可执行搜索，而不是使用一个查询查找库存不足的产品列表并使用另一个查询查找这些产品的订单。

子查询生成一个 `product` 表内 `id` 列中符合 `WHERE` 子句搜索条件的所有值的列表。

请牢记有关子查询的下列几点：

- 如果在构造连接时遇到问题（例如，查询使用 `NOT EXISTS` 谓词），子查询也可能会非常有用。
- 子查询只能返回一个列。
- 子查询只能用作比较参数、`IN` 或 `EXISTS` 子句。
- 不能在外连接 `ON` 子句内使用子查询。

本章提供了查询与删除性能的建议

主题	页码
导致 SQL Anywhere 进行处理的条件	31
计划查询	36
控制查询处理	40
优化删除操作	43

实现结构化查询提示

下表说明了改善查询结构的提示：

- 有些情况下，包含子查询的命令语句也可表示为连接，从而有可能加快运行速度。
- 如果对 **GROUP BY** 子句中的多列进行分组，按唯一值的数目降序列出这些列（如果可以）。这将提供最佳的查询性能。
- 与即席连接相比，连接索引通常能使连接查询的执行速度更快，不过其代价是要使用更大的磁盘空间且显著增加装载时间。但是，当连接查询不引用多表连接索引中的最大表时，或较小和较大表之间的行计数差异很大时，即席连接的性能通常优于连接索引。
- 您可以通过使用附加列改进性能，以便存储频繁计算的结果。

注释 涉及包含大量空值的列的查询比先前版本中的运行速度快。但是，在将大量空值插入到表的情况下，在该表中插入或更新数据这一过程可能会花费较长时间（同先前版本相比）。

GROUP BY 与 UNION ALL 配合使用时对查询性能的影响

为了改进装载性能，非常大的表有时被分割为多个小表，并通过在视图中使用 UNION ALL 进行访问。对于一些使用此类视图和 GROUP BY 的非常特殊的查询，Sybase IQ 优化程序能够通过将一些 GROUP BY 操作复制到此类 UNION ALL 的各个分支，以执行并行操作然后合并结果，从而增强性能。此种方法称为拆分 GROUP BY，可以减少由顶级 GROUP BY 处理的数据量，因此减少了查询处理时间。

在查询中，只有将 GROUP BY 与 UNION ALL 配合使用时才能提高性能。例如，下面这个简单的查询就受益于拆分 GROUP BY：

```
CREATE VIEW vtable (v1 int, v2 char(4)) AS
SELECT a1, a2 FROM tableA
UNION ALL
SELECT b1, b2 FROM tableB;

SELECT COUNT(*), SUM(v1) FROM vtable GROUP BY v2;
```

让我们来分析此查询，优化程序首先对 tableA 执行 COUNT(*) GROUP BY，对 tableB 执行 COUNT(*) GROUP BY，然后将结果传递至顶级 GROUP BY。顶级 GROUP BY 对这两个 COUNT(*) 结果执行 SUM 操作，以获得最终查询结果。请注意，顶级 GROUP BY 的角色会更改：顶级 GROUP BY 使用的集合为 SUM 而不是 COUNT。

针对拆分 GROUP BY 的限制

对于可通过拆分 GROUP BY 提高性能的情况和查询存在一些限制：

- 如果查询使用 UNION ALL 而不是 UNION，就可以通过拆分 GROUP BY 提高性能。下面这个查询结合使用了 GROUP BY 和 UNION，因此它不能通过 GROUP BY 分割提高性能：

```
CREATE VIEW viewA (va1 int, va2 int, va3 int,
va4 int) AS
SELECT b1, b2, b3, b4 FROM tableB
UNION
SELECT c1, c2, c3, c4 FROM tableC;

SELECT SUM(va1) FROM viewA GROUP BY va3;
```

- 如果查询中的集合不包含 **DISTINCT**，查询就可以通过拆分 **GROUP BY** 提高性能。下面这个查询结合使用了 **SUM DISTINCT**，因此它不能通过拆分 **GROUP BY** 提高性能：

```
CREATE VIEW viewA (va1 int, va2 int, va3 int,
va4 int) AS
SELECT b1, b2, b3, b4 FROM tableB
UNION ALL
SELECT c1, c2, c3, c4 FROM tableC;

SELECT SUM(DISTINCT va1) FROM viewA GROUP BY va3;
```

- 为使查询能够得益于拆分 **GROUP BY**，您的临时共享缓冲区高速缓存需具有足够的内存大小，以便存储用于处理附加 **GROUP BY** 运算符的集合信息和数据。

```
CREATE VIEW viewA (va1 int, va2 int, va3 int,
va4 int) AS
SELECT b1, b2, b3, b4 FROM tableB
UNION ALL
SELECT c1, c2, c3, c4 FROM tableC
UNION ALL
SELECT d1, d2, d3, d4 FROM tableD
UNION ALL
SELECT e1, e2, e3, e4 FROM tableE
UNION ALL
SELECT f1, f2, f3, f4 FROM tableF
UNION ALL
SELECT g1, g2, g3, g4 FROM tableG;

SELECT SUM(va1) FROM viewA GROUP BY va3;
```

在这个示例中，Sybase IQ 优化程序拆分 **GROUP BY** 并将六个 **GROUP BY** 运算符插入到查询计划中。因此，查询需要更大的临时高速缓存来存储集合信息和数据。如果系统不能分配足够的高速缓存，优化程序便不能拆分 **GROUP BY**。

如果有可用内存，您可以使用 **TEMP_CACHE_MEMORY_MB** 数据库选项来增加临时高速缓存大小。有关设置缓冲区高速缓存大小的信息，请参见第 50 页的“确定缓冲区高速缓存大小”。

- 为使查询能够得益于拆分 GROUP BY，AGGREGATION_PREFERENCE 数据库选项应设置为其缺省值 0。使用此值，Sybase IQ 优化程序可确定要向 GROUP BY 应用的最佳算法。如果 AGGREGATION_PREFERENCE 的值迫使 Sybase IQ 优化程序选择排序算法处理 GROUP BY，则查询不能得益于拆分 GROUP BY。可使用 AGGREGATION_PREFERENCE 选项覆盖优化程序为处理 GROUP BY 所做的算法选择，此时不应将其设置为 1 或 2。

拆分 GROUP BY 的示例

此例中，名为 tableA 的大表被分割为四个小表：tabA1、tabA2、tabA3 和 tabA4。使用四个小表和 UNION ALL 创建视图 unionTab：

```
CREATE VIEW unionTab (v1 int, v2 int, v3 int, v4 int) AS
SELECT a, b, c, d FROM tabA1
UNION ALL
SELECT a, b, c, d FROM tabA2
UNION ALL
SELECT a, b, c, d FROM tabA3
UNION ALL
SELECT a, b, c, d FROM tabA4;
```

Sybase IQ 优化程序将 GROUP BY 操作分割到下列查询中，改进了查询性能：

```
SELECT v1, v2, SUM(v3), COUNT(*) FROM unionTab
GROUP BY v1, v2;

SELECT v3, SUM(v1*v2) FROM unionTab
GROUP BY v3;
```

增强 ORDER BY 查询性能

可以通过对单个表查询中多列的引用使用多列 HG 索引增强 ORDER BY 查询的性能。此更改对于用户来说是透明的，但改进了查询性能。

使用多列 HG 索引，具有 ORDER BY 子句中多列的查询可能会加快运行速度。例如，如果在表 T 上用户具有多列索引 HG(x,y,z)，则此索引用于有序投影：

```
SELECT abs (x) FROM T
ORDER BY x, y
```

在上述示例中，HG 索引按排序顺序垂直投影 x 和 y。

如果 ROWID() 函数位于 SELECT 列表表达式，则也会使用多列 HG 索引。
例如：

```
SELECT rowid()+x, z FROM T
ORDER BY x,y,z
```

如果 ROWID() 存在于 ORDER BY 列表的末尾，而且该列表的列（ROWID() 除外）位于该索引内，且排序键与前导 HG 列依序匹配，则会在查询中使用多列索引。例如：

```
SELECT z,y FROM T
ORDER BY x,y,z,ROWID()
```

增强了查询内的并行度

Sybase IQ 早期版本在运行单个查询时利用多个 CPU 的能力十分有限。Sybase IQ 现在显著增加了由 Sybase IQ 查询引擎自动并行执行的运算符的类型。此功能缺省处于启用状态，且无需更改查询语法或进行特定调优。

改进了子查询性能

子查询展平是一种优化技术，使用此技术优化程序会将包含子查询的查询重新写入使用连接的查询中。Sybase IQ 会展平多个子查询但不是所有子查询。新数据库选项 SUBQUERY_FLATTENING_PREFERENCE 和 SUBQUERY_FLATTENING_PERCENT 可控制在什么情况下优化程序选择使用此优化技术。

在 Sybase IQ 15.0 中，不建议使用 FLATTEN_SUBQUERIES 选项。

使用高速缓存方法

相关子查询包含对子查询范围外的一个或多个表的引用，且相关子查询在每次被引用列中的值更改时都会进行重新执行。Sybase IQ 使用户能够为使用 SUBQUERY_CACHING_PREFERENCE 选项执行相关子查询选择高速缓存方法。

导致 SQL Anywhere 进行处理的条件

Sybase IQ 使用组件集成服务 (CIS) 查询远程服务器上的表。CIS 允许 Sybase IQ 处理不受 Sybase IQ 语义直接支持的查询。

CIS 会处理具有以下特点的查询：

- 引用用户定义的函数
- 包含某些系统函数
- 引用 Catalog 存储表

有关 Sybase IQ 和 SQL Anywhere 之间的差异的详细信息，请参见《参考：构件块、表和过程》中的附录 A “与其它 Sybase 数据库的兼容性”。

计划查询

如果您已创建正确的索引，Sybase IQ 查询优化程序通常能够以最有效的方式执行查询，有时即使您并未使用最有效的语法也可以做到这一点。但是适当的查询设计依然很重要。在您仔细计划查询时，您所做的计划可能会对速度和结果的正确性产生重大影响。

执行任何查询之前，Sybase IQ 查询优化程序会创建查询计划。Sybase IQ 使您能够通过下面几节介绍的选项来检查和影响查询计划，以帮助您评估查询。有关如何指定这些选项的详细信息，请参见《参考：语句和选项》。

注释 对于所有接受整数值的数据库选项，Sybase IQ 会截去 *option-value* 设置的小数部分，只保留整数值。例如，值 3.8 将被截断为 3。

查询评估选项

下列选项有助您评估查询计划。有关详细信息，请参见《参考：语句和选项》。

- **INDEX_ADVISOR** — 当此选项设置为 ON 时，索引顾问会在 Sybase IQ 查询计划中打印索引建议，如果未启用查询计划，则将其作为 Sybase IQ 消息日志文件中单独的消息进行打印。这些消息以字符串“Index Advisor:”开头，并且您可以使用该字符串在 Sybase IQ 消息文件中对其进行搜索和过滤。此选项以 OWNER.TABLE.COLUMN 格式输出消息，缺省情况下为 OFF。

另请参见《参考：构件块、表和过程》的第 7 章“系统过程”中的“Sp_iqindexadvice 过程”。

- **INDEX_ADVISOR_MAX_ROWS** — 用于限制由索引顾问所存储的消息数。一旦达到指定限制，INDEX_ADVISOR 就不再存储新的建议。但是它将继续为现有设备更新计数和时间戳。
- **NOEXEC** — 当设置为 ON 时，Sybase IQ 会生成查询计划，但不会执行整个查询。当 EARLY_PREDICATE_EXECUTION 选项为 ON 时，仍会执行查询的某些部分。

如果 EARLY_PREDICATE_EXECUTION 为 OFF，查询计划可能与正常情况下运行查询时很不同，因此不建议关闭该选项。

- **QUERY_DETAIL** — 当此选项以及 QUERY_PLAN 或 QUERY_PLAN_AS_HTML 都为 ON 时，Sybase IQ 会在生成查询计划时显示有关查询的其它信息。当 QUERY_PLAN 和 QUERY_PLAN_AS_HTML 为 OFF 时，则忽略此选项。
- **QUERY_PLAN** — 当设置为 ON（缺省值）时，Sybase IQ 会生成有关查询的消息。其中包括针对查询的有关使用连接索引、连接顺序及连接算法的消息。
- **QUERY_PLAN_TEXT_ACCESS** — 当打开此选项时，您可以从 Interactive SQL 客户端查看、保存和打印 IQ 查询计划。当关闭 QUERY_PLAN_ACCESS_FROM_CLIENT 时，不会对查询计划进行高速缓存，并且与查询计划相关的其它数据库选项对 Interactive SQL 客户端的查询计划显示不起任何作用。缺省情况下此选项为 OFF。
另请参见《参考：构件块、表和过程》的“HTML_PLAN 函数 [字符串]”和“GRAPHICAL_PLAN 函数 [字符串]”。
- **QUERY_PLAN_AFTER_RUN** — 当设置为 ON 时，会在查询结束运行之后输出查询计划。它允许查询计划包括其它信息，如从查询中各节点传递的实际行数。为使此选项有效，QUERY_PLAN 必须为 ON。缺省情况下此选项为 OFF。

- **QUERY_PLAN_AS_HTML** — 生成 HTML 格式的图形查询计划，从而可以通过 Web 浏览器查看该计划。节点之间的超链接使得 *.iqmsg* 文件中的 HTML 格式比文本格式更易于使用。使用 **QUERY_NAME** 选项，以便将查询名称包含在查询计划文件名称中。缺省情况下此选项为 **OFF**。
- **QUERY_PLAN_AS_HTML_DIRECTORY** — 当 **QUERY_PLAN_AS_HTML** 为 **ON** 并用 **QUERY_PLAN_AS_HTML_DIRECTORY** 指定一个目录时，Sybase IQ 会将 HTML 查询计划写入该指定目录中。
- **QUERY_PLAN_TEXT_CACHING** — 为用户提供了控制对计划进行高速缓存的资源的机制。在此选项为 **OFF**（缺省值）的情况下，不会针对该用户连接对查询计划进行高速缓存。

如果针对某个用户关闭 **QUERY_PLAN_TEXT_ACCESS** 选项，则对于来自该用户的连接，无论 **QUERY_PLAN_TEXT_CACHING** 如何设置都不会对查询计划进行高速缓存。

另请参见《参考：构件块、表和过程》中的“**GRAPHICAL_PLAN** 函数 [字符串]”和“**HTML_PLAN** 函数 [字符串]”。

- **QUERY_TIMING** — 控制对查询引擎中有关子查询及其它某些重复函数的计时统计信息的收集活动。通常该选项应为 **OFF**（缺省值），这是因为对于非常短的相关子查询来说，就性能而言每个子查询执行的计时成本可能很高。

注释 查询计划可向您的 *.iqmsg* 文件添加大量文本。当 **QUERY_PLAN** 为 **ON**（缺省值）时，尤其是如果 **QUERY_DETAIL** 为 **ON**，可能需要启用消息日志回卷或消息日志存档，以避免填满消息日志文件。有关详细信息，请参见《系统管理指南，卷 1》的第 1 章“Sybase IQ 系统管理概述”中的“消息日志回卷”一节。

查询树

优化程序可创建查询“树”来表示查询中的数据流。查询计划在 *.iqmsg* 文件中以文本格式呈现查询树，也可以提供图形形式的查询树。

查询树由节点组成。每个节点代表一个工作阶段。树上最低的节点是叶节点。每个叶节点代表查询中的一个表或预连接索引组。

在查询计划的顶部是运算符树的根。信息流从表开始向上流动，流经任何表示连接、排序、过滤、存储、集合和子查询的运算符。

使用查询计划

使用查询计划开始的最好方式是将 `QUERY_PLAN_AS_HTML` 选项设置为 `ON`。此选项将查询计划的图形版本置于与 `.iqmsg` 文件相同的目录中。您可以通过大部分 Web 浏览器查看此文件。

在 HTML 查询计划中，树中的每个节点都是连接至详细信息的超链接。每个框都被链接至上部的树。您可以单击任何节点，以便快速浏览整个计划。

授权用户现在可在基于 Java 的 Interactive SQL 计划窗口中显示查询计划。用户还可以从 Interactive SQL 保存和打印查询计划，而不是访问服务器上的 `.iqmsg` 文件或查询计划文件。

SQL 函数 `GRAPHICAL_PLAN` 和 `HTML_PLAN` 分别以 XML 和 HTML 格式返回 IQ 查询计划，作为字符串结果集。数据库选项 `QUERY_PLAN_TEXT_ACCESS` 和 `QUERY_PLAN_TEXT_CACHING` 控制新函数的行为。

采用以下方式从 Interactive SQL 计划窗口中查看：

- 执行查询并打开计划窗口。根据从“计划”选项（“工具” > “选项” > “计划”）中选择的计划类型，计划窗口显示相应计划。

仅当选择 `GRAPHICAL_PLAN` 选项时，该窗口才会显示 IQ 查询计划。其它计划会返回错误消息“Plan type is not supported”（计划类型不受支持）。

- 在 SQL 语句窗口中输入查询，并从菜单“SQL” > “获取计划”中进行选择。根据从“计划”选项（“工具” > “选项” > “计划”）中选择的计划类型，计划窗口显示相应计划。

仅当选择 `GRAPHICAL_PLAN` 选项时，该窗口才会显示 IQ 查询计划。其它计划会返回错误消息“Plan type is not supported”（计划类型不受支持）。

- 使用 SQL 函数 `GRAPHICAL_PLAN` 和 `HTML_PLAN` 返回查询计划，作为字符串结果。

要访问查询计划，请对下列查询使用 SQL 函数 `GRAPHICAL_PLAN` 和 `HTML_PLAN`：SELECT、UPDATE、DELETE、INSERT SELECT 和 SELECT INTO。

若要从 Interactive SQL 保存查询计划，请使用 `GRAPHICAL_PLAN` 或 `HTML_PLAN` 检索查询计划并使用 `OUTPUT` 语句将输出保存到文件中。

要查看保存的计划，请从 Interactive SQL 客户端菜单中选择“文件” > “打开”，导航到保存计划的目录。还可以通过选择“文件” > “打印”来打印计划窗口中显示的计划。

有关详细信息，请参见《参考：构件块、表和过程》中的“GRAPHICAL_PLAN 函数 [字符串]”和“HTML_PLAN 函数 [字符串]”。有关支持这些查询计划函数的选项，请参见《参考：语句和选项》中的“QUERY_PLAN_TEXT_ACCESS”和“QUERY_PLAN_TEXT_CACHING 选项”。

控制查询处理

任何用户都可以对用在处理特定查询上的时间量设置限制。具有 DBA 特权的用户能够给予某些用户高于其他用户的查询优先级，或者更改处理算法以影响查询处理速度。有关本节中介绍的选项的详细信息，请参见《参考：语句和选项》。

设置查询时间限制

通过设置 MAX_QUERY_TIME 选项，用户能够禁止进行长查询操作。如果查询的执行时间长于期望值，Sybase IQ 会通过显示相应的错误来停止查询。

注释 Sybase IQ 会截去 *option-value* 设置的小数部分，只保留整数值。例如，值 3.8 将被截断为 3。

设置查询优先级

队列中等待处理的查询首先按照提交查询的用户优先级顺序排队以等待运行，其次是按照查询提交的顺序进行排队。在高优先级查询全部执行完毕之前，不会执行低优先级查询。

用户通过下列选项为查询分配处理优先级。

- IQGOVERN_PRIORITY — 为队列中等待处理的查询分配数字优先级（1、2 或 3，1 为最高级）。
- IQGOVERN_MAX_PRIORITY — 允许 DBA 为用户或组设置 IQGOVERN_PRIORITY 的上限。

- `IQ_GOVERN_PRIORITY_TIME` — 如果高优先级（优先级为 1）查询在 `-iqgovern` 队列中等候的时间超过了指定时间，则允许高优先级用户开始执行查询。

若要查看查询的优先级，请查看 `sp_iqcontext` 存储过程返回的 `IQGovernPriority` 属性。

设置查询优化选项

下列选项影响查询处理速度：

- `AGGREGATION_PREFERENCE` — 控制处理集合的算法选择（`GROUP BY`、`DISTINCT`、`SET` 函数）。此选项主要供内部使用；除非您是经验丰富的数据库管理员，否则请不要使用它。
- `DEFAULT_HAVING_SELECTIVITY_PPM` — 设置查询中所有 `HAVING` 谓词的选择性，以覆盖优化程序对 `HAVING` 子句要过滤的行数所做的估计。
- `DEFAULT_LIKE_MATCH_SELECTIVITY_PPM` — 可为一般 `LIKE` 谓词设置缺省选择性，例如 `LIKE 'string%string'`，其中 `%` 是通配符。如果未提供其它选择性信息，并且匹配字符串不是以一组常量字符后跟单个通配符开头，则优化程序会使用此选项。
- `DEFAULT_LIKE_RANGE_SELECTIVITY_PPM` — 设置前导为常量字符的 `LIKE` 谓词的缺省选择性，形式如 `LIKE 'string%'`，其中匹配字符串是一组常量字符，后接单个通配符（`%`）。如果未提供其它选择性信息，优化程序会使用此选项。
- `EARLY_PREDICATE_EXECUTION` — 控制在执行连接优化前是否执行简单本地谓词。在大多数情况下，不应对其进行更改。
- `IN_SUBQUERY_PREFERENCE` — 控制处理 `IN` 子查询的算法选择。此选项主要供内部使用；除非您是经验丰富的数据库管理员，否则请不要使用它。
- `INDEX_PREFERENCE` — 为查询处理设置要使用的索引。Sybase IQ 优化程序一般会选择最佳可用索引来处理本地 `WHERE` 子句谓词和其它能够在 IQ 索引内部完成的操作。此选项用于取代供测试使用的优化程序选择；在大多数情况下，不应对其进行更改。
- `JOIN_PREFERENCE` — 控制处理连接时的算法选择。此选项主要供内部使用；除非您是经验丰富的数据库管理员，否则请不要使用它。
- `JOIN_SIMPLIFICATION_THRESHOLD` — 控制在应用任何连接优化程序简化之前要连接到一起的最少表数。通常，不需要更改此值。

- **MAX_HASH_ROWS** — 设置查询优化程序将考虑执行散列算法的最大预计行数。缺省值为 1,250,000 行。例如，如果两表之间有连接，而这两个表中参与该连接的估计行数超过该选项值，则优化程序将不会考虑散列连接。在 **TEMP_CACHE_MEMORY_MB** 值超过每用户 50MB 的系统上，需要为该选项考虑更高的值。
- **MAX_JOIN_ENUMERATION** — 设置在应用优化程序简化之后要进行连接顺序优化的表的最大数量。通常无需设置该选项。

设置用户提供的条件提示

Sybase IQ 查询优化程序使用来自可用索引的信息选择用于执行查询的适当策略。对于查询中的每个条件，优化程序将决定是否可以使用索引执行该条件，如果可以，优化程序会选择使用的索引以及相对于该表中其它条件的执行顺序。在这些决策中最重要的因素是条件的选择性：即，表的行中满足该条件的部分。

优化程序通常在没有用户干预的情况下做出决定，并且一般可提供最佳决策。但在某些情况下，优化程序可能无法在条件执行之前准确确定其选择性。通常只有以下两种条件之一成立时，才会出现这些情况：一种是条件所处的列没有适当的可用索引，另一种是条件涉及某些算术或函数表达式，因而过于复杂，使优化程序无法准确估计。

有关语法、参数和示例，请参见《参考：构件块、表和过程》的第 2 章“SQL 语言元素”中的“用户提供的条件提示”。

监控负载

通常，创建索引的目的是为了提供优化元数据以及强制唯一性和主 / 外键关系。但是，创建索引之后，DBA 面临对索引带来的优势进行定量的挑战。

通常，表是在 IQ main 存储区中创建的，用于临时存储必须由多个连接访问或在较长时间内访问的数据。您可能会遗忘这些表，然而它们却继续使用宝贵的磁盘空间。此外，数据仓库中的表的数目过大且负载非常复杂，从而无法手动分析使用情况。

因此，未使用的索引和表会浪费磁盘空间，增加备份时间，并降低 DML 性能。

Sybase IQ 提供了用于收集和分析已定义负载的统计信息的工具。DBA 可快速确定查询正在引用并应该由此而保留的数据库对象。可删除未使用的表 / 列 / 索引，以便减少浪费的空间，提高 DML 性能，并缩短备份时间。

负载监控是使用存储过程实现的，这些存储过程控制对表、列和索引信息的详细使用情况的收集和报告。这些过程是对 INDEX_ADVISOR 功能的补充，该功能生成相应消息，建议可提高一个或多个查询性能的其它列索引。添加推荐的索引之后，即可跟踪其使用情况，以确定这些索引是否值得保留。

有关负载监控过程的详细信息，请参见《参考：构件块、表和过程》中的和“sp_iqcolumnuse 过程”、“sp_iqindexadvice 过程”、“sp_iqindexuse 过程”、“sp_iqtableuse 过程”、“sp_iqunusedcolumn 过程”、“sp_iqunusedindex 过程”、“sp_iqunusedtable 过程”和“sp_iqworkmon 过程”。

另请参见《参考：语句和选项》中的“INDEX_ADVISOR 选项”。

优化删除操作

Sybase IQ 在三种可能的算法中选择最佳算法处理带有 HG 和 WD 索引的列的删除操作。

HG 删除操作

Sybase IQ 在三种算法中选择任意一种算法来处理带有 HG (High_Group) 索引的列的删除操作：

- **小型删除** 当从非常少的组中删除行时，少量删除会表现出最佳性能。当只删除 1 行或删除操作对带有 HG 索引的列使用相等谓词时，通常会选择它。少量删除算法可以随机访问 HG。最坏情况下，I/O 也会与访问的组数成正比。
- **中型删除** 当从若干组中删除行，而组数又足够少使得所访问的 HG 页面不多时，中型删除会表现出最佳性能。中型删除算法可提供对 HG 的有序访问。最坏情况下，I/O 也不会超过索引页面的数量。中型删除会因为对要删除的记录进行排序而产生新的开销。

- **大型删除** 从大量组中删除行时，大型删除会表现出最佳性能。大型删除按顺序扫描 HG，直至删除所有行为止。最坏情况下，I/O 也不会超过索引页面的数量。大型删除也为并行操作，但其并行操作受索引内部结构和待删除记录所属组的分布的约束。HG 列上的范围谓词可用于缩小大型删除的扫描范围。

HG 删除的开销

在 Sybase IQ 12.6 之前，HG 删除开销模型只考虑了最坏情况下的 I/O 性能，因此大多数情况下首选的是大型删除。当前的开销模型将考虑多种因素，包括 I/O 开销、CPU 开销、可用资源、索引元数据、并行处理和查询中可用的谓词。

在带 HG 索引的列上指定谓词，会大大改善开销。若要让 HG 开销选择大型删除以外的算法，必须能够确定受删除影响的不同的值（组）的数量。离散值个数初始假定为索引组数和被删除行数中的较小者。谓词可以提供更完善、更精确的离散值个数估计值。

当前开销不考虑范围谓词对大型删除的影响。这会导致在采用大型删除将更快的情况下选择中型删除。在这种情况下，您可以根据需要按照下一节所述方法强制使用大型删除算法。

使用 HG 删除性能选项

您可以使用 HG_DELETE_METHOD 选项来控制 HG 删除的性能。

通过 HG_DELETE_METHOD 选项指定的参数值会强制使用指定的删除算法，如下所示：

- 1 = 少量删除
- 2 = 大型删除
- 3 = 中型删除

有关 HG_DELETE_METHOD 数据库选项的详细信息，请参见《参考：语句和选项》的第 2 章“数据库选项”中的“HG_DELETE_METHOD 选项”。

WD 删除操作

Sybase IQ 在三种算法中选择任意一种算法来处理带有 WD (Word) 索引的删除操作。

- **小型删除** 当被删除行中几乎不包含任何非重复字时，并不需要访问很多 WD 页面，因此 WD 少量删除会表现出最佳性能。WD 少量删除算法执行 WD 有序访问。最坏情况下，I/O 也不会超过索引页面的数量。少量删除包含对要删除记录中的文字和记录 ID 进行排序的开销。
- **中型删除** WD 中型删除是 WD 少量删除的变换形式，适用于与少量删除相同的条件，即，当被删除行几乎不包含任何非重复字时。WD 中型删除仅对要删除记录中的文字进行排序。排序为并行操作，其并行操作受文字数和 CPU 可用线程数的约束。对于文字索引，中型删除方法通常比少量删除更快。
- **大型删除** 当被删除行包含大量非重复文字时，需要访问索引中的大量“组”，此时 WD 大型删除会表现出最优性能。大型删除按顺序扫描 WD，直至删除所有行为止。最坏情况下，I/O 也不会超过索引页面的数量。大型删除也为并行操作，但其并行操作受索引内部结构和待删除记录所属组的分布的约束。

WD 删除的开销

WD 删除的开销模型会考虑很多因素，包括 I/O 开销、CPU 开销、可用资源、索引元数据和并行。

您可以使用 `WD_DELETE_METHOD` 数据库选项来控制 WD 删除的性能。

使用 WD 删除性能选项

通过 `WD_DELETE_METHOD` 选项指定的参数值会强制使用指定的删除算法，如下所示：

- 0 = 表示开销模型选定的中型删除或大型删除
- 1 = 少量删除
- 2 = 大型删除
- 3 = 中型删除

有关 `WD_DELETE_METHOD` 数据库选项的详细信息，请参见《参考：语句和选项》的第 2 章“数据库选项”中的“`WD_DELETE_METHOD` 选项”。

TEXT 删除操作

Sybase IQ 在两种算法中选择任意一种算法来处理带有 TEXT 索引的列的删除操作：

- **小型删除** 当被删除行中几乎不包含任何非重复字时，并不需要访问很多 TEXT 页面，因此 TEXT 少量删除会表现出最佳性能。TEXT 少量删除算法执行 TEXT 有序访问。最坏情况下，I/O 也不会超过索引页面的数量。少量删除包含对要删除记录中的文字和记录 ID 进行排序的开销。
- **大型删除** 当被删除行包含大量非重复文字时，需要访问索引中的大量“组”，此时 TEXT 大型删除会表现出最佳性能。大型删除按顺序扫描 TEXT，直至删除所有行为止。最坏情况下，I/O 也不会超过索引页面的数量。大型删除也为并行操作，但其并行操作受索引内部结构和待删除记录所属组的分布的约束。

TEXT 删除开销

TEXT 删除的开销模型会考虑很多因素，包括 I/O 开销、CPU 开销、可用资源、索引元数据和并行。

您可以使用 TEXT_DELETE_METHOD 数据库选项来控制 TEXT 删除性能。

使用 TEXT 删除性能选项

通过 TEXT_DELETE_METHOD 选项指定的参数值会强制使用指定的删除算法，如下所示：

- 0 = 表示开销模型选定的中型删除或大型删除
- 1 = 少量删除
- 2 = 大型删除
- 3 = 中型删除

有关 TEXT_DELETE_METHOD 数据库选项的详细信息，请参见《非结构化数据分析》的第 2 章“TEXT 索引和文本配置对象”中的“TEXT_DELETE_METHOD 选项”。

管理系统资源

本章介绍 Sybase IQ 使用内存、磁盘 I/O 和 CPU 的方式，以及这些因素之间的关系。它还解释了 DBA 如何通过调整资源使用来调优性能。

本章中的建议是通用的。您需要调整它们，以适应具体的硬件和软件配置。有关针对每个平台的建议，请参见其《安装和配置指南》。

主题	页码
“性能术语简介”	47
“性能设计”	48
“内存使用概述”	48
“进程线程模型”	62
“平衡 I/O”	63
“用于调优资源使用的选项”	69
“改进资源使用的其它方式”	72
“索引提示”	74
“管理数据库大小和结构”	75
“使用 UNION ALL 视图以便更快装载”	78
“网络性能”	81

性能术语简介

性能是对计算机化业务应用程序或运行于相同环境中的多个应用程序的效率的度量。它通常以 *响应时间* 和 *吞吐量* 进行度量。

响应时间是完成单个任务所用的时间。影响它的因素是：

- 减少争用和等待时间，特别是磁盘 I/O 等待时间
- 使用更快的组件
- 减少需要资源的时间（提高并发性）

吞吐量是指在固定时间段内完成的工作量。吞吐量通常以每秒的事务数 (tps) 进行度量，也可以按每分钟、每小时、每天等进行度量。

性能设计

大多数性能增益源自良好的数据库设计、透彻的查询分析以及妥当的索引编排。最大的性能增益可以通过建立良好的设计和选择正确的索引策略来实现。

其它考查事项（如硬件和网络分析）可确定安装中的瓶颈。

有关详细信息，请参见第 3 章“优化查询和删除”。

内存使用概述

Sybase IQ 出于以下几个目的使用内存：

- 解析查询时从磁盘读取数据的缓冲区
- 从平面文件中装载时从磁盘读取数据的缓冲区
- 用于管理连接、事务、缓冲区和数据库对象的开销

后面的几节解释操作系统如何支持 Sybase IQ 的内存使用，Sybase IQ 如何为各种目的而分配内存，您如何调整内存分配以获得更好的性能，以及您可能需要做什么来配置操作系统以便 Sybase IQ 有足够的可用内存。

分页增加可用内存

当系统没有足够内存时，性能会严重降低。如果发生这种情况，则需要找到某种方式，使更多内存可用。与任何 RDBMS 软件一样，Sybase IQ 需要很多内存。分配给 Sybase IQ 的内存越多越好。

但是，系统的内存数量始终存在固定限制，因此有时操作系统只能将一部分数据放在内存中，而将其余部分放在磁盘上。如果操作系统为了满足一次内存请求而必须转向磁盘并检索任何数据，则这种情况称为分页或交换。好的内存管理的主要目标就是避免分页或交换，或者使其最大程度减少分页或交换。

最频繁使用的操作系统文件是交换文件。当内存用完时，操作系统就会将内存页交换到磁盘，以便为新数据腾出空间。当再次调用被交换的页时，就会交换其它页，并返回所需的内存页。对于有很高磁盘使用率的用户，这会非常耗时。通常，应当尝试组织内存，以避免交换，从而最大程度减少对操作系统文件的使用。有关配置内存以最大程度减少交换的信息，请参见第 58 页的“特定于平台的内存选项”。

为了最大程度利用物理内存， Sybase IQ 对*所有*数据库读取和写入均使用缓冲区高速缓存。

注释 您的磁盘上的交换空间必须至少有足够多的空间，才能容纳所有物理内存。

监控交换的实用程序

可以使用 UNIX `vmstat` 命令、UNIX `sar` 命令或 Windows 任务管理器来获取正在运行的进程数和页交换次数的统计信息。使用此信息可以发现系统是否分页过多。然后进行任何必要的调整。可能需要将交换文件放在特殊的快速磁盘上。

有关 `vmstat` 输出的示例，请参见“在 UNIX 系统上监控分页”。

服务器内存

Sybase IQ 为缓冲区、事务、数据库和服务器分配堆内存。也许还会使用共享内存，但数量上少得多。

在操作系统级别， Sybase IQ 服务器内存由堆内存组成。在大多数情况下，您不需要关注 Sybase IQ 所使用的内存是堆内存还是共享内存。所有内存分配都会自动处理。但是，在运行 Sybase IQ 之前，可能需要确保正确配置操作系统内核以使用共享内存。有关详细信息，请参见适用于您的平台的《安装和配置指南》。

管理用于 Multiplex 的内存

Multiplex 中的每个服务器都可以在其自己的主机上，也可以与其它服务器共享主机。相同系统上的两个或更多服务器不会比处理相同负载的单个组合服务器消耗更多 CPU 时间，但分开的服务器可能比单个组合服务器需要更多物理内存，因为每个服务器都不会与其它任何服务器共享所使用的内存。

注销进程影响共享内存

警告！ 在 UNIX 系统上终止进程可能导致留下信号或共享内存而不是将其自动清理掉。在 UNIX 中关闭 Sybase IQ 服务器的正确方式是通过 `stop_iq` 实用程序进行关闭，《系统管理指南，卷 1》的第 2 章“运行 Sybase IQ”中的“停止数据库服务器”一节对此进行了说明。有关异常退出后使用 `ipcs` 和 `ipcrm` 执行清理的信息，请参见《系统管理指南，卷 1》中的第 14 章“故障排除提示”。

管理缓冲区高速缓存

Sybase IQ 对用于缓冲区高速缓存的内存量超出了用于任何其它用途的内存量。Sybase IQ 有两个缓冲区高速缓存，一个用于 IQ 存储，另一个用于临时存储。它将这两个缓冲区高速缓存用于所有数据库 I/O 操作——分页、插入数据库以及备份和还原。只要数据在内存中，它就会存储在两个高速缓存的某一个之中。所有用户连接都共享这些缓冲区高速缓存。Sybase IQ 会一直跟踪与每个连接关联的数据。

阅读下面几节，了解有关管理缓冲区高速缓存的详细信息：

- 有关如何计算内存要求的信息，请参见“确定缓冲区高速缓存大小”。
- 有关了解缓冲区高速缓存大小后如何设置它们的信息，请参见“设置缓冲区高速缓存大小”。

确定缓冲区高速缓存大小

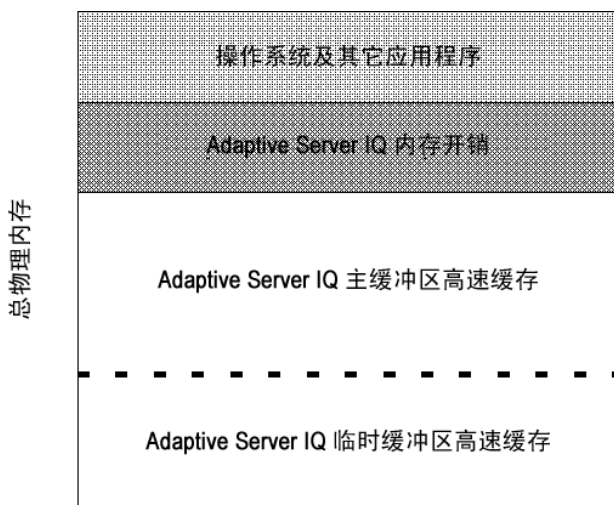
为 IQ 存储和临时存储指定的缓冲区高速缓存大小将基于几个因素而变化。对于大多数数据库来说，缺省值（主高速缓存为 16MB，临时高速缓存为 12MB）太低。应用程序所需的实际值取决于：

- 系统上的物理内存总量
- Sybase IQ、操作系统和其它应用程序执行其任务时需要此内存的量
- 是否正在执行装载、查询或同时执行这两个操作
- schema 配置和查询负载

阅读以下几节，了解确定站点最佳设置的准则。

下图显示了缓冲区高速缓存和其它内存使用量之间的关系。

图 4-1: 缓冲区高速缓存与物理内存相对比



以下几节更详细地介绍了每个部分，并提供帮助您确定每个部分需要多少内存的准则。

操作系统和其它应用程序

对于不同平台和系统使用方式，此内存数量将各不相同。例如，UNIX 文件系统会比 UNIX 裸分区执行更多的文件缓冲，因此操作系统有更高的内存要求。大多数操作系统会将大部分可用内存用于文件系统缓冲。您应了解针对特定操作系统的缓冲策略以避免过度分配内存。

此外，与 Sybase IQ 一起运行的其它应用程序（例如查询工具）都有自己的内存需要。有关这些内存要求的信息，请参见应用程序和操作系统文档。

Sybase IQ 内存开销

在确定操作系统和其它应用程序使用多少物理内存之后，就可以计算出 Sybase IQ 需要多少剩余内存来执行其任务。以下几节介绍影响此开销的因素。

裸分区与文件系统

对于 UNIX 系统，使用文件系统而不是裸分区的数据库可能需要剩余内存的另外 30% 来处理操作系统的文件缓冲。在 Windows 上，应当通过将 OS_FILE_CACHE_BUFFERING 设置为 'OFF'（对于新数据库，为缺省值）来禁用文件系统高速缓存。有关详细信息，请参见适用于具体平台的《安装和配置指南》。

多用户数据库访问

对于数据库的多用户查询，Sybase IQ 需要对每个“活动”用户使用大约 10MB。活动用户被定义为同时访问或查询数据库的用户。例如，30 个用户可能连接到 Sybase IQ，但仅 10 个左右用户可能正在任何时间同时频繁使用数据库。

用于线程堆栈的内存

处理线程需要少量的内存。您使用的 Sybase IQ 处理线程越多，需要的内存就越多。-iqmt 服务器开关控制 Sybase IQ 的线程数。-iqtss 和 -gss 服务器开关控制为每个线程分配的堆栈内存量。为 IQ 堆栈分配的总内存量大致等于：

$(-gn * (-gss + -iqtss)) + (-iqmt * -iqtss)$ 。

如果拥有大量用户，则处理线程所需的内存将增加。-gn 开关控制数据库服务器可以并发执行的任务（用户和系统请求）数。-gss 开关在一定程度上控制执行这些任务的服务器执行线程的堆栈大小。IQ 使用以下公式计算这些工作线程的堆栈大小： $(-gss + -iqtss)$ 。

线程总数（-iqmt 加 -gn）不得超过平台允许的数目。有关详细信息，请参见《实用程序指南》中的第 1 章“运行数据库服务器”。

其它内存使用

所有命令和事务都会使用一些内存。以下操作是除了前面讨论的内存使用以外最重要的内存使用：

备份。 用于备份的虚拟内存数量是在创建数据库时所指定的 IQ PAGE SIZE 函数。它大约是 $2 * \text{CPU 数} * 20 * (\text{IQ PAGE SIZE}/16)$ 。在某些平台上，通过在 BACKUP 命令中调整 BLOCK FACTOR，可能能够提高备份性能，但增加 BLOCK FACTOR 也会增加所使用的内存数量。请参见《系统管理指南，卷 1》的第 12 章“数据备份、恢复和存档”中的“增加备份过程中占用的内存”一节。

数据库检验和修复。 在检查整个数据库时，`sp_iqcheckdb` 过程将在启动任何处理之前打开所有 Sybase IQ 表及其各自的字段和索引。根据 Sybase IQ 表的个数以及这些表中的累积列数和索引数，`sp_iqcheckdb` 可能需要非常少或非常多的虚拟内存。要限制需要的内存数量，请使用 `sp_iqcheckdb` 选项，以检查或修复单个索引或表。

删除泄漏块。 删除漏洞操作也需要打开所有 Sybase IQ 表、文件和索引，因此它会使用像 `sp_iqcheckdb` 在检查整个数据库时所使用的那么多虚拟内存。它使用 Sybase IQ 临时缓冲区高速缓存来跟踪所使用的块。

Sybase IQ 主要和临时缓冲区高速缓存

在确定 Sybase IQ 需要多少内存开销之后，必须决定如何在主 Sybase IQ 和临时缓冲区高速缓存之间拆分剩余的内存。在图 4-1 中划分两个区域的虚线指示此拆分基于几个因素可能对不同数据库有所不同。

与大多数其它数据库不同，Sybase IQ 的一般准则是将 40% 分给主缓冲区高速缓存，而将 60% 分给临时缓冲区高速缓存。然而，此准则只是一个开始。当某些操作（如具有涉及 HG 索引的大型排序合并连接或插入的查询）可能需要临时缓冲区高速缓存大于 `main` 高速缓存，则其它应用程序可能有不同需求。

注释 这些准则假定系统上每次有一个活动数据库（即任何 Sybase IQ 用户一次只访问一个数据库）。如果有多个活动数据库，则需要在期望使用的数据库中进一步拆分剩余的内存。

Sybase 强烈建议您首先了解此处提供的一般准则，并通过使用 Sybase IQ 的监控工具（如第 101 页的“[监控缓冲区高速缓存](#)”中所述）和适用于您的平台的《[安装和配置指南](#)》中所述的任何特定工具，观察其性能。

缓冲区高速缓存和物理内存

用于 Sybase IQ 主和临时缓冲区高速缓存的内存总量，加上 Sybase IQ 内存开销，以及用于操作系统和其它应用程序的内存，不得超过系统的物理内存。

几乎在所有情况下，8MB 的缺省临时缓冲区高速缓存都是过低的。为了获得最佳性能，请尽可能多地为 IQ 的主缓冲区高速缓存和临时缓冲区高速缓存分配内存。例如，如果计算机上有 4GB 的物理内存可供 Sybase IQ 使用，则可拆分该内存量供主缓冲区高速缓存和临时缓冲区高速缓存使用。

注释 在某些 UNIX 平台上，可能需要设置其它服务器开关，以便使更多内存可用作缓冲区高速缓存。有关详细信息，请参见第 58 页的“[特定于平台的内存选项](#)”。

其它注意事项

基于使用方式，Sybase IQ 缓冲区高速缓存大小可能对于不同数据库有所不同。若要获取最大性能，您需要在插入、查询数据库和混合使用之间更改设置。但是，在混合使用环境中，为了重置缓冲区高速缓存选项而要求所有用户退出数据库并不始终可行。在这些情况下，可能需要照顾到装载或查询性能。

缓冲区高速缓存和内存开销准则也可能对于不同平台而有所不同。有关任何其它问题，请参见《安装和配置指南》。

设置缓冲区高速缓存大小

缺省情况下，Sybase IQ 分别将主和临时缓冲区高速缓存的大小设置为 16MB 和 12MB。大多数应用程序都需要更高的值（受物理内存总量限制）。请参见前面几节，以确定系统的正确设置。

知道需要什么设置以后，请使用表 4-1 中所述的选项设置缓冲区高速缓存大小。

表 4-1: 更改缓冲区高速缓存大小的设置

方法	何时使用它	设置一直有效的时间	有关详细信息，请参见
-iqmc 和 -iqtc 服务器开关	推荐的方法。当数据库和服务器未运行时设置高速缓存大小。允许高速缓存大小 >4GB。 对于 64 位平台，或在将高速缓存大小数据库选项设置为超过系统容量的情况下，尤其有用。	从服务器启动时到服务器停止 必须重新启动服务器才能更改缓冲区高速缓存大小。-iqmc 和 -iqtc 服务器启动选项仅在服务器正在运行时保持有效，因此每次重新启动服务器时都需要包括它们。	《实用程序指南》的第 1 章“运行数据库服务器”中的“启动数据库服务器”一节。

指定页大小

创建数据库时，可设置 Sybase IQ 页大小。该参数与缓冲区高速缓存的大小配合使用可影响该数据库的内存使用和磁盘 I/O 吞吐量。

注释 页大小无法更改，并且决定某些数据库对象的大小上限以及是否可以使用 LOB 功能。

设置页大小

Sybase IQ 以页为单位与内存双向交换数据。创建数据库时，应为 Catalog 存储和 IQ 存储指定单独的页大小。临时存储的页大小与 IQ 存储相同。

有关最佳性能的推荐 Sybase IQ 页大小，请参见《系统管理指南，卷 1》的第 5 章“使用数据库对象”中的“选择 IQ 页大小”。

因为 Catalog 存储帐户仅占 I/O 的很小一部分，所以 Catalog 存储的页大小对性能不会产生实际影响。缺省值 4096 字节应当足够。

IQ 页大小确定其它两个性能因素：缺省 I/O 传输块大小和数据库最大数据压缩。这些因素将在后面的几节中讨论。

块大小

所有 I/O 均以块为单位发生。在创建 Sybase IQ 数据库时，将设置这些块的大小；除非重新创建数据库，否则不能更改它。缺省情况下，IQ 页大小确定 I/O 传输块大小。例如，128KB 的缺省 IQ 页大小导致 8192 字节的缺省块大小。通常，Sybase IQ 使用此缺省块大小与页大小的比率，但它还会考虑其它因素。

缺省块大小应当导致大多数系统的 I/O 传输率和磁盘空间使用量达到最佳平衡。但是，与性能相比，它更有益于节省空间。如果缺省块大小表现不佳，则可以按照每页中不能有少于 2 个和多于 16 个块的约束，将它设置为 4096 和 32,768 之间的 2 的任何次幂。在某些情况下，可能需要显式设置块大小：

- 对于使用磁盘阵列的裸磁盘安装，更大的块可能获得更好的磁盘空间使用性能。
- 对于文件系统安装，若要通过磁盘空间优化性能，IQ 块大小应当大于或等于操作系统的本机块大小（如果有）。如果 IQ 块大小与文件系统的块大小匹配，则可能获得更好的 I/O 率。

表 4-2 显示每个 IQ 页大小的缺省块大小。

表 4-2: 缺省块大小

IQ 页大小 (KB)	缺省块大小 (字节)
64	4096
128 (新数据库的缺省值)	8192
256	16384
512	32768

数据压缩

Sybase IQ 在磁盘上存储数据时，将压缩所有数据。数据压缩会减少磁盘空间要求，还会提高性能。压缩量基于 IQ 页大小自动确定。

节省内存

如果计算机没有足够内存，为了节省内存，可以尝试以下调整措施。

减少缓冲区高速缓存设置

通过减少缓冲区高速缓存大小，可能能够节省内存。请记住，如果缓冲区高速缓存减少得太多，则由于缓冲区不足，可能使数据装载或查询效率降低或无法完成。

为大量用户进行优化

Sybase IQ 在 32 位平台上最多处理 200 个用户连接，在 64 位平台上最多处理 1000 个用户连接。要在 64 位系统上支持最大用户数，则可能需要调整操作系统参数和 `start_iq` 服务器参数。有关建议，请参见《安装和配置指南》及后面几节。

更改 Sybase IQ 命令行选项以满足大量用户

以下 `start_iq` 开关影响有大量用户的操作：

`-gm #_connections_to_support`

`-iqgovern #_ACTIVE_queries_to_support`

`-gn` 数据库服务器可以并发执行的任务（用户和系统请求）数

- `-c catalog_store_cache_size`
- `-ch size`
- `-cl size`
- `-gm` 这是服务器将支持连接总数。根据统计，其中某些应当已连接并处于空闲状态，而其它的部分则已连接并频繁使用数据库。
- `-iqgovern` `-iqgovern` 值限制同时执行的最大查询数。如果已提交查询的用户数超过了 `-iqgovern` 限制，则新查询将排队，直到某个活动查询完成。
- `-iqgovern` 的最佳值取决于查询的性质、CPU 数和 Sybase IQ 缓冲区高速缓存的大小。缺省值为 $2 * \text{numCPU} + 10$ 。在有很多连接用户的情况下，您可能发现将此选项设置为 $2 * \text{numCPU} + 4$ 可以提供更好的吞吐量。
- `-gn` `-gn` 的正确值取决于 `-gm` 的值。`start_iq` 实用程序将计算 `-gn` 并进行相应设置。将 `-gn` 设置得太低会阻止服务器正确操作。建议不要将 `-gn` 设置为超过 480。
- `-c` Catalog 存储缓冲区高速缓存还是 Catalog 存储的常规内存池。若要以 MB 为单位指定，请使用 `-c nM` 格式，例如 `-c 64M`。Sybase 建议使用下列值：

表 4-3: Catalog 缓冲区高速缓存设置

针对此数目的用户	在这些平台上	将 <code>-c</code> 设置为此最小值或更高
最高 1000	仅 64 位	64 MB
最高 200	64 位	48MB (64 位系统的 <code>start_iq</code> 缺省值) ; 更多用户可能受益于 64MB
最高 200	32 位	32MB (32 位系统的 <code>start_iq</code> 缺省值)

例如，在某些情况下，标准目录高速缓存大小可能太小，无法适合某些需要很多解析的查询。在此类情况下，设置 `-cl` 和 `-ch` 也许能有所帮助。例如，在 32 位平台上，尝试这些设置

```
-cl 128M
-ch 256M
```

不得将 `-c` 和 `-ch` 或 `-cl` 用于同一配置文件或命令行中。相关信息，请参见 `-ch cache-size` 选项。

警告！若要显式控制目录存储的高速缓存大小，必须在配置文件 (.cfg) 中或用于启动服务器的 UNIX 命令行上执行以下操作之一（但不要同时执行这两项操作）：

- 设置 -c 参数
 - 使用 -ch 和 -cl 参数为目录存储高速缓存大小设置具体上限和下限
- 如不按上述说明指定参数组合，则可能产生意外结果。
-

-iqmt

如果对 -gm 设置来说设置的 -iqmt 太低，则可能会导致线程匮乏。请参见《实用程序指南》的第 1 章“运行数据库服务器”中的“启动数据库服务器”一节。

为大量用户增加 Sybase IQ 临时空间

您可能需要增加临时 dbspace 以满足更多用户。

新连接和现有连接的相对优先级

如果 Sybase IQ 正忙于处理已连接的用户，则响应新连接请求的速度可能会变慢。在极端情况（例如，测试脚本在服务器忙于执行插入时触发数百个循环连接）下，新连接可能会出现连接请求超时的情况。在此情况下，服务器可能看似已停机，而实际上它只是非常繁忙。遇到此行为的用户应尝试再次连接并应考虑增加连接超时参数。

特定于平台的内存选项

在所有平台上，Sybase IQ 均为四个主要目的而使用内存：

- 主缓冲区高速缓存
- 临时缓冲区高速缓存
- Sybase IQ 内存开销（包括线程堆栈）
- 装载缓冲区

有关 Sybase IQ 内存使用情况的图表，请参见第 51 页的图 4-1。

在所有 64 位平台上，可用内存总量实际上是无限的。唯一限制的是系统的虚拟内存。

有关 HP-UX 系统上的性能调优提示，请参见适用于该平台的《安装和配置指南》。

在 32 位平台上限制适用；有关详细信息，请参见下表。

表 4-4：32 位平台上的总计可用内存

平台	总计可用内存
RedHat Linux 2.1	大约有 1.7GB 可供 Sybase IQ 使用
RedHat Linux 3.0	大约有 2.7GB 可供 Sybase IQ 使用
Windows 2000/2003/XP ^a	2.75GB 可用于 Sybase IQ

^a 您需要 Windows 2000 Advanced Server 或 Datacenter Server、Windows Server 2003 Standard、Enterprise 或 Datacenter Edition 或者 Windows XP Professional，才能获得这么多内存，并且必须设置 /3GB 开关。如果没有开关，则限制为 2GB。此数量是可用于进程的总计内存。在 Windows 服务器上，即使有 /3GB 设置，缓冲区高速缓存的总计大小也不得超过 2GB。有关详细信息，请参见适用于 Windows 的《安装和配置指南》。

由于在 Sybase IQ 服务器上使用了虚拟内存模式，虚拟内存碎片可能会导致 Windows 平台上的处理工作大幅增加。为了减少发生此情况的可能性，Sybase IQ 支持在 Windows XP 和 Windows Server 2003 上使用 Microsoft 的低碎片堆 (LFH)。

有关 Windows 平台上的更多性能调优提示，请参见第 6 章“在 32 位 Windows 系统上调优服务器”。

仅对于 UNIX 系统，Sybase IQ 提供了可以帮助您管理内存的两个命令行选项。

连系内存池

在 HP 和 Sun 平台上，可以指派指定数量的内存作为“连线”内存。连系内存是锁定到物理内存中的共享内存。内核不能从物理内存中对该部分内存进行分页。

当其它应用程序正同时运行于相同计算机上时，连系内存可能提高 Sybase IQ 性能。但是，为 Sybase IQ 分配专用连系内存会使它对该计算机上的其它应用程序不可用。

若要仅在这些 UNIX 平台上创建“连线”内存池，请指定 `-iqwmem` 命令行开关，以指示连系内存的 MB 数。（除在 Sun 上以外，您必须是用户 `root`，才能设置 `-iqwmem`。）在 64 位平台上，`-iqwmem` 的唯一上限是计算机的物理内存。

例如，在有 14GB 内存的计算机上，您可能能够留出 10GB 连系内存。为此，请指定：

```
-iqwmem 10000
```

警告！ 仅当您有足够内存可专用于为此目的而指定的数量时，才使用此开关。否则，可能导致性能大幅下降。

注释 对于此版本：

- 在 Sun Solaris 上，-iqwmem 始终提供连系内存。
 - 在 HP 上，如果您作为 root 启动服务器，则 -iqwmem 提供连系内存。如果启动服务器时您不是 root，则它提供非连系内存。此行为可能在未来版本中有更改。
-

其它应用程序和数据库的影响

请记住，用于服务器的内存来自由所有应用程序和数据库使用的内存池。如果尝试同时在相同计算机上运行多个服务器或多个数据库，或者有其它正在运行的应用程序，则可能需要减少服务器所请求的内存数量。

还可以发出 UNIX 命令 `ipcs -mb` 以查看实际段数。

HP 内存问题疑难解答

如果在 HP-UX 上有内存问题，请检查 `maxdsiz_64bit` 内核参数的值。此参数限制在 64 位 HP 处理器上可用于 Sybase IQ 的虚拟内存数量。有关建议值，请参见《安装和配置指南》。

控制文件系统缓冲

仅在 Sun Solaris UFS、Linux、Linux IBM、AIX 和 Windows 文件系统上，可以控制是否打开文件系统缓冲。如果关闭文件系统缓冲，则将数据副本从文件系统缓冲区高速缓存保存到 IQ 的主缓冲区高速缓存。通常，这样做会减少分页，并因此提高性能。但有一个例外：如果数据库的 IQ 页大小小于文件系统的块大小（通常仅在测试情况下），那么，尤其是在多用户操作期间，关闭文件系统缓冲可能降低性能。

对于最近创建的 Sybase IQ 数据库，缺省情况下会关闭文件系统缓冲。

若要对现有数据库的 IQ 主 `dbspace` 禁用文件系统缓冲，请发出以下语句：

```
SET OPTION "PUBLIC".OS_FILE_CACHE_BUFFERING = OFF
```

若要对现有数据库的 IQ 临时 `dbspace` 禁用文件系统缓冲，请发出以下语句：

```
SET OPTION "PUBLIC".OS_FILE_CACHE_BUFFERING_TEMPDB = OFF
```

尝试使用此选项以确定不同情况的最佳设置。只能为 PUBLIC 组设置此选项。数据库必须关闭并重新启动，才能使更改生效。

此直接 I/O 性能选项仅在 Sun Solaris UFS、Linux、Linux IBM、AIX 和 Windows 文件系统中可用。此选项对于 HP-UX 和 HP-UXi 不起作用，并且不影响裸磁盘上的数据库。在 Linux 上，直接 I/O 在内核版本 2.6.x 中受支持。

要在 Linux 内核版本 2.6 和 AIX 上启用直接 I/O，还要将环境变量 `IQ_USE_DIRECTIO` 设置为 1。缺省情况下，直接 I/O 在 Linux 内核版本 2.6 和 AIX 上处于禁用状态。`IQ_USE_DIRECTIO` 对 Sun Solaris 和 Windows 不起作用。

注意

- 在 Linux 内核版本 2.4 上，Sybase IQ 不支持直接 I/O。如果在 Linux 内核版本 2.4 上设置了 `IQ_USE_DIRECTIO` 环境变量，则 Sybase IQ 服务器不启动。报告错误 “Error:Invalid Block I/O argument, maybe <pathname> is a directory, or it exceeds maximum file size limit for the platform, or trying to use Direct IO on unsupported OS”。
- Solaris 没有内核参数来约束其文件系统缓冲区高速缓存的大小。一段时间以后，文件系统缓冲区高速缓存将增长并转移 IQ 缓冲区高速缓存页，从而导致过度的操作系统分页活动，并使 Sybase IQ 性能降低。
- Windows 可以通过耗用文件系统来调整分页算法的偏差，以使应用程序受益。为了提高 Sybase IQ 性能，建议使用此偏差。有关详细信息，请参见第 6 章 “在 32 位 Windows 系统上调优服务器”。

另请参见《参考：语句和选项》的第 2 章 “数据库选项” 中的 “`OS_FILE_CACHE_BUFFERING` 选项”。

获取更多内存的其它方式

在某些环境中，可能能够调整其它选项，以使更多内存可用于 Sybase IQ。

启用 Java 的数据库的选项

`SET OPTION` 命令的 `JAVA_HEAP_SIZE` 选项可以设置基于每个连接而分配给 Java 应用程序的那部分内存的最大大小（字节）。通常，基于每个连接的内存分配由所分配的 Java 变量和 Java 应用程序堆栈空间的用户工作集组成。Java 应用程序在连接上执行时，每个连接分配会来自数据库服务器的固定高速缓存，所以，防止无法控制的 Java 应用程序占用过多内存是很重要的。

进程线程模型

Sybase IQ 使用操作系统内核线程，以获得最佳性能。线程可以在用户级别和在内核级别找到。轻型进程是内核所支持的基础控制线程。操作系统决定哪些轻型进程 (LWP) 应当运行在哪个处理器上，以及何时运行。它不知道用户线程是什么，但如果用户线程正在等待或能够运行，则它会知道。

操作系统内核将 LWP 调度到 CPU 资源上。它使用它们的调度类和优先级。每个 LWP 均由内核单独进行调度，执行独立的系统调用，调用独立的页错误并在多处理器系统上并行运行。

单个、高度线程化的进程服务于所有 Sybase IQ 用户。Sybase IQ 基于该连接所完成的处理类型、可用线程总数和各种选项设置，将不同数目的内核线程分配给每个用户连接。

线程不足错误

没有足够的服务器线程来启动您已经发出的查询时，您将得到错误：

```
Not enough server threads available for this query
```

这种情况可能是临时性的。当其它某些查询完成时，线程将变为可用，在您下一次发出查询时，查询可能成功。如果这种情况持续存在，则可能需要重新启动服务器，并指定更多 Sybase IQ 线程，如下一节所述。还有可能是 `-iqmt` 为连接数设置的值过低。

用于管理线程使用的 Sybase IQ 选项

Sybase IQ 提供了以下选项，帮助您管理线程使用。

- 若要设置可用于 Sybase IQ 的最大线程数，请设置服务器启动选项 `-iqmt`。缺省值是从连接数和 CPU 数计算得到的，通常已足够。请参见《实用程序指南》的第 1 章“运行数据库服务器”中的“启动数据库服务器”一节。
- 若要设置服务器中的内部执行线程的堆栈大小，请设置服务器启动选项 `-iqtss`。缺省值通常已足够，但如果复杂查询返回一条指示堆栈的深度已超过此限制的错误的消息，则可能会增加。请参见《实用程序指南》的第 1 章“运行数据库服务器”中的“启动数据库服务器”一节。

- 若要设置单个用户将使用的最大线程数，请发出命令 `SET OPTION MAX_IQ_THREADS_PER_CONNECTION`。某些操作试图分配和使用一“组”线程。要设置“组”中可用的线程数，请执行命令 `SET OPTION MAX_IQ_THREADS_PER_TEAM`。以上选项可用于控制特定操作使用的资源数量。例如，DBA 可以先设置此选项，然后再发出 `INSERT`、`LOAD`、`BACKUP` 或 `RESTORE` 命令。

平衡 I/O

本节解释在系统中平衡 I/O 的重要性。它解释如何在单独的磁盘上使用磁盘分条以及如何如何在单独的磁盘上找到文件，以获得更好的性能。还会讨论对消息日志文件大小的控制。

原始 I/O（在 UNIX 操作系统上）

大多数 UNIX 文件系统将磁盘划分成固定大小分区。分区是由操作系统单独访问的磁盘的物理子集。磁盘分区的访问模式通常有两种：文件系统模式（通过 UFS 文件系统）或原始模式。原始模式（有时称为字符模式）执行无缓冲的 I/O，通常使用每个读取或写入系统调用与设备进行双向数据传输。UFS 模式是一种 UNIX 文件系统和缓冲 I/O 系统，它每次在缓冲区中收集数据，直到它可以传输整个缓冲区。

在创建数据库或 `dbspace` 时，可以将它放在裸设备或文件系统文件上。Sybase IQ 自动按照您指定的路径名确定它是裸分区还是文件系统文件。裸分区可以是任意大小。

有关详细信息，请参见《系统管理指南，卷 1》的第 5 章“使用数据库对象”中的“使用数据库对象”。

使用磁盘分条

磁盘分条是一种将单个文件的数据分散到多个磁盘驱动器的常规方法。此方法允许在分条的磁盘驱动器上存在连续的磁盘块。分条将一个或多个物理磁盘（或磁盘分区）组合成单个逻辑磁盘。分条磁盘跨组件物理设备对 I/O 传输进行拆分，从而并行执行这些传输。与单个磁盘相比，分条磁盘大幅提高了性能。

磁盘分条允许将数据块放在不同磁盘上。第一个块位于第一个驱动器上。第二个块位于第二个驱动器上，依此类推。所有驱动器均已使用后，此过程再次循环，并使用驱动器上的其它块。磁盘分条的实际效果是数据随机分布在多个磁盘驱动器上。针对存储于分条磁盘上的文件的随机操作趋向于使在分条集中的所有驱动器保持同等繁忙状态，从而使每秒的磁盘操作总数最大化。这是数据库环境中非常有效的技术。

可以使用由操作系统和硬件提供的磁盘分条，也可以使用 Sybase IQ 内部磁盘分条。

在 UNIX 上设置磁盘分条

提供分条磁盘的 UNIX 系统提供了相应的实用程序，用于将物理磁盘配置成分条设备。有关详细信息，请参见 UNIX 或存储管理系统文档。

在 Windows 上设置磁盘分条

在 Windows 系统上，可通过适当的 SCSI-2 磁盘控制器来使用硬件磁盘分条。如果计算机不支持硬件分条，但您有可用于数据库的多个磁盘，则可以使用 Windows 分条将磁盘 I/O 分散到多个磁盘上。请使用磁盘管理工具设置 Windows 分条。

磁盘分条的建议

下面是磁盘分条的一些一般规则：

- 为了获得最高性能，应当将分条文件系统中的各个磁盘分散到几个磁盘控制器上。但请小心，不要用太多磁盘使磁盘控制器饱和。通常，在大多数 SCSI 计算机上，每个控制器可以处理 2-3 个磁盘。有关详细信息，请参见硬件文档。
- 不要将磁盘与较慢的设备（例如磁带驱动器或 CD-ROM）放在相同控制器上。这会减慢磁盘控制器的速度。
- 在带区中每个服务器 CPU 分配 4 个磁盘。
- 单个磁盘必须是相同的设备。也就是，它们必须有相同大小、相同格式，并且通常是相同品牌。如果布局不同，则通常使用最小那个磁盘的大小，并会浪费其它磁盘空间。而且，通常会使用最慢磁盘的速度。
- 通常，用于文件分条的磁盘不应当用于任何其它目的。例如，不要将文件分条磁盘用作交换分区。

- 永远不要将包含根文件系统的磁盘用作分条设备的一部分。
- 使用裸分区以达到最大性能。

通常，应当尽可能使用磁盘分条。

注释 为了在装载数据时获得最佳结果，请将数据转储到位于分条磁盘上的平面文件，然后用 `LOAD TABLE` 命令将数据读取到 Sybase IQ 中。

内部分条

Sybase IQ 将其信息按块存储于一系列 `dbspace`（设备的文件或裸分区）中。假定正在使用磁盘分条，则 Sybase IQ 会将数据分散到有可用空间的所有 `dbspace` 中。此方式允许您一次利用多个磁盘主轴，并提供并行磁盘写入的速度。

磁盘分条选项

本节解释如何使用 Sybase IQ 提供的选项，而不使用第三方软件，进行磁盘分条。如果您已经有通过第三方软件和硬件实现的磁盘分条解决方案，则应当使用该方法。可以通过为 `CREATE DBSPACE` 命令指定 `STRIPING ON` 选项启用磁盘分条。

打开或关闭磁盘分条

创建 `dbspace` 时用于更改缺省分条的语法为：

```
SET OPTION "PUBLIC".DEFAULT_DISK_STRIPING = { ON | OFF }
```

`DEFAULT_DISK_STRIPING` 选项对于所有平台的缺省值均为 `ON`。当磁盘分条是 `ON` 时，传入数据将分散到有可用空间的所有 `dbspace`。当磁盘分条是 `OFF` 时，`dbspace`（磁盘段）在逻辑文件上从前面填充，每次填充一个磁盘段。

如果更改 `DEFAULT_DISK_STRIPING` 的值，将影响不指定分条首选项的所有后续 `CREATE DBSPACE` 操作。

磁盘分条处于打开状态时，可以使用 `ALTER DBSPACE DROP` 命令从 `dbspace` 中删除文件。但是，在删除 `dbspace` 之前，必须使用 `sp_iqemptyfile` 存储过程在此 `dbspace` 中重新定位所有数据。因为磁盘分条使数据分散在多个文件中，因此 `sp_iqemptyfile` 进程可能需要重新定位多个表和索引。请使用 `sp_iqdbspaceinfo` 和 `sp_iqdbspace` 存储过程来确定哪些表和索引驻留在 `dbspace` 上。

使用多个文件

在 `dbspace` 中使用多个文件允许跨多个操作系统文件或分区分布 Sybase IQ 和临时数据。多个文件将改进吞吐量并缩短 `dbspace` 的平均延迟时间。

可以通过 `ALTER DBSPACE` 命令向 `dbspace` 添加其它文件。

何时添加文件

如果可以，请在创建 `dbspace` 时分配所有文件以确保数据分布均匀。

如果稍后添加文件，Sybase IQ 将跨新旧 `dbspace` 对新数据进行分条。分条可能是均匀的，也可能是不均匀的，具体取决于更新的类型。由于版本控制而“翻转”的页数会对是否重新均衡分条产生重要影响。

策略文件位置

通过增加专门用于存储随机访问的文件的磁盘驱动器数，并由此增加每秒对这些文件执行的操作数，可以提高与这些文件相关的性能。随机文件包括用于 IQ 存储、临时存储、Catalog 存储的文件，以及程序（包括 Sybase IQ 可执行文件、用户及存储过程和应用程序）和操作系统文件。

相反，通过将这些文件放在专用磁盘驱动器上，由此消除与其它进程的争用，可以提高与顺序访问的文件相关的性能。顺序文件包括事务日志文件和消息日志文件。

若要避免磁盘瓶颈，请遵循以下这些建议：

- 将随机磁盘 I/O 和顺序磁盘 I/O 分开。
- 将 Sybase IQ 数据库 I/O 与其它数据库（例如 Adaptive Server Enterprise）中的代理表的 I/O 隔离。
- 将事务日志和消息日志放在单独的磁盘上，与 IQ 存储、Catalog 存储和临时存储分离，并与任何代理数据库（例如 Adaptive Server Enterprise）分离。
- 将数据库文件、临时 `dbspace` 和事务日志文件放在与数据库服务器相同的物理计算机上。

事务日志文件

事务日志文件包含允许 Sybase IQ 从系统故障恢复的信息。还需要对事务日志进行审计。此文件的缺省文件扩展名为 `.log`。

若要移动或重命名事务日志文件，请使用事务日志实用程序 (dblog)。请参见《实用程序指南》>“数据库管理实用程序”>“事务日志实用程序 (dblog)”。

警告！ Sybase IQ 事务日志文件与大多数关系数据库事务日志文件不同。如果由于某个原因而丢失了数据库文件，则会失去数据库（除非丢失的是日志文件）。但是，如果有合适的备份，则可以重新装载数据库。

截断事务日志

Sybase IQ 在事务日志中记录从系统故障恢复所需的信息。尽管所记录的信息对于每个提交的事务很小，但事务日志的大小却持续增长。如果系统中有大量会更改数据的事务，则在一段时间后，日志可能增长到非常大。

何时截断日志应当由支持 Sybase IQ 系统的 DBA 负责，并取决于日志文件的生长情况和在现场的操作过程。

表 4-5 显示了在 Sybase IQ 中截断事务日志的方法。

表 4-5: 截断事务日志

如果您的数据库是 ...	则使用此方法 ...	有关详细信息，请参见 ...
已停止	“-m 开关，此开关导致事务日志在所有数据库的每个检查点之后被截断	“截断停止的数据库的事务日志”
正在运行	带有 -xo 开关或 -r 开关的 dbbackup 命令行实用程序	《实用程序指南》>“数据库管理实用程序”>“备份实用程序 (dbbackup)”

截断停止的数据库的事务日志

可以使用 `-m` 服务器启动开关截断数据库的事务日志。请注意，建议不要将 `-m` 服务器启动开关永久保持已设置状态。此开关应当仅用于启动 Sybase IQ 以执行事务日志截断。如何执行该操作由 DBA 负责，但以下过程提供了一个建议。

❖ 截断停止的数据库的事务日志

- 1 以一个将文件标识为日志截断配置设置的名称，创建服务器开关 .cfg 文件的副本，并编辑此文件副本以添加 -m 开关。
- 2 用包含 -m 选项的配置文件启动 Sybase IQ。请注意，这时不允许有任何用户访问或事务发生。
- 3 关闭 Sybase IQ，并使用未设置 -m 选项的配置文件重新启动。

消息日志

每个数据库都有消息日志文件。此文件的缺省名称为 *dbname.iqmsg*，但在创建数据库时可以指定其它名称。消息日志文件实际是在创建数据库后首次启动该数据库时创建的。

缺省情况下，Sybase IQ 会在消息日志文件中记录所有消息，包括错误、状态和插入通知消息。可以在 LOAD 和 INSERT 语句中使用参数关闭通知消息。

在某些站点上，由于插入次数、LOAD 选项和 NOTIFY_MODULUS 数据库选项设置或其它某些条件，消息日志文件常常会迅速增长。Sybase IQ 允许通过在活动 IQ 消息日志已满时回卷消息日志或设置最大文件大小并存档日志文件来限制此文件的大小。

有关设置最大日志文件大小、存档消息日志文件以及启用消息日志回卷的信息，请参见《系统管理指南，卷 1》的第 1 章“Sybase IQ 系统管理概述”中的“消息日志记录”一节。

用于插入、删除和同步的工作空间

插入或删除数据以及同步连接索引时，Sybase IQ 在 IQ 存储中需要一些工作空间。在需要此空间的事务提交后，将收回此空间以用于其它目的。

通常，只要在 IQ 存储中维护合理百分比的可用空间，就会有足够的可用空间。但是，对于某些删除（具体取决于数据的大小及其在数据库页中的分布情况），您可能需要大量的工作空间。在即将删除数据库的主要部分且数据稀疏分散在很多页中的情况下，可以将数据库的大小临时加倍。

长事务（如在一个事务中对大表逐行执行更新）可能会消耗大量主 dbspace。将保存每次更新的版本控制信息，直到提交事务。

设置保留空间选项

两个数据库选项 `MAIN_RESERVED_DBSPACE_MB` 和 `TEMP_RESERVED_DBSPACE_MB` 控制 Sybase IQ 为某些操作而保留的空间数量。有关详细信息，请参见《系统管理指南，卷 1》的第 5 章“使用数据库对象”中的“IQ main 存储和 IQ 临时存储空间管理”一节。

用于调优资源使用的选项

Sybase IQ 数据库的并发用户数、这些用户运行的查询以及可用于这些用户的处理线程和内存可以对性能、内存使用和磁盘 I/O 产生重大影响。Sybase IQ 提供了几个选项用于调整资源使用情况，以适应不同数目的用户和查询类型。这些选项可能是：

- 仅影响当前数据库的 `SET OPTION` 命令选项。
- 影响整个数据库服务器的命令行选项。
- 仅影响当前连接的连接参数。

有关所有这些选项的详细信息，包括参数、选项何时生效以及是否可以作为单个连接和 `PUBLIC` 组设置它们，请参见《参考：语句和选项》。

有关特定于优化表的信息，请参见《系统管理指南，卷 1》的第 5 章“使用数据库对象”中的“优化存储和查询性能”一节。

限制并发查询

通过 `-iqgovern` 开关，您可以指定特定服务器上的并发查询数量。这与连接数不同，连接数是由许可证控制的。通过指定 `-iqgovern` 开关，您可以帮助 IQ 优化将缓冲区数据分页调度到磁盘上，避免过量使用内存。`-iqgovern` 的缺省值是 $(2 \times \text{CPU 数}) + 10$ 。您可能需要试验确定理想值。对于使用大量活动连接的站点，请尝试将 `-iqgovern` 设置得略低些。

设置可用 CPU 数

Sybase IQ 启动命令的 `-iqnumbercpus` 开关允许您指定可用的 CPU 数。建议仅在以下情况下使用此开关：

- 在使用 Intel® CPU 并启用超线程的计算机上
- 在已使用操作系统实用程序将 Sybase IQ 限制为仅可使用其内 CPU 子集的计算机上

有关详细信息，请参见《系统管理指南，卷 1》的第 2 章“运行 Sybase IQ”中的“设置 CPU 数”一节。

限制查询使用的临时 dbspace

如果查询的估计临时空间使用量超出指定大小，则 `QUERY_TEMP_SPACE_LIMIT` 选项将导致查询被拒绝。缺省情况下，对查询使用的临时存储没有限制。

当您发出查询时，Sybase IQ 会估计完成该查询所需的临时空间。如果估计用于排序、散列和行存储的总计临时结果空间超过了当前 `QUERY_TEMP_SPACE_LIMIT` 设置，则查询会被拒绝，并且您会收到如下所示的消息：

```
Query rejected because it exceeds total space resource
limit
```

如果此选项设置为 0（缺省值），则表示没有限制，并且没有任何查询基于其临时空间需求被拒绝。

要限制每个连接的实际临时存储使用量，DBA 可以为包括查询在内的所有 DML 语句设置 `MAX_TEMP_SPACE_PER_CONNECTION` 选项。`MAX_TEMP_SPACE_PER_CONNECTION` 监控和限制语句的实际运行时临时存储使用情况。如果连接超出 `MAX_TEMP_SPACE_PER_CONNECTION` 选项设置的限额，将返回错误并回退当前语句。

限制按行返回的查询

`QUERY_ROWS_RETURNED_LIMIT` 选项告诉查询优化程序拒绝可能消耗太多资源的查询。如果查询优化程序估计来自查询的结果集将超过此选项的值，则它会拒绝该查询，并返回消息：

```
Query rejected because it exceed resource:
Query_Rows_Returned_Limit
```

如果使用此选项，则设置它，以便它仅拒绝消耗大量资源的查询。

强制游标为非滚动游标

使用没有声明宿主变量的滚动游标时， Sybase IQ 将创建临时存储节点，用于缓冲查询结果。此存储与临时存储缓冲区高速缓存分隔开。通过临时存储节点，可以在应用程序搜索结果集时有效向前和向后滚动。然而，如果查询返回很多（数百万）行输出，并且如果您的应用程序主要执行向前滚动操作，则临时存储节点的内存要求可能会降低查询性能。若要提高性能，请通过发出以下命令消除临时存储节点：

```
SET TEMPORARY OPTION FORCE_NO_SCROLL_CURSORS = 'ON'
```

注释 如果您的应用程序通常执行向后滚动，则将 `FORCE_NO_SCROLL_CURSORS` 选项设置为 `ON` 实际上可能会降低查询性能，因为如果不存在临时高速缓存，将强制 Sybase IQ 对每个向后滚动重新执行查询。

如果您的应用程序很少执行向后滚动，请将 `FORCE_NO_SCROLL_CURSORS = 'ON'` 作为永久 `PUBLIC` 选项。它将使用更少内存，并提高查询性能。

限制游标数

`MAX_CURSOR_COUNT` 选项指定相应资源调控器，以限制连接可以一次使用的最大游标数。缺省值为 50。如果将此选项设置为 0，则允许无限数目的游标。

限制语句数

`MAX_STATEMENT_COUNT` 选项指定相应资源调控器，以限制连接可以一次使用的最大就绪语句数。

预取高速缓存页

`SET` 选项 `PREFETCH_BUFFER_LIMIT` 可以定义 Sybase IQ 可用于预取（在数据库页之前的读取）的高速缓存页数。此选项的缺省值为 0。仅当 Sybase 技术支持部门建议这样做时，才应设置此选项。有关详细信息，请参见《参考：语句和选项》中的“`PREFETCH_BUFFER_LIMIT` 选项”。

SET 选项 `BT_PREFETCH_MAX_MISS` 可以确定是否继续为给定查询预取页。如果使用 `HG` 索引的查询的运行速度比预期慢很多，请尝试逐步增大此选项的值。有关详细信息，请参见《参考：语句和选项》中的“`BT_PREFETCH_MAX_MISS` 选项”。

针对典型使用进行优化

Sybase IQ 可跟踪打开的游标数，并相应地分配内存。在某些环境中，可以设置 `USER_RESOURCE_RESERVATION` 选项，以调整认为当前正在使用该产品的当前最少游标数，并因此更节省地从临时高速缓存分配内存。

应当仅在仔细的分析显示实际需要它之后，才设置此选项。如果需要设置此选项，请与 Sybase 技术支持部门联系，以了解详细信息。

控制预取行数

预取用于提高仅提取 `relative 1` 或 `relative 0` 的游标的性能。两个连接参数允许您更改游标预取缺省值。`PrefetchRows (PROWS)` 可以设置预取行数；`PrefetchBuffer (PBUF)` 可以设置此连接可用来存储预取行的内存。在某些情况下，增加预取的行数有可能提高性能：

- 应用程序使用非常少的绝对位置提取操作来提取许多行（几百或更多）。
- 应用程序以高速率提取行，客户端和服务端位于同一台计算机上或通过快速网络连接。
- 客户端 / 服务器通信是通过速度较慢的网络（如拨号链接或广域网）进行的。

改进资源使用的其它方式

本节描述调整系统以获得最大性能或更好利用磁盘空间的几种方式。

管理 Multiplex 数据库中的磁盘空间

由于任何服务器上的任何用户都可能处在可能需要旧版表的事务中，所以 Sybase IQ 不能删除这些旧版表。在 Multiplex 数据库中同时发生表更新和查询时，Sybase IQ 可能因此消耗非常多的磁盘空间。所消耗的空间数量取决于数据和索引的性质以及更新率。

通过允许写入服务器删除查询不再需要的过时版本，可以释放磁盘块。所有服务器上的所有用户都应当定期提交其当前事务，以允许恢复旧版表。服务器可能始终正常工作，并完全可用。sp_iqversionuse 存储过程可用于显示远程服务器的版本使用情况。

查询服务器间的负载均衡

您可能能够使用 IQ 网络客户端在 Multiplex 查询服务器间平衡查询负载。此方法需要一个能够将客户端连接调度到池内计算机的中间系统，具体取决于计算机的负载。

若要使用此方法，请通过使用此中间负载均衡系统的 IP 地址和端口号、常规服务器名称以及设置为 NO 的 VerifyServerName 连接参数，在客户端系统上创建特殊 ODBC DSN。当客户端使用此 DSN 进行连接时，负载均衡器将与它确认的有最低负载的计算机建立连接。

有关如何定义在查询服务器负载均衡中使用的 ODBC DSN 的详细信息，请参见《系统管理指南，卷 1》的第 4 章“连接和通信参数”中的“VerifyServerName 参数 [Verify]”一节。

限制数据库访问

为了获得更好的查询性能，如果可能，请将数据库设置为只读，或使其在低峰使用时间进行重要更新。当您正在表中进行插入或删除时，Sybase IQ 允许多个查询用户从该表读取数据。但是，在数据库并发更新期间，性能会降低。

磁盘高速缓存

*磁盘高速缓存*是操作系统临时用于存储磁盘块副本的内存。所有基于磁盘读取和写入的文件系统通常都会通过磁盘高速缓存执行操作。从应用程序的角度来看，涉及磁盘高速缓存的所有读取和写入等同于实际的磁盘操作。

操作系统使用两种不同方法向磁盘高速缓存分配内存：固定和动态。在固定分配中使用预置数量的内存；通常会留出 10–15% 的内存分配量。操作系统通常使用 LRU（最近使用最少的）算法来管理此工作空间。对于动态分配，操作系统将在运行时确定磁盘高速缓存分配量。目标是保持尽可能多的内存处于活动备用状态，从而使对磁盘数据的需求与对真实内存的需求保持平衡。

索引提示

以下几节为选择和管理索引提供某些提示。有关这些主题的详细信息，请参见《系统管理指南，卷 1》中的第 6 章“使用 Sybase IQ 索引”。

选择正确索引类型

应该为列数据选择正确索引类型。Sybase IQ 自动提供某些索引（在所有列上用于优化投影的索引，以及用于 UNIQUE、PRIMARY KEYS 和 FOREIGN KEYS 的 HG 索引）。虽然这些索引对某些用途来说非常有用，但您可能需要其它索引以尽快处理特定查询。

Sybase IQ 查询优化程序有一个索引顾问，它可以在查询中生成优化器何时会受益于一个或多个列的其它索引的消息。若要激活索引顾问，请将 INDEX_ADVISOR 选项设置为 ON。如果不启用查询计划，则消息作为查询计划的一部分或作为单独消息显示在消息日志 (.iqmsg) 中，并且输出采用 OWNER.TABLE.COLUMN 格式。有关详细信息，请参见《参考：语句和选项》的第 2 章“数据库选项”中的“INDEX_ADVISOR 选项”。

如果列未使用枚举 FP 存储，则应考虑对连接查询中的 WHERE 子句引用的分组列创建一个 LF 或 HG 索引。Sybase IQ 优化程序可能需要枚举 FP 或 HG/LF 索引中的元数据以生成最佳查询计划。HAVING 子句中引用的非集合列也可能受益于 LF 或 HG 索引以帮助优化查询。例如：

```
SELECT c.name, SUM(l.price * (1 - l.discount))
FROM customer c, orders o, lineitem l
WHERE c.custkey = o.custkey
      AND o.orderkey = l.orderkey
      AND o.orderdate >= "1994-01-01"
      AND o.orderdate < "1995-01-01"
GROUP by c.name
HAVING c.name NOT LIKE "I%"
```

```
AND SUM(l.price * (1 - l.discount)) > 0.50  
ORDER BY 2 desc
```

请注意，添加索引会增加存储要求和装载时间，并且只应在对查询性能有益的情况下执行这一操作。

使用连接索引

用户经常需要一次查看多个表中的数据。此数据可以在查询时连接，也可以提前通过创建连接索引进行连接。有时，可以通过为以一致方式连接的列创建连接索引，提高查询性能。

因为连接索引需要大量时间和空间进行装载，所以应当只为经常需要的连接创建连接索引。Sybase IQ 连接索引支持一对多和一对一连接关系。

为删除留出足够磁盘空间

删除数据行时，Sybase IQ 将为包含任何被删除数据的每个数据库页创建一个版本页。这些版本会一直保留，直到删除事务提交。因此，在删除数据时，可能需要添加磁盘空间。有关详细信息，请参见第 394 页的“重叠版本和删除”。

管理数据库大小和结构

本节提供关于改进数据库设计和管理数据的理念。

管理数据库大小

数据库的大小主要取决于您创建的索引，以及所维护的数据数量。通过创建需要用于用户所发出的查询类型的所有索引，可以获得更快的查询处理速度。但是，如果发现不需要某些表或索引，则可以删除它们。通过这样做，可以释放磁盘空间，提高装载和备份的速度，并减少备份所需的存档存储量。

若要控制在给定表中存储的数据数量，请考虑如何以最佳方式消除不再需要的数据行。如果数据库包含最初来自 SQL Anywhere 数据库的数据，则通过简单重播 Anywhere 删除操作，可能能够根除不需要的数据；命令语法是兼容的。可以对 Adaptive Server Enterprise 数据库中的数据执行相同操作，因为 Sybase IQ 提供了 Transact-SQL 兼容性。

控制索引碎片

当索引页未被使用到最大量时，就会出现内部索引碎片。

当删除行时，可能会出现行碎片。如果删除行的整个页，则该页将被释放，但如果页上的某些行是未使用的，则未使用的空间会留在磁盘上。

处理表的 DML 操作（INSERT、UPDATE、DELETE）会导致索引碎片。两个存储过程会报告碎片：

- `sp_iqrowdensity` 报告缺省索引级别的行碎片。请参见《参考：构件块、表和过程》的第 7 章“系统过程”中的“`sp_iqrowdensity` 过程”。
- `sp_iqindexfragmentation` 报告在补充索引中的内部碎片。请参见《参考：构件块、表和过程》的第 7 章“系统过程”中的“`sp_iqindexfragmentation` 过程”。

数据库管理员可能创建其它索引，以补充列上的缺省索引。与从表中删除行时所需要的空间相比，这些索引可以使用更多空间。

两个过程都不建议执行进一步操作。数据库管理员必须检查所报告的信息，并确定是否要执行进一步操作，例如重新创建、重组或重新生成索引。

使 Catalog 文件增长最少

Catalog 文件的生长是正常情况，它根据应用程序和 Catalog 内容不同而各不相同。`.db` 文件的大小对性能没有影响，并且系统会根据需要重用 `.db` 文件中的可用页。最大限度地抑制 Catalog 文件的增大：

- 避免对 CREATE TABLE 语句使用 IN SYSTEM。
- 运行系统存储过程之后发出 COMMIT 语句。
- 在长时间运行的事务期间发出 COMMIT 语句。

非规范化以提高性能

已经以规范化形式创建数据库后，则可能执行若干基准，并决定有目的地脱离规范化以提高性能。非规范化：

- 可用表或列完成
- 采用前一类范式
- 需要如何使用数据的知识

非规范化的良好原因是：

- 所有查询都需要访问连接数据的“完整”集合
- 派生列的计算复杂性需要 `select` 存储

非规范化有风险

非规范化只能在您全面了解应用程序时才能成功执行，并且仅当性能问题指示需要这样做时，才应当执行它。进行非规范化时要考虑的一件事情是通过更改使数据保持最新所需的工作量。

这是决策支持应用程序（它们经常需要大量数据的摘要）与事务处理需求（它们执行不连续的数据修改）的差异的良好示例。非规范化通常以其它处理的开销来支持某些处理。

无论选择什么形式的非规范化，都可能潜在的数据完整性问题，必须仔细记录它们，并在应用程序设计中加以解决。

非规范化的缺点

非规范化有以下这些缺点：

- 非规范化通常会使检索加速，但可以使更新减慢。这不是在 DSS 环境中需要关注的实际问题。
- 非规范化始终是特定于应用程序的，如果应用程序更改，则需要对其重新评估。
- 非规范化可以增加表的大小。这不是 Sybase IQ 中的问题，因为您可以优化列数据的存储。有关详细信息，请参见《参考：语句和选项》的 `CREATE TABLE` 语句和“`MAX_QUERY_TIME` 选项”中的 `IQ UNIQUE` 列约束。
- 在某些实例中，非规范化会简化编码；在其它情况下，则会使编码更复杂。

非规范化的性能优势

非规范化可以通过以下方式提高性能：

- 最小化连接需求
- 预先计算集合值，即在数据修改时，而不是在选择时，计算它们
- 在某些情况下，减少表的数目

决定非规范化

当决定是否进行非规范化时，需要分析环境中应用程序的数据访问需求，及其实际性能特征。考虑非规范化时要考查的一些问题包括：

- 什么是关键查询，以及什么是期望的响应时间？
- 它们使用什么表或列？每个访问有多少行？
- 常用排序顺序是什么？
- 什么是并发预期？
- 最常访问的表有多大？
- 所有进程都计算摘要吗？
- 是否应当创建连接索引以提高性能？

使用 UNION ALL 视图以便更快装载

UNION ALL 视图可用于在为表中的所有行保留次级索引代价过高的情况下提高装载性能。Sybase IQ 允许您将数据拆分至多个单独的基表中（例如按日期）。数据将装载到这些较小的表中。然后，通过 UNION ALL 视图，将表重新连接在一起，形成一个逻辑整体，之后可以对其执行查询。

此策略可以提高装载性能，但是可能会对一些类型的查询的性能产生负面影响。针对单个基表或较小基表的 UNION ALL 视图执行的大多数类型的查询都具有大致相似的性能，只要视图定义满足第 80 页的“优化引用 UNION ALL 视图的查询”中说明的所有约束。然而，与针对单个大型基表执行相比，针对 UNION ALL 视图执行时，一些类型的查询的执行速度可能会显著降低，特别是那些涉及 DISTINCT 或涉及多个连接列的连接查询。在选择使用此策略之前，请确定装载性能的提高是否值得以降低应用程序的查询性能为代价。

UNION ALL 视图对管理员来说非常高效。例如，如果数据按月分区，则通过删除表和正确更新 UNION ALL 视图定义，就可以删除整个月份的数据值。您可以有年份、季度等的许多视图定义，而不必添加额外的日期范围谓词。

若要创建 UNION ALL 视图，请选择可以将基表划分到多个单独物理表中的逻辑方式。最常见的划分方式是按月划分。

例如，若要创建包括第一季度的所有月份的视图，请输入：

```
CREATE VIEW
SELECT * JANUARY
UNION ALL
SELECT * FEBRUARY
UNION ALL
SELECT * MARCH
UNION ALL
```

在每个月，可以将数据装载到单个基表中（在此示例中为 JANUARY、FEBRUARY 或 MARCH）。在下一月，则将数据装载到有相同列和相同索引类型的新表中。

有关语法详细信息，请参见《参考：语句和选项》中的 UNION 操作。

注释 不能在 UNION ALL 视图中执行 INSERT...SELECT。UNION ALL 运算符在此版本中不是完全并行的。使用这些运算符可能会限制查询并行度。

优化引用 UNION ALL 视图的查询

UNION ALL 视图中的所有分区都必须有一组为工作优化定义的完整索引。

有 DISTINCT 的查询在使用 UNION ALL 视图时通常运行速度比基表更慢。

Sybase IQ 包括针对 UNION ALL 视图的专利优化技术，包括：

- 在 UNION ALL 视图上拆分 GROUP BY
- 下推连接至 UNION ALL 视图

如果您需要调整引用 UNION ALL 视图的查询的性能，则可能需要设置 JOIN_PREFERENCE 数据库选项，该选项影响 UNION ALL 视图之间的连接。有关这些选项的详细信息，请参见《参考：语句和选项》中的第 2 章“数据库选项”。

仅当 UNION 满足以下所有约束时，才能将它视为分区表：

- 它只包含一个或多个 UNION ALL。
- UNION 的每个分支在其 FROM 子句中只有一个表，并且该表是物理基表。
- UNION 的任何分支都没有 DISTINCT、RANK、集合函数或 GROUP BY 子句。
- UNION 的每个分支中的 SELECT 子句中的每一项都是一列。
- 在第一个 UNION 分支的 SELECT 列表中的列的数据类型序列与 UNION 的每个随后分支中的序列相同。

另请参见

《参考：语句和选项》中的“SELECT 语句”。

管理 UNION ALL 视图性能

当 ORDER BY 为 DESC 时，不会应用某些特定优化（例如，将 DISTINCT 运算符推入 UNION ALL 视图），因为评估 UNION 下的 DISTINCT 的优化不适用于 DESC 顺序。例如，下面的查询将对性能产生负面影响：

```
SELECT DISTINCT state FROM testVU ORDER BY state DESC;
```

要解决此性能问题，查询应先评估 DISTINCT 运算符，然后才能执行 ORDER BY，在此种情况下，排序顺序为 ASC，并且可应用优化：

```
SELECT c.state FROM (SELECT DISTINCT state
                     FROM testVUA) c
ORDER BY c.state DESC;
```

另请参见

《参考：语句和选项》中的“SELECT 语句”。

改进的大型单（事实）表装载

为满足呈指数级增长的信息挑战和数据实时访问需求，Sybase IQ 显著增强了装载 High_Group (HG) 索引和包含（也称为 WORD）(WD) 索引的性能，同时仍然允许事务访问正在装载的表。针对 HG 和 WD 索引的此装载性能提升将影响：

- INSERT...SELECT
- INSERT...LOCATION
- LOAD
- UPDATE
- CREATE INDEX
- 可更新游标

LOAD TABLE 语句现在执行 HG 和 WD 索引的并行装载，因此比上一版本的执行速度更快。

网络性能

以下几节提供解决某些网络性能问题的建议。

改进大型数据传输

大型数据传输会同时减少总体吞吐量，并增加平均响应时间。下面是一些在这些传输期间提高性能的建议：

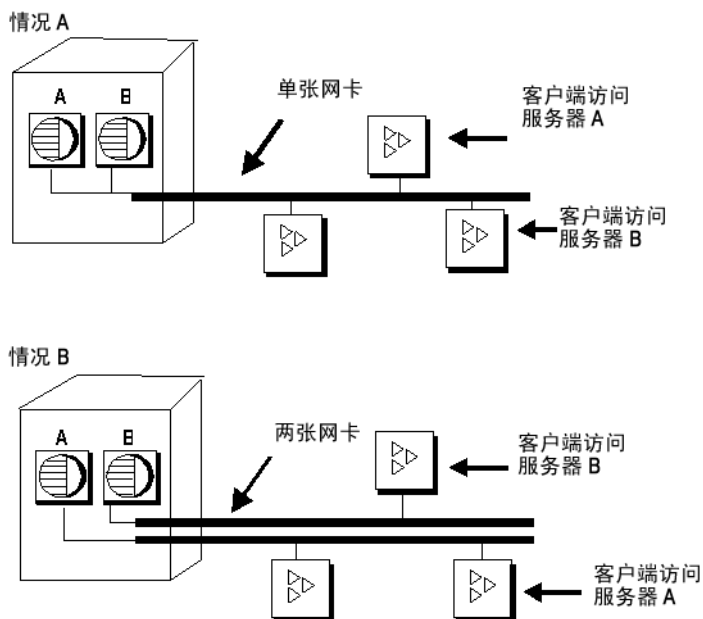
- 如果可能，在下班时段执行大型传输。
- 在大型传输期间限制并发查询数。
- 不要在大型传输期间并发运行查询和插入。
- 使用存储过程以减少通信总量。
- 使用行缓冲，以便通过网络移动大批数据。
- 如果经常使用大型传输，请考虑安装适合这类传输的更好的网络硬件。例如：
 - 令牌环 — 在高负荷利用时段比以太网硬件有更好的响应能力。
 - 光纤 — 提供非常宽的带宽，但它通常特别昂贵，难以在整个网络中使用。
 - 单独的网络 — 可以用来处理最高容量工作站与服务器之间的网络通信量。

隔离重网络负荷用户

在图 12-4 的示例 A 中，访问两个不同数据库服务器的客户端使用一个网卡。也就是，访问服务器 A 和 B 的客户端必须争用网络并通过该网卡。在示例 B 中，访问服务器 A 的客户端使用的网卡与访问服务器 B 的客户端使用的网卡不同。

将数据库服务器放在不同计算机上甚至会更好。还可以将不同数据库的高负荷用户放在不同计算机上。

图 4-2: 隔离重网络负荷用户



将少量数据放在较小包中

如果通过网络发送少量数据，请保持缺省网络包大小较小（缺省值为 512 字）。-p 服务器启动选项允许您指定最大包大小。客户端应用程序还可能允许您设置包大小。

将大量数据放在较大包中

如果大多数应用程序发送和接收大量数据，请增加缺省网络包大小。这将导致更少（但更大）的传输。

在服务器级处理

在服务器级过滤尽可能多的数据。

本章介绍用于确定系统是否正在很好地利用可用资源的工具。

主题	页码
查看 Sybase IQ 环境	85
监控缓冲区高速缓存	101
缓冲区高速缓存结构	112
避免缓冲区管理器抖动	112
缓冲区高速缓存监控清单	116
监控 CPU 使用的系统实用程序	119

查看 Sybase IQ 环境

优化 Sybase IQ 性能的第一步是查看环境。可以选择各种选项：

- 使用系统监控工具（每个系统和站点都有不同的现成工具）。
- 使用 Sybase Central 中的动态性能监控器。请参见第 86 页的“[监控性能统计信息](#)”。有关 Multiplex 性能监控，请参见《使用 Sybase IQ Multiplex》。
- 使用一个可显示有关 Sybase IQ 的信息的存储过程。请参见第 92 页的“[使用存储过程获取信息](#)”。
- 使用过程分析以跟踪存储过程、函数和事件的执行次数。请参见第 93 页的“[分析数据库过程](#)”。
- 确定合适的索引类型。有关选择索引类型的详细信息，请参见《系统管理指南，卷 1》中的第 6 章“[使用 Sybase IQ 索引](#)”。
- 有关操作界面的信息，请查看插入和删除通知消息。《系统管理指南，卷 1》中的第 7 章“[将数据移入和移出数据库](#)”提供了有关这些消息的详细信息。
- 查看 Sybase IQ 消息文件，缺省情况下该文件名为 *dbname.iqmsg*。

监控性能统计信息

Sybase Central 中的性能监控器显示一个或多个涉及的节点的统计信息的集合。统计信息实时显示在动态图表中。

可在两个不同级别访问性能监控器：

- **Multiplex 级** — 只能在多服务器间监控一项统计信息。
- **服务器级** — 在单服务器或 Multiplex 服务器上，一次最多可以监控十项统计信息。

本节介绍了仅在单服务器上进行的服务器级访问。若要对 Multiplex 服务器使用 Multiplex 级和服务器级，请参见《使用 Sybase IQ Multiplex》。

❖ 在服务器级监控性能

- 在 Sybase Central 树视图中单击服务器名，然后切换到“性能监控器”选项卡。

自定义统计信息显示

可以更改性能监控器图形的类型或内容。

❖ 更改要监控的统计信息

从服务器级性能监控器启动此对话框时，一次最多可选择十项要监控的统计信息。（对于 Multiplex 级监控，此对话框行为有所不同。请参见《使用 Sybase IQ Multiplex》。）

- 1 单击服务器，然后切换到“性能监控器”选项卡。
- 2 在“性能监控器”选项卡中，右键单击图表区域，从快捷方式菜单中选择“更改统计信息”。
- 3 在“更改统计信息”对话框中，选择要监控的统计信息。

❖ 将图表另存为位图

可以将图表另存为 .JPEG 图像文件。

- 1 右键单击“性能监控器”图表。
- 2 从弹出菜单中，选择“另存图表为”。
- 3 指定 .JPEG 文件的文件名。
- 4 单击“保存”。

❖ 打印图表

可以打印性能监控器图表。

- 1 右键单击“性能监控器”图表。
- 2 从弹出菜单中，选择“打印图表”。
- 3 查看打印选项并将页面输出到所需打印机。

❖ 切换图表视图

可以在时序图表视图、2D 条形图视图和 3D 条形图视图之间切换。

- 1 右键单击“性能监控器”图表。
- 2 从对应菜单中选择“切换图表视图”。
- 3 选择“时序图表”、“条形图 2D 垂直”或“条形图 3D 垂直”。

❖ 自定义图表

更改图表设置和图表刷新率（也称为监控器 GUI 心跳率）。

- 1 右键单击性能监控器图表区域。
- 2 选择“自定义图表”。

“自定义图表”对话框包括以下组成部分：

 - 时间窗口 — 只有在图表为时序图表时才会显示。指定数据显示的时间段。最小值为 1 分钟。最大值为 240 分钟（4 小时）。
 - 图表刷新率 — 指定图表中数据的刷新频率，以秒为单位。该值还显示在性能监控器窗格的底部。
 - 实际值与规范化值 — 实际值（缺省值）反映实际数据。规范化值会将图表数据缩放到固定范围。此选项仅用于显示目的，以便更好地在一个图表中显示具有不同范围的统计信息。
 - 图例 — 选择是显示还是隐藏图例。
- 3 若要保存任何更改，请单击“确定”。

统计信息的类别

统计信息分组为以下类别：

- [CPU 使用率统计信息](#)
- [内存使用统计信息](#)
- [高速缓存统计信息](#)
- [线程统计信息](#)
- [连接统计信息](#)
- [请求统计信息](#)
- [事务统计信息](#)
- [存储 I/O 统计信息](#)
- [DBspace 使用情况](#)
- [网络统计信息](#)

CPU 使用率统计信息

表 5-1: CPU 使用率

名称	说明	缺省情况下是否监控?
CPU 使用率	IQ 进程 CPU 使用百分比，包括系统和用户使用率。	是
CPU 系统使用率	IQ 进程 CPU 系统使用百分比。	否
CPU 用户使用率	IQ 进程 CPU 用户使用百分比。	否

内存使用统计信息

表 5-2: 内存使用

名称	说明	缺省情况下是否监控?
分配的内存	由 IQ 服务器分配的内存（以兆字节计）	是
分配的最大内存	由 IQ 服务器分配的最大内存（以兆字节计）	否

高速缓存统计信息

表 5-3: 高速缓存统计信息

名称	说明	缺省情况下是否监控?
目录高速缓存命中数	每秒目录高速缓存命中数。	否
临时高速缓存命中数	每秒临时高速缓存命中数。	否
主高速缓存命中数	每秒主高速缓存命中数。	否
目录高速缓存读取数	每秒查找目录高速缓存页的次数。	是
临时高速缓存读取数	每秒查找临时高速缓存页的次数。	否
主高速缓存读取数	每秒的主高速缓存页查找次数。	否
目录高速缓存当前大小	当前目录高速缓存大小(以兆字节计)。	否
临时高速缓存当前大小	当前临时高速缓存大小(以兆字节计)。	否
主高速缓存当前大小	当前主高速缓存大小(以兆字节计)。	否
正在使用的目录高速缓存百分比	正在使用的目录高速缓存百分比。	否
正在使用的临时高速缓存百分比	正在使用的临时高速缓存百分比。	否
正在使用的主高速缓存百分比	正在使用的主高速缓存百分比。	否
已固定的目录高速缓存	已固定的目录高速缓存页数。	否
已固定的临时高速缓存	已固定的临时高速缓存页数。	否
已固定的主高速缓存	已固定的主高速缓存页数。	否
已固定的目录高速缓存百分比	已固定的目录高速缓存的百分比。	否
已固定的临时高速缓存百分比	已固定的临时高速缓存的百分比。	否
已固定的主高速缓存百分比	已固定的主高速缓存百分比。	否
目录高速缓存脏页百分比	目录高速缓存脏页的百分比。	否
临时高速缓存脏页百分比	临时高速缓存脏页的百分比。	否
主高速缓存脏页百分比	主高速缓存脏页的百分比。	否

线程统计信息

表 5-4: 线程统计信息

名称	说明	缺省情况下是否监控?
正在使用的 IQ 线程	IQ 服务器使用的线程数	否
可用 IQ 线程	IQ 服务器中可用的线程数	否
正在使用的 SA 线程数	SQL Anywhere 引擎使用的线程数。	否

连接统计信息

表 5-5: 连接统计信息

名称	说明	缺省情况下是否监控?
连接总计	连接总数包括用户和 INC 连接。	是
用户连接数	用户连接数。	否
INC 进来的连接数	INC 进来的连接数	否
INC 外发的连接数	INC 外发的连接数	否
每分钟的用户连接数	每分钟的用户连接数	否
每分钟断开的用户连接数	每分钟断开的用户连接数	否

请求统计信息

表 5-6: 请求统计信息

名称	说明	缺省情况下是否监控?
请求	为使服务器能够处理新请求或继续处理现有的请求每秒进入服务器的次数。	否
未调度的请求	当前排队等待可用服务器线程的请求的数量。	否
IQ 正在等待操作	正在等待资源调控器的 IQ 操作数	否
IQ 活动的操作	活动的 IQ 操作数	否

事务统计信息

表 5-7: 事务统计信息

名称	说明	缺省情况下是否监控?
事务总数	包括用户事务和 INC 事务在内的活动事务的总数。	否
用户事务计数	活动的用户事务数	否
INC 事务计数	活动的 INC 事务数	否
活动 Load Table 语句	活动 LOAD TABLE 语句数	否

存储 I/O 统计信息

表 5-8: 存储 I/O 统计信息

名称	说明	缺省情况下是否监控?
Catalog 存储磁盘读取数	每秒从 catalog 存储读取的千字节数。	否
临时存储磁盘读取数	每秒从临时存储读取的千字节数。	否
Main 存储磁盘读取数	每秒从 main 存储读取的千字节数。	否
Catalog 存储磁盘写入数	每秒写入 catalog 存储的千字节数。	否
临时存储磁盘写入数	每秒写入临时存储的千字节数。	否
Main 存储磁盘写入数	每秒写入 main 存储的千字节数。	否

DBSpace 使用情况

表 5-9: DBSpace 使用情况

名称	说明	缺省情况下是否监控?
正在使用的 DBSpace 文件大小	正在使用的 DBSpace 大小。每个 dbspace 均有一项此类统计信息。	否
可用的 DBSpace 大小的百分比	可用于每个 dbspace 文件的可用空间百分比。每个文件的每个 dbspace 均有一项此类统计信息。	否

网络统计信息

表 5-10: 网络统计信息

名称	说明	缺省情况下是否监控?
收到的字节	客户端 / 服务器通信过程中每秒收到的字节数。	是
收到的未压缩字节	禁用压缩时, 在客户端 / 服务器通信过程中每秒收到的字节数。	否
发送的字节	客户端 / 服务器通信过程中每秒发送的字节数。	是
发送的未压缩字节	禁用压缩时, 在客户端 / 服务器通信过程中每秒发送的字节数。	否
可用通信缓冲区数	可用网络通信缓冲区的数量。	否
通信缓冲区总数	网络通信缓冲区的总数。	否

使用存储过程获取信息

Sybase IQ 提供的几个存储过程可以显示有关数据库的信息:

- `sp_iqconnection` 显示有关用户连接和版本的统计信息
- `sp_iqcontext` 显示有关语句正在执行什么的信息
- `sp_iqcheckdb` 检查当前数据库的有效性
- `sp_iqdbstatistics` 报告最近 `sp_iqcheckdb` 的结果
- `sp_iqdbsize` 给出当前数据库的大小
- `sp_iqspaceinfo` 显示数据库中每个对象的空间使用情况
- `sp_iqstatus` 显示有关数据库的杂项状态信息。
- `sp_iqtablesize` 给出您指定的表的大小。
- `sp_iqgroupsize` 列出指定组的成员。

有关所有 Sybase IQ 存储过程的语法详细信息和示例, 请参见的《参考: 构件块、表和过程》。

分析数据库过程

过程分析显示执行存储过程、函数、事件、系统触发器和触发器需要多长时间。您也可以查看过程中每一行的执行时间。利用数据库分析信息，您可以确定微调哪些过程就可以提高数据库的性能。

启用分析后，Sybase IQ 将监控哪些存储过程、函数、事件、系统触发器以及触发器被使用，从而跟踪执行它们用了多少时间，以及它们分别被调用了多少次。

分析信息由服务器存储在内存中，可以在 Sybase Central 中通过“分析”选项卡或在 Interactive SQL 中进行查看。分析功能一旦启用，数据库就会收集分析信息，直到您禁用分析功能或服务器关机为止。

有关在 Interactive SQL 中获取分析信息的详细信息，请参见第 99 页的“在 Interactive SQL 中查看过程分析信息”。

启用过程分析

过程分析跟踪所有连接对过程和触发器的使用。您可以在 Sybase Central 或 Interactive SQL 中启用分析。若要启用并使用过程分析，必须具有 DBA 授权。

❖ 启用分析 (Sybase Central):

- 1 以具有 DBA 授权的用户身份连接到数据库。
- 2 在左窗格中选择数据库。
- 3 从“文件”菜单中选择“属性”。
随即出现“数据库”属性表。
- 4 在“分析”选项卡上选择“对此数据库启用分析”(Enable Profiling on This Database)。
- 5 单击“确定”以关闭该属性表。

注释

您也可以通过在 Sybase Central 中右键单击数据库来启用分析。从弹出菜单中，选择“分析”>“开始分析”。

❖ 启用分析 (SQL):

- 1 以具有 DBA 授权的用户身份连接到数据库。
- 2 用 ON 设置调用 sa_server_option 存储过程。

例如，输入：

```
CALL sa_server_option ( 'procedure_profiling', 'ON')
```

如有必要，您能够看到特定用户正在使用什么过程，并且不会阻止其它连接使用数据库。当连接已存在或有多个用户使用同一用户 ID 进行连接时，这是非常有用的。

❖ **过滤用户的过程分析:**

- 1 以具有 DBA 授权的用户身份连接到数据库。
- 2 调用以下过程:

```
CALL sa_server_option  
('ProfileFilterUser','userid')
```

userid 的值是正被监控的用户的名称。

重置过程分析

重置分析后，数据库会清除旧信息，并立即开始收集有关过程、函数、事件和触发器的新信息。

下面几节假定您已经作为具有 DBA 授权的用户身份连接到数据库并且该过程分析已启用。

❖ **重置分析 (Sybase Central):**

- 1 在左窗格中选择数据库。
- 2 从“文件”菜单中选择“属性”。
随即出现“数据库”属性表。
- 3 在“分析”选项卡上单击“立即重置”。
- 4 单击“确定”以关闭该属性表。

注释

您也可以通过在 Sybase Central 中右键单击数据库来重置分析。从弹出菜单中，单击“分析” > “重置分析信息”。

❖ **重置分析 (SQL):**

- 使用 RESET 设置调用 `sa_server_option` 存储过程。

例如，输入：

```
CALL sa_server_option ('procedure_profiling',  
'RESET')
```

禁用过程分析

当收集完分析信息后，可以禁用分析或清除分析。如果禁用分析，数据库将停止收集分析信息，而在此刻之前收集的信息会留在 Sybase Central 中的“分析”选项卡上。如果清除分析，数据库将关闭分析功能，并且删除 Sybase Central 中“分析”选项卡上的所有分析数据。

❖ 禁用分析 (Sybase Central):

- 1 在左窗格中选择数据库。
- 2 从“文件”菜单中选择“属性”。
随即出现“数据库”属性表。
- 3 在“分析”选项卡上，清除“对此数据库启用分析”(Enable Profiling on This Database)选项。
- 4 单击“确定”以关闭该属性表。

注释

您也可以通过在 Sybase Central 中右键单击数据库来禁用分析。从弹出菜单中，选择“分析”>“停止分析”。

❖ 禁用分析 (SQL):

- 用 OFF 设置调用 sa_server_option 存储过程。
例如，输入：

```
CALL sa_server_option ('procedure_profiling',  
                        'OFF')
```

❖ 清除分析 (Sybase Central):

- 1 在左窗格中选择数据库。
- 2 从“文件”菜单中选择“属性”。
随即出现“数据库”属性表。
- 3 在“分析”选项卡上，单击“立即清除”。
只有在分析已启用的情况下，才能清除分析。
- 4 单击“确定”以关闭该属性表。

注释

您可以通过在 Sybase Central 中右键单击数据库来清除分析。从弹出菜单中，选择“分析” > “清除分析信息”。

❖ **清除分析 (SQL):**

- 用 CLEAR 设置调用 sa_server_option 存储过程。

例如，输入：

```
CALL sa_server_option ('procedure_profiling',  
'CLEAR')
```

在 Sybase Central 中查看过程分析信息

根据您的选择查看有关整个数据库、特定类型的对象的信息还是选择查看有关特定过程的信息，过程分析会为您提供不同的信息。该信息可以通过以下方式显示：

- 数据库中所有分析过的对象的详细信息
- 所有存储过程和函数的详细信息
- 所有事件的详细信息
- 所有触发器的详细信息
- 所有系统触发器的详细信息
- 单个分析过的对象的详细信息

若要查看分析信息，必须连接到数据库并启用分析。

当您查看有关整个数据库的分析信息时，以下列将出现：

- **Name** 列出对象的名称。
- **Owner** 列出对象的所有者。
- **Table** 列出触发器属于哪个表（此列仅出现在数据库“分析”选项卡上）。
- **Event** 显示系统触发器的触发器类型。可以是 Update 或 Delete。
- **Type** 列出对象的类型，例如，过程。
- **# Exes.** 列出每个对象已被调用的次数。
- **#msecs.** 列出各个对象的总执行时间。

这些列提供已经在数据库中执行的所有过程的分析信息的摘要。一个过程可以调用其它过程，因此列出的项目可能会多于用户专门调用的项目。

❖ 查看存储过程和函数的摘要分析信息：

- 1 在左窗格中选择“过程和函数”文件夹。
- 2 单击右窗格中的“分析”选项卡。

有关数据库中所有存储过程和函数的分析信息随即出现在“分析”选项卡上。

❖ 查看事件的摘要分析信息：

- 1 在左窗格中打开“事件”文件夹。

数据库中所有事件的列表将出现在右窗格中的“事件”选项卡上。

- 2 单击右窗格中的“分析”选项卡。

有关数据库中所有事件的分析信息随即出现在“分析”选项卡上。

❖ 查看触发器的摘要分析信息：

- 1 在左窗格中打开“触发器”文件夹。

数据库中所有触发器的列表会出现在“触发器”选项卡上。

- 2 单击右窗格中的“分析”选项卡。

有关数据库中所有触发器的分析信息随即出现在“分析”选项卡上。

❖ 查看系统触发器的摘要分析信息

- 1 在左窗格中打开“系统触发器”文件夹。

数据库中的所有触发器的列表会出现在“系统触发器”选项卡上。

- 2 单击右窗格中的“分析”选项卡。

有关数据库中所有系统触发器的分析信息随即出现在“分析”选项卡上。

查看特定过程的分析信息

Sybase IQ 提供有关单个存储过程、函数、事件和触发器的过程分析信息。Sybase Central 所显示的有关单个过程的信息不同于它所显示的有关数据库中所有存储过程、函数、事件或触发器的信息。

当您查看有关特定过程的分析信息时，以下列将出现：

- **Calls** 列出对象已被调用的次数。
- **Milliseconds** 列出各个对象的总执行时间。
- **Line** 列出每个过程行旁边的行号。
- **Source** 逐行显示 SQL 过程。

过程会分行显示，您可以查看哪些行具有较长的执行时间并因此可以通过更改来提高过程的性能。若要访问过程分析信息，您必须连接到数据库，启用分析并具有 DBA 授权。

❖ **查看存储过程或函数的分析信息：**

- 1 在左窗格中展开数据库。
- 2 在左窗格中选择 “过程和函数” 文件夹。
数据库中所有存储过程和函数的列表会出现在右窗格中的 “过程和函数” 选项卡上。
- 3 在左窗格中单击要分析的存储过程或函数。
- 4 单击右窗格中的 “分析” 选项卡。
有关特定存储过程或函数的分析信息随即出现在右窗格中的 “分析” 选项卡上。

❖ **查看事件的分析信息：**

- 1 在左窗格中展开数据库。
- 2 在左窗格中选择 “事件” 文件夹。
数据库中所有事件的列表会出现在右窗格中的 “事件” 选项卡上。
- 3 在左窗格中单击要分析的事件。
- 4 单击右窗格中的 “分析” 选项卡。
有关特定事件的分析信息随即出现在右窗格中的 “分析” 选项卡上。

❖ **查看触发器的分析信息：**

- 1 在左窗格中展开数据库。
- 2 在左窗格中打开 “触发器” 文件夹。
所有触发器的列表会出现在右窗格中的 “触发器” 选项卡上。
- 3 在右窗格中选择您要分析的触发器。
- 4 单击右窗格中的 “分析” 选项卡。
有关特定触发器的分析信息随即出现在右窗格中的 “分析” 选项卡上。

❖ 查看系统触发器的分析信息

- 1 在左窗格中展开数据库。
- 2 在左窗格中打开“系统触发器”文件夹。
所有系统触发器的列表会出现在右窗格中的“系统触发器”选项卡上。
- 3 在右窗格中选择您要分析的系统触发器。
- 4 单击右窗格中的“分析”选项卡。
有关特定系统触发器的分析信息随即出现在右窗格中的“分析”选项卡上。

在 Interactive SQL 中查看过程分析信息

您可以使用存储过程来查看过程分析信息。无论您是在 Sybase Central 中还是在 Interactive SQL 中查看分析信息，该信息都是相同的。

`sa_procedure_profile_summary` 存储过程提供有关数据库中所有过程的信息。您可以使用该过程在同一结果集中查看存储过程、函数、事件、系统触发器和触发器的分析数据。以下参数将限制该过程返回的行。

- **p_object_name** 指定要分析的对象名称。
- **p_owner_name** 指定要分析其对象的所有者。
- **p_table_name** 指定要分析触发器的表。
- **p_object_type** 指定要分析的对象类型。您可以从以下五个选项中进行选择。通过选择其中一个值，可以将结果集限制为指定类型的对象。
 - **P** 存储过程
 - **F** 函数
 - **T** 触发器
 - **E** 事件
 - **S** 系统触发器
- **p_ordering** 指定结果集的排序顺序。

请注意，由于一个过程可以调用其它过程，因此所列出的项目可能会多于用户专门调用的项目。

下面几节假定您已经作为具有 DBA 授权的用户连接到数据库并且过程分析已启用。

❖ **查看所有过程的摘要分析信息：**

- 1 执行 `sa_procedure_profile_summary` 存储过程。

例如，输入：

```
CALL sa_procedure_profile_summary
```

- 2 从“SQL”菜单中选择“执行”。

“结果”窗格中会出现一个结果集，该结果集包含有关数据库中所有过程的信息。

有关 `sa_procedure_profile_summary` 存储过程的详细信息，请参见《SQL Anywhere Server — SQL 参考》。

在 Interactive SQL 中查看特定过程的分析信息

`sa_procedure_profile` 存储过程提供特定过程中每一行的相关信息。结果集包括行号、执行时间以及占过程中所有行的总执行时间的百分比。您可以使用以下参数来限制该过程返回的行：

- **p_object_name** 指定要分析的对象名称。
- **p_owner_name** 指定要分析其对象的所有者。
- **p_table_name** 指定要分析触发器的表。

如果查询中没有包含任何参数，则该过程将返回已调用的所有过程的分析信息。

❖ **查看过程中特定行的分析信息：**

- 1 执行 `sa_procedure_profile` 存储过程。

例如，输入：

```
CALL sa_procedure_profile
```

- 2 从“SQL”菜单中选择“执行”。

“结果”窗格中会显示一个结果集，该结果集包含各个过程行的分析信息。

有关 `sa_procedure_profile` 存储过程的详细信息，请参见《SQL Anywhere Server — SQL 参考》。

监控缓冲区高速缓存

Sybase IQ 提供了相应工具以监控缓冲区高速缓存的性能。此监控器将收集与 Sybase IQ 中执行的缓冲区高速缓存、内存和 I/O 功能有关的统计信息，并将其存储在日志文件中。

缓冲区高速缓存性能是 Sybase IQ 的总体性能中的关键因素。使用监控器提供的信息，可以对分配给主和临时缓冲区高速缓存的内存数量进行微调。如果一个高速缓存正在执行明显比另一个高速缓存更多的 I/O，则重新分配少量内存，例如以迭代方式分配百分之十的高速缓存。在重新分配之后，请重新运行负载，并监控性能更改。

启动缓冲区高速缓存监控器

从 Interactive SQL 运行 Sybase IQ 缓冲区高速缓存监控器。在每次启动监控器时，它都会在 Sybase IQ 中作为单独内核线程运行。

使用此语法启动监控器：

```
IQ UTILITIES { MAIN | PRIVATE }  
INTO dummy_table_name  
START MONITOR 'monitor_options [ ... ]'
```

MAIN 启动对主缓冲区高速缓存的监控，监控针对您所连接的数据库的 IQ 存储中的所有表进行。

PRIVATE 启动对临时缓冲区高速缓存的监控，监控针对您所连接的数据库的临时存储区中的所有表进行。

您需要发出单独的命令，以监控每个缓冲区高速缓存。在监控器收集结果时，必须使每个会话保持打开状态；关闭其连接时，监控器的运行会停止。一个连接可以运行最多两个监控器运行，一个用于主缓冲区高速缓存，另一个用于临时缓冲区高速缓存。

dummy_table_name 可以是任何 Sybase IQ 基表或临时表。为了与其它 IQ UTILITIES 命令语法兼容，必须提供表名称。最好使用仅用于监控的表。

若要控制监控器输出文件的存放目录，请设置 **MONITOR_OUTPUT_DIRECTORY** 选项。如果不设置此选项，则监控器会将输出发送到数据库所在的相同目录。所有监控器输出文件均用于监控器运行期间。在监控器运行已停止之后，这些文件会保留下来。

在创建任何 **Multiplex** 查询服务器之前，要么声明在监控中使用的临时表，要么在创建新数据库时创建永久性伪表。这些解决方案避免了 DDL 更改，以便在生产运行期间数据在查询服务器上一直保持工作状态。

提示

若要简化监控器的使用，请创建存储过程以声明伪表，并指定其输出位置，然后启动监控器。

'*monitor_options*' 可以包括以下值中的一个或多个：

- **-summary** 显示主和临时缓冲区高速缓存的摘要信息。如果不指定任何监控器选项，则会收到摘要报告。所显示的字段是为其它选项描述的字段，外加以下字段：
 - *Users*: 连接到缓冲区高速缓存的用户数
 - *IO*: 由缓冲区高速缓存执行的组合物理读取数和写入数
- **-cache** 详细显示主或临时缓冲区高速缓存的活动。关键字段是 *Finds*、*HR%* 和 *BWaits*。显示的字段如下：
 - *Finds*: 对缓冲区高速缓存的查找请求数。如果 *Finds* 值突然降到零并保持不变，则服务器被死锁。当服务器有任何活动时，*Finds* 值应当是非零值。
 - *Creates*: 在数据库中创建页的请求数
 - *Dests*: 在数据库中销毁页的请求数
 - *Dirty*: 缓冲区脏化（修改）的次数
 - *HR%*: 命中率，在不请求任何 I/O 的情况下，缓冲区高速缓存满足查询的百分比。命中率越高越好，如果高速缓存设置得足够大，通常是 90% - 100%。对于大型查询，命中率可能一开始很低，但预取开始工作时则会增加。
 - *BWaits*: 强制等待繁忙页（页帧争用）的查找请求数。通常它很低，但在某些特殊情况下它可能很高。例如，如果同时启动相同查询，则都需要相同页，因此第二个请求必须等待第一个请求从磁盘获取页。
 - *ReReads*: 在相同事务中存储区的相同部分需要重读到高速缓存中的大约次数。应当始终很低，但对 Sybase IQ 12.4.2 和以上版本来说，高数字也没关系。
 - *FMiss*: 假丢失数，缓冲区高速缓存需要多次查找才能在内存中找到页的次数。此数字应当是 0 或非常小。如果该值很高，则可能发生了回退，并且需要重复某些操作。
 - *Cloned*: Sybase IQ 为了给写入程序建立新版本而需要的缓冲区数，而它必须为并发读取程序保留早期版本。仅当其它用户正在查看某页时，才复制该页。

- *Reads/Writes*: 由缓冲区高速缓存执行的物理读取和写入。
- *PF/PFRead*: 预取请求数和为预取完成的读取数。
- *GDirty*: LRU 缓冲区被脏抢夺并且 Sybase IQ 在使用它之前必须将其写出的次数。此值不应当长时间大于 0。如果是, 则可能需要增加清理程序线程数, 或移动清洗标记。
- *Pin%*: 缓冲区高速缓存中正在使用和锁定的页的百分比。
- *Dirty%*: 被修改的缓冲块的百分比。不要尝试让此值超过 85-90%; 否则, *GDirty* 将变为大于 0。
- *-cache_by_type* 可生成与 *-cache* 相同的结果, 但是前者通过 IQ 页类型进行细分。(例外是 *Bwaits* 列, 此列仅显示总计。) 当需要向 Sybase 技术支持部门提供信息时, 此格式最有用。
- *-file_suffix suffix* 可创建名为 `<dbname>.<connid>-<main_or_temp>-<suffix>` 的监控输出文件。如果不指定后缀, 则缺省情况下为 *iqmon*。
- *-io* 显示在指定时段内主或临时 (专用) 缓冲区高速缓存 I/O 率和压缩率。这些计数器表示服务器的所有活动; 这些信息不会被设备断开。显示的字段如下:
 - *Reads*: 由缓冲区高速缓存执行的物理读取数
 - *Lrd(KB)*: 读入的逻辑 KB 数 (页大小乘以请求数)
 - *Prd(KB)*: 读入的物理 KB 数
 - *Rratio*: 逻辑对物理读入数据的压缩率, 用于衡量磁盘读取的压缩效率
 - *Writes*: 缓冲区高速缓存执行的物理写入数
 - *Lwrt(KB)*: 逻辑写入 KB 数
 - *Pwrt(KB)*: 物理写入 KB 数
 - *Wratio*: 逻辑与物理写入数据的压缩率
- *-bufalloc* 显示有关主缓冲区分配器或临时缓冲区分配器的信息, 该分配器可在缓冲区高速缓存中保留空间以供排序、散列和位图等对象使用。
 - *OU*: *User_Resource_Reservation* 选项设置 (以前是 *Optimize_For_This_Many_Users*)
 - *AU*: 当前活动用户数
 - *MaxBuf*: 缓冲区分配器控制下的缓冲区数

- *Avail*: pin 配额分配的当前可用缓冲区数
- *AvPF*: 预取配额分配的当前可用缓冲区数
- *Slots*: 使用缓冲区高速缓存配额的当前注册对象数
- *PinUser*: 使用 pin 配额的对象数 (例如散列、排序和 B 树对象)
- *PFUsr*: 使用预取配额的对象数
- *Posted*: 预先计划配额用户的对象数
- *UnPost*: 作为特别配额用户的对象数
- *Locks*: 在缓冲区分配器上执行的互斥锁定数
- *Waits*: 线程必须等待锁定的次数
- *-contention* 显示很多关键的缓冲区高速缓存和内存管理器锁。这些锁定和互斥计数器显示在缓冲区高速缓存和堆内存中的活动, 以及解除这些锁定的速度。监视超时数。如果系统时间超过 20%, 则它指示出现问题。

注释 由于操作系统的改进, Sybase IQ 不再使用旋转锁定。因此, 很少使用 *woTO*、*Loops* 和 *TOs* 统计信息。

- *AU*: 当前活动用户数
- *LRULks*: LRU 的锁定次数 (对临时高速缓存重复)
- *woTO*: 授予锁定而没有超时的次数 (对临时高速缓存重复)
- *Loops*: 在授予锁定之前 Sybase IQ 重试的次数 (对临时高速缓存重复)
- *TOs*: Sybase IQ 超时并且必须等待锁定的次数 (对临时高速缓存重复)
- *BWaits*: 高速缓存中的缓冲区“忙等待”数 (对临时高速缓存重复)
- *IOLock*: Sybase IQ 锁定压缩 I/O 池的次数 (对临时高速缓存重复); 可以被忽略
- *IOWait*: Sybase IQ 必须等待对压缩 I/O 池的锁定的次数 (对临时高速缓存重复); 可以被忽略
- *HTLock*: Sybase IQ 锁定块映射散列表的次数 (对临时高速缓存重复)

- *HTWait*: Sybase IQ 必须等待块映射散列表的次数（对临时高速缓存重复）；*HTLock* 和 *HTWait* 指示您正在使用多少块映射
- *FLLock*: Sybase IQ 必须锁定空闲列表的次数（对临时高速缓存重复）
- *FLWait*: Sybase IQ 必须等待对空闲列表的锁定的次数（对临时高速缓存重复）
- *MemLks*: Sybase IQ 取得内存管理器（堆）锁定的次数
- *MemWts*: Sybase IQ 必须等待内存管理器锁定的次数
- *-threads* 显示处理线程管理器所使用的计数器。值是全服务器范围的（即无论对主或专用选择此选项都没关系）。它们表示在报告的最后一页后出现的新事件。
 - *cpus*: Sybase IQ 正在使用的 CPU 数；它可能小于系统上的数字
 - *Limit*: Sybase IQ 可以使用的最大线程数
 - *NTeams*: 当前在使用中的线程组数
 - *MaxTms*: 曾经在使用中的最大组数
 - *NThrds*: 当前的现有线程数
 - *Resrvd*: 为系统（连接）使用而保留的线程数
 - *Free*: 可供分配的线程数。请监控此值，如果它非常低，则指示线程匮乏
 - *Locks*: 对线程管理器使用的锁定数
 - *Waits*: Sybase IQ 必须等待对线程管理器的锁定的次数

注释 对象或查询需要工作时， Sybase IQ 将分配一组名为线程组的处理线程。在调整线程使用时有用的选项包括数据库选项 `MAX_IQ_THREADS_PER_CONNECTION` 和 `MAX_IQ_THREADS_PER_TEAM`，以及用于指定 Sybase IQ 可以使用的线程数的服务器选项 `-iqmt`。

- *-interval* 可指定报告间隔（秒）。缺省值为每 60 秒。最小值为每 2 秒。通过在查询期间或有性能问题的时间按缺省间隔运行监控器，通常可以获取有用的结果。非常短的间隔可能不会给出有意义的结果。间隔应当与作业时间成比例；通常，一分钟已足够。第一个显示内容显示自服务器启动以来的计数器。后续显示则显示与前一显示不同的地方。

- `-append` | `-truncate` 分别是追加到现有输出文件的末尾，或截断现有输出文件。Truncate 是缺省值。
- `-debug` 主要用于向 Sybase 技术支持部门提供信息。它显示对性能监控器可用的所有信息，无论是否有覆盖相同信息的标准显示模式。页首是按磁盘块类型分类的统计信息阵列。其后是其它缓冲区高速缓存统计信息、内存管理器统计信息、线程管理器统计信息、空闲列表统计信息、CPU 利用率和最终缓冲区分配器统计信息。然后，缓冲区分配器统计信息按客户端类型（散列、排序等）分类，并显示最近分配的缓冲区柱状图。请注意，内存分配内容指示有多少是在报告的最后一页后分配的。

注释 间隔（有两个例外）应用于每行输出，而不是每页。例外是 `-cache_by_type` 和 `-debug`，这时，每个显示内容以新页开始。

在监控器运行时检查结果

在 UNIX 系统上，可以在查询正在运行时监视监控器输出。

例如，可以使用以下命令启动监控器：

```
iq utilities main into monitor_tab
start monitor "-cache -interval 2 -file_suffix iqmon"
```

此命令将输出发送到名为 `dbname.conn#-[main|temp]-iqmon` 的 ASCII 文件。因此，对于数据库 `iqdemo`，结果会发送到 `iqdemo.2-main-iqmon`。

若要监视结果，请在系统提示下发出以下命令：

```
$ tail -f iqdemo.2-main-iqmon
```

停止缓冲区高速缓存监控器

用于停止监控器运行的命令与用于启动它的命令相似，只是不需要指定任何选项。使用此语法可以停止 Sybase IQ 缓冲区高速缓存监控器：

```
IQ UTILITIES { MAIN | PRIVATE }
INTO dummy_table_name STOP MONITOR
```

注释 为了让某些选项设置生效，必须重新启动数据库。如果监控器正在运行，则需要关闭它，以便数据库可以重新启动。

检查并保存监控器结果

监控器在普通文本文件中存储结果。此文件缺省为：

- `dbname.connection#-main-iqmon`（对于主缓冲区高速缓存结果）
- `dbname.connection#-temp-iqmon`（对于临时缓冲区高速缓存结果）

前缀 `dbname.connection#` 表示数据库名称和连接号。如果看见多个连接号，并且不能确定哪个是您的，则可以运行 `Catalog` 存储过程 `sa_conn_info`。此过程将显示与数据库的每个活动连接的连接号、用户 ID 和其它信息。

可以在 `IQ UTILITIES` 命令中使用 `-file_suffix` 参数，将后缀 `iqmon` 更改为您选择的后缀。

若要查看监控器运行的结果，请使用您通常用于显示或打印文件的文本编辑器或任何其它方法。

当再次从相同数据库和连接号运行监控器时，缺省情况下它将覆盖以前的结果。如果需要保存监控器运行的结果，请在从相同数据库再次启动监控器之前将文件复制到另一个位置，或使用 `-append` 选项。

监控器结果的示例

本节使用不同监控器选项来显示示例结果。

`-summary` 选项产生像下面这样的结果。注意，无论在 `IQ UTILITIES` 命令中请求哪个，它都会同时显示主和临时缓冲区高速缓存统计信息：

```
Sybase Adaptive Server IQ Performance Monitor
-----
Version 3.2

Options string for Main cache: "-summary -interval 5"

Summary
2004-07-16 13:53:24

Active|          Main Cache          |          Temp Cache
Users| Finds  HR% Reads/Writes GDirty Pin% Dirty% InUse%| Finds  HR%
Reads/Writes GDirty Pin% Dirty% InUse%
      0   286 99.3    2/34      0  0.0  1.6  26.2   608 99.7
2/47      0  0.0    3.6  20.0
```

监控缓冲区高速缓存

1	2621	99.4	16/155	0	5.6	8.7	81.7	4121	99.6
16/163		0	11.4	23.2	67.3				
1	2646	99.8	6/48	0	1.6	13.5	100.0	3388	99.8
6/70		1	4.1	40.9	94.5				
1	2684	99.9	7/78	0	5.6	14.3	100.0	3497	99.9
8/103		1	10.9	42.3	99.1				
1	1993	99.9	17/22	0	4.0	31.0	100.0	3342	98.7
122/149		0	8.2	41.4	91.4				
1	2479	99.9	32/110	0	5.6	13.5	100.0	3370	99.8
55/112		0	11.4	45.5	95.9				
1	3273	100.0	0/0	0	5.6	23.8	100.0	3951	100.0
0/108		1	13.6	49.1	100.0				
1	2512	99.9	2/0	0	1.6	31.0	100.0	3916	98.9
88/173		0	5.5	48.6	100.0				
1	1264	99.9	66/131	0	4.0	45.2	100.0	4317	98.9
378/305		0	6.4	40.0	77.3				
1	2122	99.8	30/125	0	5.6	12.7	99.2	3122	99.7
67/127		0	12.3	40.0	90.5				
1	3370	100.0	2/0	0	5.6	23.0	100.0	4034	100.0
2/98		2	13.2	46.4	98.2				
1	2981	99.9	2/0	0	5.6	31.7	100.0	3715	99.9
2/110		0	14.1	53.2	100.0				
1	3351	99.6	13/3	0	5.6	39.7	100.0	4131	99.7
13/123		0	14.1	57.7	100.0				
1	3286	99.6	13/13	0	5.6	40.5	100.0	4135	99.6
15/139		0	12.3	55.9	97.7				
1	296	100.0	0/0	0	1.6	41.3	100.0	3646	96.9
366/320		0	7.3	53.2	100.0				
1	1230	99.4	71/129	0	6.3	58.7	100.0	4221	98.9
390/297		0	9.5	59.1	91.8				
1	1900	100.0	125/279	0	4.0	50.0	100.0	4102	100.0
344/279		0	7.7	38.6	72.3				

Sybase Adaptive Server IQ Performance Monitor

Shutting Down


```

0      422 98.8    16/99      0  0.0    0.8    99.2    853 98.9
34/101      0  0.0    1.8    59.1

```

`-cache` 选项产生类似下面的结果，它针对临时缓冲区高速缓存。

Options string for Temp cache: "-cache -interval 10"

```

Temp Shared Buffer Cache
2001-02-18 17:43:55
Finds Creats DestS Dirty  HR% BWaitS ReReadS FMiss Cloned Reads/  PF/
GDirty Pin% Dirty%
Writes PFRead
Tm:  640    82   57   84 99.4    0    4    0    0    4/0  0/0
0 0.0  2.8
Tm: 1139   109   83  109 100.0    0    0    0    0    0/0  0/0
0 0.0  5.5
Tm: 6794   754  749  754 100.0    0    0    0    0    0/0  0/0
0 0.0  6.1
Tm: 10759 1646 1646 1646 100.0    0    0    0    0    0/0  0/0
0 0.0  6.1

```

`-io` 选项产生类似下面的结果，它针对主要缓冲区高速缓存：

Options string for main cache:"-IO -interval 5"

```

Main Buffer Cache
2001-02-18 13:58:48
Input
Reads  Lrd(KB) Prd(KB)  Rratio  Writes  Output
Lwrt(KB) Pwrt(KB)  Wratio
Mn:   10    40    34    1.18    14     56    23    2.43
Mn:    0     0     0    0.00    21     84    34    2.43
Mn:    0     0     0    0.00     7     28    11    2.43
Mn:    0     0     0    0.00    22     88    35    2.48
Mn:    0     0     0    0.00    63    252   100    2.51
Mn:    0     0     0    0.00    54    216    93    2.32
Mn:    0     0     0    0.00    64    256   101    2.52
Mn:    0     0     0    0.00    62    248    94    2.62
Mn:    0     0     0    0.00    73    292   110    2.65
Mn:    0     0     0    0.00   105   420   121    3.47

```

`-bufalloc` 选项产生类似下面的结果。

Options string for Main cache: "-bufalloc -file_suffix bufalloc-iqmon -append -interval 10"

```

Buffer Allocation
2001-02-18 10:58:39
OU/AU MaxBuf Avail AvPF Slots PinUsr PFUsr Posted UnPost Quota Locks Waits

```

1/0	1592	1592	20	0	0	0	0	0	0	1	0
1/1	1592	1592	20	0	0	0	0	0	0	1	0
1/1	1592	1592	20	0	0	0	0	0	0	1	0

注释 实际的 `-contention` 输出将在同一行上显示 Main 高速缓存、临时高速缓存和内存管理器。因为此格式非常宽，所以这些列集中的每一个都在这里单独显示。

main 高速缓存的 `-contention` 结果是：

Options string for Main cache:

```
"-contention -file_suffix contention-iqmon -append -interval 10"
      Contention
      2001-02-18 10:57:03
```

Main Cache											
AU	LRULks	woTO	Loops	TOs	BWaits	IOLock	IOWait	HTLock	HTWait	FLLock	FLWait
0	66	0	0	0	0	1	0	5	0	4	0
1	2958	0	0	0	0	160	0	1117	0	6	0
1	1513	0	0	0	1	378	0	2	0	8	0
1	370	0	0	0	0	94	0	2	0	10	0
1	156	0	0	0	0	46	0	2	0	12	0
1	885	0	0	0	0	248	0	2	0	14	0
1	1223	0	0	0	0	332	1	2	0	16	0
1	346	0	0	0	0	66	0	2	0	18	0

临时高速缓存的 `-contention` 结果是：

Temp Cache											
LRULks	woTO	Loops	TOs	BWaits	IOLock	IOWait	HTLock	HTWait	FLLock	FLWait	
70	0	0	0	0	1	0	4	0	5	0	
466	0	0	0	0	2	0	15	0	12	0	
963	0	0	0	0	2	0	8	0	20	1	
1186	0	0	0	0	2	0	2	0	23	1	
357	0	0	0	0	2	0	2	0	25	1	
444	0	0	0	0	2	0	3	0	29	0	
884	0	0	0	0	2	0	2	0	31	1	
1573	0	0	0	0	2	0	5	0	37	1	

内存管理器的结果是：

```
| Memory Mgr
MemLks MemWts
55483   13
5705    0
2048    0
186     4
2        0
137     0
22      0
203     3
```

-threads 选项的结果像下面这样：

```
Options string for Main cache: "--threads -file_suffix threads-iqmon -append -
interval 10"
```

Threads

2001-02-18 10:59:24

CPUs	Limit	NTeams	MaxTms	NThrds	Resrvd	Free	Locks	Waits
10	100	4	12	100	13	68	106	590
10	100	6	12	100	12	63	4	6
10	100	6	12	100	12	63	0	0
10	100	7	12	100	12	62	1	1
10	100	7	12	100	12	62	0	0
10	100	7	12	100	12	58	1	5
10	100	7	12	100	12	58	0	0

缓冲区高速缓存结构

Sybase IQ 按照系统的 CPU 数自动计算缓冲区高速缓存的高速缓存分区数。如果多 CPU 配置中的装载或查询性能比期望值慢，可能能够通过更改 `CACHE_PARTITIONS` 数据库选项的值来提高它。有关详细信息，请参见《参考：语句和选项》中的“`CACHE_PARTITIONS` 选项”。

当缓冲区接近高速缓存的最近使用最少 (LRU) 末尾时，它们将越过清洗标记。Sybase IQ 会将最旧页（那些越过清洗标记的页）写出到磁盘中，以便可以重用它们占据的高速缓存空间。一组 Sybase IQ 处理线程（名为清理程序线程）将清除（写入）最旧的缓冲区。

当 Sybase IQ 需要将一页数据读取到高速缓存中时，它会抢夺 LRU 缓冲区。如果缓冲区仍然是“脏的”（已修改），则它必须首先写入到磁盘中。监控器 `-cache` 报告中的 `Gdirty` 列会显示 LRU 缓冲区被脏抢夺并且 Sybase IQ 在使用它之前必须先将它写出到磁盘的次数。

通常，Sybase IQ 能够使 `Gdirty` 值保持为 0。如果此值在较长时段内大于 0，则可能需要调整某个用于控制清理程序线程数和清洗标记的数据库选项。请参见《参考：语句和选项》中的“`SWEEPER_THREADS_PERCENT` 选项”或“`WASH_AREA_BUFFERS_PERCENT` 选项”。

避免缓冲区管理器抖动

系统必须先写入脏页然后才能读取请求页的情况下会发生抖动，这会显著降低系统速度。为获得最佳性能，请始终分配足够可用的内存，从而使页写入程序满足可用空间需求。

缓冲区高速缓存抖动与系统抖动类似，且在没有足够可用于读取的清洁缓冲区的情况下发生。这导致在高速缓存中出现相同类型的“先写入然后读取”延迟，且可能会在没有足够大的缓冲区高速缓存来容纳查询中引用的所有对象的情况下发生。

若要消除缓冲区高速缓存抖动，必须为缓冲区高速缓存分配更多内存。请不要过多分配缓冲区高速缓存。分配过多内存可能会导致为数据库缓冲区高速缓存分配内存时发生系统抖动。在极端情况下，没有解决缓冲区高速缓存抖动问题时分配过多内存可能会引起多级别的抖动。

在多用户环境中，或者查询复杂性导致的曲解或不确定性使优化程序在某情况下（在该情况中需要用明显多于查询可用的高速缓存所容纳的数值目生成 HASH 对象）选择 HASH 算法时可能会出现其它不易发现形式的缓冲区高速缓存抖动。

设置缓冲区大小时，请记住以下事项：

- 如果 Sybase IQ 缓冲区高速缓存太大，则在 Sybase IQ 尝试使用所有这部分内存时，操作系统将强制分页。
- 如果 Sybase IQ 缓冲区高速缓存太小，那么 Sybase IQ 会发生抖动，因为它不能将足够的查询数据填到高速缓存中。

如果正在遇到剧烈的性能问题，则应当监视分页，以确定是否是抖动问题。如果是，请按照“[管理缓冲区高速缓存](#)”中的描述重置缓冲区大小。

如果您监控分页并确定存在抖动问题，您还可以限制在执行包含涉及散列算法的查询的语句过程中的抖动量。通过调整 `HASH_THRASHING_PERCENT` 数据库选项，可以控制在语句回退且返回错误之前允许的硬盘 I/O 百分比。

`HASH_THRASHING_PERCENT` 的缺省值是 10%。增大 `HASH_THRASHING_PERCENT` 可允许在回退之前对磁盘进行更多的分页，而减小 `HASH_THRASHING_PERCENT` 可允许在回退之前对磁盘进行较少的分页。

如果查询涉及在 Sybase IQ 的更早版本中执行的散列算法，则这些查询现在可能在达到缺省的 `HASH_THRASHING_PERCENT` 限制时发生回滚。Sybase IQ 会报告错误 `Hash insert thrashing detected` 或 `Hash find thrashing detected`。执行以下一项或多项操作可为查询提供执行所需的资源：

- 通过增大 `HASH_THRASHING_PERCENT` 的值来放宽分页限制。
- 增加临时高速缓存（仅限 DBA）的大小。请记住，增加临时高速缓存的大小要求在 `main` 高速缓存分配中减少相等大小以防止系统抖动的可能性。
- 对于此语句，尝试确定 Sybase IQ 错误估计一个或多个散列大小的原因并缓解该错误估计所产生的后果。例如，请检查需要 LF 或 HG 索引的所有列都有一个。而且，请考虑多列索引是否合适。
- 减少数据库选项 `HASH_PINNABLE_CACHE_PERCENT` 的值。

有关这些数据库选项的详细信息，请参见《参考：语句和选项》中的“`HASH_THRASHING_PERCENT` 选项”或“`HASH_PINNABLE_CACHE_PERCENT` 选项”。

若要发现查询中可能的问题，请用临时数据库选项 `QUERY_PLAN = 'ON'` 和 `QUERY_DETAIL = 'ON'` 运行查询，以生成查询计划，然后检查查询计划中的估计值。生成的查询计划存在于消息日志文件中。

监控 Windows 系统上的分页

Windows 提供了系统监控器，以帮助监控分页。若要访问它，请选择对象“Logical Disk”（包含文件 *PAGEFILE.SYS* 的磁盘实例）和计数器 Disk Transfers/Sec。将 Windows 页面文件放在不同于数据库 *dbspace* 设备的磁盘上。还可以监控对象“Memory”和计数器 Pages/Sec。但是，该值是包括软错误和硬错误的所有内存错误的总和。

监控 UNIX 系统上的分页

UNIX 提供的系统命令 *vmstat* 可以帮助您监控诸如分页这样的系统活动。简短命令语法是：

vmstat interval count

interval 是输出行之间的时间，*count* 是输出行的显示次数。有关 *vmstat*（包括其选项和字段说明）的详细信息，请参见操作系统的文档。下面是一个示例：

```
> vmstat 2 3
procs      memory                page          disk          faults        cpu
r  b  w  swap      free  re  mf  pi  po  fr  de  sr  s0  s1  sd  in  sy  cs  us  sy  id
0  0  0  3312376  31840  0  8  0  0  0  0  0  0  0  0  297  201  472  82  4  14
0  0  0  3312376  31484  2  3  0  0  0  0  0  0  0  0  260  169  597  80  3  17
0  0  0  3312368  31116  0  8  0  0  0  0  0  0  0  0  205  1202  396  67  4  29
```

上面的输出显示稳定的 Sybase IQ 查询状态，其中，计算机的物理内存尚未过度分配。几乎没有发生系统页错误。下一组示例显示指示问题的 *vmstat* 输出。（显示的输出省略了上面的一些字段，以便更好地适应显示页面。）

```
procs      memory                page          faults        cpu
r  b  w  swap      free  re  mf  pi  po  fr  de  sr  in  sy  cs  us  sy  id
0  0  0  217348  272784  0  148  11  3  9  0  2  251  1835  601  6  3  91
0  0  0  3487124  205572  0  5  0  0  0  0  0  86  131  133  0  1  99
0  0  0  3487124  205572  0  5  0  0  0  0  0  71  162  121  0  0  100
0  0  0  3483912  204500  0  425  36  0  0  0  0  169  642  355  2  2  96
0  0  0  3482740  203372  0  17  6  0  0  0  0  158  370  210  1  3  97
0  0  0  3482676  203300  0  4  10  0  0  0  0  160  1344  225  1  2  97
0  0  0  3343272  199964  1  2123  36  0  0  0  0  213  131  399  7  8  85
0  0  0  3343264  185096  0  194  84  0  0  0  0  283  796  732  1  6  93
0  0  0  3342988  183972  0  17  58  0  0  0  0  276  1051  746  2  4  94
0  0  0  3342860  183632  0  119  314  0  0  0  0  203  1660  529  3  4  94
```

```

0 0 0 3342748 182316 2 109 184 0 0 0 0 187 620 488 4 2 95
0 0 0 3342312 181104 2 147 96 0 0 0 0 115 256 260 9 2 89
0 0 0 3340748 179180 0 899 26 0 0 0 0 163 836 531 4 4 92
0 0 0 3328704 167224 0 2993 6 0 0 0 0 82 2195 222 4 7 89

```

上面的第一行输出提供了启动计算机后的系统活动摘要。前三行显示大约有 200MB 可用物理内存，并且计算机处于空闲。第四行对应于 Sybase IQ 第一次启动。从第八行开始，可用内存的数量开始迅速减少。这对应于正在分配的 Sybase IQ 缓冲区高速缓存和正在从磁盘读取的数据库页（注意 CPU 使用量已增加）。这时，几乎没有用户 CPU 时间，因为还没有开始执行查询。

```

procs      memory      page      faults      cpu
r  b w swap   free  re mf pi  po fr de sr in  sy  cs  us sy id

 7 0 0 3247636 58920 0 1880 1664 0 0 0 0 1131 442 1668 80 18 3
18 0 0 3246568 43732 0 709 1696 0 0 0 0 1084 223 1308 90 10 1
12 0 0 3246604 37004 0 358 656 0 0 0 0 600 236 722 95 5 0
15 0 0 3246628 32156 0 356 1606 0 0 0 0 1141 226 1317 91 9 0
19 0 0 3246612 26748 0 273 1248 0 0 0 0 950 394 1180 92 7 0

```

当随后查询开始时，上面的输出有轻微变化。这在用户模式 CPU 级别（*us* 字段）很明显。缓冲区高速缓存尚未填满，因为页进入错误（*pi* 字段或 KB 分页进入）仍然正在发生，并且可用内存的数量仍然在降低。

```

procs      memory      page      faults      cpu
r  b w swap   free  re mf pi  po fr de sr in  sy  cs  us sy id

21 0 0 3246608 22100 0 201 1600 0 0 0 0 1208 1257 1413 88 12 0
18 0 0 3246608 17196 0 370 1520 0 464 0 139 988 209 1155 91 8 0
11 0 0 3251116 16664 0 483 2064 138 2408 0 760 1315 218 1488 88 12 0
30 0 0 3251112 15764 0 475 2480 310 4450 0 1432 1498 199 1717 87 13 0

```

上面的输出在随后更晚出现。在第三行输出上，它显示系统已经达到它可以维护的可用内存数量的阈值。在此点上，发生了分页（*po* 字段或 KB 被分页），并且系统模式 CPU（*sy* 字段）的级别随即增加。产生此情况是因为过度分配了物理内存：Sybase IQ 缓冲区高速缓存对该计算机太大。若要解决此问题，请减少两个缓冲区高速缓存中的某一个的大小，或二者都减少。

缓冲区高速缓存监控清单

下表总结了要在监控器结果中查找的最常见项，并提供了如果行为在正常范围以外则可能需要执行的建议操作。“统计信息”列列出了在标准监控器报告中显示的名称；如果此统计内容与调试报告中的内容不同，还会列出调试统计。

请记住，对于任何监控器统计，当系统更改状态时（比如当新查询正在启动时），都可能发生临时的非正常行为。

表 5-11: 缓冲区高速缓存监控清单

统计信息	正常行为	需要调整的行为	建议操作
HR%（高速缓存命中率）	<p>高于 90%。</p> <p>对于单个内部数据结构，比如 garray、barray、位图 (bm)、散列对象、排序对象、变长 btree (btreev)、定长 btree (btreef)、位向量 (bv)、dbext、dbid、vdo、存储、检查点块 (ckpt)，在查询运行时命中率应当是超过 90%。它可能首先低于 90%。一旦预取开始工作（PF 或 PrefetchReqs > 0），则命中率应当逐渐增长到超过 90%。</p>	<p>在预取工作之后，命中率低于 90%。</p> <hr/> <p>注释 某些对象不执行预取，因此其命中率在正常情况下可能很低。</p>	<p>尝试通过调整 -iqmc 和 -iqtc 重新平衡 main 高速缓存和临时高速缓存的大小。</p> <p>还应通过调整 PREFETCH_THREADS_PERCENT 选项，尝试增加预取线程数。</p>
Gdirty（脏抢夺）	<p>在有中等高速缓存大小 (< 10GB) 的系统中为 0。</p>	<p>GDirty > 0</p> <hr/> <p>注释 仅当脏页的数目达到清洗区域的某个百分比时，才会激活清理程序线程。如果 GDirty/GrabbedDirty 大于 0 并且 I/O 率（写入）很低，则系统可能直接被轻装载，并且不需要执行操作。</p>	<p>调整 SWEEPER_THREADS_PERCENT 选项（缺省值 10%）或 WASH_AREA_BUFFERS_PERCENT 选项（缺省值 20%），以增加清洗区域的大小。</p>

统计信息	正常行为	需要调整的行为	建议操作
BWaits (缓冲区忙等待)	0	持久 > 0, 表示多个作业正在争夺同一个缓冲区。	如果 I/O 率 (写入) 很高, 则忙等待可能是高速缓存抖动导致的。请检查高速缓存报告中的命中率, 以确定是否需要重新平衡 main 高速缓存与临时高速缓存。如果批量作业正在同时启动很多几乎相同的查询, 请尝试错开启动时间。
LRU 等待 (调试报告中的 LRUNum TimeOuts 百分比)	20% 或更少	> 20%, 指示发生严重争用问题。	检查操作系统修补程序级别和其它环境设置。此问题常常是操作系统问题。
IOWait (IOWaits)	10% 或更低	> 10%	检查是否有磁盘错误或 I/O 重试
FLWait (FLMutex Waits)	20% 或更低	> 20%	检查 dbSPACE 配置: 数据库空间即将不足? DISK_STRIPING 是否为 ON? sp_iqcheckdb 是否报告大于 15% 的碎片?
HTWait (BmapHTNum Waits) MemWts (MemNtimes Waited) (PFMgrCondVar Waits)	10% 或更低	> 10%	与 Sybase 技术支持部门联系。

统计信息	正常行为	需要调整的行为	建议操作
CPU 时间 (调试报告中的 CPU Sys Seconds, CPU Total Seconds)	CPU 系统时间 (秒) < 20%	CPU 系统时间 (秒) > 20% 如果 CPU Total Seconds 还报告低利用率, 并有让系统忙的足够作业, 则高速缓存可能正在抖动, 或可能丢失平行性。	调整 <code>-iqgovern</code> 以减少允许的并发查询总数。 检查高速缓存报告中的命中率和 I/O 率, 确定是否有高速缓存抖动。 还应查看 <code>cache_by_type</code> (或调试) 报告中散列对象的命中率, 以确定散列对象是否正在抖动: 在 I/O 率 (写入) 很高时它是否为 <90%? 检查查询计划中所尝试的平行性。是否有足够线程可用? 系统是否有非常庞大的 CPU 数? 可能需要诸如 Multiplex 配置这样的策略。
InUse% (使用中的缓冲区)	除了启动期间以外等于或接近 100%	小于大约 100%	缓冲区高速缓存可能太大。 尝试通过调整 <code>-iqmc</code> 和 <code>-iqtc</code> 重新平衡 main 高速缓存和临时高速缓存的大小。
Pin% (已占用的缓冲区)	< 90%	> 90 到 95%, 指示系统正在危险地接近缓冲区不足条件, 而这会导致事务回滚	尝试重新平衡 main 高速缓存与临时高速缓存的大小。 如果不可能重新平衡缓冲区高速缓存大小, 请尝试减少 <code>-iqgovern</code> , 以限制并发运行的作业数。

统计信息	正常行为	需要调整的行为	建议操作
可用线程 (ThrNumFree)	可用 > 保留	如果可用线程数降低至保留计数，则系统可能发生线程空闲。	尝试下列操作之一： 通过设置 <code>-iqmt</code> ，增加线程数。 减少与线程相关的选项： <code>MAX_IQ_THREADS_PER_CONNECTION</code> ， <code>MAX_IQ_THREADS_PER_TEAM</code> 。 通过设置 <code>USER_RESOURCE_RESERVATION</code> ，限制查询引擎资源的分配。 通过设置 <code>-iqgovern</code> ，限制作业数。
FIOutOfSpace (仅调试)	0，指示此存储区的空闲列表不完整；未分配的页可用	1，指示已完全分配此存储 (main 或临时)	将更多 <code>dbspace</code> 添加到该存储

监控 CPU 使用的系统实用程序

除了 Sybase Central 中的性能监控器外，还可使用操作系统实用程序监视在使用 Sybase IQ 时的 CPU 使用率。

操作系统	实用程序	说明
UNIX	<code>top</code> (Sun、Linux、HP-UX)， <code>topas</code> (IBM-AIX)	实时地持续查看处理器活动。
	<code>ps</code>	报告进程状态。
	<code>vmstat</code>	显示有关系统进程、内存、分页、块 IQ、捕获和 CPU 活动的信息。
	<code>iostat -x</code>	显示磁盘子系统信息。
Windows	系统监控器 任务管理器	提供有关计算机性能和正在运行的应用程序、进程、CPU 使用率及其它系统服务的详细信息。

在 32 位 Windows 系统上调优服务器

本章介绍了在 Windows 系统上运行 Sybase IQ 的性能和调优准则。阅读本章时，请结合阅读第 4 章“管理系统资源”。

主题	页码
一般性能准则	121
监控性能	123
使用 NTFS 高速缓存	125
对插入和查询进行调优	125
对备份操作进行调优	126

一般性能准则

以下为应用于装载和查询数据的一般准则。在 Windows 下运行 Sybase IQ 的推荐最低内存 (RAM) 为 512MB。对于大多数应用程序，建议采用 4GB 以获得最佳性能。在 boot.ini 中设置 /3GB 和 /PAE 开关，以允许 Sybase IQ 在 Windows 32 位系统上寻址尽可能多的内存。

最大化吞吐量

如果正在 Windows 上运行，请确保网络服务服务器选项“最大化网络应用程序数据吞吐量”已启用。

❖ 最大化网络应用程序数据吞吐量

- 1 在“控制面板”上，双击“网络连接”，右键单击“本地连接”，选择“属性”。
- 2 选择“Microsoft 网络的文件和打印机共享”，然后单击“属性”。
- 3 在“优化”下，选择“最大化网络应用程序数据吞吐量”。

注释 在某些 Windows 版本上，可能需要安装 Microsoft Internet Information Services (IIS) 才能将服务器属性设置为最大化网络应用程序数据吞吐量。

防止内存过度分配

系统页面错误过多是由于计算机上物理内存 (RAM) 的过度分配造成的。过多的页面错误可大幅降低 Sybase IQ 的性能。可通过认真分配 Sybase IQ 缓冲区、监控 Sybase IQ 进程的虚拟地址空间以及监控可用的物理内存，防止内存的过度分配。此节介绍监控 Sybase IQ 对计算机物理内存的使用情况的准则。

监控物理内存

可供应用程序 (Sybase IQ) 使用的物理内存量显示在“物理内存 (K)”下方。如果“可用数”值始终低于 5000，则计算机的物理内存可能存在过度分配。这是因为，在达到 5000(K) 标志时，Windows 会产生页面错误以维持最小 5MB 可用内存。

若要监控物理内存，请从“任务管理器”小程序中，选择“性能”选项卡。

文件系统

Windows 文件系统支持在文件、目录和卷级别进行压缩。*检查压缩选项，并对存储 Sybase IQ 数据库的所有磁盘和卷禁用 Windows 文件系统压缩。*这是因为 Sybase IQ 提供内置压缩功能。文件系统压缩不但不能进一步减少数据库大小，而且可能会在执行读取或写入时增加 CPU 开销。

监控性能

您可将 Sybase IQ 性能监视器作为监控 Sybase IQ 性能的首选工具，第5章“[监控和调优性能](#)”对它进行了介绍。不过，您还可以使用 Windows 任务管理器和 Windows 性能工具监控系统性能。

任务管理器

Windows 任务管理器提供有关计算机上正在运行的应用程序和进程的信息。任务管理器还显示有关计算机性能和资源的详细信息。

- 若要打开“任务管理器”，请右键单击任务栏上的空白空间，选择“任务管理器”。
- 有关使用 Windows 任务管理器的详细信息，请在“Windows 任务管理器”中单击“帮助”，然后选择“任务管理器帮助主题”。

性能

Windows 性能工具包括系统监视器与性能日志和警报。系统监视器显示有关内存、磁盘、处理器和网络活动的实时数据。通过性能日志和警报，您可以记录性能数据以及为特定性能活动设置警报。

- 若要打开 Windows 性能工具，请在“控制面板”上，单击“管理工具”，然后选择“性能”。
- 有关使用性能工具的详细信息，请在“性能”中单击“帮助”，然后选择“帮助主题”。

监控虚拟地址空间和工作集

进程的虚拟地址空间为该进程的总大小。进程的工作集为当前分配给该进程的物理内存大小。在大多数情况下，为了防止出现过多的系统页面错误，Sybase IQ 进程的虚拟地址空间应小于计算机的物理内存。

由于在 Sybase IQ 服务器上使用了虚拟内存模式，虚拟内存碎片可能会导致 Windows 平台上的处理工作大幅增加。为了减少出现这种情形的可能性，Sybase IQ 支持在 Windows XP 和 Windows Server 2003 上使用 Microsoft 的低碎片堆 (LFH)。

❖ **监控虚拟地址空间和内存使用情况**

- 1 右键单击任务栏上的空白空间，选择“任务管理器”，然后单击“进程”选项卡。
- 2 单击“查看”，然后选择“选择列”。
- 3 在“选择列”对话框上，选择以下列：
 - 内存使用
 - 内存使用增量
 - 内存使用高峰值
 - 页面错误
 - 页面错误增量
 - 虚拟内存大小

监控页面错误

从 Windows 性能监视器中，按照上述说明选择 Sybase IQ 进程。选择计数器“Page Faults/sec”。此计数器的结果中包括“软”页面错误和“硬”页面错误。硬页面错误是因磁盘 I/O 导致的页面错误。软页面错误则一般不是性能问题。

若要确定硬页面错误的数量，请选择对象“LogicalDisk”和文件 *pagefile.sys* 所在位置的实例（该实例应该与 Sybase IQ 数据库不在同一个卷上）。选择计数器“Disk Transfers/sec”。将此值与“Page Faults/sec”值比较，即可了解硬页面错误在整个页面错误中所占的百分比。理想情况下，应该仅有极少或者甚至没有任何针对页面文件的 I/O 活动。不过，在低内存配置下，可能会发生分页。

如果硬页面错误率持续高于每秒 20，则表明计算机的物理内存已过度分配。

使用 NTFS 高速缓存

启用第 121 页的“最大化网络应用程序数据吞吐量”网络服务服务器选项后，即可使用 NTFS 及其关联的高速缓存来提高 Sybase IQ 数据库在插入和查询时的性能。这主要归功于 NTFS 在使用同等大小物理内存的情况下可存储比 Sybase IQ 缓冲区高速缓存多很多的数据。通过降低 main 高速缓存的大小，将此内存提供给可以更有效地使用它的 NTFS，可以最有效地利用这一优点。在 IQ 15.x 中，在装载和查询处理中将使用更多临时高速缓存。因此，使用 NTFS 高速缓存对于 main 高速缓存仍是一项非常有用的技术，但不会提供临时高速缓存的优点。

Sybase IQ 缓冲区高速缓存以未压缩的形式来存储 Sybase IQ 数据（页面）。因此，100MB 的 Sybase IQ 缓冲区高速缓存可存储相当于 100MB 的数据。与之相反的是，NTFS 高速缓存以压缩后的形式管理 Sybase IQ 数据。因此，如果压缩比率为 2:1，则 100MB 的 NTFS 高速缓存可能会存储 200MB 的 Sybase IQ 数据。因此，NTFS 高速缓存有可能保持更高的高速缓存命中率，从而导致 I/O 减少。节省的 I/O 超过将数据从 NTFS 高速缓存移动到 Sybase IQ 缓冲区高速缓存时对数据解压缩所需的计算开销。

对插入和查询进行调优

本节介绍在 Windows 平台上对插入和查询进行调优的其它准则。

合理调优的插入操作的特征

合理调优的 Sybase IQ 插入操作具有特定的特征。您可以从 Windows 任务管理器和 Windows 性能监视器中查看这些特征。

- 插入操作通常为 CPU 密集型。系统内的所有 CPU 在运行时使用率都应接近 100%，其中 CPU 的 95% 或更高使用率处于用户模式。通过单击 Windows 任务管理器的“性能”选项卡并设置“查看”——“显示内核时间”选项，便可轻松查看到此信息。
- 物理内存不应过度分配，特别是，Sybase IQ 进程的虚拟地址空间应小于计算机的物理内存 (RAM)。

- 硬页面错误（针对 *pagefile.sys* 所在卷的 I/O）应很低，理想情况下应接近于 0（零）。
- 针对 IQ 存储的 I/O 操作应保持稳定，并处于磁盘子系统的 I/O 能力之内。

Sybase IQ 使用 Windows CreateFile 选项（用于创建和打开文件）来指定要读取以便后续访问的文件。此选项用于在 LOAD TABLE 命令中指定的文件。这样，由于可提前读取并降低了对 NTFS 高速缓存内存的利用，装载的性能得到了提高。

可通过将 Sybase IQ 的 main 缓冲区高速缓存大小设置为比在第 53 页的“Sybase IQ 主要和临时缓冲区高速缓存”中介绍的计算推荐值小，来进一步提高装载性能，有时甚至可获得显著提高。第 125 页的“使用 NTFS 高速缓存”中介绍了这一性能提高的原因所在。您可将 Sybase IQ 的 main 缓冲区高速缓存的大小设置为比计算推荐值小 50% 左右。

对查询进行调优

您也可通过减少 main 缓冲区高速缓存的大小（如上段所述）来提高查询性能。请参见第 3 章“优化查询和删除”了解有关查询计划、结构和选项的详细信息。

对备份操作进行调优

Windows 仅支持固定大小的 I/O 设备。这就是说，对磁带的每次读取或写入的大小必须与其前后的读取或写入的大小相同。如有任何读取 / 写入操作超出了硬件设备的容量，则相应操作将失败。这意味着，对于备份和恢复操作，只有所有写入（或读取）都以相应硬件所配置的大小进行，备份（或恢复）才可能成功。

Sybase IQ 缺省值设计用于在每一平台上都实现尽可能高效的读取和写入操作。不过，如果您在创建 Sybase IQ 数据库时覆盖了缺省的块大小，则需要在备份该数据库时调整块因子。

对于任何备份或恢复：

$$\text{block size} \times \text{block factor} \approx \text{I/O size}$$

若要在 Windows 系统上调整块因子，必须知道可以通过磁带设备处理的最大物理块大小。驱动器制造商通常未针对此信息提供文字说明。若要确定此值（通常为 64KB），您需要使用 WIN32 API 调用编写一个小程序。然后，必须使用该数据库的块大小和 BACKUP 命令的 **BLOCK FACTOR** 选项，才能优化备份性能。有关完整的语法和用法，请参见《参考：语句和选项》。

使每一 I/O 操作的大小越接近于块大小的最大值，则备份性能越高。需使用与块大小相乘时乘积与相应驱动器的块大小最接近的 **BLOCK FACTOR**（整数）。

需要记住的是，Sybase IQ 在每个块被写入时会向块添加一些额外的数据，用以实现数据的完整性。因此，如果数据库块大小为 8192，磁带设备可处理的最大块大小为 128KB，即使 $8192 * 16 = 128KB$ ，也不可使用 16 作为块因子。必须考虑到 Sybase IQ 会对每一 I/O 操作添加的额外数据，从而使用 15 为 **BLOCK FACTOR**。请注意，在 Windows 上，对于 128KB 的缺省数据库块大小和缺省 IQ 页大小，缺省的块因子是 15。

性能统计信息

sp_iqsysmon 是可以监控多个 Sybase IQ 组件（包括缓冲区高速缓存管理、内存、线程、锁、I/O 功能和 CPU 利用率）的存储过程。

sp_iqsysmon 在批处理或文件模式下运行。性能统计信息按部分输出。

本文档定义 sp_iqsysmon 的各个部分、统计信息和字段名称。

主题	页码
字段引用	129
统计信息	137

字段引用

表 A-1 按部分说明 sp_iqsysmon 统计信息；表 A-2 按统计信息类型提供指向字段名称定义的链接。

表 A-1: 部分名称

缓冲区管理器	缓冲池
预取管理器	IQ 存储空闲列表
缓冲区分配器	内存管理器
线程管理器	CPU 时间统计信息
事务管理器	上下文服务器统计信息
目录、数据库日志和存储库统计信息	

表 A-2: 字段名称定义

统计信息	页类型
锁统计信息	缓冲区分配器 / 预取客户端类型

注释 请参见《参考：构件块、表和过程》中的“系统存储过程”，以了解有关 sp_iqsysmon 的详细信息。

缓冲区管理器

缓冲区管理器管理缓冲区高速缓存的内存内缓冲池。sp_iqsysmon 生成主要（表 A-3）和临时（表 A-4）缓冲区高速缓存统计信息。统计名称按字母顺序列出。

表 A-3: 缓冲区管理器主要统计信息

BlockmapMutexs NLocks	BlockmapMutexs N Waits
BlockmapRegEver	BlockmapRegisters
BlockmapUID	BlockmapUIDnallocs
BmapHTNMaxEntries	BmapHTNClears
BmapHTNCollisn	BmapHTNEntries
BmapHTN Finds	BmapHTN Hits
BmapHTN Hits1	BmapHTN Hits2
BmapHTN Inserts	BmapHTN LChain
BmapHTN Rehash	BmapHTN NumLocks
BmapHTN NumWaits	BufAllocAvailBufs
BufAllocAvailPF	BufAllocMaxBufs
BufHTNFoiledOps	BufHTNw2orMore
BusyWait	CacheTeamNum Asleep
CacheTeamTimes Woken	Creates
Destroys	Dirtyes
FalseMiss	Finds
GrabbedDirty	HitRate
Hits	IONumLocks
IONumWaits	LRUNumLocks
LRUNumSpinLoops	LRUNumSpins WoTO
LRUNumTimeOuts	PReadBlks
PWriteKB	PrefetchInMem
PrefetchNotInMem	PrefetchReqs
PWriteBlks	PWriteKB
Reads	Realdirties
ReReads	UnOwnRR
Writes	

表 A-4: 缓冲区管理器临时统计信息

BlockmapMutexs NLocks	BlockmapMutexs N Waits
BlockmapRegEver	BlockmapRegisters
BlockmapUID	BlockmapUIDnallocs
BmapHTMaxEntries	BmapHTNClears
BmapHTNCollisn	BmapHTNEntries
BmapHTNFinds	BmapHTNHits
BmapHTNHits1	BmapHTNHits2
BmapHTNInserts	BmapHTNLChain
BmapHTNRehash	BmapHTNNumLocks
BmapHTNNumWaits	Buffer Pool main
BufHTNEntries	BufHTNFoiledOps
BufHTNw2orMore	BusyWait
CacheTeamNum Asleep	CacheTeamTimes Woken
Cloned	Creates
Destroys	Dirtyes
Finds	FalseMiss
Flushes	GetPageFrame
GetPageFrameFailure	GotEmptyFrame
GrabbedDirty	HitRate
Hits	InUse
IONumLocks	IONumWaits
LRUNumLocks	LRUNumSpinLoops
LRUNumSpinsWoTO	LRUNumTimeOuts
MovedToMRU	MovedToWash
Pages	Pinned
PWriteBlks	PWriteKB
PrefetchInMem	PrefetchNotInMem
PrefetchReqs	PWriteBlks
PWriteKB	Reads
Realdirties	RemovedFromLRU
RemovedFromWash	RemovedInScanMode
ReReads	TimesSweepers Woken
UnOwnRR	Washed
washIntensity	WashMaxSize
washNActive Sweepers	washNBuffers
washNBuffers	washSignalThreshold
washTeamSize	Writes

缓冲池

缓冲池管理缓冲区高速缓存的内存内缓冲区。sp_iqsysmon 生成主要（表 A-5）和临时（表 A-6）缓冲区高速缓存统计信息。统计名称按字母顺序列出。

表 A-5: 缓冲池主要统计信息

Dirty	FlushedBufferCount
Flushes	GetPageFrame
GetPageFrameFailure	GotEmptyFrame
InUse	MovedToMRU
MovedToWash	Pages
Pinned	RemovedFromLRU
RemovedFromWash	RemovedInScanMode
TimesSweepers Woken	Washed
washIntensity	WashMaxSize
washNActive Sweepers	washNBuffers
washNDirtyBuffers	washSignalThreshold
washTeamSize	

表 A-6: 缓冲池临时统计信息

Dirty	FlushedBufferCount
Flushes	GetPageFrame
GetPageFrameFailure	GotEmptyFrame
InUse	MovedToMRU
MovedToWash	Pages
Pinned	RemovedFromLRU
RemovedFromWash	RemovedInScanMode
TimesSweepers Woken	Washed
washIntensity	WashMaxSize
washNActive Sweepers	washNBuffers
washNDirtyBuffers	washSignalThreshold
washTeamSize	

预取管理器

预取管理器管理预取页的异步读取。sp_iqsysmon 生成主要（表 A-7）和临时（表 A-8）预取管理器统计信息。统计名称按字母顺序列出。

表 A-7: 预取管理器主要统计信息

PFMgrCondVar	PFMgrNDropped
PFMgrNRead	PFMgrNReading
PFMgrNSubmitted	PFMgrNThreads
PFMgrNValid	

表 A-8: 预取管理器临时统计信息

PFMgrCondVar	PFMgrNDropped
PFMgrNRead	PFMgrNReading
PFMgrNSubmitted	PFMgrNThreads
PFMgrNValid	

IQ 存储空闲列表

IQ 存储空闲列表映射数据库使用的磁盘块。sp_iqsysmon 生成 IQ 存储的主要（表 A-9）和临时（表 A-10）空闲列表统计信息。统计名称按字母顺序列出。

表 A-9: IQ 存储主要空闲列表统计信息

FLBitCount	FLIsOutOfSpace
FLMutexLocks	FLMutexWaits

表 A-10: IQ 存储临时空闲列表

FLBitCount	FLIsOutOfSpace
FLMutexLocks	FLMutexWaits

缓冲区分配器

缓冲区分配器管理（临时）数据结构的缓冲区固定配额。sp_iqsysmon 生成主要（表 A-11）和临时（表 A-12）统计信息。统计名称按字母顺序列出。

表 A-11: 缓冲区分配器主要统计信息

BM	BTree
BufAllocAvailBufs	BufAllocAvailPF
BufAllocMaxBufs	BufAllocMutexLocks
BufAllocMutexWaits	BufAllocNPFUsers
BufAllocNPinUsers	BufAllocNPostedUsrs
BufAllocNPostEst	BufAllocNUnPostEst
BufAllocNUnpostUsrs	BufAllocPinQuota
BufAllocReserved	BufAllocSlots
BV	ClientCountOfPinner
DBCC	FP
Garray	Hash
LOB	NActiveCommands
NumClients	PfUserRegisters
PinUserQuota	PinUserRegisters
PrefetchUserQuota	Row
RowColumn	Sort
Store	

表 A-12: 缓冲区分配器临时统计信息

BM	BTree
BufAllocAvailBufs	BufAllocAvailPF
BufAllocMaxBufs	BufAllocMutexLocks
BufAllocMutexWaits	BufAllocNPFUsers
BufAllocNPinUsers	BufAllocNPostedUsrs
BufAllocNPostEst	BufAllocNUnPostEst
BufAllocNUnpostUsrs	BufAllocPinQuota
BufAllocReserved	BufAllocSlots
BV	ClientCountOfPinner
DBCC	FP
Garray	Hash
LOB	NActiveCommands
NumClients	PfUserRegisters
PinUserQuota	PinUserRegisters
PrefetchUserQuota	Row
RowColumn	Sort
Store	

内存管理器

sp_iqsysmon 生成用于指明内容管理器如何分配服务器堆内存的统计信息（表 A-13）。统计名称按字母顺序列出。

表 A-13: 内存管理器统计信息

MemAllocated	MemAllocatedEver
MemAllocatedMax	MemNAllocated
MemNAllocatedEver	MemNTimesLocked
MemNTimesWaited	

线程管理器

sp_iqsysmon 生成用于指明线程管理器如何处理服务器的全局线程池的统计信息（表 A-14）。统计名称按字母顺序列出。

表 A-14: 线程管理器统计信息

NumTeamsAlloc	NumThrUsed
SingleThrAlloc	TeamThrAlloc
ThreadLimit	ThrMaxTeams
ThrMutexLocks	ThrMutexWaits
ThrNTeamsInUse	ThrNumFree
ThrNumOfCpus	ThrNumThreads
ThrReserved	UsedPerActiveCmd

CPU 时间统计信息

sp_iqsysmon 生成用于指明由操作系统报告的 IQ CPU 使用情况的 CPU 时间统计信息（表 A-15）。统计名称按字母顺序列出。

表 A-15: CPU 时间统计信息

CPU Sys Seconds	CPU Total Seconds
CPU User Seconds	Elapsed Seconds

事务管理器

sp_iqsysmon 生成用于指明事务管理器如何管理事务序列化和版本分配的统计信息（表 A-16）。统计名称按字母顺序列出。

表 A-16: 事务管理器统计信息

TxnMgrNBlocked	TxnMgrNPending
TxnMgrNWaiting	TxnMgrOAVI
TxnMgrPCcondvar	TxnMgrtxncblock
TxnMgrTxnIDseq	TxnMgrVersionID
TxnMgrVersionLock	

上下文服务器统计信息

sp_iqsysmon 生成用于提供资源调控器指标的上下文服务器统计信息（表 A-16）。统计名称按字母顺序列出。

表 A-17: 上下文服务器统计信息

StCntxCondVar	StCntxLock
StCntxNAdmitted	StCntxNOrigResource
StCntxNResource	StCntxNumConns
StCntxNWaited	StCntxNWaiting

目录、数据库日志和存储库统计信息

sp_iqsysmon 生成目录和统计信息存储库的统计信息（表 A-17）。统计名称按字母顺序列出。

表 A-18: 目录、数据库日志和存储库统计信息

CatalogLock	DbLogMLock
DbLogSLock	RepositoryLock
RepositoryNList	

统计信息

BlockmapMutexs NLocks	块映射锁数。
BlockmapMutexs NWaitis	等待的块映射锁数。
BlockmapRegEver	创建数据库后实例化的块映射数。
BlockmapRegisters	监控间隔期间实例化的块映射数。
BlockmapUID	当前块映射的唯一 ID。
BlockmapUIDnallocs	监控间隔期间分配的块映射数。
BmapHTMaxEntries	理想值。
BmapHTNClears	从缓冲区散列中删除的条目数。
BmapHTNCollisn	插入到非空散列桶中的缓冲区散列表条目数。
BmapHTNEntries	块映射散列表中的条目数。
BmapHTNFinds	缓冲区管理器散列表探查数。
BmapHTNHits	成功的缓冲区管理器散列表探查数。
BmapHTNHits1	第一次尝试命中的缓冲区散列查找调用数。
BmapHTNHits2	第二次尝试命中的缓冲区散列查找调用数。
BmapHTNInserts	插入到缓冲区散列表中的块映射页数。
BmapHTNLChain	缓冲区散列表中最长链的长度。
BmapHTNRehash	由于缓冲区散列表增长而进行的重新散列数。
BmapHTNumLocks	锁定缓冲区散列表的次数。
BmapHTNumWaitis	缓冲区散列表锁的等待数。
BufAllocAvailBufs	可以固定的缓冲区计数。
BufAllocAvailPF	可以预取的缓冲区计数。
BufAllocMaxBufs	理想值。
BufAllocMutexLocks	对缓冲区分配器使用的锁计数。
BufAllocMutexWaitis	等待的缓冲区分配器锁计数。
BufAllocNPFUsers	使用预取缓冲区的索引 / 结构计数。
BufAllocNPinUsers	使用固定缓冲区的索引 / 结构计数。

BufAllocNPostEst	用于通知预计将来用户的调用数。
BufAllocNPostedUsrs	提前请求的固定缓冲区计数。
BufAllocNUnPostEst	用于取消通知预计将来用户的调用数。
BufAllocNUnpostUsrs	未提前请求的固定缓冲区计数。
BufAllocPinQuota	理想值。
BufAllocReserved	理想值。
BufAllocSlots	缓冲区分配器管理的缓冲区总数。
BufHTMaxBucketSize	最长缓冲区散列链的长度。
BufHTNBuckets	缓冲区散列表表元数。
BufHTNEntries	理想值。
BufHTNFoiledOps	必须重新获取表锁的操作数。
BufHTNw2orMore	包含 2 个或更多条目的缓冲区散列表表元数。
BusyWait	在磁盘中读取或写入页时处于等待状态的查找请求。
Cloned	查询锁定早期版本后为创建新版本而复制的缓冲区。
CacheTeamNum Asleep	当前处于非活动状态的缓冲区高速缓存线程数。
CacheTeamTimes Woken	说明。
CatalogLock	用于访问 IQ 目录的锁。
ClientCountOfPinnors	具有固定缓冲区的缓冲区分配器客户端数。
CPU User Seconds	用户模式 CPU 秒数。
CPU Sys Seconds	系统模式 CPU 秒数。
CPU Total Seconds	系统和用户秒数的总和。
Creates	创建的新页数。
DbLogMLock	IQ 消息文件锁。
DbLogSLock	IQ 消息文件锁。
Destroys	损坏的页数。
Dirtyies	页修改请求数。
Dirty	已修改的页百分比。

Elapsed	运行间隔期间使用的 CPU 秒数。
FalseMiss	辅助高速缓存查找功能找到了正确的缓冲区。
FLBitCount	存储（主或临时）中分配的块数。
Finds	已位于高速缓存中的查找数。
FLIsOutOfSpace	用于指示何时存储已完全分配（已满）的标记。
FLMutexLocks	锁定空闲列表的次数。
FLMutexWaits	空闲列表的锁等待数。
FlushedBufferCount	通过刷新来写入磁盘的缓冲区数。
Flushes	调用刷新运算符的次数。
GetPageFrame	从缓冲区管理器请求的干净缓冲区数。
GetPageFrameFailure	失败的请求数（因所有缓冲区已锁定而导致错误）。
GotEmptyFrame	找到的干净缓冲区数。
GrabbedDirty	在脏状态下被捕获并同步写入到磁盘以释放缓冲区的页数。
HitRate	已位于高速缓存中的查找百分比。
IONumLocks	临时压缩缓冲池上的锁数。
IONumWaits	临时压缩缓冲池上等待的锁。
InUse	高速缓存中标记为正在使用的缓冲区数。
LRUNumLocks	锁定缓冲区管理器 LRU 的次数。
LRUNumSpinsWoTO	线程等待获取缓冲区管理器 LRU 锁的次数。
LRUNumSpinLoops	缓冲区管理器 LRU 锁经历的旋转周期数。
LRUNumTimeOuts	线程等待获取缓冲区管理器 LRU 锁时发生超时的次数。
MemAllocated	当前从堆分配的内存量。
MemAllocatedEver	服务器启动后分配的最大内存量。
MemAllocatedMax	监控间隔期间分配的最大内存量。
MemNAllocated	要求分配内存的请求数。
MemNAllocatedEver	服务器启动后要求分配内存的请求数。
MemNTimesLocked	锁定堆的次数。
MemNTimesWaited	堆的锁等待数。

MovedToMRU	使用后放回 MRU 中的缓冲区数。
MovedToWash	使用后直接放置到清洗区中的缓冲区数。
NActiveCommands	活动命令的当前计数。
NumClients	当前注册到缓冲区分配器中的客户端数。
NumTeamsAlloc	监控间隔期间的组总数。
NumThrUsed	服务器分配的实际线程数。
PFMgrCondVar	预取管理器锁的统计信息。
PFMgrNDropped	删除的预取请求数。
PFMgrNRead	启动的预取请求数。
PFMgrNReading	完成的 PF 读取数。
PFMgrNSubmitted	提交的预取请求数。
PFMgrNThreads	为预取保留的线程数。
PFMgrNValid	有效的预取请求数。
Pages	高速缓存中的缓冲区数。
PrefetchInMem	为预取请求的页已位于内存中。
PrefetchNotInMem	读取时找到为预取请求的页。
PfUserRegisters	客户端注册到预取管理器分配器以便预取的次数。
Pinned	固定在内存中（锁定）的缓冲区百分比。
PinUserQuota	可以注册到缓冲区分配器以获取固定缓冲区的最大客户端数。
PinUserRegisters	客户端注册到缓冲区分配器以获取固定缓冲区的次数。
PReadBlks	读取的数据库块数。
PReadKB	读取的数据库块的大小（以 KB 为单位）。
PrefetchReqs	为异步预取请求的页数。
PrefetchUserQuota	可以注册到缓冲区分配器以便预取的最大客户端数。
PWriteBlks	写入的数据库块数。
PWriteKB	说明。
Realdirties	说明。
RemovedFromLRU	执行查找时 LRU 中的缓冲区数。

RemovedFromWash	执行查找时清洗区中的缓冲区数。
RemovedInScanMode	执行查找时处于扫描模式的缓冲区数。
RepositoryLock	统计信息存储库锁。
RepositoryNList	状态存储库中的统计信息数。
ReReads	从磁盘多次重新读取的页。
SingleThrAlloc	仅拥有一个线程的组数。
StCntxCondVar	资源门锁。
StCntxLock	数据库上下文锁。
StCntxNAdmitted	完成的命令的总开销。
StCntxNOrigResource	最大并发命令数。
StCntxNResource	可用并发命令数。
StCntxNWaited	监控间隔期间在资源门处等待的命令数。
StCntxNWaiting	资源门处当前阻止的命令数。
StCntxNumConns	当前连接数目。
TeamThrAlloc	监控间隔期间分配的线程组总数。
ThrMaxTeams	最大线程组数。
ThrMutexLocks	锁定线程管理器的次数。
ThrMutexWaits	等待的线程管理器锁数。
ThrNTeamsInUse	当前使用的线程组数。
ThrNumFree	可供分配的线程数。
ThrNumOfCpus	IQ 使用的 CPU 数。
ThrNumThreads	服务器分配的实际线程数。
ThrReserved	说明。
ThreadLimit	为服务器请求的线程总数。
TimesSweepers Woken	清理程序线程等待工作的次数。
TxnMgrNBlocked	开始事务中阻止的事务数。
TxnMgrNPending	待提交的事务数。
TxnMgrNWaiting	开始事务中处于等待状态的事务数。

TxnMgrOAVI	使用的是最早版本。
TxnMgrPCcondvar	事务管理器提交后锁统计信息。
TxnMgrTxnIDseq	下一事务 ID。
TxnMgrVersionID	事务管理器共享主要版本 ID。
TxnMgrVersionLock	用于版本 ID 分配的锁。
TxnMgrtxncblock	事务命令期间使用的锁。
UnOwnRR	无主重新读取。因可能太旧而在查询服务器上重新读取的页数。
UsedPerActiveCmd	每个命令的平均线程数。
WashMaxSize	清洗区中可以存在的最大缓冲区数。
Washed	从清洗区刷新的缓冲区数。
washIntensity	Buffer_Cache_Wash_Intensity 选项的当前值。
washNActive Sweepers	具有要写入的缓冲区的清理程序数。
washNBuffers	清洗区中的干净缓冲区数。
washNDirtyBuffers	清洗区中的脏缓冲区数。
washSignalThreshold	清理程序苏醒之前清洗区中的脏缓冲区数。
washTeamSize	清理程序组中的线程数。
Writes	写入磁盘的页。

页类型

BTREEV	可变 B 树。供 HG、LF 用于 varchar、varbinary 数据类型。
BTREEF	固定 B 树。供 HG、LF、WD、DT、TM、DTTM 索引用于固定数据类型。供表 VDO 用于存储索引元数据。
BV	永久位向量。供连接布隆过滤器和范围谓词处理使用。
VDO	大小可变的数据对象。供表 VDO 用于存储索引元数据。
DBEXT	DBSpace 标头块。供每个 dbspace 的第一个块使用。
DBID	数据库标识。供每个数据库的一个块使用。
SORT	排序数据。供临时高速缓存中的排序对象使用。

STORE	临时数据存储。供滚动游标存储使用。临时查询存储：联合、游标、OLAP。
GARRAY	组数组结构。供 HG、WD 索引、连接索引使用。
BARRAY	字节数组。供 FP 和 LOB 索引使用。
BARRAY	块映射。供所有对象的元数据使用。
HASH	散列表数据。供临时高速缓存中的散列对象使用。
CKPT	检查点日志。供主存储中的检查点日志条目使用。
BM	位图数据。供索引、查询处理和数据结构使用。
CMID	Multiplex 提交标识。供 Multiplex 用于通知查询服务器的新版本。
RIDCA	Recid 高速缓存数据。供 sp_iqcheckdb 使用，并且仅在临时存储中使用。
LOB	Long Binary 存储。供 Long varchar 和 long binary 存储使用。

锁统计信息

Locks	获取锁的次数。
Lock-Waits	因争用等待锁的次数。
Signals	向条件变量发送信号以调度单个线程的次数。
Broadcasts	向条件变量发送信号以调度所有线程的次数。
Waits	线程等待条件变量的次数。
RdLocks	从读写互斥锁获取读取锁的次数。
RdWaits	读取锁等待读写互斥锁的次数。
RdTryFails	读取尝试锁获取读写互斥锁失败的次数。
WrLocks	写入锁等待读写互斥锁的次数。
WrWaits	写入锁等待读写互斥锁的次数。
WrTryFails	写入尝试锁获取读写互斥锁失败的次数。
Locks	从螺旋锁获取锁的次数。
SpinsWoTO	螺旋锁等待锁超时的次数。
Spins	螺旋锁旋转以等待锁的次数。
TimeOuts	螺旋锁等待锁超时的次数。

缓冲区分配器 / 预取客户端类型

BM	位图。供大多数数据库结构使用。
BTree	B 树。供 HG、LF 用于 varchar、varbinary 数据类型。
BV	(半) 永久位向量。供连接布隆过滤器和范围谓词处理使用。
DBCC	数据库一致性检查程序。供 sp_iqcheckdb 使用。
FP	快速投影索引。供主列存储使用。
Garray	组数组。供 HG、WD 索引、连接索引使用。
Hash	散列表。供查询、枚举 FP 索引使用。
LOB	Long Binary。供 Long Binary 列使用。
Row	FP 行提取。供查询行投影使用。
RowColumn	FP 列提取。供查询列投影使用。
Sort	排序。供查询、插入、更新、中型删除、DBCC 使用。
Store	临时数据存储。供滚动游标存储使用。临时查询存储：联合、游标、OLAP。

索引

英文

- AGGREGATION_ALGORITHM_PREFERENCE
 - 选项 41
- AND 关键字 7
- AVG 函数 10
- BETWEEN 条件 9
- BLANK PADDING
 - 对连接的影响 27
 - 支持 OFF 27
- Catalog 存储 4
 - 文件增长 76
- CIS 功能性补偿
 - 性能影响 36
- comparison
 - 关于 6, 7
- COUNT 函数 10, 11
- CPU
 - 对 Sybase IQ 的可用性 70
- CPU 时间统计信息 135
- CPU 使用率
 - 监控 114, 119
- CPU 使用率统计信息 88
- CREATE DBSPACE 语句 66
- data
 - 存储 125
- dbspace
 - 定位以达到最佳性能 66
 - 使用统计信息 91
- DEFAULT_HAVING_SELECTIVITY 选项 41
- DEFAULT_LIKE_MATCH_SELECTIVITY
 - 选项 41
- DEFAULT_LIKE_RANGE_SELECTIVITY
 - 选项 41
- DENSE_RANK 函数 15
- EARLY_PREDICATE_EXECUTION 选项 41
- FLATTEN_SUBQUERIES 选项 35
- FORCE_NO_SCROLL_CURSORS 71
- FROM 子句 4, 80
 - 和连接 20
- GROUP BY
 - 有关性能的建议 32
- HASH_THRASHING_PERCENT 选项 113
- HG 索引
 - 多列 34
- I/O
 - 性能建议 63
 - 直接 60
- I/O 设备 126
- IN 条件 9
- IN_SUBQUERY_PREFERENCE 选项 41
- INDEX_ADVISOR 选项 37
- INDEX_PREFERENCE 选项 41
- IQ PATH 选项
 - 选择裸设备 63
- IQ 存储
 - 缓冲区高速缓存大小 54
- IQ 存储空闲列表 133
- IQ 页大小
 - 确定 55
- iq_dummy 表 4
- iqgovern 开关
 - 限制查询以提高性能 69
- IQGOVERN_MAX_PRIORITY 选项 40
- IQGOVERN_PRIORITY 40
- IQMSG 日志
 - 设置最大大小 68
- iqnumbercpus
 - 设置 CPU 数 70
- iqwmem 开关 59
- JOIN_ALGORITHM_PREFERENCE 选项 41
- LIST 函数 10
- MAIN_CACHE_MEMORY_MB 选项 54
- MAX 函数 10
- max 函数 10
- MAX_CURSOR_COUNT 选项 71

- MAX_HASH_ROWS 选项 42
- MAX_QUERY_TIME 选项 40
- MAX_STATEMENT_COUNT** 选项 71
- MAX_TEMP_SPACE_PER_CONNECTION 选项 70
- memory
 - Multiplex 数据库 49
 - 分页 48
 - 减少要求 56
 - 开销 51
 - 连线 59
 - 另请参见 缓冲区高速缓存 54
 - 碎片 59
 - 限制查询的使用 70
- MIN 函数 10
- Multiplex
 - 性能监控器 86
- Multiplex 数据库
 - memory 49
 - 磁盘空间 73
- NOEXEC 选项 37
- NT CreateFile 选项 126
- NT 任务管理器 123
- NT 性能监视器 123
- NTFS 高速缓存
 - 挪用内存 121
 - 提高性能 125
- NTILE 函数 15
- OR 关键字 9
- ORDER BY 子句 6, 34
- OS_FILE_CACHE_BUFFERING 选项 60
- OS_FILE_CACHE_BUFFERING_TEMPDB 选项 60
- OVER 子句 16
- partition
 - 定义 63
- PERCENT_RANK 函数 15
- PERCENTILE_CONT 函数 15
- PERCENTILE_DISC 函数 15
- PREFETCH_BUFFER_LIMIT 选项 71
- QUERY_DETAIL 选项 37
- QUERY_PLAN 选项 37
- QUERY_PLAN_AFTER_RUN 选项 37
- QUERY_PLAN_AS_HTML 选项 38
- QUERY_PLAN_AS_HTML_DIRECTORY 选项 38
- QUERY_PLAN_TEXT_ACCESS 选项 37
- QUERY_PLAN_TEXT_CACHING 选项 38
- QUERY_TEMP_SPACE_LIMIT 选项 70
- QUERY_TIMING 选项 38
- range 7
- RANK 函数 15
- RAWDETECT
 - 磁盘分条选项 65
- rows
 - 关于 3
 - 选择 6
- SELECT 语句
 - 关于 1, 28
- sp_iqcolumnuse 过程
 - sp_iqindexuse 过程 43
- sp_iqsysmon
 - CPU 时间统计信息 135
 - CPU 时间统计信息, 已定义 135
 - IQ 存储空闲列表, 统计信息 133
 - IQ 存储空闲列表, 已定义 133
 - 部分名称链接 129
 - 部分统计信息定义链接 129
 - 缓冲池, 统计信息 132
 - 缓冲池, 已定义 132
 - 缓冲区分配器, 统计信息 133
 - 缓冲区分配器, 已定义 133
 - 缓冲区分配器 / 预取客户端类型 144
 - 缓冲区管理器, 定义 130
 - 缓冲区管理器, 统计信息 130
 - 目录、数据库日志和存储库统计信息 136
 - 目录、数据库日志和存储库统计信息, 已定义 136
 - 内存管理器, 统计信息 135
 - 内存管理器, 已定义 135
 - 上下文服务器统计信息 136
 - 上下文服务器统计信息, 已定义 136
 - 事务管理器, 统计信息 136
 - 事务管理器, 已定义 136
 - 锁统计信息 143
 - 统计信息 137
 - 统计信息类型定义链接 129
 - 线程管理器, 统计信息 135
 - 线程管理器, 已定义 135
 - 页类型 142
 - 预取管理器, 统计信息 133

- 预取管理器, 已定义 133
- 字段名称链接 129
- 字段引用链接 129
- sp_iqtable 过程 2
- sp_iqtableuse 过程 43
- sp_iqunusedcolumn 过程 43
- sp_iqunusedindex 过程 43
- sp_iqunusedtable 过程 43
- sp_iqworkmon 过程 43
- SUM 函数 10
- SYSTEM dbspace 4
- TEMP_CACHE_MEMORY_MB 选项 54
- UNION ALL
 - 规则 80
 - 视图 80
- vmstat 命令
 - 监控 UNIX 上的缓冲区高速缓存 114
- WHERE 子句
 - BETWEEN 条件 9
 - ORDER BY 子句 6
 - 和模式匹配 8
 - 日期比较 7
 - 示例 6
- Windows 磁带设备 126
- WITHIN GROUP 子句 16

B

- 备份
 - 调优块大小 126
- 表
 - iq_dummy 4
 - 和外键 21
 - 连接 75
 - 列表 2
 - 相关名 20
 - 折叠 75
 - 主键 21
- 并行度
 - 查询树 35
- 不等式, 测试 7

C

- 参数
 - 函数 10
- 插入操作
 - 调优 125
- 查看过程分析数据
 - Sybase Central 96
- 查询
 - 调优 126
 - 结构化 31
 - 索引建议 74
 - 限制并发 69
 - 限制内存使用 70
 - 优化 36, 41, 74
 - 由 SQL Anywhere 进行处理 4
- 查询表 4
- 查询并行度 35
- 查询服务器
 - 平衡负载 73
- 查询计划 36
 - 生成但不执行 37
 - 使用 39
 - 图形 39
- 查询性能
 - Catalog 存储表 36
 - CIS 功能性补偿影响 36
 - 跨数据库连接 36
 - 由 SQL Anywhere 规则处理 36
- 超线程
 - 服务器开关 70
- 重置过程分析
 - SQL 94
 - Sybase Central 94
- 触发器
 - 查看单个分析信息 98
 - 查看摘要分析数据 97
- 处理查询而不 4
- 磁带设备
 - Windows 126
- 磁盘分条
 - Sybase IQ 63
 - 定义 63
 - 规则 64

索引

- 内部 65
 - 在装载中使用 65
- 磁盘高速缓存
 - 定义 73
 - 性能影响 73
- 磁盘空间
 - Multiplex 数据库 73
 - 交换空间 49
- 存储 I/O 统计信息 91
- 存储过程 2
 - 查看单个分析信息 98
 - 查看分析数据 93
 - 查看摘要分析数据 97
 - 性能监控 92

D

- 低碎片堆 59
- 地址空间
 - virtual 123
- 动态性能监控器 86
- 抖动
 - HASH_THRASHING_PERCENT 选项 113
 - 要执行的操作 112
- 段
 - 数据库, 使用多个 66
- 堆
 - 低碎片 59
- 多列索引 34
- 多线程
 - 性能影响 62

F

- 非规范化
 - 缺点 77
 - 性能优势 78
 - 原因 77
- 分区表 80
- 分析数据库过程
 - 关于 93
- 分析信息

- 触发器 98
- 存储过程和函数 98
- 事件 98
- 系统触发器 99
- 分页
 - memory 48
 - 对性能的影响 48
 - 在 UNIX 上监控 114
 - 在 Windows 上监控 114
- 分组数据 10
- 服务器
 - 监控性能 86
- 负载监控 42
- 负载平衡
 - 查询服务器间 73

G

- 高速缓存
 - NTFS 125
 - 缓冲区 125
 - 另请参见 缓冲区高速缓存 101
- 高速缓存大小
 - IQ 主和临时缓冲区 54
- 高速缓存统计信息 89
- 高速缓存页
 - 预取 71
- 工作集 123
- 固定大小的 I/O
 - Windows 126
- 过程分析
 - 触发器 97, 98
 - 存储过程和函数 97, 98
 - 各个过程的信息 97, 100
 - 过程的摘要 99
 - 事件 97, 98
 - 系统触发器 97, 99
 - 用 SQL 禁用 95
 - 用 SQL 启用 93
 - 用 SQL 清除 95
 - 用 SQL 重置 94
 - 在 Interactive SQL 中查看数据 99
 - 在 Sybase Central 中查看数据 93, 96

- 在 Sybase Central 中禁用 95
- 在 Sybase Central 中启用 93
- 在 Sybase Central 中清除 95
- 在 Sybase Central 中重置 94

H

函数

- SOUNDEX 函数 9
- 查看单个分析信息 98
- 查看分析数据 93
- 查看摘要分析数据 97
- 分析 15
- 集合 10

缓冲池

- 统计信息 132

缓冲区

- 禁用操作系统缓冲 60

缓冲区分配器 133

缓冲区分配器 / 预取客户端类型

- BM 144
- BTree 144
- BV 144
- DBCC 144
- FP 144
- Garray 144
- LOB 144
- RowColumn 144
- 存储 144
- 排序 144
- 散列 144
- 统计信息 144
- 行 144

缓冲区高速缓存 54, 125

- 布局 112
- 监控 101
- 确定大小 50
- 设置大小 54

缓冲区高速缓存监控器 101

- 示例 107

缓冲区管理器

- 抖动 112
- 统计信息 130

J

监控

- 工作集 123
- 事务状态 86
- 物理内存 122
- 虚拟地址空间 123
- 页面错误 124

监控负载 42

监控器

- IQ UTILITIES 语法 101
- 启动和停止 101
- 设置输出文件位置 101

键连接

- 使用 22

交换

- memory 48
- 磁盘空间要求 49
- 对性能的影响 48

交换文件

- 对性能的影响 48

进程

- 工作集 123
- 监控 123
- 增大 59

进程线程模型 62

禁用过程分析

- SQL 95
- Sybase Central 95

K

块大小

- 与 IQ 页大小的关系 55

L

连接

- 和 BLANK PADDING 27
- 和数据类型 25
- 矢量积 19
- 使用的限制语句 71
- 优化程序简化 41

连接索引

- 性能影响 75

连接统计信息 90

连系内存 59

列

- 从表中进行选择 5
- 大量空值 5, 31
- 关于 3
- 排序 5

临时存储

- 缓冲区高速缓存大小 54

裸分区

- 内存使用 52

裸设备

- 对性能的影响 63

M

命令

- long 6

模式匹配 8

目录、数据库日志和存储库统计信息 136

N

内存管理器 135

内存使用统计信息 88

P

撇号

- 使用 6

Q

启用过程分析

- SQL 93
- Sybase Central 93

轻型进程 62

清除过程分析

- SQL 95
- Sybase Central 95

清理程序线程 112

请求统计信息 90

区分大小写 2, 6

R

任务管理器 123

日期 7, 9

S

上下文服务器统计信息 136

设备

- Windows 上的 I/O 126

事件

- 查看单个分析信息 98
- 查看分析数据 93
- 查看摘要分析数据 97

事务管理器 136

事务日志

- 关于 67
- 截断 67
- 停止的数据库 67
- 脱机数据库 67

事务统计信息 91

事务状态

- 监控 86

数据库

- 非规范化的优势 78
- 非规范化以提高性能 77
- 管理 75

数据库段

- 定位以达到最佳性能 66

- 数据库过程
 - 查看分析数据 93
- 数据库选项
 - FLATTEN_SUBQUERIES 35
- 数据库之间的连接
 - 性能影响 36
- 数据类型
 - 连接要求 25
- 数据装载改进 81
- 顺序磁盘 I/O 66
- 搜索条件
 - 简介 6
 - 快捷方式 9
 - 日期比较 7
 - 子查询 28
- 碎片 59
- 索引
 - 多列 34
- 索引顾问 74
- 索引类型
 - 针对性能选择 74
- 锁统计信息
 - Broadcasts 143
 - Lock-Waits 143
 - RdLocks 143
 - RdTryFails 143
 - RdWaits 143
 - Signals 143
 - sp_iqsysmon 143
 - Spins 143
 - SpinsWoTO 143
 - TimeOuts 143
 - WrLocks 143
 - WrTryFails 143
 - WrWaits 143
 - 等待数 143
 - 锁 143
 - 性能统计信息 143

T

- 条件
 - 和 GROUP BY 子句 11
 - 搜索 6, 7, 9
- 调优 121
 - 对于插入操作 125
 - 用于查询 126
- 统计信息
 - BlockmapMutexsNLocks 137
 - BlockmapMutexsNWAits 137
 - BlockmapRegEver 137
 - BlockmapRegisters 137
 - BlockmapUID 137
 - BlockmapUIDnallocs 137
 - BmapHTMaxEntries 137
 - BmapHTNClears 137
 - BmapHTNCollisn 137
 - BmapHTNEntries 137
 - BmapHTNFind 137
 - BmapHTNHits 137
 - BmapHTNHits1 137
 - BmapHTNHits2 137
 - BmapHTNInserts 137
 - BmapHTNLChain 137
 - BmapHTNRehash 137
 - BmapHTNumLocks 137
 - BmapHTNumWaits 137
 - BufAllocAvailBufs 137
 - BufAllocAvailPF 137
 - BufAllocMaxBufs 137
 - BufAllocMutexLocks 137
 - BufAllocMutexWaits 137
 - BufAllocNPFUsers 137
 - BufAllocNPinUsers 137
 - BufAllocNPostedUsrs 138
 - BufAllocNPostEst 138
 - BufAllocNUnPostEst 138
 - BufAllocNUnpostUsrs 138
 - BufAllocPinQuota 138
 - BufAllocReserved 138
 - BufAllocSlots 138
 - BufHTMaxBucketSize 138
 - BufHTNBuckets 138
 - BufHTNEntries 138
 - BufHTNFoiledOps 138

- BufHTNw2orMore 138
- BusyWait 138
- CacheTeamNumAsleep 138
- CacheTeamTimesWoken 138
- CatalogLock 138
- ClientCountOfPinner 138
- Cloned 138
- CPU Sys Seconds 138
- CPU Total Seconds 138
- CPU User Seconds 138
- Creates 138
- DbLogMLock 138
- DbLogSLock 138
- Destroys 138
- Dirtyies 138
- Dirty 138
- Elapsed 139
- FalseMiss 139
- Finds 139
- FLBitCount 139
- FLIsOutOfSpace 139
- FLMutexLocks 139
- FLMutexWaits 139
- FlushedBufferCount 139
- Flushes 139
- GetPageFrame 139
- GetPageFrameFailure 139
- GotEmptyFrame 139
- GrabbedDirty 139
- HitRate 139
- InUse 139
- IONumLocks 139
- IONumWaits 139
- LRUNumLocks 139
- LRUNumSpinLoops 139
- LRUNumSpinsWoTO 139
- LRUNumTimeOuts 139
- MemAllocated 139
- MemAllocatedEver 139
- MemAllocatedMax 139
- MemNAllocated 139
- MemNAllocatedEver 139
- MemNTimesLocked 139
- MemNTimesWaited 139
- MovedToMRU 140
- MovedToWash 140
- NActiveCommands 140
- NumClients 140
- NumTeamsAlloc 140
- NumThrUsed 140
- PFMgrCondVar 140
- PFMgrNDropped 140
- PFMgrNRead 140
- PFMgrNReading 140
- PFMgrNSubmitted 140
- PFMgrNThreads 140
- PFMgrNValid 140
- PfUserRegisters 140
- Pinned 140
- PinUserQuota 140
- PinUserRegisters 140
- PReadBlks 140
- PReadKB 140
- PrefetchInMem 140
- PrefetchNotInMem 140
- PrefetchReqs 140
- PrefetchUserQuota 140
- PWriteBlks 140
- PWriteKB 140
- Realdirties 140
- RemovedFromLRU 140
- RemovedFromWash 141
- RemovedInScanMode 141
- RepositoryLock 141
- RepositoryNList 141
- ReReads 141
- SingleThrAlloc 141
- sp_iqsystemon 137
- StCntxCondVar 141
- StCntxLock 141
- StCntxNAdmitted 141
- StCntxNOrigResource 141
- StCntxNResource 141
- StCntxNumConns 141
- StCntxNWaited 141
- StCntxNWaiting 141
- TeamThrAlloc 141
- ThreadLimit 141
- ThrMaxTeams 141
- ThrMutexLocks 141
- ThrMutexWaits 141
- ThrNTeamsInUse 141

ThrNumFree 141
 ThrNumOfCpus 141
 ThrNumThreads 141
 ThrReserved 141
 TimesSweepersWoken 141
 TxnMgrNBlocked 141
 TxnMgrNPending 141
 TxnMgrNWaiting 141
 TxnMgrOAVI 142
 TxnMgrPCcondvar 142
 TxnMgrtxncblock 142
 TxnMgrTxnIDseq 142
 TxnMgrVersionID 142
 TxnMgrVersionLock 142
 UnOwnRR 142
 UsedPerActiveCmd 142
 Washed 142
 washIntensity 142
 WashMaxSize 142
 washNActiveSweepers 142
 washNBuffers 142
 washNDirtyBuffers 142
 washSignalThreshold 142
 washTeamSize 142
 Writes 142
 动态 86
 性能统计信息 137
 页数 140
 吞吐量 47
 最大化 121

W

外部引用
 已定义 29
 网络
 大型传输 82
 设置 81
 性能建议 81
 网络统计信息 91
 文档
 SQL Anywhere xi
 Sybase IQ ix

文件
 定位以达到最佳性能 66
 物理内存
 监控 122

X

系统触发器
 查看单个分析信息 99
 查看分析数据 93
 查看摘要分析数据 97
 系统存储过程 2
 系统过程
 sp_iqcolumnuse 43
 sp_iqindexuse 43
 sp_iqtableuse 43
 sp_iqunusedcolumn 43
 sp_iqunusedindex 43
 sp_iqunusedtable 43
 sp_iqworkmon 43
 下推连接 80
 限制 6
 线程
 管理选项 62
 缓冲区高速缓存 112
 监控 105
 用 -iqnumbercpus 开关控制使用 70
 线程管理器 135
 线程统计信息 90
 相关名
 关于 20
 已定义 29
 响应时间 47
 消息日志
 Sybase IQ 68
 性能 121
 CIS 功能性补偿影响 36
 磁盘高速缓存 73
 定义 47
 动态监控器 86
 多用户 71
 非规范化数据库的优势 78
 监控 101

- 平衡 I/O 63
 - 设计 48
 - 提高 12
 - 选择正确索引类型 74
 - 用 iqgovern 限制查询 69
 - 子查询 35
 - 性能调优
 - 简介 85
 - 性能和调优问题 126
 - 性能监控器
 - 2D 条形图 87
 - 3D 条形图 87
 - 保存图表 86
 - 打印图表 87
 - 服务器级 86
 - 切换图表视图 87
 - 时序视图 87
 - 示例 107
 - 所监控的统计信息 88
 - 自定义统计信息 86
 - 自定义图表 87
 - 性能统计信息
 - CPU 时间统计信息 135
 - CPU 时间统计信息, 已定义 135
 - IQ 存储空闲列表, 统计信息 133
 - IQ 存储空闲列表, 已定义 133
 - sp_iqsysmon 129
 - sp_iqsysmon, 定义 129
 - 部分名称链接 129
 - 部分统计信息定义链接 129
 - 缓冲池, 统计信息 132
 - 缓冲池, 已定义 132
 - 缓冲区分配器, 统计信息 133
 - 缓冲区分配器, 已定义 133
 - 缓冲区分配器 / 预取客户端类型 144
 - 缓冲区管理器, 定义 130
 - 缓冲区管理器, 统计信息 130
 - 目录、数据库日志和存储库统计信息 136
 - 目录、数据库日志和存储库统计信息, 已定义 136
 - 内存管理器, 统计信息 135
 - 内存管理器, 已定义 135
 - 上下文服务器统计信息 136
 - 上下文服务器统计信息, 已定义 136
 - 事务管理器, 统计信息 136
 - 事务管理器, 已定义 136
 - 锁统计信息 143
 - 统计信息, 列表 137
 - 统计信息类型定义链接 129
 - 线程管理器, 统计信息 135
 - 线程管理器, 已定义 135
 - 预取管理器, 统计信息 133
 - 预取管理器, 已定义 133
 - 字段名称链接 129
 - 字段引用链接 129
 - 虚拟 Sybase IQ 表 4
 - 虚拟地址空间 123
 - 虚拟内存
 - 碎片 59
 - 选项
 - FLATTEN_SUBQUERIES 35
 - INDEX_ADVISOR 37
 - NOEXEC 37
 - QUERY_DETAIL 37
 - QUERY_PLAN 37
 - QUERY_PLAN_AFTER_RUN 37
 - QUERY_PLAN_AS_HTML 38
 - QUERY_PLAN_AS_HTML_DIRECTORY 38
 - QUERY_PLAN_TEXT_ACCESS 37
 - QUERY_PLAN_TEXT_CACHING 38
 - QUERY_TIMING 38
 - 意外行为 4
 - 选项值
 - 截断 36
- ## Y
- ### 页类型
- BARRAY 143
 - BM 143
 - BTREEF 142
 - BTREEV 142
 - BV 142
 - CKPT 143
 - CMID 143
 - DBEXT 142
 - DBID 142

- GARRAY 143
- HASH 143
- LOB 143
- RIDCA 143
- SORT 142
- sp_iqsysmon 142
- STORE 143
- VDO 142
- 页面错误 121
 - 监控 124
- 页数
 - 解压缩 125
- 引号
 - 使用 6
- 用户定义的函数
 - 性能影响 36
- 用户提供的条件
 - 用于查询 42
- 优化查询 34, 74
- 游标
 - 限制数目 71
- 预取的高速缓存页 71
- 预取管理器 133

Z

- 在 Interactive SQL 中查看过程分析信息 99
- 在 Interactive SQL 中查看特定过程的分析信息 100
- 摘要分析数据
 - 触发器 97
 - 存储过程和函数 97
 - 事件 97
 - 系统触发器 97
- 展平子查询 35
- 直接 I/O 60
- 主数据库
 - 缓冲区高速缓存大小 54
- 装载改进 81

- 装载数据
 - HG 和 WD 索引 81
 - 并行装载 81
 - 调优 125
 - 使用分条磁盘 65
 - 装载时访问 81
- 子查询
 - 使用 28
- 子查询性能 35
- 子查询展平 35
- 字母顺序 4

