

SYBASE®

非结构化数据分析

Sybase IQ

15.2

文档 ID: DC00958-01-1520-01

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他国家 / 地区的客户可通过上述传真号码与客户服务部门联系。所有其它国际客户请与 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。

美国政府使用、复制或公开本软件受 DFARS 52.227-7013 中的附属条款 (c)(1)(ii) (针对美国国防部) 和 FAR 52.227-19(a)-(d) (针对美国非军事机构) 条款的限制。

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

目录

关于本手册	vii	
第 1 章	Sybase IQ 中的非结构化数据分析简介	1
	非结构化数据分析选件	1
	全文本搜索	2
	兼容性	2
	对标准的遵从性	2
第 2 章	TEXT 索引和文本配置对象	3
	TEXT 索引	3
	WD 和 TEXT 索引的比较	3
	创建 TEXT 索引	5
	TEXT 索引大小估计指南	5
	TEXT 索引限制	6
	显示 TEXT 索引列表	6
	编辑 TEXT 索引	7
	修改位置	8
	删除 TEXT 索引	8
	刷新 TEXT 索引	9
	删除 TEXT 索引中的行	9
	文本配置对象	10
	缺省文本配置对象	11
	创建文本配置	11
	文本配置对象设置	13
	显示文本配置的列表	16
	变更文本配置	17
	修改停止列表	17
	删除文本配置	18
	文本配置对象示例	18
	限制文本搜索中的前缀词语	20

第 3 章	外部库	23
	了解外部库	23
	外部库限制	24
	在 Multiplex 服务器上使用外部库	24
	在启动时启用和禁用外部库	25
	卸载外部库	25
第 4 章	非结构化数据查询	27
	全文本搜索	27
	全文本搜索的类型	27
	查询 LONG BINARY 列	35
	查询 LONG VARCHAR 列	36
	监控 LONG BINARY 和 LONG VARCHAR 列的性能	37
第 5 章	存储过程支持	39
	管理 TEXT 索引中的词语	39
	sa_char_terms 系统过程	39
	sa_nchar_terms 系统过程	40
	sa_text_index_stats 系统过程	41
	sa_text_index_vocab 系统过程	41
	标识外部库	42
	sa_external_library_unload 系统过程	43
	sa_list_external_library 过程	43
	控制大对象数据压缩	44
	sp_iqsetcompression 过程	44
	sp_iqshowcompression 过程	45
	显示有关大对象列的信息	45
第 6 章	大对象数据装载和卸载	47
	导出大对象数据	47
	BFILE 函数	47
	装载大对象数据	48
第 7 章	大对象数据类型	53
	大对象数据类型 LONG BINARY 和 BLOB	53
	大对象数据类型 LONG VARCHAR 和 CLOB	54
	大对象变量	56
	大对象变量数据类型转换	56
	大对象列的索引支持	57
	大对象列的 TEXT 索引支持	57
	LONG VARCHAR (CLOB) 列的 WD 索引支持	57

第 8 章	SQL 语句支持	59
	ALTER TEXT CONFIGURATION 语句	59
	ALTER TEXT INDEX 语句	61
	CREATE TEXT CONFIGURATION 语句	62
	CREATE TEXT INDEX 语句	63
	DROP TEXT CONFIGURATION 语句	64
	DROP TEXT INDEX 语句	65
第 9 章	函数支持	67
	大对象数据的函数支持摘要	67
	BIT_LENGTH 函数	68
	BYTE_LENGTH 函数	69
	BYTE_LENGTH64 函数	69
	BYTE_SUBSTR64 和 BYTE_SUBSTR 函数	69
	CHAR_LENGTH 函数	70
	CHAR_LENGTH64 函数	70
	CHARINDEX 函数	71
	LOCATE 函数	71
	OCTET_LENGTH 函数	72
	PATINDEX 函数	73
	SUBSTRING 函数	73
	SUBSTRING64 函数	74
	大对象列的集合函数支持	75
附录 A	错误和警告消息	77
	错误 1000195	78
	错误 1000198	78
	错误 1000332	79
	错误 1001013	79
	错误 1001051	80
	错误 1001052	80
	错误 1001053	81
	错误 1001054	81
	警告 1001055	82
	警告 1001056	82
	错误 1001057	83
	错误 1001058	84
	错误 1009189	84
	错误 1012030	85
索引		87

关于本手册

读者

本手册的目标读者是需要有关在 Sybase IQ 中处理非结构化数据的参考材料的 Sybase® IQ 用户。本手册提供了与 Sybase IQ 中的非结构化数据分析功能相关的可用语法、参数、函数、存储过程、索引和选件等信息。将本手册与 Sybase IQ 文档集中的一些其它书籍一同用作参考资料，以了解如何在 Sybase IQ 数据库中存储和检索非结构化数据。

相关的 Sybase IQ 文档

Sybase IQ 15.2 文档集包括：

- 适用于所用平台的《发行公告》— 包含未能及时写入手册的最新信息。
最新版本的发行公告可能已提供。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 产品手册” (Sybase Product Manuals) 网站。
- 适用于所用平台的《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件。
- 《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用户的练习。
- 《性能和调优指南》— 介绍有关非常大的数据库的查询优化、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：第一卷》— 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：第二卷》— 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》— 介绍用户时序预测和分析的 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》— 解释如何在 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。

Sybase IQ 15.2 文档集可以在以下位置通过联机方式获得：位于 <http://sybooks.sybase.com> 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® 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#) 生成和配置数据库应用程序。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, 还可以参见位于 http://dcx.sybase.com/dcx_home.php 的 DocCommentXchange 以及 “Sybase 产品手册” (Sybase Product Manuals) 网站上的 SQL Anywhere 11.0.1 中的 SQL Anywhere 文档。



Sybase IQ 中的非结构化数据分析简介

本章向您介绍 Sybase IQ 中的非结构化数据分析，并讲述与 Sybase IQ 大对象数据的兼容性和一致性。

主题	页码
非结构化数据分析选件	1
兼容性	2
对标准的遵从性	2

非结构化数据分析选件

非结构化数据分析选件将 Sybase IQ 的功能扩展到允许在 Sybase IQ 数据库中存储、检索和全文搜索二进制大对象 (BLOB) 和字符大对象 (CLOB)。

注释 用户必须经过专门授权才能使用本产品文档中介绍的非结构化数据分析功能。

随着数据量的增大，在关系数据库中存储大对象 (LOB) 数据的需求也随之增长。LOB 数据可以为以下形式之一：

- 非结构化 — 数据库只用于存储和检索数据，或者
- 半结构化（如文本）— 数据库支持数据结构并会提供支持函数（例如，字符串函数）。

典型的 LOB 数据源包括图像、地图、文档（如 PDF 文件、字处理文件和演示文稿）、音频、视频和 XML 文件。Sybase IQ 可以管理各个包含千兆字节 (GB)、千吉字节 (TB)，甚至是千万亿字节 (PB) 数据的 LOB 对象。

通过允许关系数据和非结构化数据存在于同一位置，Sybase IQ 能让组织使用同样的应用程序和同样的界面访问这两种类型的数据。Sybase IQ 的全文本搜索功能支持在处理非结构化和半结构化数据时使用文本存档应用程序（文本分析）。

全文本搜索

全文本搜索使用 TEXT 索引在数据库中搜索词语和词组，而不必扫描表行。

TEXT 索引存储带索引的列中各个词的位置信息。文本配置对象控制着 TEXT 索引在生成或刷新时其中存放的词语，以及全文本查询的解释方式。

使用 TEXT 索引查找包含某个词语或词组的行通常要比扫描表中的每一行快。

兼容性

SQL Anywhere (SA) 可在数据类型为 LONG VARCHAR 或 LONG BINARY 的列中存储大对象（最大长度达 2GB）。SQL Anywhere 对这些数据类型的支持符合 SQL/2003。SQL Anywhere 不支持 BYTE_LENGTH64、BYTE_SUBSTR64、BFILE、BIT_LENGTH、OCTET_LENGTH、CHAR_LENGTH64 和 SUBSTRING64 函数。

Adaptive Server® Enterprise (ASE) 可在数据类型为 TEXT 或 IMAGE 的列中分别存储大文本对象（最大长度达 2GB）和大文本对象（最大长度达 2GB）。Adaptive Server Enterprise 对这些数据类型的支持是 ANSI SQL Transact-SQL® 扩展。

对标准的遵从性

Sybase IQ LONG BINARY 和 LONG VARCHAR 功能符合 ISO/ANSI SQL 标准的核心级别。

TEXT 索引和文本配置对象

本章介绍如何处理 TEXT 索引和文本配置对象。TEXT 索引存储带索引的列中各个词的位置信息。TEXT 索引是使用文本配置对象中存储的设置创建的。文本配置对象控制着 TEXT 索引数据的特性，如要忽略的词语，以及要在索引中包含的词语的最小和最大长度。

主题	页码
TEXT 索引	3
文本配置对象	10

TEXT 索引

在全文本搜索中，会搜索 TEXT 索引，而不是表行。必须先要在要搜索的列上创建 TEXT 索引，然后才能执行全文本搜索。TEXT 索引存储带索引的列中各个词的位置信息。与必须扫描表中所有值的查询相比，使用 TEXT 索引的查询通常更快。

当创建 TEXT 索引时，可以指定在创建和刷新 TEXT 索引时要使用的文本配置对象。文本配置对象中包含影响索引生成方式的设置。如果不指定文本配置对象，数据库服务器就会使用缺省配置对象。

可以在以下类型的列上创建 TEXT 索引：CHAR、VARCHAR 和 LONG VARCHAR，以及 BINARY、VARBINARY 和 LONG BINARY。BINARY、VARBINARY 和 LONG BINARY 列要求 TEXT 索引必须将文本配置和外部预过滤器库结合使用。

WD 和 TEXT 索引的比较

下表对 WD 和 TEXT 索引进行语法和功能方面的比较。

表 2-1: WD 和 TEXT 索引

功能	是否被 WD 索引支持?	是否被 TEXT 索引支持?
结合词语	是, 采用如下形式: tbl.col CONTAINS('great','white', 'whale')	是, 采用如下形式: CONTAINS(tbl.col,'great white whale')
常规布尔型表达式	是, 采用如下形式: tbl.col CONTAINS ('great') AND (tbl.col CONTAINS('white) OR tbl.col CONTAINS('whale') AND NOT tbl.col CONTAINS('ship'))	是, 采用如下形式: CONTAINS(tbl.col, 'great AND (white OR whale AND NOT ship)')
搜索前缀匹配的词语	否	是, 例如: CONTAINS (tbl.col,'whale*')
对 LIKE 谓词进行加速	是, 例如: tbl.col LIKE 'whale%'	否
搜索邻近的词语	否	是, 例如: CONTAINS(tbl.col, 'white BEFORE whale') CONTAINS(tbl.col, 'whale NEAR white') CONTAINS(tbl.col, ' "white whale"')
根据搜索得分对结果进行排序	否	是

在 TEXT 索引中, 搜索前缀匹配的词语和搜索 LIKE 表达式有不同的语义, 而且根据文本配置可能返回极为不同的结果。最小长度、最大长度和停止列表的指定将管控前缀处理, 但不影响 LIKE 语义。

注释 当发生词语删除时, 布尔型表达式的含义在 WD 索引和 TEXT 索引之间是不同的, 因为 TEXT 索引处理中被删除词语的影响在 WD 索引中没有等效项。

创建 TEXT 索引

必须先要在要搜索的列上创建 TEXT 索引，然后才能执行全文本搜索。TEXT 索引存储带索引的列中各个词的位置信息。

❖ 创建 TEXT 索引 (Sybase Central)

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 在左窗格中右键单击 Text Indexes 文件夹，选择“新建” (New) > “文本索引” (Text Index)。
- 3 选择创建 TEXT 索引时所基于的表。
- 4 键入 TEXT 索引的名称。单击“下一步”。
- 5 选择要在索引中包含的列。单击“下一步”。
- 6 选择在处理 TEXT 索引的数据时要使用的文本配置对象。单击“下一步”。
- 7 对于 SQL Anywhere 表，在“指定刷新类型” (Specify a Refresh Type) 对话框中单击“下一步”。
对于 Sybase IQ 表，此选项将不显示。唯一支持的刷新类型是“立即” (Immediate)。
- 8 选择用于存储 TEXT 索引的数据库空间。
- 9 单击“下一步”。
- 10 键入用于描述文本配置的注释，然后单击“完成” (Finish)。

❖ 创建 TEXT 索引 (Interactive SQL)

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 执行 CREATE TEXT INDEX 语句。

此示例在 iqdemo 数据库中的 Customers 表的 CompanyName 列上创建 TEXT 索引 myTxtIdx。用到了 default_char 文本配置对象。

```
CREATE TEXT INDEX myTxtIdx ON Customers
( CompanyName ) CONFIGURATION default_char
```

TEXT 索引大小估计指南

要估计 TEXT 索引的主存储大小，请使用以下格式：

$$\text{字节数} = (15+L)*U + U*PAGESIZE * R + T$$

其中：

- L = 词汇表的平均词语长度
- U = 词汇表中的不重复词语数
- R = 文档数（以百万计）
- T = 所有文档中的所有词语总数

TEXT 索引需要的临时空间（以字节为单位）为 $(M+20)*T$ ，其中：

- M = 文本配置对象的最大词语长度（以字节为单位）

注释 需要的临时空间受限于排序数据的可压缩性。

TEXT 索引限制

Sybase IQ 文本配置对象和 TEXT 索引天生就有以下限制：

- Sybase IQ 引擎不对跨多个列的 TEXT 索引提供支持。
- 不支持 TEXT 索引手动刷新或自动刷新选项。
- sp_iqrebuildindex 不能用于生成 TEXT 索引。
- 不能在 BEGIN PARALLEL IQ...END PARALLEL IQ 内创建 TEXT 索引。
- TEXT 索引不支持 NGRAM 分词符。Sybase IQ 当前仅支持 GENERIC 分词符以及一列一个 TEXT 索引。

显示 TEXT 索引列表

查看数据库中所有 TEXT 索引的列表。

- ❖ **显示 TEXT 索引列表 (Sybase Central)**
 - 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
 - 2 在左窗格中，选择 Text Indexes 文件夹。
所有 TEXT 索引的列表均会显示在右窗格中。
- ❖ **显示 TEXT 索引列表 (Interactive SQL)**
 - 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。

2 执行 SELECT 语句。

例如，要列出所有 Sybase IQ TEXT 索引，请输入：

```
SELECT * FROM sp_iqindex() where index_type = 'TEXT';
```

要列出所有 TEXT 索引（包括目录表上的索引），请使用：

```
select index_name,table_name,name from SYSIDX,  
SYSTEXTIDX, SYSTABLE, SYSUSERS  
where SYSIDX.object_id=SYSTEXTIDX.index_id  
and SYSIDX.table_id=SYSTABLE.table_id  
and SYSTABLE.creator=SYSUSERS.uid;
```

编辑 TEXT 索引

更改 TEXT 索引的设置，包括数据库空间和 TEXT 索引名称。

❖ 编辑 TEXT 索引 (Sybase Central)

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 在左窗格中，选择 Text Indexes 文件夹。
- 3 在“文本索引”(Text Indexes)列表中，右键单击要修改的对象，然后选择“属性”。
- 4 在“常规”选项卡中，根据需要修改设置。
- 5 单击“确定”。

❖ 编辑 TEXT 索引 (Interactive SQL)

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 执行 ALTER TEXT INDEX 语句。

例如，要将 myTxlIdx 重命名为 MyTextIndex，请输入：

```
ALTER TEXT INDEX MyTxlIdx  
ON Customers  
RENAME AS MyTextIndex;
```

修改位置

更改用于存储 TEXT 索引的数据库空间。

❖ 修改位置 (Sybase Central)

- 1 以具有 DBA 或 SPACE ADMIN 权限的用户身份或以对数据库空间具有 CREATE 权限的表所有者身份连接到数据库。
- 2 在左窗格中，选择 Text Indexes 文件夹。
- 3 在“文本索引” (Text Indexes) 列表中，右键单击要修改的对象，然后选择“属性”。
- 4 在“常规”选项卡上，从下拉列表中选择所需的数据库空间。
- 5 在数据库空间更新后，单击“确定”。

❖ 修改位置 (Interactive SQL)

- 1 以具有 DBA 或 SPACE ADMIN 权限的用户身份连接到数据库。
- 2 执行包含 MOVE TO 子句的 ALTER TEXT INDEX 语句。

例如，要将 TEXT 索引 MyTextIndex 移至名为 tispac 的数据库空间，请输入：

```
ALTER TEXT INDEX MyTextIndex ON  
GROUP0.customers MOVE TO tispac;
```

删除 TEXT 索引

从数据库中删除 TEXT 索引。

❖ 删除 TEXT 索引 (Sybase Central)

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。

- 2 在左窗格中，选择 Text Indexes 文件夹。
- 3 在“文本索引”(Text Indexes)列表中，右键单击要修改的对象，然后选择“删除”。
- 4 在确认对话框中单击“是”。

❖ 删除 TEXT 索引 (Interactive SQL)

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 执行 DROP TEXT INDEX 语句。

例如，要删除 MyTextIndex TEXT 索引，请输入：

```
DROP TEXT INDEX MyTextIndex ON Customers;
```

刷新 TEXT 索引

Sybase IQ 表上的 TEXT 索引支持的刷新类型仅有在基础表中的数据变化时发生的“立即刷新”(Immediate Refresh)。立即刷新 Sybase IQ 表上的 TEXT 索引支持隔离级别 3。它们在创建时以及每次使用 INSERT、UPDATE 或 DELETE 语句对列中的数据进行更改时被填充。在初始刷新期间，表上持有排它锁。

删除 TEXT 索引中的行

TEXT_DELETE_METHOD 数据库选项指定在 TEXT 索引中进行删除时所使用的算法。

TEXT_DELETE_METHOD 选项

函数	指定在 TEXT 索引中进行删除时所使用的算法。
允许的值	0 — 2
	0 — 开销模型选择删除方法。
	1 — 强制小型删除方法。当要删除的行数在表的总行数中所占的百分比非常小时，小型删除方法非常有用。少量删除算法可以随机访问索引，导致带有大型数据集的高速缓存抖动。
	2 — 强制大型删除方法。该算法将扫描整个索引以搜索要删除的行。当要删除的行数在表的总行数中所占的百分比非常大时，大型删除方法非常有用。

缺省值	0
作用域	设置此选项无需 DBA 权限。可针对单个连接或 PUBLIC 组进行临时设置。设置将立即生效。
说明	<p>该选项选择在 TEXT 索引中执行删除操作时所使用的算法。当该选项未设置或设置为 0 时，由开销模型选择删除方法。开销模型在选择相应的删除算法时会考虑 CPU 相关开销以及 I/O 相关开销。开销模型考虑下列各项：</p> <ul style="list-style-type: none">• 删除行• 索引大小• 索引数据类型的宽度• 索引数据的基数• 可用临时高速缓存• 与计算机相关的 I/O 和 CPU 特性• 可用 CPU 和线程
示例	<p>要在 TEXT 索引中强制大型删除方法，请执行以下操作：</p> <pre>SET TEMPORARY OPTION TEXT_DELETE_METHOD = 2</pre>
另请参见	<p>请参见《性能和调优指南》的 第 3 章“优化查询和删除” 中的 “优化删除操作”。</p>

文本配置对象

文本配置对象控制着 TEXT 索引在生成或刷新时其中存放的词语，以及全文本查询的解释方式。

当数据库服务器创建或刷新 TEXT 索引时，它使用在创建 TEXT 索引时指定的文本配置对象的设置。如果不指定文本配置对象，数据库服务器就会根据编制索引的列中的数据类型选择一个缺省的文本配置对象。在 Sybase IQ 数据库中，始终会用到 `default_char` 文本配置对象。

文本配置对象指定使用哪个预过滤器库和分词符从要编制索引的文本中生成词语。它们指定要在 TEXT 索引中存储的词语的最小和最大长度，以及不应包含的词语的列表。文本配置对象由以下参数组成：

- **文档预过滤器** 删除文档中不必要的信息，如格式设置和图像。然后，被过滤的文档将被其它模块选取以进行进一步的处理。文档预过滤器是由第三方供应商提供的。
- **文档分词符** 将进入的字节流拆分成用词语分隔符分隔的词语，或者根据指定的规则进行拆分。文档分词符是由服务器或第三方供应商提供的。
- **停止列表处理器** 指定在生成 TEXT 索引时将忽略的词语的列表。

缺省文本配置对象

Sybase IQ 提供缺省文本配置对象 `default_char`，以用于非 NCHAR 数据。此配置是在您首次创建文本配置对象或 TEXT 索引时创建的。此外，还支持文本配置对象 `default_nchar`，以用于 IN SYSTEM 表上的 TEXT 索引的 NCHAR；不能将 `default_nchar` 文本配置用于 Sybase IQ 表上的 TEXT 索引。

表 2-2 显示 `default_char` 和 `default_nchar` 的缺省设置，对于大多数基于字符的语言这是最适合的设置。Sybase 强烈建议您不要更改缺省文本配置对象中的设置。

表 2-2: 缺省文本配置设置

设置	安装的值
TERM BREAKER	GENERIC
MINIMUM TERM LENGTH	1
MAXIMUM TERM LENGTH	20
STOPLIST	(空)

如果您删除缺省文本配置对象，则在下次您创建 TEXT 索引或文本配置对象时，会自动使用缺省值重新创建它。

创建文本配置

创建文本配置来指定取决于文本配置进程的 TEXT 索引如何处理数据中的词语。

❖ 创建文本配置 (Sybase Central)

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 在左窗格中右键单击 Text Configurations Objects 文件夹，选择“新建” (New) > “文本配置对象” (Text Configuration Object)。

- 3 键入文本配置的名称。
- 4 选择文本配置的所有者。
- 5 选择文本配置的数据库归类类型。单击“下一步”。

注释 Sybase IQ TEXT 索引不支持具有 NCHAR 归类的文本配置。

- 6 选择通用分词符算法。

注释 Sybase IQ 文本配置不支持 N-gram 算法。请参见第 13 页的“文本配置对象设置”。

- 7 输入最小和最大长度。
- 8 如果使用的是外部分词符库，请选择“使用外部分词符”(Use an external term breaker) 并指定外部分词符函数和库。

以如下格式指定函数和库：

```
function-name@library-file-name.
```

- 9 单击“下一步”。
- 10 如果使用的是外部预过滤器库，请选择“使用外部预过滤器”(Use an external prefilter) 并指定外部预过滤器函数和库。

以如下格式指定函数和库：

```
function-name@library-file-name.
```

- 11 向停止列表中添加在使用此文本配置生成 TEXT 索引时要忽略的词语。用空格分隔词语。

此列表中的词语也会在查询中被忽略。

- 12 单击“下一步”。
- 13 键入用于描述文本配置的注释，然后单击“完成”(Finish)。

❖ **创建文本配置对象 (Interactive SQL)**

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 执行 CREATE TEXT CONFIGURATION 语句。

例如，要将 default_char 文本配置对象用作模板创建名为 myTxtConfig 的文本配置对象，请输入：

```
CREATE TEXT CONFIGURATION myTxtConfig FROM
default_char;
```

文本配置对象设置

下表解释文本配置对象设置、它们如何影响编制索引的内容，以及如何解释全文本搜索查询。有关文本配置对象及其对 TEXT 索引和全文本搜索的影响，请参见第 18 页的“文本配置对象设置解释”。

分词符算法 (TERM BREAKER) TERM BREAKER 设置指定用于将字符串拆分成词语的算法。Sybase IQ 支持将 GENERIC（缺省设置）用于存储词语。

注释 用于存储 n-gram 的 NGRAM 分词符（一个 n-gram 是一组长度为 n 的字符，其中 n 是 MAXIMUM TERM LENGTH 的值）仅在 IN SYSTEM 表中得到支持，不能用在 Sybase IQ TEXT 索引中。

无论您指定什么分词符，当词语插入 TEXT 索引中时，数据库服务器都会在 TEXT 索引中记录词语的原始位置信息。如果是 n-gram，则会存储 n-gram 的位置信息，而不是原词语的位置信息。

表 2-3: TERM BREAKER 影响

对于 TEXT 索引	对于查询词语
<p>GENERIC TEXT 索引 当生成 GENERIC TEXT 索引（缺省设置）时，非字母数字字符之间出现的字母数字字符组被数据库服务器当作词语处理。在定义了词语后，超过词语长度设置的词语以及位于停止列表中的词语将被计数，但不插入 TEXT 索引中。</p> <p>GENERIC TEXT 索引上的性能可能比 NGRAM TEXT 索引上的快。但在 GENERIC TEXT 索引上无法执行模糊搜索。</p>	<p>GENERIC TEXT 索引 当查询 GENERIC TEXT 索引时，对待查询字符串中的词语就好像是在对其编制索引一样。匹配是通过将查询词语与 TEXT 索引中的词语进行比较来完成的。</p>
<p>NGRAM TEXT 索引 * 在生成 NGRAM TEXT 索引时，数据库服务器会将非字母数字字符之间的任何字母数字字符组都看作词语。一旦定义了词语，数据库服务器就会将词语拆分成 n-gram。这样，长度小于 n 的词语和位于停止列表中的 n-gram 就会被丢弃。</p> <p>例如，对于 MAXIMUM TERM LENGTH 为 3 的 NGRAM TEXT 索引，字符串“my red table”在 TEXT 索引中表示为以下 n-gram：red tab abl ble。</p>	<p>NGRAM TEXT 索引 * 当查询 NGRAM TEXT 索引时，对待查询字符串中的词语就好像是在对其编制索引一样。匹配是通过将查询词语中的 n-gram 和带索引词语中的 n-gram 进行比较完成的。</p>

* 只有 IN SYSTEM 表中支持 NGRAM TEXT 索引。

最小词语长度设置 (MINIMUM TERM LENGTH) MINIMUM TERM LENGTH 设置为插入索引中或在全文本查询中搜索的词语指定最小长度（以字符数为单位）。MINIMUM TERM LENGTH 不与 NGRAM TEXT 索引相关。

MINIMUM TERM LENGTH 有特殊的前缀搜索含义。MINIMUM TERM LENGTH 的值必须大于 0。如果将其设置为大于 MAXIMUM TERM LENGTH，则 MAXIMUM TERM LENGTH 会自动调整为等于 MINIMUM TERM LENGTH。

MINIMUM TERM LENGTH 的缺省值是根据缺省文本配置对象中的设置得到的，通常为 1。

表 2-4: MINIMUM TERM LENGTH 影响

对于 TEXT 索引	对于查询词语
GENERIC TEXT 索引 对于 GENERIC TEXT 索引, TEXT 索引将不包含比 MINIMUM TERM LENGTH 短的单词。	GENERIC TEXT 索引 当查询 GENERIC TEXT 索引时, 比 MINIMUM TERM LENGTH 短的查询词语会被忽略, 因为它们不能存在于 TEXT 索引中。
NGRAM TEXT 索引* 对于 NGRAM TEXT 索引, 此设置会被忽略。	NGRAM TEXT 索引* MINIMUM TERM LENGTH 设置对 NGRAM TEXT 索引上的全文本查询没有影响。

* 只有 IN SYSTEM 表中支持 NGRAM TEXT 索引。

最大词语长度设置 (MAXIMUM TERM LENGTH) 根据分词符算法的不同, MAXIMUM TERM LENGTH 设置有不同的用法。MAXIMUM TERM LENGTH 的值必须小于等于 60。如果将其设置为小于 MINIMUM TERM LENGTH, 则 MINIMUM TERM LENGTH 会自动调整为等于 MAXIMUM TERM LENGTH。

此设置的缺省值是根据缺省文本配置对象中的设置得到的, 通常为 20。

表 2-5: MAXIMUM TERM LENGTH 影响

对于 TEXT 索引	对于查询词语
GENERIC TEXT 索引 对于 GENERIC TEXT 索引, MAXIMUM TERM LENGTH 为插入 TEXT 索引中的词语指定最大长度 (以字符数为单位)。	GENERIC TEXT 索引 对于 GENERIC TEXT 索引, 比 MAXIMUM TERM LENGTH 长的查询词语会被忽略, 因为它们不能存在于 TEXT 索引中。
NGRAM TEXT 索引* 对于 NGRAM TEXT 索引, MAXIMUM TERM LENGTH 决定词语拆分成的 n-gram 的长度。MAXIMUM TERM LENGTH 的长度有哪些适用的选项, 要取决于所用语言。典型值为英文 4 或 5 个字符或中文 2 或 3 个字符。	NGRAM TEXT 索引* 对于 NGRAM TEXT 索引, 查询词语被拆分成长度为 n 的 n-gram, 其中 n 和 MAXIMUM TERM LENGTH 相同。数据库服务器使用这些 n-gram 来搜索 TEXT 索引。比 MAXIMUM TERM LENGTH 短的词语会被忽略, 因为它们不与 TEXT 索引中的 n-gram 匹配。

* 只有 IN SYSTEM 表中支持 NGRAM TEXT 索引。

停止列表设置 (STOPLIST) 停止列表设置指定不编制索引的词语。此设置的缺省值是根据缺省文本配置对象中的设置得到的, 通常会是一个空的停止列表。

表 2-6: STOPLIST 影响

对于 TEXT 索引	对于查询词语
GENERIC TEXT 索引 对于 GENERIC TEXT 索引，位于停止列表中的词语不会插入 TEXT 索引中。	GENERIC TEXT 索引 对于 GENERIC TEXT 索引，位于停止列表中的查询词语会被忽略，因为它们不能存在于 TEXT 索引中。
NGRAM TEXT 索引 * 对于 NGRAM TEXT 索引，TEXT 索引不包含根据停止列表中的词语形成的 n-gram。	NGRAM TEXT 索引 * 停止列表中的词语被拆分成 n-gram，n-gram 用于停止列表。同样，拆分成 n-gram 的查询词语以及任何与停止列表中的 n-gram 匹配的查询词语都会被删除，因为它们不能存在于 TEXT 索引。

* 只有 IN SYSTEM 表中支持 NGRAM TEXT 索引。

请仔细考虑是否将词语放入停止列表中。特别是，不要包含其中含有非字母数字字符（如撇号和横线）的单词。这些字符充当分词符。例如，单词 you'll（必须指定为“you'll”）拆分成 you 和 ll，并以这两个词语的形式存储在停止列表中。以后对“you”或“they'll”进行全文搜索时会受到负面影响。

NGRAM TEXT 索引中的停止列表可能会导致意外结果，因为存储的停止列表实际上采用的是 n-gram 形式，而不是您指定的实际停止列表词语。例如，在 MAXIMUM TERM LENGTH 为 3 的 NGRAM TEXT 索引中，如果您指定 STOPLIST 'there'，则这些 n-gram 会存储为停止列表：the her ere。这会影响到查询任何包含 the、her 和 ere 这三个 n-gram 的词语的能力。

显示文本配置的列表

查看数据库中所有文本配置的列表。

❖ **显示文本配置的列表 (Sybase Central)**

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 在左窗格中选择“文本配置对象”(Text Configurations Objects)。所有文本配置的列表均会出现在右窗格中。

❖ **显示文本配置的列表 (Interactive SQL)**

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 执行 SELECT 语句。

例如，要列出所有文本配置对象，请使用：

```
SELECT * FROM SYSTEXTCONFIG;
```

变更文本配置

更改文本配置对象的设置，包括数据库空间和准许的词语长度范围。只能变更不被 TEXT 索引使用的文本配置对象。

❖ 变更文本配置对象 (Sybase Central)

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 在左窗格中选择“文本配置对象”(Text Configurations Objects)。
- 3 在“文本配置”(Text Configurations)列表中，右键单击要修改的对象，然后选择“属性”。
- 4 切换到“设置”选项卡，根据需要修改设置。
- 5 单击“确定”。

❖ 变更文本配置对象 (Interactive SQL)

- 1 以具有 DBA 或 RESOURCE 权限的用户身份或以文本配置对象的所有者身份连接到数据库。
- 2 执行 ALTER TEXT CONFIGURATION 语句。

例如，要变更 myTxtConfig 文本配置对象的最小词语长度，请使用：

```
ALTER TEXT CONFIGURATION myTxtConfig  
MINIMUM TERM LENGTH 2;
```

修改停止列表

停止列表中包含在使用此文本配置生成 TEXT 索引时要忽略的词语的列表。只能变更不被 TEXT 索引使用的文本配置对象。

❖ 修改停止列表 (Sybase Central)

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 在左窗格中选择“文本配置对象”(Text Configurations Objects)。
- 3 在“文本配置”(Text Configurations)列表中，右键单击要修改的对象，然后选择“属性”。
- 4 切换到“停止列表”(Stoplist)选项卡，根据需要修改停止列表单词。用空格来分隔词语。
- 5 要按字母顺序排序停止列表词语的列表并在列表中显示它们，请单击“排序词语”(Sort Terms)。
- 6 在停止列表更新后，单击“确定”。

❖ **修改停止列表 (Interactive SQL)**

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 执行包含 STOPLIST 子句的 ALTER TEXT CONFIGURATION 语句。

例如，要向 myTxtConfig 配置对象中添加停止列表，请使用：

```
ALTER TEXT CONFIGURATION myTxtConfig  
STOPLIST 'because about therefore only';
```

删除文本配置

从数据库中删除不必要的文本配置。只能删除不被 TEXT 索引使用的文本配置。

❖ **删除文本配置 (Sybase Central)**

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 在左窗格中选择“文本配置对象”(Text Configurations Objects)。
- 3 在“文本配置”(Text Configurations) 列表中，右键单击要修改的对象，然后选择“删除”。
- 4 在确认对话框中单击“是”。

❖ **删除文本配置 (Interactive SQL)**

- 1 以具有 DBA 或 RESOURCE 权限的用户身份连接到数据库。
- 2 执行 DROP TEXT CONFIGURATION 语句。

例如，要删除名为 myTxtConfig 的文本配置对象，请使用：

```
DROP TEXT CONFIGURATION myTxtConfig;
```

文本配置对象示例

查看这些示例以了解文本配置设置如何影响 TEXT 索引，以及索引是如何被解释的。

文本配置对象设置解释

表 2-7 显示了不同文本配置对象的设置，还有这些设置如何影响编制索引的内容以及全文本查询字符串如何被解释。所有示例都使用字符串“I'm not sure I understand”。

表 2-7: 文本配置设置解释

配置设置	编制索引的词语	查询解释
TERM BREAKER:GENERIC MINIMUM TERM LENGTH:1 MAXIMUM TERM LENGTH:20 STOPLIST:"	I m not sure I understand	'("I m" AND not sure) AND I AND understand'
TERM BREAKER:GENERIC MINIMUM TERM LENGTH:2 MAXIMUM TERM LENGTH:20 STOPLIST:'not and'	sure understand	'understand'
TERM BREAKER:GENERIC MINIMUM TERM LENGTH:1 MAXIMUM TERM LENGTH:20 STOPLIST:'not and'	I m sure I understand	'"I m" AND sure AND I AND understand'

文本配置对象包含查询字符串解释

表 2-8 提供文本配置对象字符串的设置如何被解释的示例。

“解释成的字符串”列中的圆括号内数字反映了为每个词语存储的位置信息。这些数字起到在文档中进行说明的作用。实际存储的词语不包括圆括号内数字。

注释 文本文档的最大位置数是 4294967295。

此表中的解释仅适用于 CONTAINS 查询。当分析数据时，AND、NOT 和 NEAR 被认为是常规标识；*、| 等符号以及其它符号会被删除，因为它们不是字母数字字符。

表 2-8: CONTAINS 字符串解释

配置设置	字符串	解释成的字符串
TERM BREAKER:GENERIC	'w*'	'"w*(1) "'
MINIMUM TERM LENGTH:3	'we*'	'"we*(1) "'
MAXIMUM TERM LENGTH:20	'wea*'	'"wea*(1) "'
	'we* -the'	'"we*(1) " -"the(1) "'
	'for* wonderl*'	'"for*(1) " "wonderl*(1) "'
	'wonderlandwonderlandwonderland*'	' '
	'"tr* weather"'	'"weather(1) "'
	'"tr* the weather"'	'"the(1) weather(2) "'
	'"wonderlandwonderlandwonderland* wonderland"'	'"wonderland(1) "'
	'"wonderlandwonderlandwonderland* weather"'	'"weather(1) "'
	'"the_wonderlandwonderlandwonderland* weather"'	'"the(1) weather(3) "'
	'the_wonderlandwonderlandwonderland* weather'	'"the(1) " & "weather(1) "'
	'"light_a* the end" & tunnel'	'"light(1) the(3) end(4) " & "tunnel(1) "'
	light_b* the end" & tunnel'	'"light(1) the(3) end(4) " & "tunnel(1) "'
	'"light_at_b* end"'	'"light(1) end(4) "'
	'and-te*'	'"and(1) te*(2) "'
	'a_long_and_t* & journey'	'"long(2) and(3) t*(4) " & "journey(1) "'

限制文本搜索中的前缀词语

MAX_PREFIX_PER_CONTAINS_PHRASE 数据库选项指定在文本搜索表达式中允许的前缀词语数目。

MAX_PREFIX_PER_CONTAINS_PHRASE 选项

函数 指定在文本搜索表达式中允许的前缀词语数目。

允许的值 0 — 300

0 — 对搜索词组中的前缀词语没有限制

	300 — 上限（这是对词组中允许的词语总数的总限制）
缺省值	1
作用域	设置此选项无需 DBA 权限。可针对单个连接或 PUBLIC 组进行临时设置。设置将立即生效。
说明	MAX_PREFIX_PER_CONTAINS_PHRASE 指定用于禁止文本搜索表达式中有多个前缀的阈值。当此选项设置为 0 时，表示任何数目都是允许的。如果查询中有的 CONTAINS 表达式含有一个其前缀词语数目超过此选项指定数目的词语，Sybase IQ 就会检测到并报告错误。
示例	<p>如果采用缺省的 MAX_PREFIX_PER_CONTAINS_PHRASE 设置：</p> <pre>SET MAX_PREFIX_PER_CONTAINS_PHRASE = 1</pre> <p>以下 CONTAINS 子句就是有效的：</p> <pre>SELECT ch1 FROM tab1 WHERE CONTAINS(ch1, "concord bed* in mass")</pre> <p>如果采用缺省的 MAX_PREFIX_PER_CONTAINS_PHRASE 设置，以下 CONTAINS 子句就会返回语法错误：</p> <pre>SELECT ch1 FROM tab1 WHERE CONTAINS (ch1, "con* bed* in mass")</pre> <p>如果将 MAX_PREFIX_PER_CONTAINS_PHRASE 设置为等于 0（没有限制）或 2，此 CONTAINS 子句就是有效的。</p>

本章介绍如何使用外部库来为文档提供预过滤和分词。

主题	页码
了解外部库	23
在 Multiplex 服务器上使用外部库	24
在启动时启用和禁用外部库	25
卸载外部库	25

了解外部库

Sybase IQ 可以在索引创建或查询处理期间使用用 C 或 C++ 编写的外部预过滤器和分词符库来预过滤并标识文档。这些库可以动态装载到数据库服务器的进程空间中。

注释 外部预过滤器和分词符库必须由 Sybase 认证的合作伙伴提供。有关已认证供应商解决方案的信息，请参见位于 http://www.sybase.com/detail_list?id=9784 的“合作伙伴认证报告” (Partner Certification Reports) 网站然后过滤认证报告以显示 Sybase IQ 认证。

外部的动态装载的预过滤器和分词符库是在文本配置中指定的，需要由数据库服务器装载。每个库中都包含一个导出的符号，用以实现在文本配置对象中指定的外部函数。此函数返回一组函数描述符，调用者会使用这些函数描述符来执行必要的任务。

外部预过滤器和分词符库是由数据库服务器在处理第一个 CREATE TEXT INDEX 请求时装载的，这种情况发生在收到的针对给定列的查询要求装载库时或者 TEXT 索引需要更新时。

在进行 ALTER TEXT CONFIGURATION 调用时不会装载这些库，在进行 DROP TEXT CONFIGURATION 调用时也不会自动卸载它们。如果服务器在启动时使用了禁止装载外部库的启动选项，则不会装载外部预过滤器和分词符库。

由于这些外部 C/C++ 库涉及将非服务器库代码装载到服务器的进程空间中，因此，编写错误或恶意编写的函数会导致数据完整性、数据安全性和服务器的稳定性存在潜在风险。为了管理这些风险，每个 Sybase IQ 服务器都可以显式启用或禁用此功能。请参见第 25 页的“在启动时启用和禁用外部库”。

ISYTEXTCONFIG 系统表存储有关与文本配置对象关联的外部库的信息。请参见《参考：构件块、表和过程》的第 8 章“系统表和视图”中的“SYTEXTCONFIG 系统视图”。

外部库限制

使用外部库的 Sybase IQ 文本配置对象和 TEXT 索引存在以下限制：

- 对于二进制列上的 TEXT 索引，您必须使用外部供应商提供的外部库来进行文档转换。Sybase IQ 不会隐式转换二进制列中存储的文本。
- 如果使用外部分词符来标识文档，就会不支持基于 N-gram 的文本搜索。
- 无法使用外部库在 SQL Anywhere 表上创建 TEXT 索引；这样做会导致错误。

在 Multiplex 服务器上使用外部库

所有 Multiplex 服务器都必须具有访问预过滤器和分词符外部库的权限。用户必须确保每个外部预过滤器和分词符库都复制到承载 Multiplex 服务器的计算机上，并且放入服务器能够装载库的位置。

每个 Multiplex 服务器都在调用外部预过滤器和分词符库时独立于其它服务器而工作。每个进程空间都可以让外部库装载进来并完成自己的执行。假定预过滤器和分词符函数的实现在每个服务器上都一样，而且将返回相同的结果。

从一个服务器进程空间中卸载某个外部库并不会将该库从其它服务器进程空间中卸载。

在启动时启用和禁用外部库

Sybase IQ 提供启动开关来允许或禁止装载外部第三方库。此开关可以在服务器启动命令行中指定，也可以在服务器配置文件中指定。

要允许装载外部第三方库，请执行以下操作：

```
-sf -external_library_full_text
```

要禁止装载外部第三方库，请执行以下操作：

```
-sf external_library_full_text
```

要查看服务器中当前装载的库的列表，请使用 `sa_list_external_library` 存储过程。

卸载外部库

当外部库不在使用中时，可以使用系统过程 `dbo.sa_external_library_unload` 来卸载该库。该过程采用一个可选的类型为 `long varchar` 的参数。该参数指定要卸载的库的名称。如果不指定参数，则所有不在使用中的外部库都会被卸载。

以下示例用于卸载外部函数库。

```
call sa_external_library_unload('library.dll')
```


本章介绍如何查询大对象数据，包括用于处理非结构化和半结构化数据的全文本搜索功能。

主题	页码
“全文本搜索”	27
“查询 LONG BINARY 列”	35
“查询 LONG VARCHAR 列”	36
“监控 LONG BINARY 和 LONG VARCHAR 列的性能”	37

全文本搜索

全文本搜索可以快速在数据库中查找某个字词的所有实例，而不必扫描表行。全文本搜索使用 TEXT 索引，该索引中存储带索引的列中各个词的位置信息。使用 TEXT 索引查找包含某个词语的行要比扫描表中的每一行快。

全文本搜索使用 CONTAINS 搜索条件，这不同于使用 LIKE、REGEXP 和 SIMILAR TO 等谓词进行的搜索，因为这种匹配是基于词语的，而不是基于模式的。请参见第 28 页的“CONTAINS 条件”和《参考：构件块、表和过程》的第 2 章“SQL 语言元素”中的“CONTAINS 条件”。

全文本搜索中的字符串比较使用数据库的所有正常归类设置。例如，将数据库配置为不区分大小写，那么全文本搜索也不区分大小写。

全文本搜索的类型

通过使用全文本搜索，可以搜索词语、前缀或词组（词语序列）。另外，还可以将多个词语、词组或前缀组合到布尔型表达式中，或者使用临近搜索要求表达式看起来彼此接近。

在 WHERE 子句或 SELECT 语句的 FROM 子句中使用 CONTAINS 来执行全文本搜索。

FROM 子句

说明 指定 SELECT 语句中涉及的数据库表或视图。

语法 ... **FROM** *table-expression* [, ...]

参数 *table-expression*:

```
{ table-spec
| table-expression join-type table-spec [ ON condition ]
| ( table-expression [, ...] ) }
```

table-spec:

```
{ [ userid. ] table-name [ [ AS ] correlation-name ]
| select-statement [ AS correlation-name ( column-name [, ...] ) ] }
```

contains-expression:

```
{ table-name | view-name } CONTAINS
( column-name [,...], contains-query ) [ [ AS ] score-correlation-name ]
```

用法 **contains-expression** 在表名后使用 CONTAINS 子句可对表进行过滤，仅返回那些与使用 *contains-query* 指定的全文本查询匹配的行。将会返回表的每个匹配行，还有可以使用 *score-correlation-name*（如果指定）引用的 score 列。如果未指定 *score-correlation-name*，就可以按缺省相关名 *contains* 引用 score 列。

除了可选的相关名参数外，CONTAINS 子句采用与 CONTAINS 搜索条件相同的参数。CONTAINS 子句中所列的列上必须有 TEXT 索引。

请参见第 28 页的“CONTAINS 条件”和《参考：构件块、表和过程》的第 2 章“SQL 语言元素”中的“CONTAINS 条件”。

有关 FROM 子句的完整语法和说明，请参见《参考：语句和选项》的第 1 章“SQL 语句”中的 FROM 子句。

CONTAINS 条件

在 SELECT 语句的 FROM 子句中使用 CONTAINS 子句，或者在 WHERE 子句中使用 CONTAINS 搜索条件（谓词）来执行全文本查询。这两种方法返回同样的行；但 CONTAINS 子句还返回匹配行的分数。

语法 **CONTAINS** (*column-name* [,...], *contains-query-string*)

```
contains-query-string:
    simple-expression
    | or-expression
```

```

simple-expression:
  primary-expression
  | and-expression

or-expression:
  simple-expression { OR | | } contains-query-string

primary-expression:
  basic-expression
  | FUZZY " fuzzy-expression "
  | and-not-expression

and-expression:
  primary-expression [ AND | & ] simple-expression

and-not-expression:
  primary-expression [ AND | & ] { NOT | - } basic-expression

basic-expression:
  term
  | phrase
  | ( contains-query-string )
  | proximity-expression

fuzzy-expression:
  term
  | fuzzy-expression term

term:
  simple-term
  | prefix-term

prefix-term:
  simple-term*

phrase:
  " phrase-string "

proximity-expression:
  term ( BEFORE | NEAR ) [ minimum distance, | maximum distance ]
  term
  | term { BEFORE | NEAR | ~ } term

phrase-string:
  term
  | phrase-string term

```

参数

- **simple-term** 一个用空格和特殊字符分隔的字符串，表示要搜索的单个带索引的词语（字）。
- **distance** 一个正整数。
- **and-expression** 使用 *and-expression* 可以指定 *primary-expression* 和 *simple-expression* 都必须出现在 TEXT 索引中。缺省情况下，如果未在词语或表达式之间指定任何运算符，就会采用 *and-expression*。例如，'a b' 被解释为 'a AND b'。与符号 (&) 可以用来替代 AND，可以连接两端的表达式或词语（例如，'a & b'）。

- **and-not-expression** 使用 *and-not-expression* 可以指定 *primary-expression* 必须出现在 TEXT 索引中, 但 *basic-expression* 不得出现在 TEXT 索引中。这也称为非。当对非使用连字符时, 连字符前面必须加一个空格, 而且连字符必须连接后续的词。例如, 'a -b' 等同于 'a AND NOT b'; 但对于 'a - b', 连字符会被忽略, 字符串等同于 'a AND b'。'a-b' 等同于词组 '"a b"'。
- **or-expression** 使用 *or-expression* 可以指定至少一个 *simple-expression* 或 *contains-query-string* 必须出现在 TEXT 索引中。例如, 'a|b' 被解释为 'a OR b'。
- **fuzzy-expression** 使用 *fuzzy-expression* 可以查找与您指定的内容类似的词。模糊匹配仅在 IN SYSTEM 表的 NGRAM TEXT 索引上得到支持。
- **proximity-expression** 使用 *proximity-expression* 可以搜索彼此接近的词。例如, 'b NEAR[2,5] c' 搜索彼此距离最多 5 个词, 最少 2 个词的 b 和 c 实例。词的顺序不重要; 'b NEAR c' 等同于 'c NEAR b'。如果指定了 NEAR 但没有 *distance*, 就会应用缺省值 10 个词。可以指定否定符号 (~) 来替代 NEAR。这等同于指定了不带 *distance* 的 NEAR, 因此会应用缺省值 10 个词。NEAR 表达式无法链接到一起 (例如, 'a NEAR[1] b NEAR[1] c')。

BEFORE 与 NEAR 类似, 不同之处是词的顺序重要。'b BEFORE c' 不等同于 'c BEFORE b', 在前者中, 词 'b' 必须在 'c' 的前面, 而在后者中, 词 'b' 必须在 'c' 的后面。BEFORE 与 NEAR 一样, 接受最小和最大距离。缺省最小值为 1。如果给定最小距离, 就必须小于或等于最大距离; 否则, 会返回错误。
- **prefix-term** 使用 *prefix-term* 可以搜索以指定前缀开头的词。例如, 'datab*' 搜索所有以 *datab* 开头的词。这也称为前缀搜索。在前缀搜索中, 匹配是针对词中星号左侧的内容执行的。

注释

CONTAINS 搜索条件采用列列表和 *contains-query-string* 作为参数。它可以用在任何可以指定搜索条件（也称为谓词）的地方，返回 **TRUE** 或 **FALSE**。*contains-query-string* 必须是常量字符串，或者变量（其值在查询时是已知的）。

如果指定了多个列，它们就必须引用一个基表；**TEXT** 索引无法跨多个基表。可以在 **FROM** 子句中直接引用基表，也可以在视图或派生表中使用它，但前提条件是视图或派生表不使用 **DISTINCT**、**GROUP BY**、**ORDER BY**、**UNION**、**INTERSECT**、**EXCEPT** 或行限制。

使用 ANSI 连接语法的查询是被支持的（**FULL OUTER JOIN**、**RIGHT OUTER JOIN**、**LEFT OUTER JOIN**），但其性能可能为次优。仅当需要每个 **CONTAINS** 子句中的 **score** 列时，才将外连接用于 **FROM** 子句中的 **CONTAINS**。否则，应将 **CONTAINS** 移到 **ON** 条件或 **WHERE** 子句。

以下查询类型是不被支持的：

- 使用 SQL Anywhere 表的远程查询，而且该表具有与远程表连接的完整 **TEXT** 索引。
- 使用 Sybase IQ 和 SQL Anywhere 表的查询，其中要使用的完整 **TEXT** 索引位于 SQL Anywhere 表上。
- 使用 TSQL 样式外部连接语法（***=***、**=*** 和 ***=**）的查询。

如果您将长度小于 32KB 的 SQL 变量用作搜索词语，并且变量的类型是 **LONG VARCHAR**，则使用 **CAST** 将该变量转换为 **VARCHAR** 数据类型。例如：

```
SELECT * FROM tabl WHERE CONTAINS(c1, cast(v1 AS
    VARCHAR(64))
```

下列警告适用于在查询字符串中使用非字母数字字符的情况：

- 词语中带星号会返回错误。
- 避免在 **fuzzy-expression** 中使用非字母数字字符（包括特殊字符），因为它们被视为空格并充当分词符。
- 尽量避免在查询字符串中使用不是特殊字符的非字母数字字符。任何不是特殊字符的非字母数字字符都会导致包含它的词语被视为词组，从而在该字符所在的位置将词语断开。例如，'things we've done' 被解释为 'things "we ve" done'。
- 在词组中，星号是唯一继续被解释为特殊字符的特殊字符。词组中的所有其它特殊字符都会被视为空格充当分词符。

contains-query-string 的解释分为两个主要步骤:

- **第 1 步: 解释运算符和优先级** 在此步骤中, 关键字被解释为运算符, 并且会应用优先级规则。
- **第 2 步: 应用文本配置对象设置** 在此步骤中, 会向词语应用文本配置对象设置。任何查询词语若超过词语长度设置或位于停止列表中, 都会被删除。

CONTAINS 搜索条件中的运算符优先级

在查询求值期间, 会使用以下优先级顺序对表达式求值:

- 1 FUZZY, NEAR
- 2 AND NOT
- 3 AND
- 4 OR

星号 (*) 的允许语法

星号用于前缀搜索。星号可以出现在查询字符串的结尾, 也可以后跟空格、与符号、竖线、右括号或右引号。星号的任何其它用法都会返回错误。

表 4-1 显示允许的星号用法:

表 4-1: 星号解释

查询字符串	等同于	被解释为
'th*&best'	'th* AND best' 和 'th* best'	查找任何以 th 开头的词语和词语 best。
'th* best'	'th* OR best'	查找任何以 th 开头的词语或词语 best。
'very&(best th*)'	'very AND (best OR th*)'	查找词语 very 和词语 best 或任何以 th 开头的词语。
'"fast auto*"'		查找词语 fast, 且其后紧跟以 auto 开头的词语。
'"auto* price"'		查找以 auto 开头的词语, 且其后紧跟词语 price。

注释 包含星号的查询字符串的解释随文本配置对象设置的不同而不同。

连字符 (-) 的允许语法

连字符可以用于词语或表达式否定, 等同于 NOT。连字符是否被解释为否定取决于其在查询字符串中的位置。例如, 当连字符紧挨在某个词语或表达式的前面时, 它会被解释为否定。如果连字符内嵌在某个词语内, 它就会被解释为连字符。

用于否定的连字符必须前面带一个空格, 而且后面紧跟一个表达式。

当用于模糊表达式的词组时，连字符被视为空格并用作分词符。

表 4-2 显示允许的连字符语法：

表 4-2：连字符解释

查询字符串	等同于	被解释为
'the -best'	'the AND NOT best'、'the AND -best'、'the & -best'、'the NOT best'	查找词语 the，而不是词语 best。
'the -(very best)'	'the AND NOT (very AND best)'	查找词语 the，而不是词语 very 和 best。
'the -"very best"'	'the AND NOT "very best"'	查找词语 the，而不是词组 very best。
'alpha-numeric'	"alpha numeric"	查找词语 alpha，且其后紧跟词语 numeric。
'wild - west'	'wild west' 和 'wild AND west'	查找词语 wild 和词语 west。

特殊字符的允许语法

表 4-3 显示除星号和连字符外的所有特殊字符的允许语法。这些字符如果位于某个词组中，就不会被认为是特殊字符，而且会被删除。

注释 有关指定字符串文字的限制也适用于查询字符串。例如，撇号必须位于转义序列内。

表 4-3：特殊字符解释

字符或语法	用法示例和注释
与符号 (&)	与符号等同于 AND，可以按如下方式指定： <ul style="list-style-type: none"> 'a & b' 'a &b' 'a& b' 'a&b'
竖线 ()	竖线等同于 OR，可以按如下方式指定： <ul style="list-style-type: none"> 'a b' 'a b' 'a b' 'a b'
双引号 (")	双引号用于包含一组顺序和相对距离很重要的词语。例如，在查询字符串 'learn "full text search"' 中，"full text search" 是一个词组。在此示例中，learn 可以来自词组的前面或后面，或者存在于另一列中（如果 TEXT 索引是以多个列为基础生成的），但完全一样的词组必须存在于单个列中。
小括号 ()	小括号用于指定表达式的求值顺序（如果不同于缺省顺序）。例如，'a AND (b c) ' 被解释为 a 和 b 或 c。

字符或语法	用法示例和注释
否定符号 (~)	否定符号等同于 NEAR，没有特殊语法规则。查询字符串 'full~text' 等同于 'full NEAR text'，而且被解释为：词语 text 的十个词语内的词语 full。
中括号 []	中括号和关键字 NEAR 一起使用可以包含 <i>distance</i> 。中括号的其它用法都会返回错误。

被删除词语的影响

TEXT 索引是根据针对用于创建 TEXT 索引的文本配置对象而定义的设置生成的。TEXT 索引排除满足以下任一条件的词语：

- 词语包含在停止列表中。
- 词语短于最小词语长度（仅 GENERIC）。
- 词语长于最大词语长度。

同样的规则适用于查询字符串。被删除的词语可以在词组的结尾或开头与零个或多个词语匹配。例如，假定词语 'the' 位于停止列表中：

- 如果该词语显示在 AND、OR 或 NEAR 的某一侧，则运算符和该词语都会被删除。例如，搜索 'the AND apple'、'the OR apple' 或 'the NEAR apple' 等同于搜索 'apple'。
- 如果该词语出现在 AND NOT 的右侧，则 AND NOT 和该词语都会被删除。例如，搜索 'apple AND NOT the' 等同于搜索 'apple'。
- 如果该词语显示在 AND NOT 的左侧，则整个表达式都会被删除。例如。搜索 'the AND NOT apple' 不返回任何行。另一个示例：'orange and the AND NOT apple' 与 'orange AND (the AND NOT apple)' 相同，而后者在 AND NOT 表达式被删除后等同于搜索 'orange'。将这个与搜索表达式 '(orange and the) and not apple' 对比，后者等同于搜索 'orange and not apple'。
- 如果该词语显示在某个词组中，该词组就可以与被删除词语所在位置的任意词语匹配。例如，搜索 'feed the dog' 与 'feed the dog'、'feed my dog'、'feed any dog' 等匹配。

注释 如果您要搜索的所有词语都被删除，Sybase IQ 就会返回错误 CONTAINS has NULL search term (CONTAINS 有 NULL 搜索词语)。SQL Anywhere 不报告任何错误，并返回零行。

CONTAINS 表表达式

当您在查询的 FROM 子句中包含 CONTAINS 子句时，每个匹配项都有一个与其关联的得分。该得分指明匹配的接近程度，您可以使用得分信息来排序数据。两个主要标准决定得分：

- 词语在带索引的行中出现的次数。词语在带索引的行中出现的次数越多，其得分就越高。
- 词语在 TEXT 索引中出现的次数。词语在 TEXT 索引中出现的次数越多，其得分就越低。

根据全文本搜索的类型，其它标准会影响得分。例如，在邻近搜索中，搜索词语的邻近会影响得分。缺省情况下，CONTAINS 子句的结果集具有相关名 `contains`，该名称有一个名为 `score` 的列。可以在 SELECT 列表、ORDER BY 子句或查询的其它部分中引用 `"contains".score`。但由于 `contains` 是 SQL 保留字，因此，必须记得将其放在双引号中。另外，还可以指定另外的相关名，例如 `CONTAINS (expression) AS ct`。全文本搜索的示例以 `ct.score` 的形式引用 `score` 列。

此语句在 `MarketingInformation.Description` 中搜索以 “stretch” 开头的词语或以 “comfort” 开头的词语：

```
SELECT ID, ct.score, Description
FROM MarketingInformation
CONTAINS ( MarketingInformation.Description,
          'stretch* | comfort*' )
AS ct ORDER BY ct.score DESC;
```

注释 SQL Anywhere 文档提供全文本搜索示例。这些示例并非全都适用于 Sybase IQ 15.2。例如，Sybase IQ 不支持属于 IF 搜索条件的文本搜索，也不允许模糊或 NGRAM 搜索。

另请参见

“SQL Anywhere 11.0.1 > SQL Anywhere Server — SQL 用法 > 查询和修改数据 > 查询数据” 中的 “全文本搜索的类型”。

查询 LONG BINARY 列

在 SELECT 语句的 WHERE 子句中，除 `BYTE_LENGTH64`、`BYTE_SUBSTR64`、`BYTE_SUBSTR`、`BIT_LENGTH`、`OCTET_LENGTH`、`CHARINDEX` 和 `LOCATE` 函数外，只能在 `IS NULL` 和 `IS NOT NULL` 表达式中使用 `LONG BINARY` 列。

LONG BINARY 列不能用在 SELECT 语句的 ORDER BY、GROUP BY 和 HAVING 子句中，也不能与 DISTINCT 关键字合用。

Sybase IQ 不支持在 LONG BINARY (BLOB) 列或变量上使用 LIKE 谓词。如果尝试使用 LIKE 谓词在 LONG BINARY 列中搜索模式，则会返回错误 Invalid data type comparison in predicate (谓词中的数据类型比较无效)。

请参见第 9 章“函数支持”。

查询 LONG VARCHAR 列

在 SELECT 语句的 WHERE 子句中，除 BIT_LENGTH、CHAR_LENGTH、CHAR_LENGTH64、CHARINDEX、LOCATE、OCTET_LENGTH、PATINDEX、SUBSTRING64 和 SUBSTRING 函数外，LONG VARCHAR 列只能用于 IS NULL 和 IS NOT NULL 表达式中。

可以使用 LIKE 谓词在 LONG VARCHAR 列上搜索模式。支持所有长度小于等于 126 个字符的模式。不支持长度大于 254 个字符的模式。根据模式的内容，支持长度在 127 和 254 个字符之间的某些模式。

LIKE 谓词支持任意数据大小的 LONG VARCHAR (CLOB) 变量。当前，一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。

LONG VARCHAR 列不能用于 SELECT 语句的 ORDER BY、GROUP BY 和 HAVING 子句，也不能与 DISTINCT 关键字 (SELECT DISTINCT 和 COUNT DISTINCT) 合用。

请参见第 9 章“函数支持”。

CONTAINS 谓词支持

可以在 LONG VARCHAR (CLOB) 列上创建 WORD (WD) 索引，并使用 CONTAINS 谓词在该列中搜索最大长度为 255 个字符的字符串常量。

在使用 WD 索引的 LONG BINARY (BLOB) 列上不支持 CONTAINS 谓词。如果尝试使用 CONTAINS 谓词在含有 WD 索引的 LONG BINARY 列中搜索字符串，则会返回错误。使用外部库的 TEXT 索引支持在二进制数据上使用 CONTAINS。

请参见《参考：构件块、表和过程》的第2章“SQL 语言元素”中的“CONTAINS 条件”。

监控 LONG BINARY 和 LONG VARCHAR 列的性能

Sybase IQ 性能监视器显示 LONG BINARY 和 LONG VARCHAR 列的性能数据。

本章介绍对数据类型为 LONG BINARY (BLOB) 和 LONG VARCHAR (CLOB) 的列的存储过程支持，以及全文本搜索。

主题	页码
管理 TEXT 索引中的词语	39
标识外部库	42
控制大对象数据压缩	44
显示有关大对象列的信息	45

管理 TEXT 索引中的词语

通过使用以下存储过程，可以将字符串拆分成词语，查找 TEXT 索引中有多少词语以及各个词语的位置，显示有关 TEXT 索引的统计信息。

sa_char_terms 系统过程

作用	将 CHAR 字符串拆分为多个术语，并在一行中返回每个术语及其位置。
语法	<pre>sa_char_terms('char-string' [, 'text-config-name' [, 'owner']])</pre>
参数	<p>char-string 您要分析的 CHAR 字符串。</p> <p>text-config-name 在处理字符串时将应用的文本配置对象。缺省值为 “default_char”。</p> <p>owner 指定的文本配置对象的所有者。缺省值为 DBA。</p>
说明	可以使用此系统过程来查明在应用文本配置对象的设置时如何解释字符串。这可以帮助您了解哪些词语将在编制索引期间被删除或者将从查询字符串中被删除。
权限	无。

示例 下列语句返回 CHAR 字符串 “the quick brown fox jumped over the fence” 中的词语：

```
CALL sa_char_terms  
  ( 'the quick brown fox jumped over the fence' );
```

表 5-1: CHAR 字符串解释

词语	位置
the	1
quick	2
brown	3
fox	4
jumped	5
over	6
the	7
fence	8

sa_nchar_terms 系统过程

作用 将 NCHAR 字符串拆分为多个术语，并在一行中返回每个术语及其位置。

语法

```
sa_nchar_terms( 'char-string' [, 'text-config-name' [, 'owner' ] ] )
```

参数 **char-string** 您要分析的 NCHAR 字符串。
text-config-name 在处理字符串时将应用的文本配置对象。缺省值为 “default_nchar”。
owner 指定的文本配置对象的所有者。缺省值为 DBA。

说明 可以使用 sa_nchar_terms 来查明在应用文本配置对象的设置时如何解释字符串。这可以帮助您了解哪些词语将在编制索引期间被删除或者将从查询字符串中被删除。

注释 只有 IN SYSTEM 表支持 NCHAR 数据类型。

权限 无

示例 sa_nchar_terms 的语法与 sa_char_terms 系统过程的语法类似。

sa_text_index_stats 系统过程

作用 返回有关数据库中的 TEXT 索引的统计信息。

语法 `sa_text_index_stats()`

说明 使用 `sa_text_index_stats` 可以查看数据库中的每个 TEXT 索引的统计信息。下表说明 `sa_text_index_stats` 返回的信息：

表 5-2: TEXT 索引的统计信息

列名	类型	说明
owner_id	UNSIGNED INT	表的所有者的 ID
table_id	UNSIGNED INT	表的 ID
index_id	UNSIGNED INT	TEXT 索引的 ID
text_config_id	UNSIGNED BIGINT	由 TEXT 索引引用的文本配置的 ID
owner_name	CHAR(128)	所有者的名称
table_name	CHAR(128)	表的名称
index_name	CHAR(128)	TEXT 索引的名称
text_config_name	CHAR(128)	文本配置对象的名称
doc_count	UNSIGNED BIGINT	TEXT 索引中带索引的列值的总数
doc_length	UNSIGNED BIGINT	TEXT 索引中数据的总长度
pending_length	UNSIGNED BIGINT	待执行更改的总长度
deleted_length	UNSIGNED BIGINT	待执行删除的总长度
last_refresh	TIMESTAMP	最后一次刷新的日期和时间

`pending_length`、`deleted_length` 和 `last_refresh` 值对于 IMMEDIATE REFRESH TEXT 索引为 NULL。

权限 需要具有 DBA 权限

示例 下列语句返回数据库中每个 TEXT 索引的统计信息：

```
CALL sa_text_index_stats( );
```

sa_text_index_vocab 系统过程

作用 列出 TEXT 索引中出现的所有词语，以及显示每个词语的带索引值的总数。

语法 `sa_text_index_vocab(`
`'text-index-name',`
`'table-name',`

```
'table-owner'
)
```

参数

text-index-name 使用此 CHAR(128) 参数可以指定 TEXT 索引的名称。

table-name 使用此 CHAR(128) 参数可以指定 TEXT 索引所基于的表的名称。

table-owner 使用此 CHAR(128) 参数可以指定表的所有者。

说明

`sa_text_index_vocab` 返回 TEXT 索引中出现的所有词语，以及显示每个词语的带索引值的总数（如果词语在某些带索引值中出现多次，则该值小于总出现次数）。

`sa_text_index_vocab` 有以下限制：

- 参数值不能是宿主变量或表达式。参数 *text-index-name*、*table-name* 和 *table-owner* 必须是约束或变量。

权限

需要具有 DBA 权限或对带索引表具有 SELECT 权限。

示例

以下示例执行 `sa_text_index_vocab` 以返回所有出现在 GROUPO 拥有的表 Customers 上的 TEXT 索引 MyTextIndex：

```
sa_text_index_vocab
('MyTextIndex', 'Customers', 'GROUPO');
```

表 5-3: 索引中的词语

词语	次数
a	1
Able	1
Acres	1
Active	5
Advertising	1
Again	1
...	...

标识外部库

`sa_list_external_library` 存储过程列出服务器中当前装载的库。在标识库后，可以使用 `sa_external_library_unload` 将库从服务器中卸载。

sa_external_library_unload 系统过程

作用	卸载外部库。
语法	sa_external_library_unload (['external-library'])
参数	external-library （可选）使用此 LONG VARCHAR 参数可以指定要卸载的库的名称。如果不指定任何库，则所有不在使用中的外部库都会被卸载。
说明	如果指定了外部库，但它在使用中或还未装载，则会返回错误。如果未指定任何参数，则在找不到任何已装载的外部库时，会返回错误。
权限	需要具有 DBA 权限。
示例	以下示例卸载名为 <i>myextlib.dll</i> 的外部库： <pre>CALL sa_external_library_unload('myextlib.dll');</pre> 以下示例卸载所有当前不在使用中的库： <pre>CALL sa_external_library_unload();</pre>

sa_list_external_library 过程

作用	列出服务器中当前装载的外部库。
语法	sa_list_external_library ()
说明	返回引擎中装载的外部库的列表以及各个库的引用计数。引用计数是引擎中库的实例数。可以通过执行程序 <code>sa_external_library_unload</code> 来卸载外部库（只要其引用计数是 0）。
权限	需要具有 DBA 权限。
示例	以下示例列出外部库及其引用计数： <pre>call sa_list_external_library()</pre>

控制大对象数据压缩

在将数据库缓冲区写入磁盘时，`sp_iqsetcompression` 存储过程可以控制数据类型为 `LONG BINARY` 和 `LONG VARCHAR` 的列的压缩。另外，还可以使用 `sp_iqsetcompression` 来禁用压缩。此功能可节省 CPU 周期，因为 `LONG BINARY` 或 `LONG VARCHAR` 列中存储的某些数据格式（例如，JPG 文件）已经压缩，即使再行压缩也不会取得任何效果。`sp_iqshowcompression` 存储过程显示大对象列的压缩设置。

sp_iqsetcompression 过程

作用 设置数据类型为 `LONG BINARY (BLOB)` 和 `LONG VARCHAR (CLOB)` 的列中的数据压缩。

语法 `sp_iqsetcompression (owner, table, column, on_off_flag)`

权限 需要 DBA 权限。

说明 `sp_iqsetcompression` 用于控制数据类型为 `LONG BINARY (BLOB)` 和 `LONG VARCHAR (CLOB)` 的列的压缩。压缩设置仅应用于 Sybase IQ 基表。

`sp_iqsetcompression` 的副作用是更改压缩设置之后将会发生 `COMMIT`。

表 5-4: sp_iqsetcompression 的参数

姓名	说明
<code>owner</code>	要设置压缩的表的所有者
<code>table</code>	要设置压缩的表
<code>column</code>	要设置压缩的列
<code>on_off_flag</code>	压缩设置：ON 表示启用压缩，OFF 表示禁用压缩

示例 例如，假定如下表定义：

```
CREATE TABLE USR.picTable (picID INT NOT NULL,
picJPG LONG BINARY NOT NULL);
```

要禁用对 LOB 列 `picJPG` 的压缩，请调用 `sp_iqsetcompression`（您必须拥有 DBA 权限）：

```
CALL sp_iqsetcompression('USR', 'picTable', 'picJPG',
'OFF');
```

此命令不返回任何行。

sp_iqshowcompression 过程

作用 显示数据类型为 LONG BINARY (BLOB) 和 LONG VARCHAR (CLOB) 的列的压缩设置。

语法 `sp_iqshowcompression (owner, table, column)`

权限 需要 DBA 权限。

说明 返回列名和压缩设置。压缩设置值为 “ON”（启用压缩）和 “OFF”（禁用压缩）。

表 5-5: sp_iqshowcompression 的参数

姓名	说明
<i>owner</i>	要设置压缩的表的所有者
<i>table</i>	要设置压缩的表
<i>column</i>	要设置压缩的列

示例 例如，假定如下表定义：

```
CREATE TABLE USR.pixTable (picID INT NOT NULL,
picJPG LONG BINARY NOT NULL);
```

要检查 pixTable 表中各列的压缩状态，请调用 sp_iqshowcompression（您必须拥有 DBA 权限）：

```
CALL sp_iqshowcompression('USR', 'pixTable',
'picJPG');
```

此命令返回一行：

```
'picJPG', 'ON'
```

显示有关大对象列的信息

存储过程 sp_iqindexsize 显示单个 LONG BINARY 和 LONG VARCHAR 列的大小。

LONG BINARY 列的大小

以下输出显示了一个具有约 42GB 数据的 LONG BINARY 列。页大小为 128KB。Info 类型 largelob 在最后一行中：

Username	Indexname	Type	Info	KBytes	Pages	Compressed Pages
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	Total	42953952	623009	622923
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	vdo	0	0	0
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	bt	0	0	0
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	garray	0	0	0
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	bm	136	2	1

DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	barray	2312	41	40
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	dpstore	170872	2551	2549
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	largelob	42780632	620415	620333

在此示例中，压缩比为 $42953952/(623009*128) = 53.9\%$ 。

LONG VARCHAR 列
的大小

以下输出显示了一个具有约 42GB 数据的 LONG VARCHAR 列。页大小为 128KB。Info 类型 largelob 在最后一行中：

Username	Indexname	Type	Info	KBytes	Pages	Compressed Pages
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	Total	42953952	623009	622923
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	vdo	0	0	0
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	bt	0	0	0
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	garray	0	0	0
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	bm	136	2	1
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	barray	2312	41	40
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	dpstore	170872	2551	2549
DBA	test10.DBA.ASIQ_IDX_T128_C3_FP	FP	largelob	42780632	620415	620333

在此示例中，压缩比为 $42953952/(623009*128) = 53.9\%$ 。

本章介绍如何在 Sybase IQ 中导出和装载大对象数据。

主题	页码
导出大对象数据	47
装载大对象数据	48

导出大对象数据

IQ 数据提取功能包含 BFILE 函数，可用于将各个 LONG BINARY 和 LONG VARCHAR 单元提取到服务器上的各个操作系统文件中。BFILE 函数不一定要与数据提取功能一起使用。

BFILE 函数

作用

将各个 LONG BINARY 和 LONG VARCHAR 单元提取到服务器上的各个操作系统文件中。

语法

BFILE(*file-name-expression*, *large-object-column*)

参数

file-name-expression LONG BINARY 或 LONG VARCHAR 数据所写入的输出文件的名称。此文件名称最大长度可以为 (32K -1) 字节，但必须是文件系统支持的有效路径名。

large-object-column LONG BINARY 或 LONG VARCHAR 列的名称。

用法

BFILE 返回:

- 如果文件成功写入，则为 1
- 如果文件未成功打开或写入，则为 0
- 如果 LONG BINARY 或 LONG VARCHAR 单元值为 NULL，则为 NULL

如果 LONG BINARY 或 LONG VARCHAR 单元值为 NULL，则不会打开任何文件，也不会写入任何数据。

文件路径相对于服务器的启动位置，且打开和写入操作使用服务器进程的权限执行。BFILE 输出文件不支持磁带设备。

使用 BFILE 之外的其它函数所检索的 LONG BINARY 和 LONG VARCHAR 单元（即，之后通过客户端 / 服务器数据库连接检索到的单元）受到最大长度为 2GB 的大小限制。使用 SUBSTRING64 或 BYTE_SUBSTR64 函数可以通过 SELECT (SELECT, OPEN CURSOR) 检索大于 2GB 的 LONG BINARY 单元。使用 SUBSTRING64 函数可以通过 SELECT (SELECT, OPEN CURSOR) 检索大于 2GB 的 LONG VARCHAR 单元。某些连接驱动程序（例如 ODBC、JDBC™ 和 Open Client™）不允许在一个 SELECT 中返回超过 2GB 的数据。

BFILE 函数不一定要与数据提取功能一起使用。

示例

本示例显示如何使用 BFILE 从 LONG BINARY 列 lobcol 中提取数据，该列是在第 49 页的“装载示例”中创建和装载的。要将数据写入可以在装载中用作辅助文件的文件，请输入：

```
SELECT c1, filename, ext,
       '../myoutput/' + TRIM(filename) + '.' + TRIM(ext) fname,
       BFILE(fname, lobcol)
FROM ltab
     WHERE lobcol IS NOT NULL
          AND ext IS NOT NULL
```

此命令为第 1 行中的 lobcol 生成带有扩展名 *boston.jpg* 的文件名，并为第 2 行中的 lobcol 生成带有扩展名 *map_of_concord.bmp* 的文件名。

装载大对象数据

使用 LOAD TABLE 语句的扩展语法来装载 LONG BINARY 和 LONG VARCHAR 数据。可以从 ASCII 或 BCP 格式的主文件装载的大对象数据没有大小限制，除非操作系统具有相关限制。从主文件加载到大对象列的固定宽度数据的最大长度为 32K - 1。

还可以在主装载文件中指定辅助装载文件。每个单独辅助数据文件正好包含一个 LONG BINARY 或 LONG VARCHAR 单元值。

扩展 LOAD TABLE 语法

```
LOAD [ INTO ] TABLE [ owner ].table-name
... ( column-name load-column-specification [, ...] )
... FROM 'filename-string' [, ...]
... [ QUOTES { ON | OFF } ]
... ESCAPES OFF
... [ FORMAT { ascii | binary | bcp } ]
```

```

... [ DELIMITED BY 'string' ]
...
load-column-specification:
...
| { BINARY | ASCII } FILE( integer )
| { BINARY | ASCII } FILE ( 'string' )

```

关键字 BINARY FILE（对于 LONG BINARY）或 ASCII FILE（对于 LONG VARCHAR）向装载指定，列的主输入文件包含辅助文件（其中包含 LONG BINARY 或 LONG VARCHAR 单元值）的路径，而不是 LONG BINARY 或 LONG VARCHAR 数据本身。辅助文件路径名可以是完全限定的也可以是相对的。如果辅助文件路径名不是完全限定的，则该路径相对于在其中启动服务器的目录。辅助文件不支持磁带设备。

Sybase IQ 支持在主装载文件中装载长度无限制的 LONG BINARY 和 LONG VARCHAR 值（遵从操作系统限制）。在将十六进制格式的二进制数据从主文件中装载到 LONG BINARY 列中时，Sybase IQ 要求十六进制数字的总数为偶数。如果单元值中包含奇数个十六进制数字，就会报错“Odd length of binary data value detected on column”（在列中检测到长度为奇数的二进制数据值）。用于 LONG BINARY 装载的输入文件应始终包含奇数个十六进制数字。

Sybase IQ 不支持使用 LOAD TABLE...FORMAT BINARY 从主文件中装载大对象列。可以从辅助文件中装载二进制格式的大对象数据。有关使用二进制格式装载数据的详细信息，请参见《系统管理指南：第一卷》的[第7章“将数据移入和移出数据库”](#)中的[“使用二进制装载格式”](#)。

对于 LOAD TABLE FORMAT BCP，装载规范只能包含列名、NULL 和 ENCRYPTED。这说明当您使用 LOAD TABLE FORMAT BCP 选项装载 LONG BINARY 和 LONG VARCHAR 列时，您不能使用辅助文件。请参见《参考：语句和选项》的[第1章“SQL 语句”](#)中的[LOAD TABLE 语句](#)。

装载示例

本示例显示使用 LONG BINARY 数据创建和装载表的 SQL 语句。

```

CREATE TABLE ltab (c1 INT, filename CHAR(64),
                  ext CHAR(6), lobcol LONG BINARY NULL);
LOAD TABLE ltab (
    c1,
    filename,
    ext NULL('NULL'),
    lobcol BINARY FILE ('') NULL('NULL')
)
FROM 'abc.inp'
QUOTES OFF ESCAPES OFF;

```

主文件 *abc.inp* 包含以下数据：

```
1,boston,jpg,/s1/loads/lobs/boston.jpg,
```

```
2,map_of_concord,bmp,/s1/loads/maps/concord.bmp,  
3,zero length test,NULL,,  
4,null test,NULL,NULL,
```

将 LONG BINARY 数据装载到表 tab 中后，列 lobcol 的第一行和第二行分别包含文件 *boston.jpg* 和 *concord.bmp* 的内容。第三行和第四行分别包含零长度值和 NULL。

控制装载错误

使用数据库选项 SECONDARY_FILE_ERROR，可以指定在打开或读取辅助 BINARY FILE 或 ASCII FILE 出现错误时执行的装载操作。

如果 SECONDARY_FILE_ERROR 为 ON，则在打开或读取辅助 BINARY FILE 或 ASCII FILE 出现错误时，装载将回退。

如果 SECONDARY_FILE_ERROR 为 OFF（缺省值），则无论在打开或读取辅助 BINARY FILE 或 ASCII FILE 时出现任何错误，装载均将继续。

LONG BINARY 或 LONG VARCHAR 单元保留为下列值之一：

- 如果列允许空值，则为 NULL
- 如果列不允许空值，则为零长度值

任何用户都可针对 PUBLIC 组或临时组设置 SECONDARY_FILE_ERROR；它会立即生效。

在将完整性约束违规记录到装载错误 ROW LOG 文件时，为 LONG BINARY 或 LONG VARCHAR 列记录的信息为：

- 如果日志记录发生在第一道装载操作内，则为从主数据文件读取的实际文本
- 如果日志记录发生在第二道装载操作内，则为零长度值

去除尾随空白

LOAD TABLE...STRIP 选项对于 LONG VARCHAR 数据没有影响。不从 LONG VARCHAR 数据去除尾随空白，即使 STRIP 选项为 ON，也是如此。

外围引号

LOAD TABLE...QUOTES 选项不适用于从辅助文件装载 LONG BINARY (BLOB) 或 LONG VARCHAR (CLOB) 数据，而无论其设置如何。前导或尾随引号将作为 CLOB 数据的一部分来装载。使用 QUOTES ON 选项，位于引号之间的两个连续引号将作为两个连续引号进行装载。

截断部分多字节字符数据

在装载期间，将根据 TRIM_PARTIAL_MBC 数据库选项的值截断部分多字节 LONG VARCHAR 数据：

- 如果 TRIM_PARTIAL_MBC 为 ON，对于主数据和带 ASCII FILE 选项的 LOAD，均会截断部分多字节字符。
- 如果 TRIM_PARTIAL_MBC 为 OFF，则带 ASCII FILE 选项的 LOAD 将根据 SECONDARY_FILE_ERROR 数据库选项的值处理部分多字节字符。

表 6-1 列出如何根据 TRIM_PARTIAL_MBC 和 SECONDARY_FILE_ERROR 的值装载尾随的多字节字符。

表 6-1: 使用 ASCII FILE 选项装载 LONG VARCHAR 时的部分多字节字符

TRIM_PARTIAL_MBC	SECONDARY_FILE_ERROR	找到尾随部分多字节字符
ON	ON/OFF	截断尾随部分多字节字符
OFF	ON	单元 — 如果允许为 null, 则为 null LOAD 错误 — 如果不允许为 null, 则回退
OFF	OFF	单元 — 如果允许为 null, 则为 null 单元 — 如果不允许为 null, 则为零长度

另请参见

有关 LOAD TABLE、INSERT...VALUES、INSERT...SELECT、INSERT...LOCATION、SELECT...INTO 和 UPDATE SQL 语句对大对象变量的支持的信息, 请参见第 7 章 “大对象数据类型”。

大对象数据类型

本章介绍大对象 LONG BINARY 和 LONG VARCHAR 数据类型列的特性以及大对象数据的索引支持。

主题	页码
大对象数据类型 LONG BINARY 和 BLOB	53
大对象数据类型 LONG VARCHAR 和 CLOB	54
大对象变量	56
大对象列的索引支持	57

大对象数据类型 LONG BINARY 和 BLOB

Sybase IQ 中的二进制大对象 (BLOB) 数据存储在全数据类型为 LONG BINARY 或 BLOB 的列中。

对于大小为 128KB 的 IQ 页，单个 LONG BINARY 数据值的长度范围可以为零 (0) 到 512TB (千吉字节)；对于大小为 512KB 的 IQ 页，可以为零 (0) 到 2PB (千万亿字节)。(最大长度为 4 GB 乘以数据库页大小。)要容纳具有 LONG BINARY 数据的表，必须使用大小至少为 128KB (131072 字节) 的 IQ 页创建 IQ 数据库。

表或数据库可以包含的 LONG BINARY 列数分别为每个表支持的最大列数和每个数据库支持的最大列数。

LONG BINARY 列可以是 NULL 也可以是 NOT NULL，且可以存储零长度的值。BLOB 域的数据类型为允许 NULL 的 LONG BINARY。

不可以对 LONG BINARY 列构造非 FP 索引或连接索引。

如果结果集包含 BLOB 列，则会禁用预取。

使用 UPDATE、INSERT、LOAD TABLE、DELETE、TRUNCATE、SELECT...INTO 和 INSERT...LOCATION SQL 语句修改 LONG BINARY 列。LONG BINARY 列不支持定位型更新和删除。

可以使用 INSERT...LOCATION 命令将 Adaptive Server Enterprise 的 IMAGE 列插入到 LONG BINARY 列中。插入的所有 IMAGE 数据均以无提示方式正好在 2147483648 字节 (2GB) 处右截断。

数据类型转换

除用于 INSERT 和 UPDATE 的 BINARY 和 VARBINARY 数据类型外，没有从 LONG BINARY 数据类型到其它非 LONG BINARY 数据类型的任何隐式数据类型转换。存在从 TINYINT、SMALLINT、INTEGER、UNSIGNED INTEGER、BIGINT、UNSIGNED BIGINT、CHAR 和 VARCHAR 数据类型到 LONG BINARY 数据类型的隐式转换。没有从 BIT、REAL、DOUBLE 或 NUMERIC 数据类型到 LONG BINARY 数据类型的任何隐式转换。可以使用 CONVERSION_MODE 数据库选项控制隐式转换。

LONG BINARY 数据类型当前支持的字节子字符串函数被接受为 INSERT 和 UPDATE 语句的隐式转换的输入。请参见第 9 章“函数支持”。

LONG BINARY 数据类型可以显式转换为 BINARY 或 VARBINARY。不存在以 LONG BINARY 数据类型为源类型或目标类型的任何其它显式数据类型转换（例如，使用 CAST 或 CONVERT 函数）。

在 LONG BINARY 转换到 BINARY 或 VARBINARY 期间，截断 LONG BINARY 数据的处理方式与截断 BINARY 和 VARBINARY 数据的处理方式相同。如果 STRING_RTRUNCATION 选项为 ON，则对二进制列执行 INSERT 或 UPDATE 操作时，对任何值（不仅是非空格字符）的任何右截断均会导致截断错误，而且事务将被回退。

大对象数据类型 LONG VARCHAR 和 CLOB

Sybase IQ 中的字符大对象 (CLOB) 数据存储在大对象数据类型为 LONG VARCHAR 或 CLOB 的列中。

对于大小为 128KB 的 IQ 页，单个 LONG VARCHAR 数据值的长度范围可以为零 (0) 到 512TB（千吉字节）；对于大小为 512KB 的 IQ 页，可以为零 (0) 到 2PB（千万亿字节）。（最大长度为 4 GB 乘以数据库页大小。）要容纳具有 LONG VARCHAR 数据的表，必须使用大小至少为 64KB（65536 字节）的 IQ 页创建 IQ 数据库。

表或数据库可以包含的 LONG VARCHAR 列数分别为每个表支持的最大列数和每个数据库支持的最大列数。

Sybase IQ 同时支持单字节和多字节 LONG VARCHAR 数据。

LONG VARCHAR 列可以是 NULL 也可以是 NOT NULL，且可以存储零长度的值。CLOB 域的数据类型为允许 NULL 的 LONG VARCHAR。若要创建非空 LONG VARCHAR 列，请在列定义中显式指定 NOT NULL。

在创建表或向现有表添加列时，您可以使用 CLOB 域创建 LONG VARCHAR 列。例如：

```
CREATE TABLE lvtab (c1 INTEGER, c2 CLOB,
                    c3 CLOB NOT NULL);
ALTER TABLE lvtab ADD c4 CLOB;
```

可以针对 LONG VARCHAR 列创建 WORD (WD) 索引。但不可以针对 LONG VARCHAR 列构造其它非 FP 索引类型和连接索引。

可以使用 UPDATE、INSERT...VALUES、INSERT...SELECT、LOAD TABLE、DELETE、TRUNCATE、SELECT...INTO 和 INSERT...LOCATION SQL 语句修改 LONG VARCHAR 列。LONG VARCHAR 列不支持定位型更新和删除。

可以通过 INSERT...LOCATION 命令将 Adaptive Server Enterprise 的 TEXT 列插入到 LONG VARCHAR 列中。插入的所有 TEXT 数据以无提示方式在 2147483648 个字节 (2GB) 处右截断。

数据类型转换

仅除用于 INSERT 和 UPDATE 的 LONG BINARY、CHAR 和 VARCHAR 外，没有从 LONG VARCHAR 数据类型到其它非 LONG VARCHAR 数据类型的任何隐式数据类型转换。存在从 CHAR 和 VARCHAR 数据类型到 LONG VARCHAR 数据类型的隐式转换。没有从 BIT、REAL、DOUBLE、NUMERIC、TINYINT、SMALLINT、INT、UNSIGNED INT、BIGINT、UNSIGNED BIGINT、BINARY、VARBINARY 或 LONG BINARY 数据类型到 LONG VARCHAR 数据类型的任何隐式转换。可以使用 CONVERSION_MODE 数据库选项控制隐式转换。

LONG VARCHAR 数据类型当前支持的字符串函数被接受为 INSERT 和 UPDATE 语句的隐式转换的输入。请参见第 9 章“函数支持”。

LONG VARCHAR 数据类型可以显式转换为 CHAR 和 VARCHAR。不存在以 LONG VARCHAR 数据类型为源类型或目标类型的任何其它显式数据类型转换（例如，使用 CAST 或 CONVERT 函数）。

在 LONG VARCHAR 转换到 CHAR 期间截断 LONG VARCHAR 数据的处理方式与截断 CHAR 数据的处理方式相同。如果 STRING_RTRUNCATION 选项为 ON 且对非空格进行字符串右截断，则会报告截断错误并回退事务。转换时尾随部分多字节字符会被替换为空格。

在 LONG VARCHAR 转换到 VARCHAR 期间截断 LONG VARCHAR 数据的处理方式与截断 VARCHAR 数据的处理方式相同。如果 STRING_RTRUNCATION 选项为 ON 且对非空格进行字符串右截断，则会报告截断错误并回退事务。转换时尾随部分多字节字符会被截断。

大对象变量

进站 LONG BINARY 和 LONG VARCHAR 变量（由 IQ 使用的主机变量或 SQL 变量）没有最大长度。

出站 LONG BINARY 和 LONG VARCHAR 变量（由 IQ 设置的变量）有最大长度，为 2GB - 1。

LOAD TABLE、INSERT...VALUES、INSERT...SELECT、INSERT...LOCATION、SELECT...INTO 和 UPDATE SQL 语句接受任意数据大小的 LONG BINARY 和 LONG VARCHAR 变量。当前，SQL 变量可以容纳最大长度为 2GB - 1 的数据。

BIT_LENGTH、BYTE_LENGTH、BYTE_LENGTH64、BYTE_SUBSTR、BYTE_SUBSTR64、CHARINDEX、LOCATE、OCTET_LENGTH 和 SUBSTRING64 函数支持 SQL 变量可以容纳的任意数据大小的 LONG BINARY 和 LONG VARCHAR 变量。此外，CHAR_LENGTH、CHAR_LENGTH64、PATINDEX、SUBSTR 和 SUBSTRING 函数支持 SQL 变量可以容纳的任意数据大小的 LONG VARCHAR 变量。

大对象变量数据类型转换

数据库选项 ENABLE_LOB_VARIABLES 控制大对象变量的数据类型转换。

ENABLE_LOB_VARIABLES 选项

函数 控制大对象变量的数据类型转换。

允许值 ON、OFF

缺省值 关闭

作用域 设置此选项无需 DBA 权限。可针对单个连接或 PUBLIC 组进行临时设置。设置将立即生效。

说明 此选项控制大对象变量的数据类型转换。当 ENABLE_LOB_VARIABLES 为 OFF 时，会隐式转换小于 32K 的大对象变量；如果大对象变量大于或等于 32K 时，会报告错误。LONG VARCHAR 变量会隐式转换为 VARCHAR 数据类型并在 32K 处截断。LONG BINARY 变量会隐式转换为 VARBINARY 数据类型并在 32K 处截断。

当 ENABLE_LOB_VARIABLES 为 ON 时，任何大小的大对象变量都会保留其原始数据类型和大小。

示例 要将大对象变量的数据类型和大小保留为大于 32K，请输入以下命名：

```
SET TEMPORARY OPTION ENABLE_LOB_VARIABLES = ON
```

大对象列的索引支持

Sybase IQ 在 LONG BINARY and LONG VARCHAR 列上支持 TEXT 索引，在 LONG VARCHAR 上支持 WORD (WD) 索引。

大对象列的 TEXT 索引支持

Sybase IQ TEXT 索引支持 LONG BINARY 和 LONG VARCHAR 列。

请参见第 8 章“SQL 语句支持”和第 2 章“TEXT 索引和文本配置对象”。

LONG VARCHAR (CLOB) 列的 WD 索引支持

Sybase IQ 在 LONG VARCHAR (CLOB) 列上为 WORD (WD) 索引提供以下支持：

- 可以在 Sybase Central 中对数据类型为 CHAR、VARCHAR 和 LONG VARCHAR 的列创建 WD 索引。
- WD 索引支持的最宽列是 LOB 列的最大宽度。（最大长度为 4 GB 乘以数据库页大小。）

Sybase IQ 支持的最大字词宽度为 255 字节。

- LONG VARCHAR (CLOB) 列还支持基于 CHAR 和 VARCHAR 列的 WD 索引的所有 sp_iqcheckdb 选项，包括分配、检查和验证模式。
- 可以使用 sp_iqrebuildindex 在 LONG VARCHAR (CLOB) 列上重建 WD 索引。

采用二进制格式的中文文本或文档要求 ETL 预处理，以便定位字词并将其转换为 WD 索引可分析的格式。

本章介绍支持对 TEXT 索引和文本配置进行处理的 SQL 语句和语法。

主题	页码
ALTER TEXT CONFIGURATION 语句	59
ALTER TEXT INDEX 语句	61
CREATE TEXT CONFIGURATION 语句	62
CREATE TEXT INDEX 语句	63
DROP TEXT CONFIGURATION 语句	64
DROP TEXT INDEX 语句	65

ALTER TEXT CONFIGURATION 语句

说明

改变文本配置对象。

语法

```
ALTER TEXT CONFIGURATION [ owner.]config-name
    STOPLIST stoplist
    | DROP STOPLIST
    | { MINIMUM | MAXIMUM } TERM LENGTH integer
    | TERM BREAKER
    | { GENERIC
      | [ EXTERNAL NAME library-and-entry-point-name-string ]
      | NGRAM }
    | PREFILTER EXTERNAL NAME library-and-entry-point-name-string
stoplist : string-expression
library-and-entry-point-name-string : [operating-system:]function-
name@library
```

示例

示例 1 以下语句将创建文本配置对象 maxTerm16，然后将最大词语长度更改为 16:

```
CREATE TEXT CONFIGURATION maxTerm16 FROM
default_char;
ALTER TEXT CONFIGURATION maxTerm16 MAXIMUM TERM
LENGTH 16;
```

示例 2 以下语句将在 `maxTerm16` 配置对象中添加停止列表词语：

```
ALTER TEXT CONFIGURATION maxTerm16
STOPLIST 'because about therefore only';
```

示例 3 此示例将更新文本配置对象 `my_text_config`，以便在外部库 `mytermbreaker.dll` 中使用入口点 `my_term_breaker` 来拆分文本。

```
CREATE TEXT CONFIGURATION my_text_config FROM
default_char;
ALTER TEXT CONFIGURATION my_text_config
TERM BREAKER GENERIC EXTERNAL NAME
'platform:my_term_breaker@mytermbreaker';
```

示例 4 此示例将更新文本配置对象 `my_text_config`，以便在外部库 `myprefilter.dll` 中使用入口点 `my_prefilter` 来预过滤文档。

```
ALTER TEXT CONFIGURATION my_text_config
PREFILTER EXTERNAL NAME
'platform:my_prefilter@myprefilter';
```

用法

TEXT 索引取决于文本配置对象。Sybase IQ TEXT 索引使用立即刷新，而且无法截断；必须先删除索引，然后才能改变文本配置对象。

要查看文本配置对象的设置，请查询 `SYSTEXTCONFIG` 系统视图

STOPLIST 子句 使用此子句可以创建或替换在生成 TEXT 索引时将忽略的词语的列表。在此列表中指定的词语也会在查询中被忽略。用空格分隔停止列表词语。

停止列表词语不能包含空格。停止列表不应包含非字母数字字符。非字母数字字符被解释为空格，并且将词语拆分成多个词语。例如，“and/or”被解释为两个词语“and”和“or”。停止列表词语的最大数目为 7999。

DROP STOPLIST 子句 使用此子句可以删除某个文本配置对象的停止列表。

MINIMUM TERM LENGTH 子句 以字符数为单位指定要在 TEXT 索引中包含的词语的最小长度。在使用 NGRAM TEXT 索引时，在 MINIMUM TERM LENGTH 子句中指定的值将被忽略。

在生成或刷新 TEXT 索引时，比此设置短的词语会被忽略。此选项的值必须大于 0。如果将此选项设置为高于 MAXIMUM TERM LENGTH，MAXIMUM TERM LENGTH 的值就会自动调整为和新的 MINIMUM TERM LENGTH 值相同。

MAXIMUM TERM LENGTH 子句 对于 GENERIC TEXT 索引，要在 TEXT 索引中包含的词语的最大长度（以字符数为单位）。在生成或刷新 TEXT 索引时，比此设置长的词语会被忽略。

MAXIMUM TERM LENGTH 的值必须小于等于 60。如果将此选项设置为低于 MINIMUM TERM LENGTH，MINIMUM TERM LENGTH 的值就会自动调整为和新的 MAXIMUM TERM LENGTH 值相同。

TERM BREAKER 子句 用于将列值切分为词语的算法的名称。IN SYSTEM 表的选项为 GENERIC（缺省值）或 NGRAM。Sybase IQ TEXT 索引仅支持 GENERIC 分词符。GENERIC 算法将由一个或多个字母数字字符组成且由非字母数字字符分隔的字符串视为一个词语。

NGRAM 算法将字符串拆分成 n-gram。一个 n-gram 是一个较大字符串中的由 n 个字符组成的子字符串。对于模糊（近似）匹配或者对于不使用空格或非字母数字字符来分隔词语的文档，NGRAM 分词符是必需的。IN SYSTEM 表支持 NGRAM。

TERM BREAKER 可以包括使用 EXTERNAL NAME 和库入口点来指定外部分词符库。

DROP PREFILTER 子句 删除外部预过滤器并将 ISYSEXTCONFIG 表中的预过滤器列设置为 NULL。

PREFILTER EXTERNAL NAME 子句 指定 entry_point 以及由外部供应商提供的外部预过滤器库的库名。

副作用

自动提交。

权限

用户必须具有 DBA 权限，才能改变文本配置对象来指定外部预过滤器或分词符的外部库和函数。

对文本配置的所有其它修改都可以由配置对象的所有者或具有 DBA 权限的用户完成。

ALTER TEXT INDEX 语句

说明 改变 TEXT 索引的定义。

语法 **ALTER TEXT INDEX** [*owner*.]**text-index-name** **ON** [*owner*.]**table-name**
alter-clause

alter-clause :
rename-object
| *move-object*

rename-object :
RENAME { **AS** | **TO** } *new-name*

move-object:

MOVE TO *dbspace-name*

示例

在此示例中，第一个语句创建 TEXT 索引 MyTextIndex，并将其定义为 IMMEDIATE REFRESH。第二个语句将 TEXT 索引重命名为 Text_index_daily。第三个语句将其移至一个名为 tispac 的数据库空间。

```
CREATE TEXT INDEX MyTextIndex ON Customers ( CompanyName
) IMMEDIATE REFRESH;
ALTER TEXT INDEX MyTextIndex ON Customers RENAME AS
Text_index_daily;
ALTER TEXT INDEX Text_Index_Daily ON Customers MOVE TO
tispac;
```

用法

RENAME 子句 重命名 TEXT 索引。

MOVE 子句 将 TEXT 索引移至指定的数据库空间。

副作用

自动提交。

权限

必须是基础表的所有者，或者具有 DBA 权限，或者对对象具有 REFERENCES 权限，才能重命名索引。

要移动 TEXT 索引，您必须具有 DBA 或 SPACE ADMIN 权限，或者必须是表所有者并对数据库空间具有 CREATE 权限。

CREATE TEXT CONFIGURATION 语句

说明

创建文本配置对象。

语法

```
CREATE TEXT CONFIGURATION [ owner.] new-config-name
FROM [ owner.] existing-config-name
```

示例

此 CREATE TEXT CONFIGURATION 语句使用 default_char 文本配置对象来创建文本配置对象 max_term_sixteen。后续的 ALTER TEXT CONFIGURATION 语句将 max_term_sixteen 的最大词语长度更改为 16:

```
CREATE TEXT CONFIGURATION max_term_sixteen FROM
default_char;
ALTER TEXT CONFIGURATION max_term_sixteen MAXIMUM TERM
LENGTH 16;
```


用法	<p>将一个文本配置对象用作模板来创建另一个文本配置对象，然后根据需要使使用 ALTER TEXT CONFIGURATION 语句改变选项。</p> <p>要查看数据库中所有文本配置对象及其设置的列表，请查询 SYSTEXTCONFIG 系统视图</p> <p><i>FROM</i> 子句 指定要用作新建文本配置对象的模板的文本配置对象的名称。缺省文本配置对象的名称是 default_char 和 default_nchar。Sybase IQ 表仅支持 default_char； default_nchar 只有在 SQL Anywhere 表上得到支持。</p> <p>副作用</p> <p>自动提交。</p>
权限	<p>必须具有 DBA 或 RESOURCE 权限。</p> <p>所有文本配置对象都具有 PUBLIC 访问权限。任何具有创建 TEXT 索引的权限的用户都可以使用任何文本配置对象。</p>

CREATE TEXT INDEX 语句

说明	创建 TEXT 索引。
语法	<pre>CREATE TEXT INDEX text-index-name ON [owner.]table-name(column-name, ...) [IN dbspace-name] [CONFIGURATION [owner.]text-configuration-name] [IMMEDIATE REFRESH]</pre>
示例	<p>此示例在 iqdemo 数据库中的 Customers 表的 CompanyName 列上创建 TEXT 索引 myTxtIdx。用到了 max_term_sixteen 文本配置对象。</p> <pre>CREATE TEXT INDEX myTxtIdx ON Customers (CompanyName); CONFIGURATION max_term_sixteen;</pre>
用法	<p>不能在视图或临时表上创建 TEXT 索引。不能在 IN SYSTEM 物化视图上创建 TEXT 索引。</p> <p>不会复制 TEXT 索引来连接索引表。可以在参与连接索引的表的列上创建 TEXT 索引。</p> <p>BEGIN PARALLEL IQÖEND PARALLEL IQ 语句不支持 CREATE TEXT INDEX。</p> <p><i>ON</i> 子句 指定 TEXT 索引所基于的表和列。</p>

IN 子句 指定 TEXT 索引所在的数据库空间。如果不指定此子句，就会在基础表所在的同一数据库空间中创建 TEXT 索引。

CONFIGURATION 子句 指定在创建 TEXT 索引时使用的文本配置对象。如果不指定此子句，就会使用 default_char 文本配置对象。

REFRESH 子句 IMMEDIATE REFRESH 用作缺省设置，而且是 Sybase IQ 中的表的唯一允许值。指定 IMMEDIATE REFRESH 以便每次基础表中的更改影响 TEXT 索引中的数据时都刷新 TEXT 索引。

IMMEDIATE REFRESH TEXT 索引在创建时被填充，只要基础表中的数据发生更改就会被刷新。一旦创建 TEXT 索引后，就无法将其改入、改出 IMMEDIATE REFRESH。

副作用

自动提交。

权限

您必须是基础表的所有者，或者具有 DBA 权限，或者具有 REFERENCES 权限。

您必须对该数据库空间具有 CREATE 权限。

DROP TEXT CONFIGURATION 语句

说明

删除文本配置对象。

语法

DROP TEXT CONFIGURATION [owner.]text-config-name

示例

以下语句将创建和删除 mytextconfig 文本配置对象：

```
CREATE TEXT CONFIGURATION mytextconfig FROM
default_char;
DROP TEXT CONFIGURATION mytextconfig;
```

用法

如果尝试删除具有相关 TEXT 索引的文本配置对象，则会导致错误。在删除文本配置对象之前，必须先删除相关的 TEXT 索引。

文本配置对象存储在 ISYSTEXTCONFIG 系统表中。

副作用

自动提交

权限

必须是文本配置对象的所有者，或者具有 DBA 权限。

DROP TEXT INDEX 语句

说明	从数据库中删除 TEXT 索引。
语法	DROP TEXT INDEX <i>text-index-name</i> ON [<i>owner</i>] <i>table-name</i>
示例	以下语句创建和删除 TextIdx TEXT 索引： <pre>CREATE TEXT INDEX TextIdx ON Customers (Street); DROP TEXT INDEX TextIdx ON Customers;</pre>
用法	<i>ON</i> 指定 TEXT 索引所基于的表。 在删除文本配置对象之前，必须先删除相关的 TEXT 索引。 副作用 自动提交
权限	必须是基础表的所有者，或者具有 DBA 权限，或者具有 REFERENCES 权限。

函数支持

本章介绍支持 LONG BINARY 和 LONG VARCHAR 数据类型的 Sybase IQ 函数。

主题	页码
大对象数据的函数支持摘要	67
BIT_LENGTH 函数	68
BYTE_LENGTH 函数	69
BYTE_LENGTH64 函数	69
BYTE_SUBSTR64 和 BYTE_SUBSTR 函数	69
CHAR_LENGTH 函数	70
CHAR_LENGTH64 函数	70
CHARINDEX 函数	71
LOCATE 函数	71
OCTET_LENGTH 函数	72
PATINDEX 函数	73
SUBSTRING 函数	73
SUBSTRING64 函数	74
大对象列的集合函数支持	75

大对象数据的函数支持摘要

表 9-1 总结了 LONG BINARY (BLOB) 和 LONG VARCHAR (CLOB) 数据类型以及 LONG BINARY 和 LONG VARCHAR 变量的函数支持。

除本章介绍的函数之外，还可以使用 BFILE 函数来提取 LOB 数据。请参见第 47 页的“导出大对象数据”。

表 9-1: LOB 数据类型和变量的函数支持

函数	是否支持 BLOB 数据?	是否支持 BLOB 变量?	是否支持 CLOB 数据?	是否支持 CLOB 变量?
BIT_LENGTH()	是	是	是	是
BYTE_LENGTH()	是 *	是 *	是 *	是 *
BYTE_LENGTH64()	是	是	是	是
BYTE_SUBSTR()	是	是	是	是
BYTE_SUBSTR64()	是	是	是	是
CHAR_LENGTH()	否	否	是	是
CHAR_LENGTH64()	否	否	是	是
CHARINDEX()	是	是	是	是
LOCATE()	是	是	是	是
OCTET_LENGTH()	是	是	是	是
PATINDEX()	否	否	是	是
SUBSTR() / SUBSTRING()	否	否	是	是
SUBSTRING64()	是	是	是	是

* 只有在查询返回小于 2GB 时，BYTE_LENGTH 函数才支持 LONG BINARY 列和变量以及 LONG VARCHAR 列和变量。如果返回的 LONG BINARY 或 LONG VARCHAR 数据的字节长度大于 2GB，BYTE_LENGTH 就会返回错误，以指明您必须使用 BYTE_LENGTH64 函数。

以下各节介绍支持 LONG BINARY 和 LONG VARCHAR 列和变量的函数。有关这些函数和示例的完整说明，请参见《参考：构件块、表和过程》中的第 4 章“SQL 函数”。

BIT_LENGTH 函数

作用 BIT_LENGTH 函数返回无符号的 64 位值，该值包含大对象列或变量参数的位长度。如果参数为 NULL，则 BIT_LENGTH 返回 NULL。

语法 `BIT_LENGTH(large-object-column)`

参数 `large-object-column` LONG VARCHAR 或 LONG BINARY 列或变量的名称。

用法 BIT_LENGTH 支持所有 Sybase IQ 数据类型以及任意数据大小的 LONG BINARY 和 LONG VARCHAR 变量。当前，一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。

BYTE_LENGTH 函数

- 作用** BYTE_LENGTH 函数返回字符串中的字节数。
- 用法** 只有在查询返回小于 2GB 时，BYTE_LENGTH 函数才支持 LONG BINARY 列和变量以及 LONG VARCHAR 列和变量。如果返回的 LONG BINARY 或 LONG VARCHAR 数据的字节长度大于或等于 2GB，BYTE_LENGTH 就会返回错误，以指明您必须使用 BYTE_LENGTH64 函数。
- 关于 BYTE_LENGTH 函数语法和用法，请参见《参考：构件块、表和过程》的第 4 章“SQL 函数”中的“BYTE_LENGTH 函数 [字符串]”。

BYTE_LENGTH64 函数

- 作用** BYTE_LENGTH64 函数返回无符号的 64 位值，该值包含大对象列或变量参数的位长度。
- 语法** `BYTE_LENGTH64(large-object-column)`
- 参数** **large-object-column** LONG VARCHAR 或 LONG BINARY 列或变量的名称。
- 用法** BYTE_LENGTH64 函数支持 LONG BINARY 和 LONG VARCHAR 列以及任意数据大小的 LONG BINARY 和 LONG VARCHAR 变量。当前，一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。

BYTE_SUBSTR64 和 BYTE_SUBSTR 函数

- 作用** BYTE_SUBSTR64 和 BYTE_SUBSTR 函数返回大对象列或变量参数的字节子字符串。
- 语法** `BYTE_SUBSTR64(large-object-column, start, length)`
`BYTE_SUBSTR(large-object-column, start, length)`
- 参数** **large-object-column** LONG VARCHAR 或 LONG BINARY 列或变量的名称。
start 表示子字符串开始位置的整数表达式。正整数表示从字符串开始处开始，第一个字节位于位置 1。负整数表示子串从字符串结尾处开始，最后一个字节位于位置 -1。
length 表示子字符串长度的整数表达式。正的 length 表示要返回的字节数且以 start 位置为起始位置。负的 length 表示要返回的字节数且以 start 位置为结束位置。

- 用法
- 函数 BYTE_LENGTH64、BYTE_SUBSTR64 和 BYTE_SUBSTR 的嵌套运算不支持大对象列或变量。
 - BYTE_SUBSTR64 和 BYTE_SUBSTR 函数支持 LONG BINARY 和 LONG VARCHAR 列以及任意数据大小的 LONG BINARY 和 LONG VARCHAR 变量。当前，一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。

CHAR_LENGTH 函数

- 作用 CHAR_LENGTH 函数返回有符号的 32 位值，该值包含 LONG VARCHAR 列或变量参数（包括尾随空白）的字符长度。
- 语法 **CHAR_LENGTH**(*long-varchar-object*)
- 参数 **long-varchar-object** LONG VARCHAR 列或 LONG VARCHAR 变量的名称。
- 用法
- CHAR_LENGTH 支持 LONG VARCHAR 列以及任意数据大小的 LONG VARCHAR 变量。当前，一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。
 - 如果参数为 NULL，则 CHAR_LENGTH 返回 NULL。
 - 如果字符长度超过 2GB - 1 (2147483647)，则返回错误。

CHAR_LENGTH64 函数

- 作用 CHAR_LENGTH64 函数返回无符号的 64 位值，该值包含 LONG VARCHAR 列或变量参数（包括尾随空白）的字符长度。
- 语法 **CHAR_LENGTH64**(*long-varchar-object*)
- 参数 **long-varchar-object** 表中的 LONG VARCHAR 列或 LONG VARCHAR 变量的名称。
- 用法
- CHAR_LENGTH64 支持 LONG VARCHAR 列以及任意数据大小的 LONG VARCHAR 变量。当前，一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。
 - 如果参数为 NULL，则 CHAR_LENGTH64 返回 NULL。

CHARINDEX 函数

作用	CHARINDEX 函数返回 64 位带符号的整数，它包含指定字符串在大对象列或变量参数中第一次出现的位置。对于 CHAR 和 VARCHAR 列，CHARINDEX 返回 32 位带符号的整数位置。
语法	CHARINDEX (<i>string-expression</i> , <i>large-object-column</i>)
参数	string-expression 您要搜索的最大长度为 255 个字节的字符串。 large-object-column LONG VARCHAR 或 LONG BINARY 列或变量的名称。
用法	<ul style="list-style-type: none"> 在 CHARINDEX 函数中返回或指定的所有位置或偏移量始终为字符偏移量，并且可能不同于多字节数据的字节偏移量。 如果搜索的大对象单元包含多个 <i>string-expression</i> 实例，CHARINDEX 仅返回第一个实例的位置。 如果列不包含任何字符串，CHARINDEX 函数返回零 (0)。 搜索长度超过 255 个字节的字符串将返回 NULL。 搜索零长度的字符串将返回 1。 如果任一参数为 NULL，结果则为 NULL。 CHARINDEX 支持搜索 LONG VARCHAR 和 LONG BINARY 列以及任意数据大小的 LONG VARCHAR 和 LONG BINARY 变量。当前，一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。
另请参见	请参见《参考：构件块、表和过程》的 第 4 章“SQL 函数” 中的“ CHARINDEX 函数 [字符串] ”。

LOCATE 函数

作用	LOCATE 函数返回 64 位带符号的整数，它包含指定字符串在大对象列或变量参数中的位置。对于 CHAR and VARCHAR 列，LOCATE 返回 32 位带符号的整数位置。
语法	LOCATE (<i>large-object-column</i> , <i>string-expression</i> [, <i>numeric-expression</i>])
参数	large-object-column 要搜索的 LONG VARCHAR 或 LONG BINARY 列或变量的名称。 string-expression 您要搜索的最大长度为 255 个字节的字符串。

numeric-expression 字符串中作为搜索起始位置的位置或偏移量。对于 LONG VARCHAR 和 LONG BINARY 列, *numeric-expression* 为 64 位带符号的整数; 对于 CHAR、VARCHAR 和 BINARY 列, 该参数为 32 位带符号的整数。第一个字符为位置 1。如果起始偏移量是负数, LOCATE 返回上一匹配字符串偏移量, 而不是第一个字符。负数偏移量表示要从搜索中排除的字符串末尾的字符数。排除的字符数的计算公式为 $(-1 * \text{偏移量}) - 1$ 。

用法

- 在 LOCATE 函数中返回或指定的所有位置或偏移量始终为字符偏移量, 并且可能不同于多字节数据的字节偏移量。
- 如果搜索的大对象单元包含多个字符串实例, 则会出现下列情况:
 - 如果指定 *numeric-expression*, LOCATE 则在字符串中的该偏移量处开始搜索。
 - 如果未指定 *numeric-expression*, LOCATE 则仅返回第一个实例的位置。
- 如果列不包含任何字符串, LOCATE 将返回零 (0)。
- 搜索长度超过 255 个字节的字符串将返回 NULL。
- 搜索零长度的字符串将返回 1。
- 如果任一参数为 NULL, 结果则为 NULL。
- LOCATE 支持搜索 LONG VARCHAR 和 LONG BINARY 列以及任意数据大小的 LONG VARCHAR 和 LONG BINARY 变量。当前, 一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。

另请参见

《参考: 构件块、表和过程》的中的第 4 章 “SQL 函数” “LOCATE 函数 [字符串]”。

OCTET_LENGTH 函数

作用

OCTET_LENGTH 函数返回无符号的 64 位值, 该值包含大对象列或变量参数的位长度。

语法

OCTET_LENGTH(*column-name*)

参数

large-object-column LONG VARCHAR 或 LONG BINARY 列或变量的名称。

用法

- 如果参数为 NULL, 则 OCTET_LENGTH 返回 NULL。
- OCTET_LENGTH 支持所有 Sybase IQ 数据类型以及任意数据大小的 LONG VARCHAR 和 LONG BINARY 变量。当前, 一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。

PATINDEX 函数

作用	PATINDEX 函数返回 64 位无符号的整数，它包含指定模式在 LONG VARCHAR 列或变量中第一次出现的位置。对于 CHAR 和 VARCHAR 列，PATINDEX 返回 32 位无符号的整数位置。
语法	PATINDEX('%pattern%', long-varchar-column)
参数	<p>pattern 要搜索的模式。对于具有通配符的模式，此字符串限制为 126 字节。如果省略前导百分比通配符，当模式出现在列值开头时，PATINDEX 则返回一 (1)，当模式未出现在列值开头时，则返回零 (0)。类似地，如果省略尾随百分比通配符，模式应出现在列值末尾。模式使用的通配符与 LIKE 比较相同。</p> <p>不带通配符（百分比 (%) 或下划线 (_) 的模式) 的长度最大可达 255 字节。</p> <p>long-varchar-column LONG VARCHAR 列或变量的名称。</p>
用法	<ul style="list-style-type: none"> 在 PATINDEX 函数中返回或指定的所有位置或偏移量始终为字符偏移量，并且可能不同于多字节数据的字节偏移量。 如果搜索的 LONG VARCHAR 单元包含多个字符串模式实例，PATINDEX 仅返回第一个实例的位置。 如果列不包含任何模式，PATINDEX 将返回零 (0)。 搜索长度超过 126 个字节的模式将返回 NULL。 搜索零长度的模式将返回 1。 如果任一参数为 NULL，结果则为零 (0)。 PATINDEX 支持任意数据大小的 LONG VARCHAR 变量。当前，一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。PATINDEX 不支持 LONG BINARY 变量或搜索 LONG BINARY 列。
另请参见	<ul style="list-style-type: none"> 《参考：构件块、表和过程》的中的第 4 章 “SQL 函数” “PATINDEX 函数 [字符串]”。 《参考：构件块、表和过程》的中的第 2 章 “SQL 语言元素” “LIKE 条件”。

SUBSTRING 函数

作用	SUBSTRING 函数返回 LONG VARCHAR 列或变量参数的可变长度字符串。如果任一参数为 NULL，则 SUBSTRING 返回 NULL。
----	--

语法	{ SUBSTRING SUBSTR } (<i>long-varchar-column</i> , <i>start</i> [, <i>length</i>])
参数	long-varchar-column LONG VARCHAR 列或变量的名称。 start 表示子字符串开始位置的整数表达式。正整数表示从字符串开始处开始，第一个字符位于位置 1。负整数表示子串从字符串结尾处开始，最后一个字符位于位置 -1。 length 表示子字符串字符长度的整数表达式。正的 length 表示要返回的字符数且以 start 位置为起始位置。负的 length 表示要返回的字符数且以 start 位置为结束位置。
用法	SUBSTRING 支持任意数据大小的 LONG VARCHAR 变量。当前，一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。SUBSTRING 不支持 LONG BINARY 变量或搜索 LONG BINARY 列。

SUBSTRING64 函数

作用	SUBSTRING64 函数返回大对象列或变量参数的可变长度字符串。
语法	SUBSTRING64 (<i>large-object-column</i> , <i>start</i> [, <i>length</i>])
参数	large-object-column LONG VARCHAR 或 LONG BINARY 列或变量的名称。 start 表示子字符串开始位置的 8 字节整数。SUBSTRING64 将值为负或零的 start 偏移量解析为用“非字符”对字符串左侧进行了填充。第一个字符从位置 1 开始。 length 表示子字符串长度的 8 字节整数。如果 length 为负，则返回错误。
示例	假定有一个名为 col1 的列，它包含字符串（“ABCDEFGH”），则 SUBSTRING64 函数将返回以下值： SUBSTRING64(col1, 2, 4) 返回字符串 “BCDE” SUBSTRING64(col1, 1, 3) 返回字符串 “BCDE” SUBSTRING64(col1, 0, 3) 返回字符串 “AB” SUBSTRING64(col1, -1, 3) 返回字符串 “A”

用法

- 如果任一参数为 NULL，则 SUBSTRING64 返回 NULL。
- 函数 SUBSTRING64、SUBSTRING、SUBSTR、BYTE_SUBSTR 和 BYTE_SUBSTR64 的嵌套运算不支持大对象列或变量。
- SUBSTRING64 支持搜索 LONG VARCHAR 和 LONG BINARY 列以及任意数据大小的 LONG VARCHAR 和 LONG BINARY 变量。当前，一个 SQL 变量可以容纳最大长度为 2GB - 1 的数据。

大对象列的集合函数支持

只有集合函数 COUNT (*) 在 LONG BINARY 和 LONG VARCHAR 类型的列中得到支持。不支持 COUNT DISTINCT 参数。如果将 LONG BINARY 或 LONG VARCHAR 类型的列用于 MIN、MAX、AVG 或 SUM 集合函数，则会返回错误。

错误和警告消息

关于本附录

本附录介绍在您处理非结构化数据（包括 LONG BINARY 和 LONG VARCHAR 列）时可能返回的错误和警告消息。

目录

主题	页码
错误 1000195	78
错误 1000198	78
错误 1000332	79
错误 1001013	79
错误 1001051	80
错误 1001052	80
错误 1001053	81
错误 1001054	81
警告 1001055	82
警告 1001056	82
错误 1001057	83
错误 1001058	84
错误 1009189	84
错误 1012030	85

错误 1000195

消息文本

LOAD specification ‘%2’ only valid for column(s) having datatype ‘%3’ .%1

项	值
SQLCode	-1000195L
常量	EMSG_BINARYFILE
SQLState	QDB95
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	20855
严重级代码	14
参数 1	发生异常的位置
参数 2	装载规范的类型
参数 3	列的数据类型

可能的原因

LOAD TABLE 语句中指定的装载规范仅对具有给定数据类型的列有效。

错误 1000198

消息文本

Cannot create join index with table(s) having column(s) of datatype %2. %1

项	值
SQLCode	-1000198L
常量	EMSG_CANNOT_CREATE_JOIN_INDEX
SQLState	QDB98
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	20858
严重级代码	14
参数 1	发生异常的位置
参数 2	列的数据类型

可能的原因

在尝试对具有一个或多个 LONG VARCHAR 或 LONG BINARY 数据类型列的表创建连接索引时会报告该错误。大部分数据类型都支持 JOIN INDEX 功能。但是，有一些数据类型（例如 LONG BINARY 和 LONG VARCHAR）不支持该功能。

错误 1000332

消息文本

Odd length of binary data value detected on column %2 %1

项	值
SQLCode	-1000332L
常量	EMSG_ODDNUMBER_NIBBLES
SQLState	QDD20
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	21206
严重级代码	14
参数 1	发生异常的位置
参数 2	列名

可能的原因

在将十六进制格式的二进制数据从主装载文件中装载到 LONG BINARY 列中时，Sybase IQ 要求十六进制数字的总数为偶数。如果单元值包含奇数个十六进制数字，就会报告此错误。用于 LONG BINARY 装载的输入文件包含的十六进制数字的总数应该始终为奇数。

错误 1001013

消息文本

Invalid data type comparison %1

项	值
SQLCode	-1001013L
常量	EMSG_TYPECOMPAREERROR
SQLState	QFA13
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	20522
严重级代码	14
参数 1	发生异常的位置

可能的原因

如果您试图使用 LIKE 谓词在 LONG BINARY 列中搜索某模式，则会报告此错误。LONG BINARY (BLOB) 列不支持 LIKE 谓词。

错误 1001051

消息文本

Query returns %3 data > 2GB.Use %2 %1

项	值
SQLCode	-1001051L
常量	EMSG_LOB_OVER_2G_W_ARG
SQLState	QFA47
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	21097
严重级代码	14
参数 1	SA 分析源代码行
参数 2	推荐的函数
参数 3	long binary 或 long varchar 数据类型

可能的原因

在查询尝试返回大于 2 GB 的 LONG BINARY 或 LONG VARCHAR 值时会报告该错误。

错误 1001052

消息文本

Parameter %2 must be long binary/varchar type. %3 %1

项	值
SQLCode	-1001052L
常量	EMSG_ONLY_SUPPORT_LOB_W_ARG
SQLState	QFA48
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	21098
严重级代码	14
参数 1	SA 分析源代码行
参数 2	LOB 参数名
参数 3	推荐的函数名

可能的原因

用于大对象 (LOB) 函数参数的数据类型无效时会报告该错误。

错误 1001053

消息文本

Wrong number of parameters to function %2 %1

项	值
SQLCode	-1001053L
常量	EMSG_WRONG_NUM_PARAMS_W_ARG
SQLState	QFA49
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	21099
严重级代码	14
参数 1	SA 分析源代码行
参数 2	函数名

可能的原因

向大对象 (LOB) 函数传递的参数数目不正确时会报告该错误。

错误 1001054

消息文本

You cannot specify long binary/varchar column in the ORDER/GROUP by clause or in an aggregate function. %1

项	值
SQLCode	-1001054L
常量	EMSG_LOB_NOT_ALLOWED_GROUP
SQLState	QFA50
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	21100
严重级代码	14
参数 1	发生异常的位置

可能的原因

尝试在 ORDER BY、GROUP BY 或集合子句中使用 LONG BINARY 列时会报告该错误。

警告 1001055

消息文本

An error occurred loading %1 column, %2, for %3, rowid %4.

项	值
SQLCode	1001055L
常量	EMSG_LOB_LOAD_ERROR_WARN
SQLState	QFA51
ODBC 2 状态	OK
ODBC 3 状态	OK
Sybase 错误代码	21101
严重级代码	10
参数 1	long binary 或 long varchar 数据类型
参数 2	FP 索引名称
参数 3	辅助文件名
参数 4	行 ID

可能的原因

在装载操作期间打开或读取 LONG BINARY 或 LONG VARCHAR 辅助文件时遇到错误的情况下会返回该警告消息。在 SECONDARY_FILE_ERROR 选项为 OFF 且有错误发生时，服务器日志和 IQ 消息文件中会返回此警告消息。

警告 1001056

消息文本

An error occurred extracting %1 column, %2, for %3.

项	值
SQLCode	1001056L
常量	EMSG_LOB_EXTRACT_ERROR_WARN
SQLState	QFA52
ODBC 2 状态	OK
ODBC 3 状态	OK
Sybase 错误代码	21102
严重级代码	10
参数 1	long binary 或 long varchar 数据类型
参数 2	FP 索引名称
参数 3	辅助文件名

可能的原因

在提取操作期间尝试提取 LONG BINARY 或 LONG VARCHAR 列且遇到错误时会返回该警告消息。在 SECONDARY_FILE_ERROR 选项为 OFF 且有错误发生时，服务器日志和 IQ 消息文件中会返回此警告消息。

错误 1001057

消息文本

You must use BFILE() to extract %2 column. %1

项	值
SQLCode	-1001057L
常量	EMSG_LOB_EXTRACT_USE_BFILE
SQLState	QFA53
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	21103
严重级代码	14
参数 1	发生异常的位置
参数 2	long binary 或 long varchar 数据类型

可能的原因

在数据库选项 TEMP_EXTRACT_NAME1 设置为 ON 的情况下执行包含 LONG BINARY 或 LONG VARCHAR 列的查询，且未指定 BFILE 函数时会报告该错误。

错误 1001058

消息文本

The secondary file name, %2, is too long. %1

项	值
SQLCode	-1001058L
常量	EMSG_LOB_SECONDARY_FILE_TOOLONG
SQLState	QFA54
ODBC 2 状态	OK
ODBC 3 状态	OK
Sybase 错误代码	21104
严重级代码	14
参数 1	发生异常的位置
参数 2	辅助文件名

可能的原因

LOAD TABLE 辅助文件路径名的长度超过操作系统的路径名长度限制时会报告该错误。报告该错误后采取的措施取决于 SECONDARY_FILE_ERROR 数据库选项的值。

错误 1009189

消息文本

Text document exceeds maximum number of terms.Support up to 4294967295 terms per document. %1

项	值
SQLCode	-1009189L
常量	EMSG_MAXTERM_ERROR
SQLState	QSB84
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	21210
严重级代码	14
参数 1	发生异常的位置

可能的原因

来自外部预过滤器或分词符库的错误。

错误 1012030

消息文本

for long binary/varchar Column ‘%2’ , database page size of (%3) must be greater than %4. %1

项	值
SQLCode	-1012030
常量	EMSG_CAT_PAGESIZETOOSMALL
SQLState	QUA30
ODBC 2 状态	ERROR
ODBC 3 状态	ERROR
Sybase 错误代码	20953
严重级代码	14
参数 1	发生异常的位置
参数 2	列号
参数 3	请求的页大小
参数 4	允许的最小页大小

可能的原因

数据库页大小太小，无法创建 LONG BINARY 或 LONG VARCHAR 列。数据库页大小必须为 128K 或更大才能创建 LONG BINARY 或 LONG VARCHAR 列。

索引

英文

Adaptive Server Enterprise

插入 IMAGE 数据 53

插入 TEXT 数据 55

ALTER TEXT CONFIGURATION 17

语法 59

ALTER TEXT INDEX 7, 8

语法 61

BEGIN PARALLEL IQ 语句 63

BFILE 函数 47

示例 48

提取功能 47

提取示例 48

语法 47

BIT_LENGTH 函数

说明 68

语法 68

BLOB

BIT_LENGTH 函数 68

BYTE_LENGTH 函数 69

BYTE_LENGTH64 函数 69

BYTE_SUBSTR 函数 69

BYTE_SUBSTR64 函数 69

CHARINDEX 函数 71

LOCATE 函数 71

LONG BINARY 53

OCTET_LENGTH 函数 72

sp_iqindexsize 45

SUBSTRING64 函数 74

TEXT 索引 57

变量 56

变量函数支持 67

插入 IMAGE 数据 53

插入数据 53

存储过程支持 39

大小 53

导出数据 47

二进制大对象 53

更新数据 53

函数支持 67

集合函数支持 75

监控性能 37

列 53

数据类型 53

数据类型转换 54

说明 53

索引 53, 57

索引支持 57

修改 53

预取 53

在查询中 27, 35

装载数据 48

BLOB 变量

数据类型转换 56

BYTE_LENGTH64 函数

说明 69

语法 69

BYTE_SUBSTR 函数

说明 69

语法 69

BYTE_SUBSTR64 函数

说明 69

语法 69

char

拆分成词语 39

CHAR_LENGTH 函数

说明 70

语法 70

CHAR_LENGTH64 函数

说明 70

语法 70

CHARINDEX 函数

说明 71

语法 71

CLOB

BIT_LENGTH 函数 68

BYTE_LENGTH 函数 69

BYTE_LENGTH64 函数 69

BYTE_SUBSTR 函数 69

- BYTE_SUBSTR64 函数 69
- CHAR_LENGTH 函数 70
- CHAR_LENGTH64 函数 70
- CHARINDEX 函数 71
- LOCATE 函数 71
- LONG VARCHAR 54
- OCTET_LENGTH 函数 72
- PATINDEX 函数 73
- sp_iqindexsize 46
- SUBSTRING 函数 73
- SUBSTRING64 函数 74
- TEXT 索引 57
- WD 索引 55, 57
- 变量 56
- 变量函数支持 67
- 插入 TEXT 数据 55
- 插入数据 55
- 词索引 55, 57
- 存储过程支持 39
- 大小 54
- 导出数据 47
- 更新数据 55
- 函数支持 67
- 集合函数支持 75
- 列 54
- 数据类型 54
- 数据类型转换 55
- 说明 53
- 索引 55, 57
- 索引支持 53, 57
- 修改 55
- 在查询中 27, 36
- 装载数据 48
- 字符大对象 54
- CLOB 变量
 - 数据类型转换 56
- CONTAINS
 - 表表达式 28, 35
- CONTAINS 条件
 - TEXT 索引 28
- contains-expression
 - FROM 子句 28
- CREATE TEXT CONFIGURATION 11
 - 语法 62
- CREATE TEXT INDEX 3, 5
 - 语法 63
- default_char 11
- default_nchar 11
- DROP TEXT CONFIGURATION 18
 - 语法 64
- DROP TEXT INDEX 8
 - 语法 65
- ENABLE_LOB_VARIABLES 选项 56
- END PARALLEL IQ
 - CREATE TEXT INDEX 63
- FROM 子句
 - CONTAINS 28, 35
 - contains-expression 28
 - 语法 28
- IMAGE 数据
 - 插入到 LONG BINARY 中 53
 - 从 ASE 插入 53
- LOAD TABLE
 - 辅助装载文件 48
 - 扩展语法 48
 - 示例 49
 - 主装载文件 48
- LOB
 - 插入数据 53, 55
 - 导出数据 47
 - 典型来源 1
 - 更新数据 53, 55
 - 函数支持 67
 - 简介 1
 - 索引支持 57
 - 装载数据 48
- LOB 变量
 - 数据类型转换 56
- LOB 的数据压缩 44
 - 更改设置 44
 - 显示设置 45
- LOB 数据的压缩 44
 - 更改设置 44
 - 显示设置 45
- LOB 压缩
 - 更改设置 44
 - 禁用 44
 - 启用 44
 - 显示设置 45
- LOCATE 函数
 - 说明 71
 - 语法 71

- LONG BINARY
 - BIT_LENGTH 函数 68
 - BLOB 53
 - BYTE_LENGTH 函数 69
 - BYTE_LENGTH64 函数 69
 - BYTE_SUBSTR 函数 69
 - BYTE_SUBSTR64 函数 69
 - CHARINDEX 函数 71
 - DELETE 53
 - INSERT 53
 - LOAD TABLE 53
 - LOCATE 函数 71
 - OCTET_LENGTH 函数 72
 - SELECT...INTO 53
 - sp_iqindexsize 45
 - SUBSTRING64 函数 74
 - TEXT 索引 57
 - TRUNCATE 53
 - UPDATE 53
 - 变量 56
 - 插入 IMAGE 数据 53
 - 插入数据 53
 - 存储过程支持 39
 - 大小 53
 - 导出数据 47
 - 二进制大对象 53
 - 更新数据 53
 - 集合函数支持 75
 - 监控性能 37
 - 列 53
 - 数据类型转换 54
 - 索引 53, 57
 - 索引支持 57
 - 修改 53
 - 在查询中 27, 35
 - 装载数据 48
- LONG BINARY 变量
 - 数据类型转换 56
- LONG VARCHAR
 - BIT_LENGTH 函数 68
 - BYTE_LENGTH 函数 69
 - BYTE_LENGTH64 函数 69
 - BYTE_SUBSTR 函数 69
 - BYTE_SUBSTR64 函数 69
 - CHAR_LENGTH 函数 70
 - CHAR_LENGTH64 函数 70
 - CHARINDEX 函数 71
 - CLOB 54
 - DELETE 55
 - INSERT 55
 - LOAD TABLE 55
 - LOCATE 函数 71
 - OCTET_LENGTH 函数 72
 - PATINDEX 函数 73
 - SELECT...INTO 55
 - sp_iqindexsize 46
 - SUBSTRING 函数 73
 - SUBSTRING64 函数 74
 - TEXT 索引 57
 - TRUNCATE 55
 - UPDATE 55
 - WD 索引 55, 57
 - 变量 56
 - 插入 TEXT 数据 55
 - 插入数据 55
 - 词索引 55, 57
 - 存储过程支持 39
 - 大小 54
 - 导出数据 47
 - 更新数据 55
 - 集合函数支持 75
 - 列 54
 - 数据类型转换 55
 - 索引 55, 57
 - 索引支持 57
 - 修改 55
 - 在查询中 27, 36
 - 装载数据 48
 - 字符大对象 54
- LONG VARCHAR 变量
 - 数据类型转换 56
- MAX_PREFIX_PER_CONTAINS_PHRASE 选项 20
- Multiplex 服务器
 - 外部库 24
- nchar
 - 拆分成词语 40
- OCTET_LENGTH 函数
 - 说明 72
 - 语法 72
- PATINDEX 函数
 - 说明 73
 - 语法 73
- sa_char_terms 存储过程 39

- sa_external_library_unload 存储过程 25, 43
- sa_list_external_library 存储过程 25, 43
- sa_nchar_terms 存储过程 40
- sa_text_index_stats 存储过程 41
- sa_text_index_vocab 存储过程 41
- SECONDARY_FILE_ERROR 选项 50
- SELECT 语句
 - FROM 子句语法 28
- sp_iqindexsize
 - BLOB 45
 - CLOB 46
 - LONG BINARY 45
 - LONG VARCHAR 46
 - 二进制大对象 45
 - 字符大对象 46
- sp_iqindexsize 存储过程 45
- sp_iqsetcompression 存储过程 44
- sp_iqshowcompression 存储过程 45
- STRING_RTRUNCATION 选项 54, 55
- SUBSTRING 函数
 - 说明 73
 - 语法 73
- SUBSTRING64 函数
 - 说明 74
 - 语法 74
- TEXT 数据
 - 插入到 LONG VARCHAR 中 55
 - 从 ASE 插入 55
- TEXT 索引 3, 57
 - CONTAINS 条件 28
 - 编辑 7
 - 变更 61
 - 创建 3, 5, 63
 - 词语 39
 - 更改数据库空间 8
 - 列表 6
 - 删除 8, 65
 - 删除行 9
 - 刷新 9
 - 统计信息 41
 - 文本配置对象 10, 11
 - 限制 6
- TEXT_DELETE_METHOD 选项 9
- TRIM_PARTIAL_MBC 选项 50
- WD 索引 57

B

- 编辑
 - TEXT 索引 7
 - 文本配置对象 17
- 变更
 - TEXT 索引 7, 61
 - 文本配置对象 17, 59
- 变量
 - BLOB 56
 - BLOB 的函数支持 67
 - BLOB 转换 56
 - CLOB 56
 - CLOB 的函数支持 67
 - CLOB 转换 56
 - LONG BINARY 56
 - LONG BINARY 转换 56
 - LONG VARCHAR 56
 - LONG VARCHAR 转换 56
 - 二进制大对象 56
 - 二进制大对象转换 56
 - 字符大对象 56
 - 字符大对象转换 56
- 标识
 - 外部库 42, 43
- 标准 2

C

- 插入
 - BLOB 53
 - CLOB 55
 - LOB 53, 55
 - LONG BINARY 53
 - LONG VARCHAR 55
 - 大对象数据 53, 55
- 查询
 - BLOB 27, 35
 - CLOB 27, 36
 - LONG BINARY 27, 35
 - LONG VARCHAR 27, 36
 - 二进制大对象 27, 35
 - 字符大对象 27, 36
- 拆分
 - 词语 39, 40

创建

- TEXT 索引 3, 5, 63
- 文本配置对象 11, 62

词语

- TEXT 索引 39
- 拆分 39, 40
- 忽略 15
- 全文本搜索 27
- 停止列表 17
- 行位置 39, 40
- 最大长度 15
- 最小长度 14

存储过程

- BLOB 39
- CLOB 39
- LONG BINARY 39
- LONG VARCHAR 39
- sa_char_terms 39
- sa_external_library_unload 43
- sa_list_external_library 43
- sa_nchar_terms 40
- sa_text_index_stats 41
- sa_text_index_vocab 41
- sp_iqindexsize 45
- sp_iqsetcompression 44
- sp_iqshowcompression 45
- 二进制大对象 39
- 字符大对象 39

错误消息 77

D

大对象数据

- 插入 53, 55
- 导出 47
- 更新 53, 55
- 索引支持 57
- 装载 48

导出

- BFILE 函数 47
- BFILE 示例 48
- BLOB 47
- CLOB 47
- LOB 47
- LONG BINARY 47
- 大对象数据 47

导出 LONG VARCHAR 47

多字节字符

- TRIM_PARTIAL_MBC 选项 50
- 删去部分 50
- 装载时截断 50

E

二进制大对象

- BIT_LENGTH 函数 68
- BLOB 53
- BYTE_LENGTH 函数 69
- BYTE_LENGTH64 函数 69
- BYTE_SUBSTR 函数 69
- BYTE_SUBSTR64 函数 69
- LONG BINARY 53
- OCTET_LENGTH 函数 72
- sp_iqindexsize 45
- SUBSTRING64 函数 74
- TEXT 索引 57
- 变量 56
- 插入 IMAGE 数据 53
- 存储过程支持 39
- 大小 53
- 集合函数支持 75
- 监控性能 37
- 列 53
- 数据类型 53
- 数据类型转换 54
- 说明 53
- 索引 53, 57
- 索引支持 57
- 修改 53
- 在查询中 27, 35
- 二进制大对象变量
- 数据类型转换 56

F

非结构化数据分析选项 1

授权 1

分词符库 10, 11, 13, 14, 15, 23

限制 24

分析

非结构化数据分析选项 1

G

更新

- BLOB 53
- CLOB 55
- LOB 53, 55
- LONG BINARY 53
- LONG VARCHAR 55
- 大对象数据 53, 55

H

函数

- BFILE 47
- BFILE 示例 48
- BIT_LENGTH 68
- BYTE_LENGTH 69
- BYTE_LENGTH64 69
- BYTE_SUBSTR 69
- BYTE_SUBSTR64 69
- CHAR_LENGTH 70
- CHAR_LENGTH64 70
- CHARINDEX 71
- LOCATE 71
- LONG BINARY 集合支持 75
- LONG VARCHAR 集合支持 75
- OCTET_LENGTH 72
- PATINDEX 73
- SUBSTRING 73
- SUBSTRING64 74
- 针对 BLOB 67
- 针对 BLOB 变量 67
- 针对 CLOB 67
- 针对 CLOB 变量 67
- 针对 LOB 67

J

兼容性

- 与 Adaptive Server Enterprise 之间 2
- 与 ASE 之间 2
- 与 SA 之间 2
- 与 SQL Anywhere 之间 2

禁用

- 外部库 25
- 压缩 44
- 警告消息 77

K

- 库, 外部 10, 11

L

- 立即刷新 9, 41

列表

- TEXT 索引 6
- 外部库 43
- 文本配置对象 16

Q

启用

- 外部库 25
- 压缩 44

前缀

- 词语限制 20
- 全文本搜索 2, 3, 27
- 类型 27

S

删除

- TEXT 索引 8, 65
- 文本配置对象 18, 64

升级

- LONG BINARY 53
- 现有的 LONG BINARY 列 53

实例

- 外部库 43

示例

- 文本配置对象 18, 19

- 授权 1

数据库空间

- TEXT 索引 8
- 修改 8

数据库选项

- ENABLE_LOB_VARIABLES 56
- MAX_PREFIX_PER_CONTAINS_PHRASE 20
- TEXT_DELETE_METHOD 9

数据类型

- BLOB 53
- CLOB 54
- LONG BINARY 53
- LONG VARCHAR 54

数据类型转换

- LONG BINARY 变量 56
- 从 LONG BINARY 到 BINARY 54
- 从 LONG BINARY 到 VARBINARY 54
- 从 LONG VARCHAR 到 CHAR 55
- 从 LONG VARCHAR 到 VARCHAR 55

刷新

- TEXT 索引 9
- 立即 9, 41

搜索

- CONTAINS 条件 28
- CONTAINS 子句 35
- TEXT 索引 3
- 前缀词语限制 20
- 全文本 2, 27

搜索条件

- CONTAINS 条件 28
- CONTAINS 子句 35

索引

- BLOB 53, 57
- CLOB 55, 57
- LOB 57
- LONG BINARY 53, 57
- LONG VARCHAR 55, 57
- TEXT 3, 57
- WD 55, 57
- WORD 55, 57
- 包含 55, 57
- 大对象数据 57
- 二进制大对象 53, 57
- 列表 6
- 全文本搜索 2, 27
- 刷新 9
- 字符大对象 55, 57

T

提取功能

- BFILE 函数 47

添加

- TEXT 索引 3, 5
- 文本配置对象 11
- 停止列表 10, 11, 15
- 修改 17
- 统计信息
 - TEXT 索引 41

W

外部库 23

- Multiplex 服务器 24

标识 42

禁用 25

列表 43

启用 25

限制 24

卸载 25, 43

文本配置对象 10

变更 17, 59

创建 11, 62

列表 16

缺省值 11

删除 18, 64

设置 13, 14, 15

示例 18, 19

文本搜索

- FROM contains-expression 28

文档

SQL Anywhere ix

Sybase IQ vii

X

限制

- TEXT 索引 6

外部库 24

消息

错误 77

警告 77

索引

卸载

外部库 25, 43

性能监控器

BLOB 37

LONG BINARY 37

二进制大对象 37

修改

数据库空间 8

停止列表 17

选件

非结构化数据分析 1

选项

ENABLE_LOB_VARIABLES 56

MAX_PREFIX_PER_CONTAINS_PHRASE 20

TEXT_DELETE_METHOD 9

Y

预过滤器库 10, 11, 13, 23

限制 24

最大词语长度 15

最小词语长度 14

预取 53

Z

装载

BLOB 48

CLOB 48

LOAD TABLE 示例 49

LOB 48

LONG BINARY 48

LONG VARCHAR 48

SECONDARY_FILE_ERROR 选项 50

TRIM_PARTIAL_MBC 选项 50

大对象数据 48

截断字符数据 50

控制错误 50

去除尾随空白 50

字符大对象

BIT_LENGTH 函数 68

BYTE_LENGTH 函数 69

BYTE_LENGTH64 函数 69

BYTE_SUBSTR 函数 69

BYTE_SUBSTR64 函数 69

CHAR_LENGTH 函数 70

CHAR_LENGTH64 函数 70

CHARINDEX 函数 71

CLOB 54

LOCATE 函数 71

LONG VARCHAR 54

OCTET_LENGTH 函数 72

PATINDEX 函数 73

sp_iqindexsize 46

SUBSTRING 函数 73

SUBSTRING64 函数 74

TEXT 索引 57

WD 索引 55, 57

变量 56

插入 TEXT 数据 55

词索引 55, 57

存储过程支持 39

大小 54

集合函数支持 75

列 54

数据类型 54

数据类型转换 55

说明 53

索引 55, 57

索引支持 53, 57

修改 55

在查询中 27, 36

字符大对象变量

数据类型转换 56