

SYBASE®

非構造化データ分析

**Sybase IQ**

15.2

ドキュメント ID : DC00285-01-1520-01

改訂 : 2010 年 4 月

Copyright © 2010 by Sybase, Inc. All rights reserved.

このマニュアルは Sybase ソフトウェアの付属マニュアルであり、新しいエディションまたはテクニカル・ノートで特に示されない限り、後続のリリースにも付属します。このマニュアルの内容は、予告なく変更されることがあります、このマニュアルに記載されているソフトウェアはライセンス契約に基づいて提供されるものであり、無断で使用することはできません。

#### マニュアルの注文

マニュアルの注文を承ります。ご希望の方は、サイバース株式会社営業部または代理店までご連絡ください。マニュアルの変更は、弊社の定期的なソフトウェア・リリース時にのみ提供されます。このマニュアルの内容を Sybase, Inc. の書面による事前の許可なく複製、転載、翻訳することは、電子的、機械的、手作業、光学的、その他、形態や手段を問わず禁じられています。

Sybase の商標は、[Sybase trademarks ページ](http://www.sybase.com/detail?id=1011207) (<http://www.sybase.com/detail?id=1011207>) で確認できます。Sybase およびこのリストに掲載されている商標は、米国法人 Sybase, Inc. の商標です。® は、米国における登録商標であることを示します。

Java および Java 関連の商標は、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Unicode と Unicode のロゴは、Unicode, Inc. の登録商標です。

このマニュアルに記載されている上記以外の社名および製品名は、当該各社の商標または登録商標の場合があります。

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.

# 目次

はじめに .....	vii	
<b>第 1 章</b>	<b>Sybase IQ の非構造化データ分析の概要 .....</b>	<b>1</b>
	非構造化データ分析オプション .....	1
	全文検索 .....	2
	互換性 .....	2
	標準への準拠 .....	2
<b>第 2 章</b>	<b>TEXT インデックスとテキスト設定オブジェクト .....</b>	<b>3</b>
	TEXT インデックス .....	3
	WD インデックスと TEXT インデックスの比較 .....	4
	TEXT インデックスの作成 .....	5
	TEXT インデックスのサイズの見積もりに関するガイドライン .....	6
	TEXT インデックスの制限 .....	6
	TEXT インデックスのリストの表示 .....	7
	TEXT インデックスの編集 .....	7
	ロケーションの変更 .....	8
	TEXT インデックスの削除 .....	9
	TEXT インデックスの更新 .....	9
	TEXT インデックスのローの削除 .....	9
	テキスト設定オブジェクト .....	11
	デフォルトのテキスト設定オブジェクト .....	11
	テキスト設定の作成 .....	12
	テキスト設定オブジェクトの設定 .....	13
	テキスト設定のリストの表示 .....	16
	テキスト設定の変更 .....	17
	トップリストの変更 .....	18
	テキスト設定の削除 .....	18
	テキスト設定オブジェクトの例 .....	19
	テキスト検索でのプレフィックス単語の制限 .....	21

---

<b>第 3 章</b>	<b>外部ライブラリ .....</b>	<b>23</b>
	外部ライブラリについて .....	23
	外部ライブラリの制限 .....	24
	マルチプレックス・サーバでの外部ライブラリの使用 .....	25
	起動時の外部ライブラリの有効化と無効化 .....	25
	外部ライブラリのアンロード .....	25
<b>第 4 章</b>	<b>非構造化データのクエリ .....</b>	<b>27</b>
	全文検索 .....	27
	全文検索のタイプ .....	27
	LONG BINARY カラムのクエリ .....	36
	LONG VARCHAR カラムのクエリ .....	36
	LONG BINARY カラムと LONG VARCHAR カラムのパフォーマンスのモニタリング .....	37
<b>第 5 章</b>	<b>ストアド・プロシージャのサポート .....</b>	<b>39</b>
	TEXT インデックス内の単語の管理 .....	39
	sa_char_terms システム・プロシージャ .....	39
	sa_nchar_terms システム・プロシージャ .....	40
	sa_text_index_stats システム・プロシージャ .....	41
	sa_text_index_vocab システム・プロシージャ .....	42
	外部ライブラリの確認 .....	43
	sa_external_library_unload システム・プロシージャ .....	43
	sa_list_external_library プロシージャ .....	43
	ラージ・オブジェクト・データの圧縮の制御 .....	44
	sp_iqsetcompression プロシージャ .....	44
	sp_iqshowcompression プロシージャ .....	45
	ラージ・オブジェクト・カラムについての情報の表示 .....	46
<b>第 6 章</b>	<b>ラージ・オブジェクト・データのロードとアンロード .....</b>	<b>47</b>
	ラージ・オブジェクト・データのエクスポート .....	47
	BFILE 関数 .....	47
	ラージ・オブジェクト・データのロード .....	48
<b>第 7 章</b>	<b>ラージ・オブジェクト・データ型 .....</b>	<b>53</b>
	ラージ・オブジェクト・データ型、LONG BINARY および BLOB .....	53
	ラージ・オブジェクト・データ型、LONG VARCHAR および CLOB .....	54
	ラージ・オブジェクト変数 .....	56
	ラージ・オブジェクト変数のデータ型変換 .....	57
	ラージ・オブジェクト・カラムのインデックスのサポート .....	57
	ラージ・オブジェクト・カラムの TEXT インデックスのサポート .....	57
	LONG VARCHAR (CLOB) カラムの WD インデックスのサポート .....	58

---

第 8 章	SQL 文のサポート .....	59
	ALTER TEXT CONFIGURATION 文 .....	59
	ALTER TEXT INDEX 文 .....	61
	CREATE TEXT CONFIGURATION 文 .....	62
	CREATE TEXT INDEX 文 .....	63
	DROP TEXT CONFIGURATION 文 .....	64
	DROP TEXT INDEX 文 .....	65
第 9 章	関数のサポート .....	67
	ラージ・オブジェクト・データをサポートする関数の概要 .....	68
	BIT_LENGTH 関数 .....	69
	BYTE_LENGTH 関数 .....	69
	BYTE_LENGTH64 関数 .....	69
	BYTE_SUBSTR64 関数と BYTE_SUBSTR 関数 .....	70
	CHAR_LENGTH 関数 .....	70
	CHAR_LENGTH64 関数 .....	71
	CHARINDEX 関数 .....	71
	LOCATE 関数 .....	72
	OCTET_LENGTH 関数 .....	73
	PATINDEX 関数 .....	73
	SUBSTRING 関数 .....	74
	SUBSTRING64 関数 .....	75
	ラージ・オブジェクト・カラムの集合関数のサポート .....	75
付録 A	エラー・メッセージと警告メッセージ .....	77
	エラー 1000195 .....	78
	エラー 1000198 .....	78
	エラー 1000332 .....	79
	エラー 1001013 .....	79
	エラー 1001051 .....	80
	エラー 1001052 .....	80
	エラー 1001053 .....	81
	エラー 1001054 .....	81
	警告 1001055 .....	82
	警告 1001056 .....	82
	エラー 1001057 .....	83
	エラー 1001058 .....	84
	エラー 1009189 .....	84
	エラー 1012030 .....	85
索引 .....	87	



# はじめに

## 対象読者

このマニュアルは、Sybase IQ で非構造化データを操作するためのリファレンス資料を必要としている Sybase® IQ ユーザを対象としています。このマニュアルには、Sybase IQ の非構造化データ分析機能に関する構文、パラメータ、関数、ストアド・プロシージャ、インデックス、オプションなどの情報が掲載されています。このマニュアルを、Sybase IQ マニュアル・セットの他のマニュアルと一緒にリファレンスとして使用し、Sybase IQ データベース内の非構造化データの格納と取得について理解してください。

## 関連 Sybase IQ マニュアル

Sybase IQ 15.2 のマニュアル・セットの構成は次のとおりです。

- 使用しているプラットフォームの『リリース・ノート』-マニュアルには記載できなかった最新の情報が記載されています。  
リリース・ノートの最新バージョンを利用する場合があります。製品の CD がリリースされた後で、製品またはマニュアルに関する重要な情報が追加されているかを確認するには、Sybase Product Manuals Web サイトを使用してください。
- 使用しているプラットフォームの『インストールおよび設定ガイド』-Sybase IQ のインストール、アップグレード、およびいくつかの設定の手順について説明しています。
- 『新機能の概要 Sybase IQ 15.2』-最新バージョンの新機能と動作変更がまとめられています。
- 『Sybase IQ による高度なセキュリティ』-Sybase IQ データ・レポジトリ内でのユーザによるカラムの暗号化の使用について説明しています。この製品のオプションをインストールするには、別途ライセンスが必要です。
- 『エラー・メッセージ』-Sybase IQ エラー・メッセージ (Sybase エラー・コード、SQLCode、SQLState、および SQL プリプロセッサのエラーと警告) を示します。
- 『IMSL 数値関数ライブラリ・ユーザ・ガイド：第 2/2 卷 C 統計ライブラリ』-IMSL C 統計ライブラリの時系列 C 関数の簡潔な説明が記載されています。このマニュアルは、RAP - The Trading Edition® Enterprise のユーザのみが入手できます。
- 『Sybase IQ の概要』-Sybase IQ や Sybase Central™ データベース管理ツールの操作に慣れていない場合に参照してください。操作の練習ができます。

- 
- ・『パフォーマンス&チューニング・ガイド』－巨大なデータベースのクエリ最適化、設計、チューニングについて説明しています。
  - ・『クイック・スタート』－Sybase IQ のソフトウェア・インストールを確定するために Sybase IQ に付属のデモ・データベースの構築とクエリを行う方法が記載されています。デモ・データベースのマルチプレックスへの変換についても説明しています。
  - ・『リファレンス・マニュアル』－Sybase IQ のリファレンス・ガイドです。
    - ・『リファレンス：ビルディング・ブロック、テーブル、およびプロセージャ』－Sybase IQ でサポートされる SQL、ストアド・プロシージャ、データ型、およびシステム・テーブルについて説明しています。
    - ・『リファレンス：文とオプション』－Sybase IQ でサポートされる SQL 文およびオプションについて説明しています。
  - ・『システム管理ガイド』－以下で構成されています。
    - ・『システム管理ガイド第 1 卷』－起動、接続、データベース作成、自動入力とインデックス作成、バージョン管理、照合、システムのバックアップとリカバリ、トラブルシューティング、およびデータベースの修復方法について説明しています。
    - ・『システム管理ガイド第 2 卷』－プロセージャとバッチの作成および実行、OLAP でのプログラミング、リモート・データへのアクセス、Open Server™ としての IQ の設定について説明しています。このマニュアルでは、スケジューリングとイベント処理、XML でのプログラミング、デバッグについても説明しています。
  - ・『時系列ガイド』－時系列の予測と分析に使用する SQL 関数について説明しています。この製品オプションを使用するには、RAP – The Trading Edition™ Enterprise が必要です。
  - ・『Sybase IQ の非構造化データ分析の概要』－Sybase IQ データベースでの非構造化データの格納と取得について説明しています。この製品のオプションをインストールするには、別途ライセンスが必要です。
  - ・『ユーザ定義関数ガイド』－ユーザ定義関数、パラメータ、考えられる使用事例が記載されています。
  - ・『Sybase IQ マルチプレックスの使用』－複数のノードにまたがって発生する大きなクエリの負荷を管理するマルチプレックス機能の使用方法について説明しています。
  - ・『ユーティリティ・ガイド』－Sybase IQ ユーティリティ・プログラムのリファレンス項目(使用可能な構文、パラメータ、オプションなど)について説明しています。

Sybase IQ 15.2 のマニュアル・セットは、Product Manuals からオンラインで利用できます。(<http://sybooks.sybase.com>)

**関連 SQL Anywhere  
マニュアル**

Sybase IQ は SQL Anywhere® パッケージのコンポーネントである SQL Anywhere Server と多くのコンポーネントを共有しているため、Sybase IQ は SQL Anywhere Server と同じ機能を数多くサポートしています。IQ のマニュアル・セットは、SQL Anywhere のマニュアルの該当する箇所を参照しています。

SQL Anywhere には、次のマニュアルがあります。

- ・ 『SQL Anywhere サーバー - データベース管理』では、SQL Anywhere データベースの実行、管理、構成方法について説明します。このマニュアルでは、データベース接続、データベース・サーバ、データベース・ファイル、バックアップ手順、セキュリティ、高可用性、Replication Server® での複写、管理ユーティリティおよびオプションについても説明します。
- ・ 『SQL Anywhere サーバー - プログラミング』では、C、C++、Java、PHP、Perl、Python、および Visual Basic や Visual C# などの .NET プログラミング言語を使用したデータベース・アプリケーションの構築および展開方法について説明します。このマニュアルでは、ADO.NET や ODBC などの各種プログラミング・インターフェースについても説明します。
- ・ 『SQL Anywhere サーバー - SQL リファレンス』には、システム・プロシージャおよびカタログ（システム・テーブルおよびビュー）に関する参照情報が記載されています。また、SQL 言語（検索条件、構文、データ型、関数）の SQL Anywhere の実装についても説明します。
- ・ 『SQL Anywhere サーバー - SQL の使用法』では、データベースの設計／作成方法、データのインポート／エクスポート／変更方法、データの検索方法、およびストアド・プロシージャとトリガの作成方法について説明します。

DocCommentXchange および Sybase Product Manuals Web サイトでも、SQL Anywhere 11.0.1 コレクションの SQL Anywhere のマニュアルを参照できます。[\(http://dcx.sybase.com/dcx\\_home.php\)](http://dcx.sybase.com/dcx_home.php)



この章では、Sybase IQ の非構造化データ分析について紹介し、Sybase IQ ラージ・オブジェクト・データの標準との互換性と準拠について説明します。

トピック	ページ
非構造化データ分析オプション	1
互換性	2
標準への準拠	2

## 非構造化データ分析オプション

非構造化データ分析オプションは、Sybase IQ の機能を拡張することで、Sybase IQ データベース内のバイナリ・ラージ・オブジェクト (BLOB) とキャラクタ・ラージ・オブジェクト (CLOB) の格納、取得、および全文検索を可能にします。

---

**注意** この製品マニュアルで説明する非構造化データ分析機能を使用するには、正規のライセンスを取得している必要があります。

---

データのボリュームが増えるにつれ、リレーショナル・データベースにラージ・オブジェクト (LOB) データを格納するニーズも増大します。LOB データには次の種類があります。

- 非構造化 – データベースはデータを単に格納および抽出する。
- 半構造化 (テキストなど) – データベースはデータ構造をサポートし、操作を支援する関数 (文字列関数など) を提供する。

一般的な LOB データ・ソースとしては、イメージ、マップ、ドキュメント (PDF ファイル、ワード・プロセッサ・ファイル、プレゼンテーションなど)、オーディオ、ビデオ、XML ファイルが挙げられます。Sybase IQ では、ギガバイト (GB)、テラバイト (TB)、さらにはペタバイト (PB) のデータが含まれる個々の LOB オブジェクトを管理できます。

リレーショナル・データと非構造化データを同じロケーションに格納できるため、Sybase IQ を使用することで、同じアプリケーションとインターフェースを使用して両方のタイプのデータにアクセスできます。Sybase IQ の全文検索機能は、非構造化データおよび半構造化データの処理において、テキスト・アーカイブ・アプリケーション (テキスト分析) をサポートします。

## 全文検索

全文検索では、TEXT インデックスを使用して、テーブルのローをスキャンせずに、データベース内の単語および語句を検索します。

TEXT インデックスには、インデックス・カラム内の単語の位置情報が格納されます。テキスト設定オブジェクトによって、TEXT インデックスの構築または更新時にインデックスに配置される単語と、全文クエリの解釈方法が制御されます。

一般的に、TEXT インデックスを使用して、単語または語句が含まれるローを検索する方法は、ほとんどの場合、テーブル内の各ローをスキャンする方法よりも高速です。

## 互換性

SQL Anywhere (SA) では、ラージ・オブジェクト (最大長 2GB) を LONG VARCHAR データ型または LONG BINARY データ型のカラムに格納できます。SQL Anywhere は SQL/2003 標準に準拠して、これらのデータ型をサポートしています。SQL Anywhere は、BYTE\_LENGTH64、BYTE\_SUBSTR64、BFILE、BIT\_LENGTH、OCTET\_LENGTH、CHAR\_LENGTH64、SUBSTRING64 の各関数をサポートしていません。

Adaptive Server® Enterprise (ASE) では、テキスト・ラージ・オブジェクト (最大長 2GB) を TEXT データ型のカラムに、バイナリ・ラージ・オブジェクト (最大長 2GB) を IMAGE データ型のカラムにそれぞれ格納できます。Adaptive Server Enterprise は、ANSI SQL Transact-SQL® 拡張に準拠して、これらのデータ型をサポートしています。

## 標準への準拠

Sybase IQ LONG BINARY 機能と LONG VARCHAR 機能は、ISO/ANSI SQL 標準の Core レベルに準拠しています。

# TEXT インデックスとテキスト設定オブジェクト

この章では、TEXT インデックスとテキスト設定オブジェクトの操作について説明します。TEXT インデックスには、インデックス・カラム内の単語の位置情報が格納されます。TEXT インデックスは、テキスト設定オブジェクトに格納されている設定を使用して作成されます。テキスト設定オブジェクトによって、無視する単語、インデックスに含める単語の最小長と最大長などの TEXT インデックス・データの特性が制御されます。

トピック	ページ
TEXT インデックス	3
テキスト設定オブジェクト	11

## TEXT インデックス

全文検索では、テーブル・ローではなく、TEXT インデックスが検索されます。全文検索を実行するには、検索するカラムに TEXT インデックスを作成する必要があります。TEXT インデックスには、インデックス・カラム内の単語の位置情報が格納されます。TEXT インデックスを使用するクエリは、ほとんどの場合、テーブル内のすべての値をスキヤンする必要のあるクエリよりも高速です。

TEXT インデックスを作成するときに、TEXT インデックスの作成および更新時に使用するテキスト設定オブジェクトを指定できます。テキスト設定オブジェクトには、インデックスの構築方法に影響を与える設定が格納されます。テキスト設定オブジェクトを指定しない場合、データベース・サーバはデフォルトの設定オブジェクトを使用します。

TEXT インデックスを作成できるカラムのデータ型は、CHAR、VARCHAR、LONG VARCHAR、BINARY、VARBINARY、LONG BINARY です。BINARY、VARBINARY、LONG BINARY カラムの場合は、TEXT インデックスで、テキスト設定と外部プレフィルタ・ライブラリを使用する必要があります。

## WD インデックスと TEXT インデックスの比較

次の表では、WD インデックスと TEXT インデックスの構文と機能を比較しています。

表 2-1: WD インデックスと TEXT インデックス

機能	WD インデックスによるサポート	TEXT インデックスによるサポート
単語の連結	あり、構文： tbl.col CONTAINS('great', 'white' , 'whale')	あり、構文： CONTAINS(tbl.col, 'great white whale')
一般的なブール式	あり、構文： tbl.col CONTAINS ( 'great' ) AND (tbl.col CONTAINS('white') OR tbl.col CONTAINS('whale') AND NOT tbl.col CONTAINS('ship'))	あり、構文： CONTAINS(tbl.col, 'great AND ( white OR whale AND NOT ship )')
プレフィクスと一致する単語の検索	不可	あり、構文例： CONTAINS (tbl.col, 'whale*')
LIKE 述部による高速化	あり、構文例： tbl.col LIKE 'whale%'	不可
近接する単語の検索	不可	あり、構文例： CONTAINS(tbl.col, 'white BEFORE whale')  CONTAINS(tbl.col, 'whale NEAR white')  CONTAINS(tbl.col, ' "white whale" ')
検索スコアに基づいた結果の順序付け	不可	可

TEXT インデックスでは、プレフィックスと一致する単語の検索と、LIKE 式を使用した検索では、セマンティックが異なり、テキスト設定に応じてまったく異なる結果が返される場合があります。最小長、最大長、およびストップリストの指定によってプレフィックスの処理は制御されますが、LIKE のセマンティックは影響を受けません。

**注意** 単語の削除が発生する場合、ブール式の意味は、WD インデックスと TEXT インデックスで異なります。これは、TEXT インデックスの処理では削除された単語による影響が生じますが、WD インデックスではこれに相当する影響が生じないためです。

## TEXT インデックスの作成

全文検索を実行するには、検索するカラムに TEXT インデックスを作成する必要があります。TEXT インデックスには、インデックス・カラム内の単語の位置情報が格納されます。

### ❖ TEXT インデックスの作成 (Sybase Central)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 左側のウィンドウ枠で、[Text Indexes] フォルダを右クリックし、[新規作成] > [Text Index] を選択します。
- 3 TEXT インデックスを作成するテーブルを選択します。
- 4 TEXT インデックスの名前を入力します。[次へ] をクリックします。
- 5 インデックスに含めるカラムを選択します。[次へ] をクリックします。
- 6 TEXT インデックスのデータを処理するときに使用するテキスト設定オブジェクトを選択します。[次へ] をクリックします。
- 7 SQL Anywhere テーブルの場合は、[Specify a Refresh Type] ダイアログで [次へ] をクリックします。  
Sybase IQ テーブルの場合は、このオプションは表示されません。サポートされる更新のタイプは [即時] のみです。
- 8 TEXT インデックスを格納する DB 領域を選択します。
- 9 [次へ] をクリックします。
- 10 テキスト設定について説明するコメントを入力し、[完了] をクリックします。

❖ TEXT インデックスの作成 (Interactive SQL)

1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。

2 CREATE TEXT INDEX 文を実行します。

次の例では、iqdemo データベース内の Customers テーブルの CompanyName カラムに、TEXT インデックス myTxtIdx が作成されます。default\_char テキスト設定オブジェクトが使用されます。

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

## TEXT インデックスのサイズの見積もりに関するガイドライン

TEXT インデックスのメイン・ストアのサイズを見積もるには、次の式を使用します。

バイト数 =  $(15+L)*U + U*PAGESIZE * R + T$

各オブジェクトの意味は、次のとおりです。

- L = 語彙における単語の平均長
- U = 語彙におけるユニークな単語の数
- R = 無数のドキュメントの数
- T = すべてのドキュメントに含まれるすべての単語の合計数

TEXT インデックス用に必要な一時領域 (バイト単位) は、 $(M+20)* T$  です。

- M = テキスト設定のための単語の最大長 (バイト単位)

---

**注意** 必要な一時領域は、ソート・データの圧縮率によって異なります。

---

## TEXT インデックスの制限

Sybase IQ テキスト設定オブジェクトおよび TEXT インデックスには、次の設計上の制限があります。

- Sybase IQ エンジンは、複数のカラムにまたがる TEXT インデックスをサポートしない。
- TEXT インデックスの手動更新オプションまたは自動更新オプションはサポートされていない。
- sp\_iqrebuildindex を使用して TEXT インデックスを構築できない。

- TEXT インデックスを BEGIN PARALLEL IQ...END PARALLEL IQ 内で作成できない。
- TEXT インデックスでは、NGRAM 単語分割はサポートされない。Sybase IQ で現在サポートされているのは、GENERIC 単語分割、およびカラムあたり1つのTEXT インデックスのみ。

## TEXT インデックスのリストの表示

データベースのすべての TEXT インデックスのリストを表示します。

### ❖ TEXT インデックスのリストの表示 (Sybase Central)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 左側のウィンドウ枠で、[Text Indexes] フォルダを選択します。  
すべての TEXT インデックスのリストが右側のウィンドウ枠に表示されます。

### ❖ TEXT インデックスのリストの表示 (Interactive SQL)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 SELECT 文を実行します。

たとえば、すべての Sybase IQ TEXT インデックスを表示するには、次のように入力します。

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

カタログ・テーブルの TEXT インデックスを含むすべての TEXT インデックスを表示するには、次の構文を使用します。

```
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;
```

## TEXT インデックスの編集

DB 領域や TEXT インデックスの名前などの、TEXT インデックスの設定を変更します。

### ❖ TEXT インデックスの編集 (Sybase Central)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 左側のウィンドウ枠で、[Text Indexes] フォルダを選択します。

- 3 [Text Indexes] のリストで、変更するオブジェクトを右クリックし、[プロパティ] を選択します。
- 4 [一般] タブで、必要に応じて設定を変更します。
- 5 [OK] をクリックします。

❖ **TEXT インデックスの編集 (Interactive SQL)**

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 ALTER TEXT INDEX 文を実行します。  
たとえば、myTxtIdx を MyTextIndex に名前を変更するには、次のように入力します。

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

## ロケーションの変更

TEXT インデックスが格納される DB 領域を変更します。

❖ **ロケーションの変更 (Sybase Central)**

- 1 DBA または SPACE ADMIN 権限のあるユーザとして、または DB 領域に対する CREATE 権限のあるテーブル所有者としてデータベースに接続します。
- 2 左側のウィンドウ枠で、[Text Indexes] フォルダを選択します。
- 3 [Text Indexes] のリストで、変更するオブジェクトを右クリックし、[プロパティ] を選択します。
- 4 [一般] タブで、ドロップダウン・リストから DB 領域を選択します。
- 5 DB 領域が更新されたら、[OK] をクリックします。

❖ **ロケーションの変更 (Interactive SQL)**

- 1 DBA または SPACE ADMIN 権限のあるユーザとしてデータベースに接続します。
- 2 MOVE TO 句を含む ALTER TEXT INDEX 文を実行します。  
たとえば、TEXT インデックス MyTextIndex を、tispace という名前の DB 領域に移動するには、次のように入力します。

```
ALTER TEXT INDEX MyTextIndex ON
  GROUPO.customers MOVE TO tispace;
```

## TEXT インデックスの削除

データベースから TEXT インデックスを削除します。

### ❖ TEXT インデックスの削除 (Sybase Central)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 左側のウィンドウ枠で、[Text Indexes] フォルダを選択します。
- 3 [Text Indexes] のリストで、変更するオブジェクトを右クリックし、[削除] を選択します。
- 4 確認ダイアログで [はい] をクリックします。

### ❖ TEXT インデックスの削除 (Interactive SQL)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 `DROP TEXT INDEX` 文を実行します。

たとえば、`MyTextIndex` という名前の TEXT インデックスを削除するには、次のように入力します。

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

## TEXT インデックスの更新

Sybase IQ テーブルの TEXT インデックスでサポートされる更新のタイプは即時更新のみです。即時更新は、基になるテーブルのデータが変更された場合に実行されます。Sybase IQ テーブルの TEXT インデックスの即時更新は、独立性レベル 3 をサポートしています。作成時、および `INSERT` 文、`UPDATE` 文、または `DELETE` 文を使用してカラムのデータが変更されるたびに値が設定されます。初期更新の間、テーブルに排他ロックが保持されます。

## TEXT インデックスのローの削除

`TEXT_DELETE_METHOD` データベース・オプションは、TEXT インデックスでの削除処理中に使用するアルゴリズムを指定します。

**TEXT\_DELETE\_METHOD オプション**

ファンクション TEXT インデックスの削除処理で使用されるアルゴリズムを指定します。

指定できる値 0 - 2

0 - 削除方法はコスト・モデルにより選択される。

1 - スモール・デリートが強制される。削除されるローの数が、テーブルの全ロー数に比べて非常に少ないときは、スモール・デリートが便利。スモール・デリートはインデックスにランダムにアクセス可能、大きいデータ・セットでキャッシュがスラッシングされる。

2 - ラージ・デリートが強制される。このアルゴリズムは、削除するロー検索のため全インデックスをスキャンする。削除されるローの数が、テーブルの全ロー数に比べてかなり多いときは、ラージ・デリートが便利。

デフォルト 0

スコープ このオプションを設定するために、DBA パーミッションは必要ありません。個々の接続または PUBLIC グループに一時的に設定することもできます。すぐに有効になります。

説明 このオプションは、TEXT インデックスの削除操作中に使用するアルゴリズムを指定するために使用します。このオプションを設定しないか 0 に設定した場合、削除方法はコスト・モデルにより選択されます。コスト・モデルは、適切な削除アルゴリズムを選択する際に、CPU 関連のコストと I/O 関連のコストを考慮します。コスト・モデルで考慮されるのは以下の要素です。

- 削除されたロー
- インデックス・サイズ
- インデックス・データ型の幅
- インデックス・データのカーディナリティ
- 利用可能なテンポラリ・キャッシュ
- マシンに関連する I/O と CPU の特性
- 利用可能な CPU とスレッド

例 TEXT インデックスからのラージ・デリートを強制するには、次のように入力します。

```
SET TEMPORARY OPTION TEXT_DELETE_METHOD = 2
```

参照 『パフォーマンス&チューニング・ガイド』の「第3章 クエリと削除の最適化」にある「削除オペレーションの最適化」を参照してください。

## テキスト設定オブジェクト

テキスト設定オブジェクトによって、TEXT インデックスの構築または更新時にインデックスに配置される単語と、全文クエリの解釈方法が制御されます。

データベース・サーバは、TEXT インデックスの作成時または更新時に、TEXT インデックスが作成されたときに指定されたテキスト設定オブジェクトの設定を使用します。テキスト設定オブジェクトが指定されていない場合、データベース・サーバは、インデックスが作成されるカラムのデータ型に基づいて、デフォルトのテキスト設定オブジェクトを選択します。Sybase IQ データベースでは、`default_char` テキスト設定オブジェクトが常に使用されます。

テキスト設定オブジェクトによって、インデックスを作成するドキュメントから単語を生成するために使用されるプレフィルタ・ライブラリと単語分割ライブラリが指定されます。テキスト設定オブジェクトでは、TEXT インデックス内に格納する単語の最小長と最大長、および除外する単語のリストを指定します。テキスト設定オブジェクトは次のパラメータで構成されます。

- **ドキュメント・プレフィルタ** フォーマットやイメージなどの不要な情報をドキュメントから削除する。その後、フィルタされたドキュメントが他のモジュールによって選択され、さらなる処理が行われる。ドキュメント・プレフィルタはサード・パーティ・ベンダによって提供される。
- **ドキュメント単語分割** 受信バイト・ストリームを、単語セパレータによって区切られた、または指定された規則に従って区切られた単語に分割する。ドキュメント単語分割は、サーバまたはサード・パーティ・ベンダによって提供される。
- **ストップリスト・プロセッサ** TEXT インデックスの構築中に無視する単語のリストを指定する。

## デフォルトのテキスト設定オブジェクト

Sybase IQ には、非 NCHAR データで使用するデフォルトのテキスト設定オブジェクト `default_char` が用意されています。この設定は、テキスト設定オブジェクトまたは TEXT インデックスを初めて作成するとときに作成されます。また、テキスト設定オブジェクト `default_nchar` では IN SYSTEM テーブルの TEXT インデックスの NCHAR での使用がサポートされています。Sybase IQ テーブルの TEXT インデックスに `default_nchar` テキスト設定を使用することはできません。

表 2-2 は、ほとんどの文字ベースの言語に最適な、`default_char` および `default_nchar` のデフォルト設定を示しています。デフォルトのテキスト設定オブジェクトの設定を変更しないことを強くおすすめします。

表 2-2: デフォルトのテキスト設定オブジェクトの設定

設定値	インストールされている値
TERM BREAKER	GENERIC
MINIMUM TERM LENGTH	1
MAXIMUM TERM LENGTH	20
STOPLIST	(空)

デフォルトのテキスト設定オブジェクトを削除した場合、次に TEXT インデックスまたはテキスト設定オブジェクトを作成したときに、自動的にデフォルト値で再作成されます。

## テキスト設定の作成

テキスト設定を作成し、テキスト設定プロセスに依存する TEXT インデックスがデータ内の単語を処理する方法を指定します。

### ❖ テキスト設定の作成 (Sybase Central)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 左側のウィンドウ枠で、[Text Configurations Objects] フォルダを右クリックし、[新規作成] > [Text Configurations Object] を選択します。
- 3 テキスト設定の名前を入力します。
- 4 テキスト設定の所有者を選択します。
- 5 テキスト設定のデータベース照合のタイプを選択します。[次へ] をクリックします。

**注意** NCHAR 照合を使用するテキスト設定は、Sybase IQ TEXT インデックスではサポートされません。

- 6 GENERIC 単語分割アルゴリズムを選択します。

**注意** Sybase IQ テキスト設定では、N-gram アルゴリズムはサポートされていません。詳細については、「[テキスト設定オブジェクトの設定](#)」(13 ページ) を参照してください。

- 7 単語の最小長と最大長を入力します。
- 8 外部の単語分割ライブラリを使用する場合は、[Use an external term breaker] を選択して、外部の単語分割関数およびライブラリを指定します。  
関数とライブラリを次の形式で指定します。  
function-name@library-file-name

- 9 [ 次へ ] をクリックします。
- 10 外部のプレフィルタ・ライブラリを使用する場合は、[Use an external prefilter] を選択して、外部のプレフィルタ関数およびライブラリを指定します。  
関数とライブラリを次の形式で指定します。  
function-name@library-file-name
- 11 このテキスト設定を使用して TEXT インデックスを構築するときに無視する単語をストップリストに追加します。単語をスペースで区切れます。  
このリストに含まれている単語は、クエリでも無視されます。
- 12 [ 次へ ] をクリックします。
- 13 テキスト設定について説明するコメントを入力し、[ 完了 ] をクリックします。

#### ❖ テキスト設定オブジェクトの作成 (Interactive SQL)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 CREATE TEXT CONFIGURATION 文を実行します。

たとえば、`default_char` テキスト設定オブジェクトをテンプレートとして使用して `myTxtConfig` というテキスト設定オブジェクトを作成するには、次のように入力します。

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

## テキスト設定オブジェクトの設定

次の表では、テキスト設定オブジェクトの設定と、それらがインデックスの作成対象にどのような影響を与えるか、および全文検索クエリがどのように解釈されるかを説明しています。テキスト設定オブジェクトと、それらが TEXT インデックスおよび全文検索に与える影響の例については、「[テキスト設定オブジェクトの設定の解釈](#)」(19 ページ) を参照してください。

**単語分割アルゴリズム (TERM BREAKER)** TERM BREAKER 設定では、文字列を単語に分割するために使用するアルゴリズムを指定します。Sybase IQ では、単語の格納に関して GENERIC (デフォルト) がサポートされています。

---

**注意** NGRAM 単語分割による n-gram (n-gram は長さ  $n$  の一連の文字であり、 $n$  は MAXIMUM TERM LENGTH の値です) の格納がサポートされるのは IN SYSTEM テーブルでのみであり、Sybase IQ TEXT インデックスでは使用できません。

---

指定する単語分割にかかわらず、データベース・サーバは、単語が TEXT インデックスに挿入されるときに、単語の元の位置情報を TEXT インデックスに記録します。n-gram の場合は、元の単語の位置情報ではなく、n-gram の位置情報が格納されます。

表 2-3: TERM BREAKER の影響

TEXT インデックスに対して	クエリ単語に対して
<b>GENERIC TEXT インデックス</b> GENERIC TEXT インデックス (デフォルト) を構築する場合、英数字以外の文字の間の一連の英数字は、データベース・サーバによって単語として処理されます。単語の定義後に、単語の長さの設定を超える単語と、ストップリストに含まれている単語は、カウントはされますが、TEXT インデックスには挿入されません。 GENERIC TEXT インデックスにおけるパフォーマンスは、NGRAM TEXT インデックスよりも高速である場合があります。ただし、GENERIC TEXT インデックスではファジー検索は実行できません。	<b>GENERIC TEXT インデックス</b> GENERIC TEXT インデックスに対してクエリする場合、クエリ文字列内の単語は、インデックスが作成される場合と同じように処理されます。クエリ単語と TEXT インデックスに含まれる単語を比較して照合が実行されます。
<b>NGRAM TEXT インデックス*</b> NGRAM TEXT インデックスを構築する場合、英数字以外の文字の間の一連の英数字は、データベース・サーバによって1つの単語として処理されます。単語が定義されると、データベース・サーバが単語を n-gram に分割します。こうすることで、n よりも短い単語と、ストップリストに含まれている n-gram は破棄されます。 たとえば、MAXIMUM TERM LENGTH が 3 の NGRAM TEXT インデックスの場合、文字列 'my red table' は、TEXT インデックスでは、red、tab、abl、ble の n-gram として表されます。	<b>NGRAM TEXT インデックス*</b> NGRAM TEXT インデックスに対してクエリする場合、クエリ文字列内の単語は、インデックスが作成される場合と同じように処理されます。クエリ単語の n-gram とインデックスが付けられた単語の n-gram を比較して照合が実行されます。

\*NGRAM TEXT インデックスは、IN SYSTEM テーブルでのみサポートされます。

**単語の最小長の設定 (MINIMUM TERM LENGTH)** MINIMUM TERM LENGTH 設定で、インデックスに挿入される、または全文クエリで検索される単語の最小長 (文字数) を指定します。MINIMUM TERM LENGTH は、NGRAM TEXT インデックスには関係しません。

MINIMUM TERM LENGTH は、特にプレフィクス検索に関係します。MINIMUM TERM LENGTH の値は 0 よりも大きくする必要があります。MINIMUM TERM LENGTH に MAXIMUM TERM LENGTH よりも大きい値を設定した場合、MAXIMUM TERM LENGTH は、MINIMUM TERM LENGTH と等しい値に自動的に調整されます。

MINIMUM TERM LENGTH のデフォルト値は、デフォルトのテキスト設定オブジェクトの設定から取得されます。通常は 1 です。

表 2-4: MINIMUM TERM LENGTH の影響

TEXT インデックスに対して	クエリ単語に対して
<b>GENERIC TEXT インデックス</b> GENERIC TEXT インデックスの場合、TEXT インデックスには、MINIMUM TERM LENGTH よりも短い単語は格納されません。	<b>GENERIC TEXT インデックス</b> GENERIC TEXT インデックスに対してクエリする場合、MINIMUM TERM LENGTH よりも短いクエリ単語は、TEXT インデックスに存在している可能性がないため無視されます。
<b>NGRAM TEXT インデックス *</b> NGRAM TEXT インデックスの場合、この設定は無視されます。	<b>NGRAM TEXT インデックス *</b> MINIMUM TERM LENGTH 設定は、NGRAM TEXT インデックスに対する全文クエリには影響しません。

\*NGRAM TEXT インデックスは、IN SYSTEM テーブルでのみサポートされます。

#### 単語の最大長の設定 (MAXIMUM TERM LENGTH) MAXIMUM TERM

LENGTH 設定は、単語分割アルゴリズムに応じて異なります。MAXIMUM TERM LENGTH の値は 60 以下である必要があります。MAXIMUM TERM LENGTH に MINIMUM TERM LENGTH よりも小さい値を設定した場合、MINIMUM TERM LENGTH は、MAXIMUM TERM LENGTH と等しい値に自動的に調整されます。

この設定のデフォルト値は、デフォルトのテキスト設定オブジェクトの設定から取得されます。通常は 20 です。

表 2-5: MAXIMUM TERM LENGTH の影響

TEXT インデックスに対して	クエリ単語に対して
<b>GENERIC TEXT インデックス</b> GENERIC TEXT インデックスの場合、MAXIMUM TERM LENGTH は、TEXT インデックスに挿入される単語の最大長 (文字数) を指定します。	<b>GENERIC TEXT インデックス</b> GENERIC TEXT インデックスの場合、MAXIMUM TERM LENGTH よりも長いクエリ単語は、TEXT インデックスに存在している可能性がないため無視されます。
<b>NGRAM TEXT インデックス *</b> NGRAM TEXT インデックスの場合、MAXIMUM TERM LENGTH によって、単語が分割される n-gram の長さが決まります。MAXIMUM TERM LENGTH の適切な長さは、言語によって異なります。一般的な値は、英語の場合は 4 または 5 文字、中国語の場合は 2 または 3 文字です。	<b>NGRAM TEXT インデックス *</b> NGRAM TEXT インデックスの場合、クエリ単語は長さ n の n-gram に分割されます。n は MAXIMUM TERM LENGTH と同じです。データベース・サーバは、n-gram を使用して TEXT インデックスを検索します。MAXIMUM TERM LENGTH よりも短い単語は、TEXT インデックス内の n-gram と一致しないため、無視されます。

\*NGRAM TEXT インデックスは、IN SYSTEM テーブルでのみサポートされます。

**ストップリストの設定 (STOPLIST)** ストップリストの設定では、インデックスを作成しない単語を指定します。この設定のデフォルト値は、デフォルトのテキスト設定オブジェクトの設定から取得されます。通常は、ストップリストは空です。

表 2-6: STOPLIST の影響

TEXT インデックスに対して	クエリ単語に対して
<b>GENERIC TEXT インデックス</b> GENERIC TEXT インデックスの場合、ストップリストに含まれる単語は TEXT インデックスに挿入されません。	<b>GENERIC TEXT インデックス</b> GENERIC TEXT インデックスの場合、ストップリストに含まれるクエリ単語は、TEXT インデックスに存在している可能性がないため無視されます。
<b>NGRAM TEXT インデックス*</b> NGRAM TEXT インデックスの場合、TEXT インデックスには、ストップリストに含まれている単語から形成された n-gram は格納されません。	<b>NGRAM TEXT インデックス*</b> ストップリストに含まれる単語は n-gram に分割され、n-gram がストップリスト用に使用されます。同様に、クエリ単語は n-gram に分割され、ストップリストに含まれる n-gram と一致する n-gram は、TEXT インデックスに存在している可能性が無いため削除されます。

\*NGRAM TEXT インデックスは、IN SYSTEM テーブルでのみサポートされます。

単語をストップリストに含めるかどうかは、慎重に検討してください。特に、アポストロフィやダッシュなど、英数字以外の文字を含む単語は含めないでください。これらの文字は、単語の分割記号として機能します。たとえば、you'll という単語 ('you'll' と指定する必要があります) は、you と ll に分割され、2つの単語としてストップリストに格納されます。以降の 'you' または 'they'll' の全文検索が悪影響を受けます。

NGRAM TEXT インデックスのストップリストによって、予期しない結果が生じる場合があります。これは、格納されるストップリストが、実際には n-gram 形式であり、指定したストップリストの単語ではないためです。たとえば、MAXIMUM TERM LENGTH が 3 である NGRAM TEXT インデックスでは、STOPLIST に 'there' と指定した場合、the her ere の n-gram がストップリストとして格納されます。これは、the、her、ere という n-gram を含む単語をクエリする能力に影響します。

## テキスト設定のリストの表示

データベースのすべてのテキスト設定のリストを表示します。

### ❖ テキスト設定のリストの表示 (Sybase Central)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
  - 2 左側のウィンドウ枠で、[Text Configurations Objects] を選択します。
- すべてのテキスト設定のリストが右側のウィンドウ枠に表示されます。

❖ テキスト設定のリストの表示 (Interactive SQL)

1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。

2 SELECT 文を実行します。

たとえば、すべてのテキスト設定オブジェクトを表示するには、次の構文を使用します。

```
SELECT * FROM SYSTEXTCONFIG;
```

## テキスト設定の変更

DB 領域や、単語で許可する長さの範囲などの、テキスト設定オブジェクトの設定を変更します。TEXT インデックスで使用されていないテキスト設定オブジェクトのみを変更できます。

❖ テキスト設定オブジェクトの変更 (Sybase Central)

1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。

2 左側のウィンドウ枠で、[Text Configurations Objects] を選択します。

3 [Text Configurations] のリストで、変更するオブジェクトを右クリックし、[プロパティ] を選択します。

4 [設定] タブに切り替え、必要に応じて設定を変更します。

5 [OK] をクリックします。

❖ テキスト設定オブジェクトの変更 (Interactive SQL)

1 DBA または RESOURCE 権限のあるユーザとして、またはテキスト設定オブジェクトの所有者としてデータベースに接続します。

2 ALTER TEXT CONFIGURATION 文を実行します。

たとえば、myTxtConfig テキスト設定オブジェクトの単語の最小長を変更するには、次の構文を使用します。

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

## ストップリストの変更

ストップリストには、このテキスト設定を使用して TEXT インデックスを構築するときに無視する単語のリストを格納します。TEXT インデックスで使用されていないテキスト設定オブジェクトのみを変更できます。

### ❖ ストップリストの変更 (Sybase Central)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 左側のウィンドウ枠で、[Text Configurations Objects] を選択します。
- 3 [Text Configurations] のリストで、変更するオブジェクトを右クリックし、[プロパティ] を選択します。
- 4 [Stoplist] タブに切り替えて、必要に応じてストップリストの単語を変更します。スペースを使用して単語を区切ります。
- 5 ストップリストの単語のリストをアルファベット順にソートし、それらをリストに表示するには、[Sort Terms] をクリックします。
- 6 ストップリストが更新されたら、[OK] をクリックします。

### ❖ ストップリストの変更 (Interactive SQL)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 STOPLIST 句を含む ALTER TEXT CONFIGURATION 文を実行します。

たとえば、myTxtConfig 設定オブジェクトにストップリストを追加するには、次の構文を使用します。

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

## テキスト設定の削除

不要なテキスト設定をデータベースから削除します。TEXT インデックスで使用されていないテキスト設定のみを削除できます。

### ❖ テキスト設定の削除 (Sybase Central)

- 1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。
- 2 左側のウィンドウ枠で、[Text Configurations Objects] を選択します。
- 3 [Text Configurations] のリストで、変更するオブジェクトを右クリックし、[削除] を選択します。
- 4 確認ダイアログで [はい] をクリックします。

❖ テキスト設定の削除 (Interactive SQL)

1 DBA または RESOURCE 権限のあるユーザとしてデータベースに接続します。

2 DROP TEXT CONFIGURATION 文を実行します。

たとえば、`myTxtConfig` というテキスト設定オブジェクトを削除するには、次の構文を使用します。

```
DROP TEXT CONFIGURATION myTxtConfig;
```

## テキスト設定オブジェクトの例

サンプルを確認して、テキスト設定オブジェクトの設定が TEXT インデックスにどのように影響するか、また、インデックスがどのように解釈されるかを理解します。

### テキスト設定オブジェクトの設定の解釈

表 2-7 では、さまざまなテキスト設定オブジェクトの設定と、それらの設定がインデックスの作成対象にどのような影響を与えるか、および全文検索クエリ文字列がどのように解釈されるかを説明しています。すべての例で、文字列 'I'm not sure I understand' を使用しています。

表 2-7: テキスト設定オブジェクトの設定の解釈

設定	インデックスが作成される単語	クエリの解釈
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'

テキスト設定オブジェクトの **CONTAINS** クエリ文字列の解釈

表 2-8 では、テキスト設定オブジェクトの文字列の設定がどのように解釈されるかを、例を挙げて示しています。

「文字列の解釈」列のカッコで囲まれた数値は、単語ごとに格納される位置情報を示しています。数値は、マニュアルで説明するためのものです。格納される実際の単語には、カッコで囲まれた数値は含まれません。

**注意** テキスト・ドキュメントの位置情報の最大数は 4294967295 です。

この表に示すのは、**CONTAINS** クエリの解釈のみです。データが解析されるときに、AND、NOT、NEAR は、通常のトークンと見なされます。また、\*、|などの記号は、英数字ではないため、削除されます。

表 2-8: **CONTAINS** 文字列の解釈

設定	文字列	文字列の解釈
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)"'

## テキスト検索でのプレフィクス単語の制限

**MAX\_PREFIX\_PER\_CONTAINS\_PHRASE** データベース・オプションでは、テキスト検索式で許可するプレフィクス単語の数を指定します。

### MAX\_PREFIX\_PER\_CONTAINS\_PHRASE オプション

ファンクション	テキスト検索式で許可するプレフィクス単語の数を指定します。
指定できる値	0 – 300 0 – 検索フレーズでプレフィクス単語を制限しない。 300 – 上限 ( これは、フレーズで許可する単語の合計数の総合限度 )
デフォルト	1
スコープ	このオプションを設定するために、DBA パーミッションは必要ありません。個々の接続または PUBLIC グループに一時的に設定することもできます。すぐには有効になります。
説明	MAX_PREFIX_PER_CONTAINS_PHRASE では、テキスト検索式での一定数以上のプレフィクスを禁止するためのしきい値を指定します。このオプションを 0 に設定した場合、プレフィクスはいくつでも許可されます。クエリに CONTAINS 式が存在し、フレーズに含まれているプレフィクス単語の数がこのオプションで指定された数を超えている場合、Sybase IQ によって検出され、エラーが報告されます。
例	以下のデフォルトの MAX_PREFIX_PER_CONTAINS_PHRASE 設定を使用した場合、  SET MAX_PREFIX_PER_CONTAINS_PHRASE = 1  次の CONTAINS 句は有効です。  SELECT ch1 FROM tab1 WHERE CONTAINS(ch1, '"concord bed*' in mass")  デフォルトの MAX_PREFIX_PER_CONTAINS_PHRASE 設定を使用した場合、次の CONTAINS 句では、構文エラーが返されます。  SELECT ch1 FROM tab1 WHERE CONTAINS (ch1, '"con* bed* in mass")  MAX_PREFIX_PER_CONTAINS_PHRASE を 0 ( 制限なし ) または 2 に設定した場合は、上記の CONTAINS 句は有効です。



## 外部ライブラリ

この章では、外部ライブラリを使用して、ドキュメントでプレフィルタおよび単語分割を提供する方法について説明します。

トピック	ページ
外部ライブラリについて	23
マルチプレックス・サーバでの外部ライブラリの使用	25
起動時の外部ライブラリの有効化と無効化	25
外部ライブラリのアンロード	25

### 外部ライブラリについて

Sybase IQ では、C または C++ で記述された外部のプレフィルタ・ライブラリおよび単語分割ライブラリを使用して、インデックスの作成中またはクエリの処理中にドキュメントをプレフィルタしてトークン化できます。これらのライブラリは、データベース・サーバのプロセス領域に動的にロードできます。

---

**注意** 外部のプレフィルタ・ライブラリおよび単語分割ライブラリは、Sybase 認定パートナーから提供を受ける必要があります。認定ベンダ・ソリューションについては、Partner Certification Reports web site にアクセスし、認定レポートをフィルタ処理して、Sybase IQ に関する認定を表示してください。([http://www.sybase.com/detail\\_list?id=9784](http://www.sybase.com/detail_list?id=9784))

---

外部の動的にロード可能なプレフィルタ・ライブラリおよび単語分割ライブラリは、テキスト設定で指定します。また、これらのライブラリは、データベース・サーバによってロードされる必要があります。各ライブラリには、テキスト設定オブジェクトで指定されている外部関数を実装するエクスポートされた記号が含まれています。この関数は、呼び出し元が必要なタスクを実行するために使用する一連の関数記述子を返します。

外部のプレフィルタ・ライブラリおよび単語分割ライブラリは、特定のカラムに対する、ライブラリのロードを必要とするクエリが受信されたとき、または TEXT を更新する必要があるときに、最初の CREATE TEXT INDEX 要求で、データベース・サーバによってロードされます。

ライブラリは、ALTER TEXT CONFIGURATION呼び出しが行われたときにはロードされません。また、DROP TEXT CONFIGURATION呼び出しが行われたときに、自動的にアンロードされるわけではありません。サーバが、外部ライブラリのロードを禁止するオプションを使用して起動された場合は、外部のプレフィルタ・ライブラリおよび単語分割ライブラリはロードされません。

これらの外部 C/C++ ライブラリは、サーバのプロセス領域への非サーバ・ライブラリ・コードのロードを行うので、関数の記述が不完全な場合や意図的に不正な場合、データの整合性やセキュリティ、およびサーバの堅牢性に関するリスクが発生する可能性があります。これらのリスクを管理するために、Sybase IQ サーバごとに、明示的にこの機能を有効または無効にできます。詳細については、「[起動時の外部ライブラリの有効化と無効化](#) (25 ページ)」を参照してください。

ISYSTEEXTCONFIG システム・テーブルには、テキスト設定オブジェクトに関する、外部ライブラリについての情報が格納されています。『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第 8 章 システム・テーブルとシステム・ビュー」にある「SYSTEEXTCONFIG システム・ビュー」を参照してください。

## 外部ライブラリの制限

外部ライブラリを使用する Sybase IQ テキスト設定オブジェクトおよび TEXT インデックスには、次の制限があります。

- バイナリ・カラムの TEXT インデックスの場合、外部ベンダによって提供された外部ライブラリを使用してドキュメント変換を行う必要がある。Sybase IQ は、バイナリ・カラムに格納されているドキュメントを暗黙的に変換しない。
- 外部の単語分割を使用してドキュメントをトーカン化する場合、n-gram ベースのテキスト検索はサポートされない。
- 外部ライブラリを使用して SQL Anywhere テーブルに TEXT インデックスを作成することはできない。実行した場合、エラーが発生する。

## マルチプレックス・サーバでの外部ライブラリの使用

すべてのマルチプレックス・サーバが、外部のプレフィルタ・ライブラリと単語分割ライブラリにアクセスできる必要があります。ユーザは、外部の各プレフィルタ・ライブラリと各単語分割ライブラリが、マルチプレックス・サーバをホストしているマシンにコピーされており、しかもサーバによってライブラリをロードできる場所に配置されていることを確認する必要があります。

外部のプレフィルタおよび単語分割の呼び出し時に、各マルチプレックス・サーバは、他のサーバとは独立して動作します。各プロセス領域では、外部ライブラリがロードされ、独自に実行できます。プレフィルタ関数および単語分割関数は、各サーバで同様に実装され、同じ結果が返されることが前提となります。

あるサーバのプロセス領域から外部ライブラリをアンロードしても、他のサーバのプロセス領域からライブラリがアンロードされるわけではありません。

## 起動時の外部ライブラリの有効化と無効化

Sybase IQ は、外部のサードパーティ・ライブラリのロードを有効または無効にする起動スイッチを提供しています。このスイッチは、サーバ起動コマンド・ラインまたはサーバ設定ファイルで指定できます。

外部のサードパーティ・ライブラリのロードを有効にする場合

```
-sf -external_library_full_text
```

外部のサードパーティ・ライブラリのロードを無効にする場合

```
-sf external_library_full_text
```

現在サーバにロードされているライブラリを一覧表示するには、`sa_list_external_library` ストアド・プロシージャを使用します。

## 外部ライブラリのアンロード

ライブラリを使用しなくなった場合は、`dbo.sa_external_library_unload` システム・プロシージャを使用して外部ライブラリをアンロードします。このプロシージャは、`long varchar` 型のオプション・パラメータを 1 つ受け取ります。このパラメータでは、アンロードするライブラリの名前を指定します。パラメータを指定しない場合、使用されていないすべての外部ライブラリがアンロードされます。

次の例では、外部関数ライブラリがアンロードされます。

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



## 非構造化データのクエリ

この章では、非構造化データと半構造化データを処理する全文検索機能を含む、ラージ・オブジェクト・データのクエリについて説明します。

トピック	ページ
全文検索	27
LONG BINARY カラムのクエリ	36
LONG VARCHAR カラムのクエリ	36
LONG BINARY カラムと LONG VARCHAR カラムのパフォーマンスのモニタリング	37

### 全文検索

全文検索では、テーブルのローをスキャンせずに、データベース内の単語（ワード）のすべてのインスタンスをすばやく見つけることができます。全文検索では、インデックス・カラム内の単語の位置情報を格納する TEXT インデックスを使用します。インデックスを使用して、単語が含まれるローを検索する方法は、テーブル内の各ローをスキャンする方法よりも高速です。

全文検索では、CONTAINS 検索条件を使用します。これは、一致がパターンベースではなく単語ベースであるため、LIKE、REGEXP、SIMILAR TO などの述部を使用する検索とは異なります。『CONTAINS 条件』(28 ページ) および『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第2章 SQL 言語の要素」の「CONTAINS 条件」を参照してください。

全文検索の文字列比較では、データベースのすべての標準照合設定が使用されます。たとえば、大文字と小文字を区別しないようにデータベースを設定すると、全文検索でも大文字と小文字は区別されません。

### 全文検索のタイプ

全文検索を使用して、単語、プレフィックス、語句（一連の単語）を検索できます。複数の単語、語句、またはプレフィックスをブール式に組み合わせたり、近接検索を使用して、式が互いに近接していることを求めたりできます。

WHERE 句または SELECT 文の FROM 句で CONTAINS 句を使用して全文検索を実行します。

**FROM 句**

説明

SELECT 文に必要なデータベース・テーブルまたはビューを指定します。

構文

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

パラメータ

*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* ]

使用法

**contains-expression** テーブル名の後に CONTAINS 句を使用してテーブルをフィルタし、*contains-query* で指定した全文クエリに一致するローのみを返します。テーブルの一致するすべてのローが、*score-correlation-name* ( 指定されている場合 ) を使用して参照できる score カラムとともに返されます。*score-correlation-name* が指定されていない場合は、デフォルトの相関名 *contains* で score カラムを参照できます。

オプションの相関名の引数を例外として、CONTAINS 句は、CONTAINS 検索条件と同じ引数を取ります。CONTAINS 句でリストされているカラムには TEXT インデックスがある必要があります。

「CONTAINS 条件」(28 ページ) および『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第 2 章 SQL 言語の要素」にある「CONTAINS 条件」を参照してください。

FROM 句の完全な構文と説明については、『リファレンス：文とオプション』の「第 1 章 SQL 文」にある「FROM 句」を参照してください。

**CONTAINS 条件**

SELECT 文の FROM 句で CONTAINS 句を使用して、または、WHERE 句で CONTAINS 検索条件(述部)を使用して、全文クエリを実行します。どちらの方法でも同じローが返されますが、CONTAINS 句では一致するローのスコアも返されます。

構文

**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

```

## パラメータ

- **simple-term** スペースと特殊文字で区切られた文字列。これは、検索対象である、単一の、インデックスが付けられた単語(ワード)を表します。
- **distance** 正の整数。
- **and-expression** *and-expression* を使用して、TEXT インデックスで *primary-expression* と *simple-expression* の両方が見つかる必要があることを指定します。デフォルトでは、単語または式の間に演算子が指定されていない場合、*and-expression* と見なされます。たとえば、'a b' は 'a AND b' と解釈されます。AND の代わりにアンパサンド(&)を使用して、両側に式または単語を隣接させることができます('a & b' など)。
- **and-not-expression** *and-not-expression* を使用して、*primary-expression* は TEXT インデックスに存在するが、*basic-expression* は TEXT インデックスで見つからないことが条件であることを示します。これは、符号反転とも呼ばれます。符号反転にハイフンを使用する場合は、ハイフンの前にスペースを挿入する必要があります、ハイフンがそれに続く単語に隣接している必要があります。たとえば、'a -b' は 'a AND NOT b' と同じですが、'a - b' の場合は、ハイフンが無視され、文字列は 'a AND b' と同じになります。'a-b' は、'"a b"' というフレーズと同じです。

- **or-expression** *or-expression* を使用して、少なくとも *simple-expression* または *contains-query-string* のいずれかが TEXT インデックスに存在する必要があることを指定します。たとえば、'a|b' は 'a OR b' と解釈されます。
- **fuzzy-expression** *fuzzy-expression* を使用して、指定した単語と似た単語を見つけます。ファジー一致は、IN SYSTEM テーブルの NGRAM TEXT インデックスでのみサポートされています。
- **proximity-expression** *proximity-expression* を使用して、互いに近くにある単語を検索します。たとえば、'b NEAR[2,5] c' は、互いに単語 2 ~ 5 個分離している b と c のインスタンスを検索します。単語の順序は重要ではありません。'b NEAR c' は 'c NEAR b' と同じです。距離を指定せずに NEAR を指定した場合、デフォルトの単語 10 個分が適用されます。NEAR の代わりに波型記号 (~) を指定できます。これは、距離を指定せずに NEAR を指定する場合と同じであるため、デフォルトの単語 10 個分の距離が適用されます。NEAR 式をつなげる ('a NEAR[1] b NEAR[1] c' など) ことはできません。

BEFORE は、単語の順序が重要であることを除き、NEAR に似ています。'b BEFORE c' は 'c BEFORE b' と同じではありません。前者の場合、単語 'b' が 'c' よりも前にある必要があり、後者の場合、単語 'b' が 'c' よりも後にある必要があります。BEFORE は、NEAR と同様に最小距離と最大距離の両方を受け入れます。デフォルトの最小距離は 1 です。最小距離を指定する場合は、最大距離以下にする必要があります。そうしないと、エラーが返されます。

- **prefix-term** *prefix-term* を使用して、指定されたプレフィックスで始まる単語を検索します。たとえば、'datab\*' では、datab で始まる単語が検索されます。これは、プレフィックス検索とも呼ばれます。プレフィックス検索では、アスタリスクの左側の部分にあたる単語の一部について照合が行われます。

#### Remarks

CONTAINS 検索条件は、カラム・リストと *contains-query-string* を引数として使用します。検索条件（述部とも呼ばれる）を指定できる任意の場所で使用でき、TRUE または FALSE が返されます。*contains-query-string* は、定数文字列か、クエリ時に既知である値を持つ変数である必要があります。

複数のカラムを指定する場合は、それらすべてが单一のベース・テーブルを参照している必要があります。TEXT インデックスは、複数のベース・テーブルにまたがることはできません。ベース・テーブルは、FROM 句で直接参照するか、ビューまたは抽出テーブルで使用できます。ただし、これは、ビューまたは抽出テーブルで DISTINCT、GROUP BY、ORDER BY、UNION、INTERSECT、EXCEPT、ローの制限が使用されていない場合に限ります。

ANSI ジョイン構文 (FULL OUTER JOIN, RIGHT OUTER JOIN, LEFT OUTER JOIN) を使用したクエリがサポートされますが、次善最適なパフォーマンスしか得られない場合があります。FROM 句で CONTAINS に外部ジョインを使用するのは、各 CONTAINS 句の score カラムが必要な場合のみにしてください。それ以外の場合は、CONTAINS を ON 条件または WHERE 句に移動してください。

次のタイプのクエリはサポートされていません。

- リモート・テーブルにジョインされている、フル TEXT インデックスを持つ SQL Anywhere テーブルを使用したリモート・クエリ。
- 使用されるフル TEXT インデックスが SQL Anywhere テーブルにある場合の、Sybase IQ および SQL Anywhere テーブルを使用したクエリ。
- TSQL スタイルの外部ジョイン構文(\*=\*、=\* および \*=)を使用したクエリ。

長さが 32KB 未満の SQL 変数を検索単語として使用し、変数のデータ型が LONG VARCHAR の場合は、CAST を使用して変数を VARCHAR データ型に変換します。例を示します。

```
SELECT * FROM tab1 WHERE CONTAINS(c1, cast(v1 AS VARCHAR(64)))
```

クエリ文字列で英数字以外の文字を使用すると、次の警告が適用されます。

- 単語の途中でアスタリスクを使用するとエラーが返される。
- fuzzy-expression での英数字以外の文字(特殊文字を含む)の使用は避ける。これらの文字はスペースとして処理され、単語の分割記号として機能するため。
- 可能な場合は、クエリ文字列での特殊文字ではない英数字以外の文字の使用は避ける。特殊文字ではない英数字以外の文字を使用すると、それらの文字の場所で単語が分割され、それらを含む単語が語句として処理される。たとえば、'things we've done' は 'things "we ve" done' と解釈される。

語句内では、アスタリスクが、特殊文字としてそのまま解釈される唯一の特殊文字である。語句内のその他のすべての特殊文字は、スペースとして処理され、単語の分割記号として機能する。

*contains-query-string* の解釈は、2 つの主要な段階を経て実行されます。

- 手順 1：演算子と優先度を解釈する** この段階では、キーワードが演算子として解釈され、優先度の規則が適用される。
- 手順 2：テキスト設定オブジェクトの設定を適用する** この段階では、テキスト設定オブジェクトの設定が単語に適用される。単語の長さの設定を超えているか、またはストップ・リストに含まれているクエリ単語は削除される。

CONTAINS 検索条件での演算子の優先度

クエリの評価中、式は、次の優先順位を使用して評価されます。

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

## アスタリスク (\*) の許可される構文

アスタリスクは、プレフィックス検索に使用します。アスタリスクは、クエリ文字列の最後に配置するか、その後ろにスペース、アンパンド、縦線、終了カッコ、終了引用符を続けることができます。その他の方法でアスタリスクを使用するとエラーが返されます。

表 4-1 は、アスタリスクの許可される使用方法を示しています。

表 4-1: アスタリスクの解釈

クエリ文字列	同等のクエリ文字列	解釈
'th*&best'	'th* AND best' OR 'th* best'	th で始まる単語、および単語 best が含まれるものを検索する。
'th* best'	'th* OR best'	th で始まる単語、または単語 best が含まれるものを検索する。
'very&(best th*)'	'very AND (best OR th*)'	単語 very、および th で始まる単語または単語 best が含まれるものを検索する。
""fast auto*""		単語 fast の直後に auto で始まる単語が続く語句が含まれるものを検索する。
""auto* price""		auto で始まる単語の直後に単語 price が続く語句が含まれるものを検索する。

**注意** アスタリスクを含むクエリ文字列の解釈は、テキスト設定オブジェクトの設定によって異なります。

## ハイフン (-) の許可される構文

ハイフンは、単語または式の反転に使用でき、NOT と同等に機能します。ハイフンが反転として解釈されるかどうかは、クエリ文字列のどこにあるかによって異なります。たとえば、ハイフンを単語または式の直前に配置した場合は、反転と解釈されます。ハイフンを単語内に配置した場合は、ハイフンとして解釈されます。

ハイフンを反転に使用する場合は、ハイフンの前にスペースを挿入し、ハイフンの直後に式を配置する必要があります。

ファジー式の語句内で使用した場合、ハイフンはスペースとして処理され、単語の分割記号として機能します。

表 4-2 は、ハイフンの許可される構文を示しています。

表 4-2: ハイフンの解釈

クエリ文字列	同等のクエリ文字列	解釈
'the -best'	'the AND NOT best', 'the AND -best', 'the & -best', 'the NOT best'	単語 the を含み、単語 best を含まないものを検索する。
'the -(very best)'	'the AND NOT (very AND best)'	単語 the を含み、単語 very および best を含まないものを検索する。
'the -"very best"'	'the AND NOT "very best"'	単語 the を含み、語句 very best を含まないものを検索する。

クエリ文字列	同等のクエリ文字列	解釈
'alpha-numerics'	""alpha numerics""	単語 alpha の直後に単語 numerics が続く語句が含まれるものを探索する。
'wild - west'	'wild west'、OR 'wild AND west'	単語 wild、および単語 west が含まれるものを探索する。

特殊文字の許可される構文

**表 4-3** は、アスタリスクとハイフンを除くすべての特殊文字の許可される構文を示しています。これらの文字は、語句内で使用した場合は特殊文字とは見なされず、削除されます。

**注意** クエリ文字列には、文字列リテラルを指定する場合の制限も適用されます。たとえば、アポストロフィは、エスケープ・シーケンス内にある必要があります。

表 4-3: 特殊文字の解釈

文字または構文	使用例および備考
アンパサンド (&)	アンパサンドは AND と同じであり、次のように指定できる。 <ul style="list-style-type: none"> <li>• 'a &amp; b'</li> <li>• 'a &amp;b'</li> <li>• 'a&amp; b'</li> <li>• 'a&amp;b'</li> </ul>
縦線 ( )	縦線は OR と同じであり、次のように指定できる。 <ul style="list-style-type: none"> <li>• 'a  b'</li> <li>• 'a  b'</li> <li>• 'a   b'</li> <li>• 'a b'</li> </ul>
二重引用符 ("")	二重引用符を使用して、順序と相対距離が重要な一連の単語を囲む。たとえば、クエリ文字列 'learn "full text search"' では、"full text search" が語句である。この例では、learn は語句の前または後にあっても、別のカラムに存在していても (TEXT インデックスが複数のカラムに構築されている場合) 構わないが、このとおりの語句が単一のカラムに存在している必要がある。
カッコ ()	カッコを使用して、式の評価の順序を指定する ( デフォルトの順序と異なる場合 )。たとえば、'a AND (b c)' は、a、および b または c と解釈される。
波型記号 (~)	波型記号は NEAR と同じであり、特別な構文規則はない。クエリ文字列 'full~text' は 'full NEAR text' と同じであり、単語 text から 10 単語内の範囲にある単語 full と解釈される。
角カッコ []	キーワード NEAR と組み合わせて角カッコを使用して、距離を囲む。その他の方法で角カッコを使用するとエラーが返される。

## 削除した単語の影響

TEXT インデックスは、TEXT インデックスの作成に使用されるテキスト設定オブジェクトに定義されている設定に従って構築されます。TEXT インデックスは、次のいずれかの条件に当てはまる単語を除外します。

- 単語がストップ・リストに含まれている。
- 単語が単語の最小長よりも短い (GENERIC のみ)。
- 単語が単語の最大長よりも長い。

同じ規則がクエリ文字列に適用されます。削除される単語は、語句の先頭または最後の 0 個以上の単語が合致する場合があります。たとえば、単語 'the' がストップ・リストに含まれているとします。

- この単語が AND、OR、または NEAR のいずれかの側にある場合、演算子と単語の両方が削除されます。たとえば、'the AND apple'、'the OR apple'、または 'the NEAR apple' を検索することは、'apple' を検索することと同じです。
- この単語が AND NOT の右側にある場合、AND NOT と単語の両方が削除されます。たとえば、'apple AND NOT the' を検索することは、'apple' を検索することと同じです。
- この単語が AND NOT の左側にある場合、式全体が削除されます。たとえば、'the AND NOT apple' を検索した場合、ローは返されません。別の例を示します。'orange and the AND NOT apple' は 'orange AND (the AND NOT apple)' と同じであり、これは、AND NOT 式が削除されて、'orange' を検索することと同じになります。これを検索式 '(orange and the) and not apple' と対比してみてください。この式は、'orange and not apple' を検索する場合と同じです。
- この単語が語句に含まれる場合、語句は、削除される単語の位置にどのような単語があっても一致します。たとえば、'feed the dog' の検索では、'feed the dog'、'feed my dog'、'feed any dog' などと一致します。

---

**注意** 検索対象のすべての単語が削除される場合、Sybase IQ は、エラー CONTAINS has NULL search term を返します。SQL Anywhere では、エラーは報告されず、ローは返されません。

---

## CONTAINS テーブル式

クエリの FROM 句に CONTAINS 句を含めると、それぞれの一致にスコアが関連付けられます。スコアは、一致の近さを示し、スコア情報を使用してデータをソートできます。次の 2 つの主な条件によってスコアが決定されます。

- インデックス・ローにおける単語の出現回数。インデックス・ローにおける単語の出現回数が多いほど、そのスコアは高くなる。
- TEXT インデックスにおける単語の出現回数。TEXT インデックスにおける単語の出現回数が多いほど、そのスコアは低くなる。

全文検索の種類によっては、その他の条件がスコアに影響します。たとえば、近接検索では、検索単語の近接性がスコアに影響します。デフォルトでは、CONTAINS 句の結果セットは相関名 `contains` を持ち、これには、`score` という単一のカラムが含まれます。SELECT リスト、ORDER BY 句、またはクエリのその他の部分で `"contains".score` を参照できます。ただし、`contains` は SQL の予約語であるため、二重引用符で囲む必要があります。または、`CONTAINS ( expression ) AS ct` のように、別の相関名を指定することもできます。全文検索の例では、`score` カラムを `ct.score` と呼んでいます。

次の文は、`MarketingInformation.Description` で、“stretch” で始まる単語、または “comfort” で始まる単語を検索します。

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

**注意** SQL Anywhere のマニュアルには、全文検索の例が記載されています。これらの例すべてが Sybase IQ 15.2 に当てはまるわけではありません。たとえば、Sybase IQ では、IF 検索条件の一部となるテキスト検索はサポートされておらず、ファジーまたは NGRAM 検索は許可されていません。

## 参照

全文検索の種類については、『SQL Anywhere サーバー-SQL の使用法』>「データのクエリと変更」>「データのクエリ」を参照してください。

## LONG BINARY カラムのクエリ

SELECT 文の WHERE 句では、BYTE\_LENGTH64、BYTE\_SUBSTR64、BYTE\_SUBSTR、BIT\_LENGTH、OCTET\_LENGTH、CHARINDEX、LOCATE 関数に加え、LONG BINARY カラムだけが、IS NULL および IS NOT NULL 式で使用できます。

LONG BINARY カラムは、SELECT 文の ORDER BY 句、GROUP BY 句、HAVING 句では使用できません。また、DISTINCT キーワードが指定されている場合も使用できません。

Sybase IQ では、LONG BINARY (BLOB) カラムまたは変数での LIKE 述部はサポートされていません。LIKE 述部を使用して、LONG BINARY カラムでパターンを検索すると、「述部内の無効なデータ型比較です」というエラーが返されます。

詳細については、「[第 9 章 関数のサポート](#)」を参照してください。

## LONG VARCHAR カラムのクエリ

SELECT 文の WHERE 句では、BIT\_LENGTH、CHAR\_LENGTH、CHAR\_LENGTH64、CHARINDEX、LOCATE、OCTET\_LENGTH、PATINDEX、SUBSTRING64、SUBSTRING 関数に加え、LONG VARCHAR カラムだけが、IS NULL および IS NOT NULL 式で使用できます。

LIKE 述部を使用して、LONG VARCHAR カラムでパターンを検索できます。126 文字以下のパターンは、すべてサポートされています。254 文字よりも長いパターンは、サポートされていません。127 ~ 254 文字の長さのパターンは、パターンの内容によってサポートされることがあります。

LIKE 述部は、任意のデータ・サイズの LONG VARCHAR (CLOB) 変数をサポートします。現時点では、SQL 変数で保持できる最大長は 2GB - 1 です。

LONG VARCHAR カラムは、SELECT 文の ORDER BY 句、GROUP BY 句、HAVING 句では使用できません。また、DISTINCT キーワード (SELECT DISTINCT、COUNT DISTINCT) が指定されている場合も使用できません。

詳細については、「[第 9 章 関数のサポート](#)」を参照してください。

### CONTAINS 述部のサポート

WORD (WD) インデックスを LONG VARCHAR (CLOB) カラムに作成し、CONTAINS 述部を使用して、カラムで最大長が 255 文字の文字列定数を検索できます。

CONTAINS 述部は、WD インデックスを使用している LONG BINARY (BLOB) カラムではサポートされません。CONTAINS 述部を使用して、WD インデックスを持つ LONG BINARY カラムの文字列を検索しようとすると、エラーが返されます。外部ライブラリを使用する TEXT インデックスは、バイナリ・データでの CONTAINS をサポートしています。

『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「[第 2 章 SQL 言語の要素](#)」にある「CONTAINS 条件」を参照してください。

## LONG BINARY カラムと LONG VARCHAR カラムのパフォーマンスのモニタリング

Sybase IQ パフォーマンス・モニタは、LONG BINARY カラムと LONG VARCHAR カラムのパフォーマンス・データを表示します。



この章では、LONG BINARY (BLOB) データ型と LONG VARCHAR (CLOB) データ型のカラム、および全文検索でのストアド・プロシージャのサポートについて説明します。

トピック	ページ
<a href="#">TEXT インデックス内の単語の管理</a>	39
<a href="#">外部ライブラリの確認</a>	43
<a href="#">ラージ・オブジェクト・データの圧縮の制御</a>	44
<a href="#">ラージ・オブジェクト・カラムについての情報の表示</a>	46

## TEXT インデックス内の単語の管理

これらのストアド・プロシージャを使用して、文字列を単語に分割したり、TEXT インデックスに含まれる単語数とそれらの位置を確認したり、TEXT インデックスに関する統計情報を表示したりできます。

### sa\_char\_terms システム・プロシージャ

機能	CHAR 文字列を単語に分割し、各単語をローとして、その位置とともに返します。
構文	<pre>sa_char_terms( 'char-string' [, 'text-config-name' [, 'owner' ] ] )</pre>
パラメータ	<p><b>char-string</b> 解析する CHAR 文字列。</p> <p><b>text-config-name</b> 文字列の処理時に適用するテキスト設定オブジェクト。デフォルト値は 'default_char' です。</p> <p><b>所有者</b> 指定されたテキスト設定オブジェクトの所有者。デフォルト値は DBA です。</p>
説明	このシステム・プロシージャを使用して、テキスト設定オブジェクトの設定が適用されるときに、文字列がどのように解釈されるのかを確認できます。これは、インデックスの作成中またはクエリ文字列からどの単語が削除されるのかを知りたい場合に便利です。
パーミッション	なし。

例 この文では、CHAR 文字列 'the quick brown fox jumped over the fence' 内の単語が返されます。

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

表 5-1: CHAR 文字列の解釈

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

## sa\_nchar\_terms システム・プロシージャ

機能 NCHAR 文字列を単語に分割し、各単語をローとして、その位置とともに返します。

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

**char-string** 解析する NCHAR 文字列。

**text-config-name** 文字列の処理時に適用するテキスト設定オブジェクト。デフォルト値は 'default\_nchar' です。

**所有者** 指定されたテキスト設定オブジェクトの所有者。デフォルト値は DBA です。

説明 `sa_nchar_terms` を使用して、テキスト設定オブジェクトの設定が適用されるときに、文字列がどのように解釈されるのかを確認できます。これは、インデックスの作成中またはクエリ文字列からどの単語が削除されるのかを知りたい場合に便利です。

---

**注意** NCHAR データ型は、IN SYSTEM テーブルでのみサポートされます。

---

パーミッション [なし]

例 `sa_nchar_terms` の構文は、`sa_char_terms` システム・プロシージャの構文と似ています。

**sa\_text\_index\_stats システム・プロシージャ**

**機能** データベース内の TEXT インデックスに関する統計情報を返します。

**構文** **sa\_text\_index\_stats()**

**説明** **sa\_text\_index\_stats** を使用して、データベース内の各 TEXT インデックスの統計情報を表示します。次の表に、**sa\_text\_index\_stats** によって返される情報を示します。

**表 5-2: TEXT インデックスの統計情報**

カラム名	種類	説明
owner_id	UNSIGNED INT	テーブルの所有者の ID
table_id	UNSIGNED INT	テーブルの ID
index_id	UNSIGNED INT	TEXT インデックスの ID
text_config_id	UNSIGNED BIGINT	TEXT インデックスによって参照されるテキスト設定の ID
owner_name	CHAR(128)	所有者の名前
table_name	CHAR(128)	テーブルの名前
index_name	CHAR(128)	TEXT インデックスの名前
text_config_name	CHAR(128)	テキスト設定オブジェクトの名前
doc_count	UNSIGNED BIGINT	TEXT インデックス内のインデックス・カラム値の総数
doc_length	UNSIGNED BIGINT	TEXT インデックス内のデータの長さの合計
pending_length	UNSIGNED BIGINT	保留中の変更の長さの合計
deleted_length	UNSIGNED BIGINT	保留中の削除の長さの合計
last_refresh	TIMESTAMP	最終更新日時

`pending_length`、`deleted_length`、`last_refresh` の各値は、IMMEDIATE REFRESH TEXT インデックスの場合は NULL です。

**パーミッション**

DBA 権限が必要です。

**例**

この文では、データベース内の各 TEXT インデックスの統計情報が返されます。

```
CALL sa_text_index_stats();
```

## sa\_text\_index\_vocab システム・プロシージャ

### 機能

TEXT インデックスに含まれるすべての単語と、各単語が含まれるインデックス値の合計数を示します。

### 構文

```
sa_text_index_vocab(  
  'text-index-name',  
  'table-name',  
  'table-owner'  
)
```

### パラメータ

**text-index-name** この CHAR(128) パラメータを使用して、TEXT インデックスの名前を指定します。

**table-name** この CHAR(128) パラメータを使用して、TEXT インデックスが構築されているテーブルの名前を指定します。

**table-owner** この CHAR(128) パラメータを使用して、テーブルの所有者を指定します。

### 説明

sa\_text\_index\_vocab は、TEXT インデックスに含まれるすべての単語と、各単語が含まれるインデックス値の合計数を返します (これは、単語が一部のインデックス値で複数回出現する場合の合計出現回数未満となります)。

sa\_text\_index\_vocab には、次の制限があります。

- パラメータ値はホスト変数またはホスト式にできません。引数 *text-index-name*、*table-name*、*table-owner* は、制約または変数である必要があります。

### パーミッション

インデックス・テーブルに対する DBA 権限である SELECT パーミッションが必要です。

### 例

この例では、sa\_text\_index\_vocab が実行され、GROUPO が所有するテーブル Customers の TEXT インデックス MyTextIndex に含まれるすべての単語が返されます。

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

表 5-3: インデックス内の単語

用語	出現回数
a	1
Able	1
Acres	1
Active	5
Advertising	1
Again	1
...	...

## 外部ライブラリの確認

`sa_list_external_library` ストアド・プロシージャは、現在サーバにロードされているライブラリを示します。確認後に、ライブラリをサーバからアンロードするには、`sa_external_library_unload` を使用します。

### **sa\_external\_library\_unload** システム・プロシージャ

#### 機能

外部ライブラリをアンロードします。

#### 構文

```
sa_external_library_unload ( [ 'external-library' ] )
```

#### パラメータ

**external-library** 必要に応じて、この LONG VARCHAR パラメータを使用して、アンロードするライブラリの名前を指定します。ライブラリが指定されていない場合、使用されていないすべての外部ライブラリがアンロードされます。

#### 説明

指定した外部ライブラリが使用中であるか、ロードされていない場合は、エラーが返されます。パラメータを指定しない場合で、ロードされている外部ライブラリが見つからないときは、エラーが返されます。

#### パーミッション

DBA 権限が必要です。

#### 例

この例では、*myextlib.dll* という外部ライブラリがアンロードされます。

```
CALL sa_external_library_unload( 'myextlib.dll' );
```

この例では、現在使用されていないすべてのライブラリがアンロードされます。

```
CALL sa_external_library_unload();
```

### **sa\_list\_external\_library** プロシージャ

#### 機能

現在サーバにロードされている外部ライブラリを一覧表示します。

#### 構文

```
sa_list_external_library()
```

#### 説明

エンジンにロードされている外部ライブラリのリストを、参照カウントとともに返します。参照カウントは、エンジン内のライブラリのインスタンス数です。`sa_external_library_unload` プロシージャを実行して外部ライブラリをアンロードできるのは、その参照カウントが 0 の場合のみです。

#### パーミッション

DBA 権限が必要です。

#### 例

この例では、外部ライブラリとその参照カウントが一覧表示されます。

```
call sa_list_external_library()
```

## ラージ・オブジェクト・データの圧縮の制御

`sp_iqsetcompression` ストアド・プロシージャは、データベース・バッファをディスクに書き込むときの LONG BINARY データ型および LONG VARCHAR データ型のカラムの圧縮を制御します。`sp_iqsetcompression` を使用して圧縮を無効にすることもできます。LONG BINARY カラムまたは LONG VARCHAR カラムに格納されている特定のデータ・フォーマット (JPG ファイルなど) は既に圧縮されており、圧縮を重ねても意味がないので、この機能を使用することで CPU サイクルを節約できます。ラージ・オブジェクト・カラムの圧縮設定の表示は、`sp_iqshowcompression` ストアド・プロシージャで行います。

### `sp_iqsetcompression` プロシージャ

機能	LONG BINARY (BLOB) データ型および LONG VARCHAR (CLOB) データ型のカラムでのデータの圧縮を設定します。
構文	<code>sp_iqsetcompression ( owner, table, column, on_off_flag )</code>
パーミッション	DBA 権限が必要です。
説明	<code>sp_iqsetcompression</code> では、LONG BINARY (BLOB) データ型と LONG VARCHAR (CLOB) データ型のカラムの圧縮を制御できます。圧縮設定は、Sybase IQ ベース・テーブルにのみ適用されます。 <code>sp_iqsetcompression</code> には、圧縮設定を変更した後に COMMIT が実行されるという二次的な影響があります。

表 5-4: `sp_iqsetcompression` パラメータ

名前	説明
<code>owner</code>	圧縮を設定するテーブルの所有者
<code>table</code>	圧縮を設定するテーブル
<code>column</code>	圧縮を設定するカラム
<code>on_off_flag</code>	圧縮設定: 圧縮を有効にする場合は ON、無効にする場合は OFF

#### 例

この例では、次のテーブル定義を想定します。

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

LOB カラム `picJPG` で圧縮を無効にするには、`sp_iqsetcompression` を呼び出します (この作業を行うには DBA パーミッションが必要です)。

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

このコマンドはローを返しません。

## sp\_iqshowcompression プロシージャ

機能	LONG BINARY (BLOB) データ型および LONG VARCHAR (CLOB) データ型のカラムの圧縮設定を表示します。
構文	<b>sp_iqshowcompression ( <i>owner</i>, <i>table</i>, <i>column</i> )</b>
パーミッション	DBA 権限が必要です。
説明	カラム名と圧縮設定を返します。圧縮設定値は 'ON' ( 圧縮は有効 ) および 'OFF' ( 圧縮は無効 ) です。

表 5-5: *sp\_iqshowcompression* パラメータ

名前	説明
<i>owner</i>	圧縮を設定するテーブルの所有者
<i>table</i>	圧縮を設定するテーブル
<i>column</i>	圧縮を設定するカラム

### 例

この例では、次のテーブル定義を想定します。

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

*pixTable* テーブルのカラムの圧縮ステータスをチェックするには、*sp\_iqshowcompression* を呼び出します ( この作業を行うには DBA パーミッションが必要です )。

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

このコマンドは 1 つのローを返します。

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

## ラージ・オブジェクト・カラムについての情報の表示

ストアド・プロシージャ `sp_iqindexsize` では、個々の LONG BINARY および LONG VARCHAR カラムのサイズを表示できます。

LONG BINARY カラムの  
サイズ 次の出力は、約 42GB のデータが格納されている LONG BINARY カラムのもの  
です。ページ・サイズは 128KB です。`largelob` Info 型は最後のローにあります。

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

この例では、圧縮率は  $42953952/(623009*128) = 53.9\%$  です。

LONG VARCHAR カラム  
のサイズ 次の出力は、約 42GB のデータが格納されている LONG VARCHAR カラムの  
ものです。ページ・サイズは 128KB です。`largelob` Info 型は最後のローにあり  
ます。

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

この例では、圧縮率は  $42953952/(623009*128) = 53.9\%$  です。

# ラージ・オブジェクト・データのロード とアンロード

この章では、ラージ・オブジェクト・データを Sybase IQ からエクスポートする方法と、Sybase IQ にロードする方法について説明します。

トピック	ページ
<a href="#">ラージ・オブジェクト・データのエクスポート</a>	47
<a href="#">ラージ・オブジェクト・データのロード</a>	48

## ラージ・オブジェクト・データのエクスポート

IQ データ抽出機能には、個別の LONG BINARY セルと LONG VARCHAR セルをサーバ上の個別のオペレーティング・システム・ファイルに抽出できる BFILE 関数が含まれています。BFILE は、データ抽出機能と一緒に使用できる他、単独でも使用できます。

### BFILE 関数

#### 機能

個別の LONG BINARY セルと LONG VARCHAR セルをサーバ上の個別のオペレーティング・システム・ファイルに抽出します。

#### 構文

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

#### パラメータ

**file-name-expression.** LONG BINARY データまたは LONG VARCHAR データが書き込まれる出力ファイルの名前。このファイル名の最大長は (32K -1) バイトですが、ファイル・システムによってサポートされている有効なパス名とします。

**large-object-column.** LONG BINARY カラムまたは LONG VARCHAR カラムの名前。

#### 使用法

BFILE は、次の値を返します。

- 1 ( ファイルの書き込みが成功した場合 )
- 0 ( ファイルが開かれていない、または書き込みが失敗した場合 )
- NULL (LONG BINARY セルまたは LONG VARCHAR セルの値が NULL の場合 )

LONG BINARY セルまたは LONG VARCHAR セルの値が NULL の場合、ファイルは開かれず、データも書き込まれません。

ファイル・パスは、サーバが開始されたロケーションから見た相対指定です。ファイルを開いて書き込むには、サーバ・プロセスのパーミッションが必要です。BFILE 出力ファイルについては、テープ・デバイスはサポートされていません。

BFILE 関数以外によって(つまり、クライアント / サーバ・データベース接続によって後で)取得された LONG BINARY セルと LONG VARCHAR セルの最大長は、2 GB です。SELECT (SELECT, OPEN CURSOR) を使用して 2GB を超える LONG BINARY セルを取得するには、SUBSTRING64 または BYTE\_SUBSTR64 を使用します。SELECT (SELECT, OPEN CURSOR) を使用して 2GB を超える LONG BINARY セルを取得するには、SUBSTRING64 を使用します。なお、ODBC、JDBC™、Open Client™ などの一部の接続ドライバの場合、2GB を超える値は単なる SELECT コマンドでは返されません。

BFILE は、データ抽出機能と一緒に使用できる他、単独でも使用できます。

### 例

この例は、「[ロードの例](#) (50 ページ)」によって作成およびロードされた LONG BINARY カラム lobcol から、BFILE を使用してデータを抽出する方法を説明しています。ロードのセカンダリ・ファイルとして使用できるファイルにデータを書き込むには、次のように入力します。

```
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
```

このコマンドは、ロー 1 の lobcol のファイル名を *boston.jpg* の拡張子付きで、ロー 2 の lobcol のファイル名を *map\_of\_concord.bmp* の拡張子付きで生成します。

## ラージ・オブジェクト・データのロード

LOAD TABLE 文の拡張構文を使用して、LONG BINARY データと LONG VARCHAR データをロードします。オペレーティング・システムによって制限されていないかぎり、無制限のサイズのラージ・オブジェクト・データを、ASCII フォーマットまたは BCP フォーマットでプライマリ・ファイルからロードできます。プライマリ・ファイルからラージ・オブジェクト・カラムにロードされる固定幅データの最大長は 32K - 1 です。

ここで、プライマリ・ロード・ファイルにセカンダリ・ロード・ファイルを指定することもできます。個々のセカンダリ・データ・ファイルには、LONG BINARY セルまたは LONG VARCHAR セルの値が 1 つだけあります。

拡張 LOAD TABLE 構文

```

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 )

```

キーワード BINARY FILE (LONG BINARY) または ASCII FILE (LONG VARCHAR) は、LONG BINARY セルまたは LONG VARCHAR セルの値が格納されているセカンダリ・ファイルのパスを含むカラムに、LONG BINARY データ自体または LONG VARCHAR データ自体ではなく、プライマリ入力ファイルをロードするように指定します。セカンダリ・ファイルのパス名は、完全修飾されたパスまたは相対パスのいずれかを使用できます。セカンダリ・ファイルのパス名が完全修飾されたパスでない場合は、パスはサーバが起動されたディレクトリからの相対パスです。セカンダリ・ファイルについては、テープ・デバイスはサポートされていません。

Sybase IQ では、プライマリ・ロード・ファイルの無制限の長さ（オペレーティング・システムの制限に従います）の LONG BINARY 値と LONG VARCHAR 値をロードできます。16 進数形式のバイナリ・データをプライマリ・ファイルから LONG BINARY カラムにロードする場合、Sybase IQ では、16 進数字の合計数は偶数である必要があります。セルの値に奇数個の 16 進数字が含まれている場合、“Odd length of binary data value detected on column” というエラーが報告されます。LONG BINARY のロードの入力ファイルには、必ず偶数個の 16 進数字が含まれている必要があります。

Sybase IQ では、LOAD TABLE...FORMAT BINARY を使用したプライマリ・ファイルからのラージ・オブジェクト・カラムのロードはサポートされていません。セカンダリ・ファイルからは、バイナリ・フォーマットでラージ・オブジェクト・データをロードできます。バイナリ・フォーマットを使用したデータのロードの詳細については、『システム管理ガイド 第1巻』の「第7章 MovingData In and Out of Databases」の「Using binary load format」を参照してください。

LOAD TABLE FORMAT BCP では、ロード仕様にカラム名、NULL、ENCRYPTED のみを含めることができます。つまり、LOAD TABLE FORMAT BCP オプションを使用して、LONG BINARY カラムと LONG VARCHAR カラムをロードする場合は、セカンダリ・ファイルは使用できません。『リファレンス：文とオプション』の「第1章 SQL 文」の「LOAD TABLE 文」を参照してください。

### ロードの例

この例は、表を作成して **LONG BINARY** データをロードする SQL 文を示します。

```
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;
```

プライマリ・ファイル *abc.inp* には次のデータがあります。

```
1,boston.jpg,/s1/loads/lobs/boston.jpg,
2,map_of_concord.bmp,/s1/loads/maprs/concord.bmp,
3,zero length test,NULL.,
4,null test,NULL,NULL,
```

**LONG BINARY** データが *ltab* 表にロードされると、*lobcol* カラムの最初のローには *boston.jpg* ファイルの中身が、2 番目のローには *concord.bmp* ファイルの中身がそれぞれ入ります。3 番目のローには長さ 0 の値が、4 番目のローには NULL がそれぞれ入ります。

### ロード・エラーの制御

データベース・オプション **SECONDARY\_FILE\_ERROR** では、セカンダリ・ファイル (BINARY FILE または ASCII FILE) を開けなかった場合、またはその読み取りに失敗した場合に取るアクションを指定できます。

**SECONDARY\_FILE\_ERROR** をオンにすると、セカンダリ・ファイル (BINARY FILE または ASCII FILE) を開けなかった場合、またはその読み取りに失敗した場合、ロードはロールバックします。

**SECONDARY\_FILE\_ERROR** をオフ (デフォルト) にすると、セカンダリ・ファイル (BINARY FILE または ASCII FILE) を開けなかった場合、またはその読み取りに失敗した場合でも、ロードは続行されます。**LONG BINARY** セルまたは **LONG VARCHAR** セルの値は次のいずれかのままです。

- カラムが NULL を許容している場合、NULL。
- カラムが NULL を許容していない場合、長さ 0 の値。

**SECONDARY\_FILE\_ERROR** は任意のユーザが PUBLIC グループに対して設定したり、一時的に設定したりできます。また、その設定はただちに有効になります。

整合性制約違反をロード・エラー ROW LOG ファイルにロギングすると、**LONG BINARY** カラムまたは **LONG VARCHAR** カラムには次の情報が記録されます。

- プライマリ・データ・ファイルから実際に読み取ったテキスト (1 回目のロード操作でロギングが発生した場合)
- 長さ 0 の値 (2 回目のロード操作でロギングが発生した場合)

後続ブランクの削除	LOAD TABLE...STRIP オプションは LONG VARCHAR データには何の影響も及ぼしません。STRIP オプションがオンでも、LONG VARCHAR データの後続ブランクは削除されません。	
引用符で囲む	LOAD TABLE...QUOTES オプションは、その設定に関係なく、セカンダリ・ファイルからの LONG BINARY (BLOB) または LONG VARCHAR (CLOB) データのロードには適用されません。開始引用符または終了引用符は、CLOB データの一部としてロードされます。引用符で囲まれている 2 つの連続した引用符は、QUOTES ON オプションを使用すると 2 つの連続した引用符としてロードされます。	
マルチバイト文字データの一部のトランケート	マルチバイト LONG VARCHAR データの一部が、TRIM_PARTIAL_MBC データベース・オプションの値に従ってロード時にトランケートされます。	
	<ul style="list-style-type: none"> <li>TRIM_PARTIAL_MBC をオンにすると、プライマリ・データと ASCII FILE オプションを指定して LOAD した値の両方で、マルチバイト文字の一部がトランケートされます。</li> <li>TRIM_PARTIAL_MBC をオフにすると、ASCII FILE オプションを指定した LOAD は、マルチバイト文字の一部を SECONDARY_FILE_ERROR データベース・オプションの値に従って処理します。</li> </ul>	
参照	表 6-1 は、ロード時に後続マルチバイト文字が TRIM_PARTIAL_MBC と SECONDARY_FILE_ERROR の値に従って処理される様子を示します。	

表 6-1: LONG VARCHAR ロード時 (ASCII FILE オプション指定) のマルチバイト文字の一部

TRIM_PARTIAL_MBC	SECONDARY_FILE_ERROR	一部の後続マルチバイト文字の処理
ON	ON/OFF	後続マルチバイト文字の一部をトランケート
OFF	ON	セル - NULL (NULL が許容されている場合) LOAD エラー - ロールバック (NULL が許容されていない場合)
OFF	OFF	セル - NULL (NULL が許容されている場合) セル - 長さ 0 の値 (NULL が許容されていない場合)

LOAD TABLE、INSERT...VALUES、INSERT...SELECT、INSERT...LOCATION、SELECT...INTO、UPDATE SQL 文でのラージ・オブジェクト変数のサポートについては、「[第7章 ラージ・オブジェクト・データ型](#)」を参照してください。



この章では、ラージ・オブジェクト LONG BINARY データ型カラムと LONG VARCHAR データ型カラムの特性、およびラージ・オブジェクト・データのインデックスのサポートについて説明します。

トピック	ページ
ラージ・オブジェクト・データ型、LONG BINARY および BLOB	53
ラージ・オブジェクト・データ型、LONG VARCHAR および CLOB	54
ラージ・オブジェクト変数	56
ラージ・オブジェクト・カラムのインデックスのサポート	57

## ラージ・オブジェクト・データ型、LONG BINARY および BLOB

Sybase IQ のバイナリ・ラージ・オブジェクト (BLOB) データは、LONG BINARY データ型または BLOB データ型のカラムに格納されます。

個々の LONG BINARY データ型の取り得る長さは、IQ ページ・サイズが 128KB の場合は 0 ~ 512TB (テラバイト)、IQ ページ・サイズが 512KB の場合は 0 ~ 2PB (ペタバイト) です。(最大長は、4GB にデータベース・ページ・サイズを掛けた値になります)。LONG BINARY データが格納されているテーブルをサポートするには、IQ ページ・サイズを最低 128KB (131072 バイト) に設定して、IQ データベースを作成します。

テーブルまたはデータベースは、それぞれ、テーブルあたりの最大カラム数、データベースあたりの最大カラム数を上限として、LONG BINARY カラムをいくつでも持つことができます。

LONG BINARY カラムでは、NULL または NOT NULL が許容され、長さゼロの値を格納できます。ドメイン BLOB は、NULL を許容する LONG BINARY データ型です。

非 FP インデックスまたはジョイン・インデックスは、LONG BINARY カラムには作成できません。

結果セットに BLOB カラムが含まれている場合、プリフェッチは無効です。

LONG BINARY カラムは、UPDATE、INSERT、LOAD TABLE、DELETE、TRUNCATE、SELECT...INTO、INSERT...LOCATION の各 SQL 文を使用して修正します。位置付け更新と位置付け削除は、LONG BINARY カラムではサポートされていません。

Adaptive Server Enterprise の IMAGE カラムは、INSERT...LOCATION コマンドを使用して、LONG BINARY カラムに挿入できます。挿入されたすべての IMAGE データは、2147483648 バイト (2GB) を超えた分が暗黙的に右トランケートされます。

### データ型変換

LONG BINARY データ型から他の非 LONG BINARY データ型 (INSERT と UPDATE における BINARY データ型と VARBINARY データ型を除く) への暗黙的なデータ型変換は行われません。TINYINT、SMALLINT、INTEGER、UNSIGNED INTEGER、BIGINT、UNSIGNED BIGINT、CHAR、VARCHAR データ型から LONG BINARY データ型への暗黙的なデータ型変換は行われます。BIT、REAL、DOUBLE、または NUMERIC データ型から LONG BINARY データ型への暗黙的なデータ型変換は行われません。暗黙的な変換は、CONVERSION\_MODE データベース・オプションを使用して制御できます。

LONG BINARY データ型に対して現在サポートされているバイト部分文字列関数は、INSERT および UPDATE 文の暗黙的な変換の入力として受け入れられます。詳細については、[「第 9 章 関数のサポート」](#) を参照してください。

LONG BINARY データ型は BINARY または VARBINARY に明示的に変換できます。LONG BINARY データ型が変換先または変換元となるその他の明示的なデータ型変換 (たとえば、CAST 関数または CONVERT 関数によるデータ型変換) は行われません。

LONG BINARY から BINARY または VARBINARY への変換中の LONG BINARY データのトランケーションは、BINARY および VARBINARY データのトランケーションと同じ方法で処理されます。STRING\_RTRUNCATION オプションが ON の場合は、バイナリ・カラムの INSERT または UPDATE の (スペース以外の文字だけでなくすべての値) 右トランケートを実行すると、トランケーションエラーが発生し、トランザクションがロールバックされます。

## ラージ・オブジェクト・データ型、LONG VARCHAR および CLOB

Sybase IQ のキャラクタ・ラージ・オブジェクト (CLOB) データは、LONG VARCHAR データ型または CLOB データ型のカラムに格納されます。

個々の LONG VARCHAR データ型の取り得る長さは、IQ ページ・サイズが 128KB の場合は 0 ~ 512TB (テラバイト)、IQ ページ・サイズが 512KB の場合は 0 ~ 2PB (ペタバイト) です (最大長は、4GB にデータベース・ページ・サイズを掛けた値になります)。LONG VARCHAR データが格納されているテーブルをサポートするには、IQ ページ・サイズを最低 64KB (65536 バイト) に設定して、IQ データベースを作成します。

テーブルまたはデータベースは、それぞれ、テーブルあたりの最大カラム数、データベースあたりの最大カラム数を上限として、LONG VARCHAR カラムをいくつでも持つことができます。

Sybase IQ は、シングルバイトおよびマルチバイトの LONG VARCHAR データをサポートしています。

LONG VARCHAR カラムでは、NULL または NOT NULL が許容され、長さゼロの値を格納できます。ドメイン CLOB は、NULL を許容する LONG VARCHAR データ型です。非 NULL LONG VARCHAR カラムを作成するには、NOT NULL をカラム定義に明示指定します。

テーブルを作成するとき、または既存のテーブルにカラムを追加するときは、ドメイン CLOB を使用して LONG VARCHAR カラムを作成できます。例を示します。

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

WORD (WD) インデックスは、LONG VARCHAR カラムに作成できます。その他の非 FP インデックスとジョイン・インデックスは、LONG VARCHAR カラムには作成できません。

LONG VARCHAR カラムは、UPDATE、INSERT...VALUES、INSERT...SELECT、LOAD TABLE、DELETE、TRUNCATE、SELECT...INTO、INSERT...LOCATION の各 SQL 文を使用して修正できます。位置付け更新と位置付け削除は、LONG VARCHAR カラムではサポートされていません。

Adaptive Server Enterprise の TEXT カラムは、INSERT...LOCATION コマンドを使用して、LONG VARCHAR カラムに挿入できます。挿入された TEXT データはすべて、2147483648 バイト (2GB) を超えた分が暗黙的に右トランケートされます。

#### データ型変換

LONG VARCHAR データ型から他の非 LONG VARCHAR データ型 (INSERT と UPDATE のみにおける LONG BINARY、CHAR、VARCHAR を除く) への暗黙的なデータ型変換は行われません。CHAR データ型および VARCHAR データ型から LONG VARCHAR データ型への暗黙的なデータ変換は行われます。BIT、REAL、DOUBLE、NUMERIC、TINYINT、SMALLINT、INT、UNSIGNED INT、BIGINT、UNSIGNED BIGINT、BINARY、VARBINARY、または LONG BINARY データ型から LONG VARCHAR データ型への暗黙的なデータ変換は行われません。暗黙的な変換は、CONVERSION\_MODE データベース・オプションを使用して制御できます。

LONG VARCHAR データ型に対して現在サポートされている文字列関数は、INSERT および UPDATE 文の暗黙的な変換の入力として受け入れられます。詳細については、「[第9章 関数のサポート](#)」を参照してください。

LONG VARCHAR データ型は CHAR および VARCHAR に明示的に変換できます。LONG VARCHAR データ型が変換先または変換元となる他の明示的なデータ型変換 (たとえば、CAST 関数または CONVERT 関数によるデータ型変換) は行われません。

LONG VARCHAR から CHAR への変換中の LONG VARCHAR データのトランケーションは、CHAR データのトランケーションと同じ方法で処理されます。STRING\_RTRUNCATION オプションが ON の場合は、スペース以外の文字の文字列の右トランケートを実行すると、トランケーション エラーが報告され、トランザクションがロールバックされます。末尾にある部分的なマルチバイト文字は、変換時にスペースに置き換えられます。

LONG VARCHAR から VARCHAR への変換中の LONG VARCHAR データのトランケーションは、VARCHAR データのトランケーションと同じ方法で処理されます。STRING\_RTRUNCATION オプションが ON の場合は、スペース以外の文字の文字列の右トランケートを実行すると、トランケーション エラーが報告され、トランザクションがロールバックされます。末尾にある部分的なマルチバイト文字は、変換時にトランケートされます。

## ラージ・オブジェクト変数

インバウンドの LONG BINARY 変数および LONG VARCHAR 変数 (IQ が使用するホスト変数または SQL 変数) に最大長はありません。

アウトバウンドの LONG BINARY 変数および LONG VARCHAR 変数 (IQ が設定する変数) の最大長は 2GB - 1 です。

LOAD TABLE、INSERT...VALUES、INSERT...SELECT、INSERT...LOCATION、SELECT...INTO、UPDATE の各 SQL 文は、任意のデータ・サイズの LONG BINARY 変数と LONG VARCHAR 変数を受け入れます。現時点では、SQL 変数で保持できる最大長は 2GB - 1 です。

BIT\_LENGTH、BYTE\_LENGTH、BYTE\_LENGTH64、BYTE\_SUBSTR、BYTE\_SUBSTR64、CHARINDEX、LOCATE、OCTET\_LENGTH、SUBSTRING64 の各関数は、SQL 変数が保持できる任意のデータ・サイズの LONG BINARY 変数と LONG VARCHAR 変数をサポートします。さらに、CHAR\_LENGTH、CHAR\_LENGTH64、PATINDEX、SUBSTR、SUBSTRING の各関数は、SQL 変数が保持できる任意のデータ・サイズの LONG VARCHAR 変数をサポートします。

## ラージ・オブジェクト変数のデータ型変換

データベース・オプション `ENABLE_LOB_VARIABLES` は、ラージ・オブジェクト変数のデータ型変換を制御します。

### `ENABLE_LOB_VARIABLES` オプション

ファンクション	ラージ・オブジェクト変数のデータ型変換を制御します。
指定できる値	ON、OFF
デフォルト	OFF
スコープ	このオプションを設定するために、DBA パーミッションは必要ありません。個々の接続または PUBLIC グループに一時的に設定することもできます。すぐには有効になります。
説明	このオプションは、ラージ・オブジェクト変数のデータ型変換を制御します。 <code>ENABLE_LOB_VARIABLES</code> が OFF の場合、32K 未満のラージ・オブジェクト変数は暗黙的に変換されます。ラージ・オブジェクト変数が 32K 以上の場合はエラーが報告されます。LONG VARCHAR 変数は VARCHAR データ型に暗黙的に変換され、32K を超えた部分がトランケートされます。LONG BINARY 変数は VARBINARY データ型に暗黙的に変換され、32K を超えた部分がトランケートされます。 ENABLE_LOB_VARIABLES が ON の場合、任意のサイズのラージ・オブジェクト変数で元のデータ型とサイズが保持されます。
例	32K を超えるラージ・オブジェクト変数のデータ型とサイズを保持するには、次のように入力します。

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

## ラージ・オブジェクト・カラムのインデックスのサポート

Sybase IQ は、LONG BINARY カラムと LONG VARCHAR カラムでの TEXT インデックス、および LONG VARCHAR カラムでの WORD (WD) インデックスをサポートします。

### ラージ・オブジェクト・カラムの TEXT インデックスのサポート

Sybase IQ の TEXT インデックスは、LONG BINARY カラムおよび LONG VARCHAR カラムをサポートします。

詳細については、「[第8章 SQL文のサポート](#)」と「[第2章 TEXT インデックスとテキスト設定オブジェクト](#)」を参照してください。

## LONG VARCHAR (CLOB) カラムの WD インデックスのサポート

Sybase IQ では、以下により、LONG VARCHAR (CLOB) カラムで WORD (WD) インデックスがサポートされます。

- Sybase Central では、CHAR、VARCHAR、LONG VARCHAR の各データ型のカラムで、WD インデックスを作成できる。
- WD インデックスによってサポートされる最大幅のカラムは、LOB カラムの最大幅（最大長は、4GB にデータベース・ページ・サイズを掛けた値になります）。

Sybase IQ によってサポートされる最大ワード長は 255 バイト。

- CHAR カラムと VARCHAR カラムでの WD インデックスの `sp_iqcheckdb` オプションもすべて、LONG VARCHAR (CLOB) カラムで、それも割り付け、チェック、検証の各モードでサポートされる。
- `sp_iqrebuildindex` を使用して、LONG VARCHAR (CLOB) カラムで WD インデックスを再構築できる。

バイナリ・フォーマットで書かれた中国語のテキストや文書では、ETL の前処理を実施し、単語を見つけ出して WD インデックスが解析できる形に変換する必要があります。

この章では、TEXT インデックスとテキスト設定の操作をサポートする SQL 文および構文について説明します。

トピック	ページ
<a href="#">ALTER TEXT CONFIGURATION 文</a>	59
<a href="#">ALTER TEXT INDEX 文</a>	61
<a href="#">CREATE TEXT CONFIGURATION 文</a>	62
<a href="#">CREATE TEXT INDEX 文</a>	63
<a href="#">DROP TEXT CONFIGURATION 文</a>	64
<a href="#">DROP TEXT INDEX 文</a>	65

## ALTER TEXT CONFIGURATION 文

### 説明

テキスト設定オブジェクトを変更します。

### 構文

```

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

```

### 例

**例 1** この文では、テキスト設定オブジェクト maxTerm16 が作成され、単語の最大長が 16 に変更されます。

```

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

```

**例 2** この文では、ストップリスト単語が maxTerm16 設定オブジェクトに追加されます。

```

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

```

**例 3** この例では、外部ライブラリ mytermbreaker.dll のエントリ・ポイント my\_term\_breaker を使用してテキストを分割するように、テキスト設定オブジェクト my\_text\_config が更新されます。

```
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';
```

**例 4** この例では、外部ライブラリ myprefilter.dll のエントリ・ポイント my\_prefilter を使用してドキュメントのプレフィルタを実行するように、テキスト設定オブジェクト my\_text\_config が更新されます。

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

## 使用法

TEXT インデックスは、テキスト設定オブジェクトに依存します。Sybase IQ TEXT インデックスでは、即時更新が使用され、トランケートすることはできません。テキスト設定オブジェクトを変更する前に、インデックスを削除する必要があります。

テキスト設定オブジェクトの設定を表示するには、SYSTEXTCONFIG システム・ビューに対してクエリを実行します。

**STOPLIST 句** この句を使用して、TEXT インデックスの構築時に無視する単語のリストを作成したり、置き換えたりします。このリストで指定されている単語は、クエリでも無視されます。ストップリスト単語はスペースで区切ります。

ストップリスト単語にスペースを含めることはできません。ストップリスト単語には、英数字以外の文字は含めないでください。英数字以外の文字はスペースとして解釈され、単語が複数の単語に分割されます。たとえば、“and/or”は、2つの単語 “and” と “or” として解釈されます。ストップリスト単語の最大数は 7999 です。

**DROP STOPLIST 句** この句を使用して、テキスト設定オブジェクトのストップリストを削除します。

**MINIMUM TERM LENGTH 句** TEXT インデックスに含める単語の最小長( 文字数 )を指定します。NGRAM TEXT インデックスを使用する場合は、MINIMUM TERM LENGTH 句に指定されている値は無視されます。

この設定よりも短い単語は、TEXT インデックスの構築または更新時に無視されます。このオプションの値は 0 よりも大きい値である必要があります。このオプションに MAXIMUM TERM LENGTH よりも大きい値を設定した場合、MAXIMUM TERM LENGTH の値は、新しい MINIMUM TERM LENGTH の値と同じになるように自動的に調整されます。

**MAXIMUM TERM LENGTH 句** GENERIC TEXT インデックスの場合、TEXT インデックスに含める単語の最大長( 文字数 )です。この設定よりも長い単語は、TEXT インデックスの構築または更新時に無視されます。

MAXIMUM TERM LENGTH の値は 60 以下である必要があります。このオプションに MINIMUM TERM LENGTH よりも小さい値を設定した場合、MINIMUM TERM LENGTH の値は、新しい MAXIMUM TERM LENGTH の値と同じになるように自動的に調整されます。

*TERM BREAKER* 句 カラム値を単語に分割するために使用するアルゴリズムの名前です。IN SYSTEM テーブルでの選択肢は、**GENERIC** (デフォルト) または **NGRAM** です。Sybase IQ TEXT インデックスでサポートされるのは **GENERIC** 単語分割のみです。**GENERIC** アルゴリズムでは、英数字以外の文字で区切られた 1 つ以上の英数字の文字列は、1 つの単語として扱われます。

**NGRAM** アルゴリズムでは、文字列が n-gram に分割されます。n-gram は、大きな文字列の n 文字の部分文字列です。**NGRAM** 単語分割が必要となるのは、ファジー (近接) 一致の場合、または、スペースもしくは英数字以外の文字を使用せずに単語を区切るドキュメントの場合です。**NGRAM** は IN SYSTEM テーブルでサポートされます。

*TERM BREAKER* には、**EXTERNAL NAME** とライブラリのエントリ・ポイントを使用して、外部の単語分割ライブラリの指定を含めることができます。

*DROP PREFILTER* 句 外部のプレフィルタを削除し、**ISYSTEXTCONFIG** テーブルのプレフィルタ・カラムに NULL を設定します。

*PREFILTER EXTERNAL NAME* 句 外部ベンダによって提供される外部のプレフィルタ・ライブラリのエントリ・ポイントとライブラリ名を指定します。

#### 関連する動作

オートコミット。

#### パーミッション

テキスト設定オブジェクトを変更して、外部プレフィルタまたは単語分割のための外部ライブラリおよび関数を指定するには、ユーザに DBA 権限がある必要があります。

テキスト設定に対するその他すべての変更は、設定オブジェクトの所有者か、DBA 権限を持つユーザが実行できます。

## ALTER TEXT INDEX 文

### 説明

TEXT インデックスの定義を変更します。

### 構文

**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**

**例** この例では、1つ目の文によって TEXT インデックス MyTextIndex が作成され、IMMEDIATE REFRESH として定義されます。2つ目の文によって、TEXT インデックスの名前が Text\_index\_daily に変更されます。3つ目の文によって、この TEXT インデックスが tispace という名前の DB 領域に移動されます。

```
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 tispace;
```

**使用法** *RENAME* 句 TEXT インデックスの名前を変更します。

*MOVE* 句 TEXT インデックスを、指定の DB 領域に移動します。

#### 関連する動作

オートコミット。

**パーミッション** インデックスの名前を変更するには、基になるテーブルの所有者であるか、DBA 権限または REFERENCES パーミッションを持っている必要があります。

TEXT インデックスを移動するには、DBA 権限または SPACE ADMIN 権限を持っているか、テーブル所有者であり、かつ DB 領域に対する CREATE パーミッションを持っている必要があります。

## CREATE TEXT CONFIGURATION 文

**説明**

テキスト設定オブジェクトを作成します。

**構文**

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

**例**

この CREATE TEXT CONFIGURATION 文では、default\_char テキスト設定オブジェクトを使用して、テキスト設定オブジェクト max\_term\_sixteen が作成されます。その後の ALTER TEXT CONFIGURATION 文によって、max\_term\_sixteen の単語の最大長が 16 に変更されます。

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

**使用法**

別のテキスト設定オブジェクトをテンプレートとして使用してテキスト設定オブジェクトを作成し、必要に応じて ALTER TEXT CONFIGURATION 文を使用してオプションを変更します。

データベース内のすべてのテキスト設定オブジェクトとその設定のリストを表示するには、SYSTEXTCONFIG システム・ビューに対してクエリを実行します。

**FROM句** 新しいテキスト設定オブジェクトを作成するためのテンプレートとして使用するテキスト設定オブジェクトの名前を指定します。デフォルトのテキスト設定オブジェクトの名前は、`default_char` および `default_nchar` です。`default_char` は Sybase IQ テーブルでのみサポートされており、`default_nchar` は SQL Anywhere テーブルでのみサポートされています。

#### 関連する動作

オートコミット。

#### パーミッション

DBA 権限または RESOURCE 権限が必要です。

すべてのテキスト設定オブジェクトに PUBLIC アクセスが設定されています。TEXT インデックスを作成するパーミッションを持つユーザは、任意のテキスト設定オブジェクトを使用できます。

## CREATE TEXT INDEX 文

#### 説明

TEXT インデックスを作成します。

#### 構文

```
CREATE TEXT INDEX text-index-name
  ON [ owner:table-name( column-name, ... )
    [ IN dbspace-name ]
    [ CONFIGURATION [ owner:text-configuration-name ] ]
    [ IMMEDIATE REFRESH ] ]
```

#### 例

この例では、`iqdemo` データベース内の `Customers` テーブルの `CompanyName` カラムに、TEXT インデックス `myTxtIdx` が作成されます。`max_term_sixteen` テキスト設定オブジェクトが使用されます。

```
CREATE TEXT INDEX myTxtIdx ON Customers (CompanyName );
  CONFIGURATION max_term_sixteen;
```

#### 使用法

ビューまたはテンポラリ・テーブルに TEXT インデックスを作成することはできません。IN SYSTEM マテリアライズド・ビューに TEXT インデックスを作成することはできません。

TEXT インデックスは、ジョイン・インデックス・テーブルには複写されません。TEXT インデックスは、ジョイン・インデックスの一部となっているテーブルのカラムに作成できます。

BEGIN PARALLEL IQ...END PARALLEL IQ 文では、CREATE TEXT INDEX はサポートされません。

**ON句** TEXT インデックスを構築するテーブルおよびカラムを指定します。

**IN句** TEXT を格納する DB 領域を指定します。この句を指定しない場合、TEXT インデックスは、基になるテーブルと同じ DB 領域に作成されます。

*CONFIGURATION* 句 TEXT インデックスの作成時に使用するテキスト設定オブジェクトを指定します。この句を指定しない場合、*default\_char* テキスト設定オブジェクトが使用されます。

*REFRESH* 句 **IMMEDIATE REFRESH** がデフォルト値として使用され、Sybase IQ のテーブルではこの値のみが許可されます。**IMMEDIATE REFRESH** を指定すると、基になるテーブルでの変更により TEXT インデックスのデータが影響を受けるたびに TEXT インデックスが更新されます。

**IMMEDIATE REFRESH TEXT** インデックスは、作成時に値が設定され、元になるカラムのデータが変更されるたびに更新されます。TEXT インデックスが作成された後は、**IMMEDIATE REFRESH** に変更することも、**IMMEDIATE REFRESH** から変更することもできません。

### 関連する動作

オートコミット。

#### パーミッション

基になるテーブルの所有者であるか、DBA 権限または REFERENCES パーミッションを持っている必要があります。

DB 領域に対する CREATE パーミッションを持っている必要があります。

## DROP TEXT CONFIGURATION 文

### 説明

テキスト設定オブジェクトを削除します。

### 構文

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

### 例

次の文では、*mytextconfig* テキスト設定オブジェクトが作成されて、削除されます。

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

### 使用法

依存する TEXT インデックスがあるテキスト設定オブジェクトを削除しようとすると、エラーが発生します。テキスト設定オブジェクトを削除する前に、依存する TEXT インデックスを削除する必要があります。

テキスト設定オブジェクトは、**ISYSTEEXTCONFIG** システム・テーブルに格納されます。

### 関連する動作

オートコミット

#### パーミッション

テキスト設定オブジェクトの所有者であるか、DBA 権限を持っている必要があります。

## DROP TEXT INDEX 文

説明	データベースから TEXT インデックスを削除します。
構文	<b>DROP TEXT INDEX</b> <i>text-index-name</i> <b>ON</b> [ <i>owner</i> ] <i>table-name</i>
例	次の文では、TextIdx という TEXT インデックスが作成されて、削除されます。
	<pre>CREATE TEXT INDEX TextIdx ON Customers ( Street ); DROP TEXT INDEX TextIdx ON Customers;</pre>
使用法	<i>ON</i> TEXT インデックスが構築されているテーブルを指定します。 テキスト設定オブジェクトを削除する前に、依存する TEXT インデックスを削除する必要があります。
	<b>関連する動作</b>
	オートコミット
パーミッション	基になるテーブルの所有者であるか、DBA 権限または REFERENCES パーミッションを持っている必要があります。



## 関数のサポート

この章では、LONG BINARY データ型と LONG VARCHAR データ型をサポートする Sybase IQ 関数について説明します。

トピック	ページ
<a href="#">ラージ・オブジェクト・データをサポートする関数の概要</a>	68
<a href="#">BIT_LENGTH 関数</a>	69
<a href="#">BYTE_LENGTH 関数</a>	69
<a href="#">BYTE_LENGTH64 関数</a>	69
<a href="#">BYTE_SUBSTR64 関数と BYTE_SUBSTR 関数</a>	70
<a href="#">CHAR_LENGTH 関数</a>	70
<a href="#">CHAR_LENGTH64 関数</a>	71
<a href="#">CHARINDEX 関数</a>	71
<a href="#">LOCATE 関数</a>	72
<a href="#">OCTET_LENGTH 関数</a>	73
<a href="#">PATINDEX 関数</a>	73
<a href="#">SUBSTRING 関数</a>	74
<a href="#">SUBSTRING64 関数</a>	75
<a href="#">ラージ・オブジェクト・カラムの集合関数のサポート</a>	75

## ラージ・オブジェクト・データをサポートする関数の概要

表 9-1 は、LONG BINARY (BLOB) データ型と LONG VARCHAR (CLOB) データ型および LONG BINARY 変数と LONG VARCHAR 変数に対する関数のサポートをまとめたものです。

この章で説明する関数の他に、BFILE 関数を使用して LOB データを抽出できます。詳細については、「[ラージ・オブジェクト・データのエクスポート](#)」(47 ページ) を参照してください。

表 9-1: LOB データ型および変数に対する関数のサポート

機能	BLOB データに対するサポート	BLOB 変数に対するサポート	CLOB データに対するサポート	CLOB 変数に対するサポート
BIT_LENGTH()	あり	あり	あり	あり
BYTE_LENGTH()	あり *	あり *	あり *	あり *
BYTE_LENGTH64()	あり	あり	あり	あり
BYTE_SUBSTR()	あり	あり	あり	あり
BYTE_SUBSTR64()	あり	あり	あり	あり
CHAR_LENGTH()	なし	なし	あり	あり
CHAR_LENGTH64()	なし	なし	あり	あり
CHARINDEX()	あり	あり	あり	あり
LOCATE()	あり	あり	あり	あり
OCTET_LENGTH()	あり	あり	あり	あり
PATINDEX()	なし	なし	あり	あり
SUBSTR() / SUBSTRING()	なし	なし	あり	あり
SUBSTRING64()	あり	あり	あり	あり

\*BYTE\_LENGTH 関数が LONG BINARY カラムと変数および LONG VARCHAR カラムと変数の両方をサポートするのは、クエリの戻り値が 2GB 未満の場合のみです。返された LONG BINARY データまたは LONG VARCHAR データのバイト長が 2GB を超える場合は、BYTE\_LENGTH によって、BYTE\_LENGTH64 関数を使用する必要があることを示すエラーが返されます。

これ以降の項では、LONG BINARY カラムと変数および LONG VARCHAR カラムと変数をサポートする関数について説明します。これらの関数の詳細と使用例については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第 4 章 SQL 関数」を参照してください。

## BIT\_LENGTH 関数

### 機能

BIT\_LENGTH 関数は、ラージ・オブジェクト・カラムまたは変数のパラメータのビット長を表す符号なし 64 ビット値を返します。引数が NULL の場合、BIT\_LENGTH は NULL を返します。

### 構文

**BIT\_LENGTH( *large-object-column* )**

### パラメータ

**large-object-column** LONG VARCHAR カラムまたは変数、もしくは LONG BINARY カラムまたは変数の名前。

### 使用法

BIT\_LENGTH は、すべての Sybase IQ データ型と、任意のデータ・サイズの LONG BINARY 変数および LONG VARCHAR 変数をサポートします。現時点では、SQL 変数で保持できる最大長は 2GB - 1 です。

## BYTE\_LENGTH 関数

### 機能

BYTE\_LENGTH 関数は、文字列のバイト数を返します。

### 使用法

BYTE\_LENGTH 関数が LONG BINARY カラムと変数および LONG VARCHAR カラムと変数の両方をサポートするのは、クエリの戻り値が 2GB 未満の場合のみです。返された LONG BINARY データまたは LONG VARCHAR データのバイト長が 2GB 以上の場合には、BYTE\_LENGTH によって、BYTE\_LENGTH64 関数を使用する必要があることを示すエラーが返されます。

BYTE\_LENGTH 関数の構文と使用法については、『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第4章 SQL 関数」の「BYTE\_LENGTH 関数 [ 文字列 ]」を参照してください。

## BYTE\_LENGTH64 関数

### 機能

BYTE\_LENGTH64 関数は、ラージ・オブジェクト・カラムまたは変数のパラメータのバイト長を表す符号なし 64 ビット値を返します。

### 構文

**BYTE\_LENGTH64( *large-object-column* )**

### パラメータ

**large-object-column** LONG VARCHAR カラムまたは変数、もしくは LONG BINARY カラムまたは変数の名前。

### 使用法

BYTE\_LENGTH64 関数は、LONG BINARY カラムと LONG VARCHAR カラムおよび任意のデータ・サイズの LONG BINARY 変数と LONG VARCHAR 変数の両方をサポートします。現時点では、SQL 変数で保持できる最大長は 2GB - 1 です。

## BYTE\_SUBSTR64 関数と BYTE\_SUBSTR 関数

### 機能

BYTE\_SUBSTR64 関数と BYTE\_SUBSTR 関数は、ラージ・オブジェクト・カラムまたは変数のパラメータのバイト部分文字列を返します。

### 構文

**BYTE\_SUBSTR64( *large-object-column*, *start*, *length* )**  
**BYTE\_SUBSTR( *large-object-column*, *start*, *length* )**

### パラメータ

**large-object-column** LONG VARCHAR カラムまたは変数、もしくは LONG BINARY カラムまたは変数の名前。

**start** 部分文字列の始まりを表す整数式。正の整数は部分文字列が文字列の先頭から始まり、最初のバイトがポジション 1 であることを、負の整数は部分文字列が文字列の末尾から始まり、最後のバイトがポジション 1 であることを示します。

**length** 部分文字列の長さを表す整数式。正の値は、*start* ポジションを始点として返される部分文字列のバイト数を示します。負の値は、*start* ポジションを終点として返される部分文字列のバイト数を示します。

### 使用法

- BYTE\_LENGTH64、BYTE\_SUBSTR64、BYTE\_SUBSTR の各関数のネストされた演算では、ラージ・オブジェクト・カラムまたは変数はサポートされない。
- BYTE\_SUBSTR64 関数と BYTE\_SUBSTR 関数は、LONG BINARY カラムと LONG VARCHAR カラムおよび任意のデータ・サイズの LONG BINARY 変数と LONG VARCHAR 変数の両方をサポートする。現時点では、SQL 変数で保持できる最大長は 2GB - 1。

## CHAR\_LENGTH 関数

### 機能

CHAR\_LENGTH 関数は、LONG VARCHAR カラムまたは変数パラメータの文字長(後続ブランクを含む)を表す符号付き 32 ビット値を返します。

### 構文

**CHAR\_LENGTH( *long-varchar-object* )**

### パラメータ

**long-varchar-object** LONG VARCHAR カラムまたは LONG VARCHAR 変数の名前。

### 使用法

- CHAR\_LENGTH は、LONG VARCHAR カラムと任意のデータ・サイズの LONG VARCHAR 変数をサポートします。現時点では、SQL 変数で保持できる最大長は 2GB - 1 です。
- 引数が NULL の場合、CHAR\_LENGTH は NULL を返します。
- 文字長が 2GB - 1 (2147483647) を超える場合はエラーが返されます。

## CHAR\_LENGTH64 関数

機能	CHAR_LENGTH64 関数は、LONG VARCHAR カラムまたは変数パラメータの文字長(後続ブランクを含む)を表す符号なし 64 ビット値を返します。
構文	<b>CHAR_LENGTH64( <i>long-varchar-object</i> )</b>
パラメータ	<b>long-varchar-object</b> テーブルの LONG VARCHAR カラムまたは LONG VARCHAR 変数の名前。
使用法	<ul style="list-style-type: none"> <li>CHAR_LENGTH64 は、LONG VARCHAR カラムと任意のデータ・サイズの LONG VARCHAR 変数をサポートする。現時点では、SQL 変数で保持できる最大長は 2GB - 1。</li> <li>引数が NULL の場合、CHAR_LENGTH64 は NULL を返す。</li> </ul>

## CHARINDEX 関数

機能	CHARINDEX 関数は、指定された文字列がラージ・オブジェクト・カラムまたは変数のパラメータで最初に出現する位置を格納した 64 ビット符号付き整数を返します。CHAR カラムと VARCHAR カラムの場合、CHARINDEX は 32 ビット符号付き整数で位置を返します。
構文	<b>CHARINDEX( <i>string-expression</i>, <i>large-object-column</i> )</b>
パラメータ	<b>string-expression</b> 検索対象の最大 255 バイトの文字列。 <b>large-object-column</b> LONG VARCHAR カラムまたは変数、もしくは LONG BINARY カラムまたは変数の名前。
使用法	<ul style="list-style-type: none"> <li>CHARINDEX 関数で返されるか指定される位置またはオフセットはすべて、常に文字オフセットであり、マルチバイト・データの場合はバイト・オフセットとは異なることがある。</li> <li>検索されるラージ・オブジェクト・セルに、<i>string-expression</i> のインスタンスが 2 つ以上含まれる場合、CHARINDEX は最初のインスタンスの位置だけを返す。</li> <li>カラムに文字列が含まれない場合、CHARINDEX 関数はゼロ (0) を返す。</li> <li>長さが 255 バイトを超える文字列を検索すると、NULL が返される。</li> <li>長さが 0 の文字列を検索すると、1 が返されます。</li> <li>引数のどれか 1 つでも NULL の場合、結果は NULL になる。</li> <li>CHARINDEX は、検索する LONG VARCHAR カラムと LONG BINARY カラムおよび任意のデータ・サイズの LONG VARCHAR 変数と LONG BINARY 変数をサポートする。現時点では、SQL 変数で保持できる最大長は 2GB - 1。</li> </ul>
参照	『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第4章 SQL 関数」の「CHARINDEX 関数 [文字列]」を参照してください。

## LOCATE 関数

### 機能

LOCATE 関数は、ラージ・オブジェクト・カラムまたは変数のパラメータでの指定された文字列の位置を格納した 64 ビット符号付き整数を返します。CHAR カラムと VARCHAR カラムの場合、LOCATE は、32 ビット符号付き整数で位置を返します。

### 構文

```
LOCATE( large-object-column, string-expression  
[ , numeric-expression ] )
```

### パラメータ

**large-object-column** 検索する LONG VARCHAR カラムまたは変数、もしくは LONG BINARY カラムまたは変数の名前。

**string-expression** 検索対象の最大 255 バイトの文字列。

**numeric-expression** 文字列内で検索を開始する文字位置またはオフセットです。*numeric-expression* は LONG VARCHAR カラムと LONG BINARY カラムの場合は 64 ビット符号付き整数であり、CHAR カラム、VARCHAR カラム、BINARY カラムの場合は 32 ビット符号付き整数です。最初の文字の位置は 1 です。開始オフセットが負の場合、LOCATE は、最初ではなく最後にマッチする文字列のオフセットを返します。負のオフセットは、文字列の末尾から何文字を検索から除外するかを示します。除外する文字数は、(-1 \* オフセット) - 1 で計算します。

### 使用法

- LOCATE 関数で返されるか指定される位置またはオフセットはすべて、常に文字オフセットであり、マルチバイト・データの場合はバイト・オフセットとは異なることがある。
- 検索されるラージ・オブジェクト・セルに、文字列のインスタンスが 2 つ以上含まれる場合、次のように処理される。
  - numeric-expression* を指定した場合、LOCATE は文字列内のそのオフセット位置から検索を開始する。
  - numeric-expression* を指定しなかった場合、LOCATE は最初のインスタンスの位置だけを返す。
- カラムに文字列が含まれない場合、LOCATE はゼロ (0) を返す。
- 長さが 255 バイトを超える文字列を検索すると、NULL が返される。
- 長さが 0 の文字列を検索すると、1 が返されます。
- 引数のどれか 1 つでも NULL の場合、結果は NULL になる。
- LOCATE は、検索する LONG VARCHAR カラムと LONG BINARY カラムおよび任意のデータ・サイズの LONG VARCHAR 変数と LONG BINARY 変数をサポートする。現時点では、SQL 変数で保持できる最大長は 2GB - 1。

### 参照

『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第 4 章 SQL 関数」の「LOCATE 関数 [ 文字列 ]」を参照してください。

## OCTET\_LENGTH 関数

### 機能

OCTET\_LENGTH 関数は、ラージ・オブジェクト・カラムまたは変数のパラメータのバイト長を表す符号なし 64 ビット値を返します。

### 構文

**OCTET\_LENGTH( *column-name* )**

### パラメータ

**large-object-column** LONG VARCHAR カラムまたは変数、もしくは LONG BINARY カラムまたは変数の名前。

### 使用法

- 引数が NULL の場合、OCTET\_LENGTH は NULL を返す。
- OCTET\_LENGTH は、すべての Sybase IQ データ型と、任意のデータ・サイズの LONG VARCHAR 変数および LONG BINARY 変数をサポートする。現時点では、SQL 変数で保持できる最大長は 2GB - 1。

## PATINDEX 関数

### 機能

PATINDEX 関数は、指定されたパターンが LONG VARCHAR カラムまたは変数で最初に出現する位置を格納した 64 ビット符号なし整数を返します。CHAR カラムと VARCHAR カラムの場合、PATINDEX は、32 ビット符号なし整数で位置を返します。

### 構文

**PATINDEX( %*pattern*%, *long-varchar-column* )**

### パラメータ

**pattern** 検索するパターンです。パターンは、ワイルドカードを使用して 126 バイトまでの文字列を指定してください。先頭の % ワイルドカードを省略すると、PATINDEX はパターンがカラム値の最初に出現する場合は 1 を、そうでない場合は 0 を返します。同様に、末尾の % ワイルドカードを省略した場合、パターンがカラム値の最後に出現する必要があります。パターンは LIKE 比較と同じワイルドカードを使用します。

パターンにワイルドカード(パーセント(%)またはアンダースコア(\_))を使用しない場合は、255 バイトの長さまで指定できます。

**long-varchar-column** LONG VARCHAR カラムまたは変数の名前。

### 使用法

- PATINDEX 関数で返されるか指定される位置またはオフセットはすべて、常に文字オフセットで、マルチバイト・データの場合はバイト・オフセットとは異なることがある。
- 検索される LONG VARCHAR セルに、文字パターンのインスタンスが 2 つ以上含まれる場合、PATINDEX は最初のインスタンスの位置だけを返す。
- カラムに文字パターンが含まれない場合、PATINDEX はゼロ (0) を返す。

- 長さが 126 バイトを超えるパターンを検索すると、NULL が返されます。
- 長さが 0 のパターンを検索すると、1 が返される。
- 引数のどれか 1 つでも NULL の場合、結果はゼロ (0) となる。
- PATINDEX は、任意のデータ・サイズの LONG VARCHAR 変数をサポートする。現時点では、SQL 変数で保持できる最大長は 2GB - 1。PATINDEX は、LONG BINARY 変数または検索する LONG BINARY カラムをサポートしていない。
- 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第 4 章 SQL 関数」の「PATINDEX 関数 [ 文字列 ]」を参照してください。
- 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第 2 章 SQL 言語の要素」の「LIKE 条件」を参照してください。

**参照**

- 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第 4 章 SQL 関数」の「PATINDEX 関数 [ 文字列 ]」を参照してください。
- 『リファレンス：ビルディング・ブロック、テーブル、およびプロシージャ』の「第 2 章 SQL 言語の要素」の「LIKE 条件」を参照してください。

## SUBSTRING 関数

**機能**

SUBSTRING 関数は、LONG VARCHAR カラムまたは変数のパラメータの可変長文字列を返します。引数のどれか 1 つでも NULL の場合、SUBSTRING は NULL を返します。

**構文**

{ **SUBSTRING** | **SUBSTR** } ( *long-varchar-column*, *start* [ , *length* ] )

**long-varchar-column** LONG VARCHAR カラムまたは変数の名前。

**start** 部分文字列の始まりを表す整数式。正の整数は部分文字列が文字列の先頭から始まり、最初の文字がポジション 1 であることを、負の整数は部分文字列が文字列の末尾から始まり、最後の文字がポジション 1 であることを示します。

**length** 部分文字列の文字長を表す整数式。正の値は、*start* ポジションを始点として返される文字数を示します。負の値は、*start* ポジションを終点として返される文字数を示します。

**使用法**

SUBSTRING は、任意のデータ・サイズの LONG VARCHAR 変数をサポートします。現時点では、SQL 変数で保持できる最大長は 2GB - 1 です。

SUBSTRING は、LONG BINARY 変数または検索する LONG BINARY カラムをサポートしていません。

## SUBSTRING64 関数

### 機能

SUBSTRING64 関数は、ラージ・オブジェクト・カラムまたは変数のパラメータの可変長文字列を返します。

### 構文

**SUBSTRING64** ( *large-object-column*, *start* [ , *length* ] )

### パラメータ

**large-object-column** LONG VARCHAR カラムまたは変数、もしくは LONG BINARY カラムまたは変数の名前。

**start** 部分文字列の開始を表す 8 バイト整数です。SUBSTRING64 では、*start* オフセット値が負またはゼロの場合、文字列の左側に非文字が埋め込まれているものと解釈します。最初の文字はポジション 1 から始まります。

**length** 部分文字列の長さを表す 8 バイト整数式です。*length* の値が負の場合、エラーが返されます。

### 例

たとえば、*col1* カラムに文字列 “ABCDEFG” が格納されている場合、SUBSTRING64 関数は次の値を返します。

SUBSTRING64( *col1*, 2, 4 ) は文字列 “BCDE” を返します。

SUBSTRING64( *col1*, 1, 3 ) は文字列 “ABC” を返します。

SUBSTRING64( *col1*, 0, 3 ) は文字列 “AB” を返します。

SUBSTRING64( *col1*, -1, 3 ) は文字列 “A” を返します。

### 使用法

- 引数のどれか 1 つでも NULL の場合、SUBSTRING64 は NULL を返す。
- SUBSTRING64、SUBSTRING、SUBSTR、BYTE\_SUBSTR、BYTE\_SUBSTR64 の各関数のネストされた演算では、ラージ・オブジェクト・カラムまたは変数はサポートされない。
- SUBSTRING64 は、検索する LONG VARCHAR カラムと LONG BINARY カラムおよび任意のデータ・サイズの LONG VARCHAR 変数と LONG BINARY 変数をサポートする。現時点では、SQL 変数で保持できる最大長は 2GB - 1。

## ラージ・オブジェクト・カラムの集合関数のサポート

LONG BINARY カラムおよび LONG VARCHAR カラムでサポートされている唯一の集合関数が、COUNT (\*) です。COUNT DISTINCT パラメータはサポートされていません。MIN 集合関数、MAX 集合関数、AVG 集合関数、または SUM 集合関数に LONG BINARY カラムまたは LONG VARCHAR カラムを指定すると、エラーが返されます。



## エラー・メッセージと警告メッセージ

### この付録について

この付録では、LONG BINARY カラムや LONG VARCHAR カラムなどの非構造化データの操作で返される可能性のあるエラー・メッセージと警告メッセージについて説明します。

### 内容

トピック	ページ
<a href="#">エラー 1000195</a>	78
<a href="#">エラー 1000198</a>	78
<a href="#">エラー 1000332</a>	79
<a href="#">エラー 1001013</a>	79
<a href="#">エラー 1001051</a>	80
<a href="#">エラー 1001052</a>	80
<a href="#">エラー 1001053</a>	81
<a href="#">エラー 1001054</a>	81
<a href="#">警告 1001055</a>	82
<a href="#">警告 1001056</a>	82
<a href="#">エラー 1001057</a>	83
<a href="#">エラー 1001058</a>	84
<a href="#">エラー 1009189</a>	84
<a href="#">エラー 1012030</a>	85

**エラー 1000195**

## メッセージ・テキスト

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

項目	値
SQLCode	-1000195L
定数	EMSG_BINARYFILE
SQLState	QDB95
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	20855
重大度コード	14
Parameter 1	例外の発生場所
Parameter 2	ロード仕様の種類
Parameter 3	カラムのデータ型

## 考えられる原因

LOAD TABLE 文のロード仕様が有効なのは、このデータ型のカラムだけです。

**エラー 1000198**

## メッセージ・テキスト

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

項目	値
SQLCode	-1000198L
定数	EMSG_CANNOT_CREATE_JOIN_INDEX
SQLState	QDB98
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	20858
重大度コード	14
Parameter 1	例外の発生場所
Parameter 2	カラムのデータ型

## 考えられる原因

このエラーは、1つまたは複数の LONG VARCHAR データ型カラムまたは LONG BINARY データ型カラムを持つテーブルにジョイン・インデックスを作成しようとすると報告されます。JOIN INDEX はほとんどのデータ型に対応しています。しかし、一部には、この機能が対応していないデータ型 (LONG BINARY、LONG VARCHAR など) もあります。

## エラー 1000332

### メッセージ・テキスト

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

項目	値
SQLCode	-1000332L
定数	EMSG_ODDNUMBER_NIBBLES
SQLState	QDD20
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	21206
重大度コード	14
Parameter 1	例外の発生場所
Parameter 2	カラム名

### 考えられる原因

16 進数形式のバイナリ・データをプライマリ・ロード・ファイルから LONG BINARY カラムにロードする場合、Sybase IQ では、16 進数の合計数は偶数である必要があります。セルの値に奇数個の 16 進数字が含まれている場合に、このエラーが報告されます。LONG BINARY のロードの入力ファイルには、必ず偶数個の 16 進数字が含まれている必要があります。

## エラー 1001013

### メッセージ・テキスト

Invalid data type comparison %1

項目	値
SQLCode	-1001013L
定数	EMSG_TYPECOMPAREERROR
SQLState	QFA13
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	20522
重大度コード	14
Parameter 1	例外の発生場所

### 考えられる原因

LIKE 述部を使用して、LONG BINARY カラムでパターンを検索しようと、このエラーが報告されます。LIKE 述部は、LONG BINARY (BLOB) カラムではサポートされていません。

## エラー 1001051

## メッセージ・テキスト

Query returns %3 data &gt; 2GB. Use %2 %1

項目	値
SQLCode	-1001051L
定数	EMSG_LOB_OVER_2G_W_ARG
SQLState	QFA47
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	21097
重大度コード	14
Parameter 1	SA 解析ソース・コード行
Parameter 2	推奨される関数
Parameter 3	long binary または long varchar データ型

## 考えられる原因

このエラーは、クエリが2ギガバイトを超える LONG BINARY 値または LONG VARCHAR 値を返そうとすると報告されます。

## エラー 1001052

## メッセージ・テキスト

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

項目	値
SQLCode	-1001052L
定数	EMSG_ONLY_SUPPORT_LOB_W_ARG
SQLState	QFA48
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	21098
重大度コード	14
Parameter 1	SA 解析ソース・コード行
Parameter 2	LOB 引数名
Parameter 3	推奨される関数の名前

## 考えられる原因

このエラーがレポートされるのは、Large Object (LOB) 関数のパラメータとして無効なデータ型が使用された場合です。

## エラー 1001053

### メッセージ・テキスト

Wrong number of parameters to function %2 %1

項目	値
SQLCode	-1001053L
定数	EMSG_WRONG_NUM_PARAMS_W_ARG
SQLState	QFA49
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	21099
重大度コード	14
Parameter 1	SA 解析ソース・コード行
Parameter 2	関数名

### 考えられる原因

このエラーがレポートされるのは、Large Object (LOB) 関数に渡された引数の個数が不正な場合です。

## エラー 1001054

### メッセージ・テキスト

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

項目	値
SQLCode	-1001054L
定数	EMSG_LOB_NOT_ALLOWED_GROUP
SQLState	QFA50
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	21100
重大度コード	14
Parameter 1	例外の発生場所

### 考えられる原因

このエラーは、ORDER BY 句、GROUP BY 句、または集約関連の句に LONG BINARY カラムが指定されていた場合に報告されます。

## 警告 1001055

## メッセージ・テキスト

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

項目	値
SQLCode	1001055L
定数	EMSG_LOB_LOAD_ERROR_WARN
SQLState	QFA51
ODBC 2 State	OK
ODBC 3 State	OK
Sybase エラー・コード	21101
重大度コード	10
Parameter 1	long binary または long varchar データ型
Parameter 2	FP インデックス名
Parameter 3	セカンダリ・ファイル名
Parameter 4	rowid

## 考えられる原因

この警告メッセージは、ロード操作中に LONG BINARY 型または LONG VARCHAR 型のセカンダリ・ファイルを開こう（または読み込もう）としてエラーが発生すると返されます。この警告メッセージは、SECONDARY\_FILE\_ERROR オプションがオフの状態でエラーが発生すると、サーバ・ログと IQ メッセージ・ファイルに返されます。

## 警告 1001056

## メッセージ・テキスト

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

項目	値
SQLCode	1001056L
定数	EMSG_LOB_EXTRACT_ERROR_WARN
SQLState	QFA52
ODBC 2 State	OK
ODBC 3 State	OK
Sybase エラー・コード	21102
重大度コード	10
Parameter 1	long binary または long varchar データ型
Parameter 2	FP インデックス名
Parameter 3	セカンダリ・ファイル名

## 考えられる原因

この警告メッセージは、LONG BINARY カラムまたは LONG VARCHAR カラムを抽出しようとしてエラーが発生すると返されます。この警告メッセージは、SECONDARY\_FILE\_ERROR オプションがオフの状態でエラーが発生すると、サーバ・ログと IQ メッセージ・ファイルに返されます。

**エラー 1001057**

## メッセージ・テキスト

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

項目	値
SQLCode	-1001057L
定数	EMSG_LOB_EXTRACT_USE_BFILE
SQLState	QFA53
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	21103
重大度コード	14
Parameter 1	例外の発生場所
Parameter 2	long binary または long varchar データ型

## 考えられる原因

このエラーは、LONG BINARY カラムまたは LONG VARCHAR カラムを対象とするクエリで、データベース・オプション TEMP\_EXTRACT\_NAME1 が ON に設定されていたが BFILE 関数が指定されていなかった場合に報告されます。

**エラー 1001058**

## メッセージ・テキスト

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

項目	値
SQLCode	-1001058L
定数	EMSG_LOB_SECONDARY_FILE_TOOLONG
SQLState	QFA54
ODBC 2 State	OK
ODBC 3 State	OK
Sybase エラー・コード	21104
重大度コード	14
Parameter 1	例外の発生場所
Parameter 2	セカンダリ・ファイル名

## 考えられる原因

このエラーは、LOAD TABLE セカンダリ・ファイルのパス名がオペレーティング・システムのパス名の長さ制限を超えている場合に報告されます。このエラーの発生時に実行されるアクションは、SECONDARY\_FILE\_ERROR データベース・オプションの値によって異なります。

**エラー 1009189**

## メッセージ・テキスト

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

項目	値
SQLCode	-1009189L
定数	EMSG_MAXTERM_ERROR
SQLState	QSB84
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	21210
重大度コード	14
Parameter 1	例外の発生場所

## 考えられる原因

外部のプレフィルタ・ライブラリまたは単語分割ライブラリによるエラー。

## エラー 1012030

### メッセージ・テキスト

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

項目	値
SQLCode	-1012030
定数	EMSG_CAT_PAGESIZETOOSMALL
SQLState	QUA30
ODBC 2 State	ERROR
ODBC 3 State	ERROR
Sybase エラー・コード	20953
重大度コード	14
Parameter 1	例外の発生場所
Parameter 2	カラム番号
Parameter 3	要求ページ・サイズ
Parameter 4	最小許容ページ・サイズ

### 考えられる原因

データベースのページ・サイズが小さすぎるため、LONG BINARY カラムまたは LONG VARCHAR カラムを作成できません。LONG BINARY カラムまたは LONG VARCHAR カラムを作成するのに必要なページ・サイズは 128K 以上です。



# 索引

## A

Adaptive Server Enterprise  
  IMAGE データの挿入 54  
  TEXT データの挿入 55  
ALTER TEXT CONFIGURATION 17, 18  
  構文 59  
ALTER TEXT INDEX 7, 8  
  構文 61

## B

BEGIN PARALLEL IQ 文 63  
BFILE 関数 47  
  構文 47  
  抽出機能 47  
  抽出例 48  
  例 48  
BIT\_LENGTH 関数  
  構文 69  
  説明 69  
BLOB  
  BIT\_LENGTH 関数 69  
  BYTE\_LENGTH64 関数 69  
  BYTE\_LENGTH 関数 69  
  BYTE\_SUBSTR64 関数 70  
  BYTE\_SUBSTR 関数 70  
  CHARINDEX 関数 71  
  IMAGE データの挿入 54  
  LOCATE 関数 72  
  LONG BINARY 53  
  OCTET\_LENGTH 関数 73  
  sp\_iqindexsize 46  
  SUBSTRING64 関数 75  
  TEXT インデックス 57  
  インデックス 53, 57  
  インデックスのサポート 57  
  カラム 53  
  関数のサポート 68  
  クエリ 27, 36  
  サイズ 53

集合関数のサポート 75  
ストアド・プロシージャのサポート 39  
説明 53  
データ型 53  
データ型変換 54  
データのエクスポート 47  
データの更新 53  
データの挿入 53  
データのロード 48  
バイナリ・ラージ・オブジェクト 53  
パフォーマンスのモニタリング 37  
プリフェッチ 53  
変更 53  
変数 56  
変数の関数のサポート 68  
関数  
  BLOB 変数 68  
  BLOB 変数  
    データ型変換 57  
  BYTE\_LENGTH64 関数  
    構文 69  
    説明 69  
  CHAR\_LENGTH64 関数  
    説明 71  
  BYTE\_SUBSTR64 関数  
    構文 70  
    説明 70  
  BYTE\_SUBSTR 関数  
    構文 70  
    説明 70

## C

char  
  単語への分割 39  
CHAR\_LENGTH64 関数  
  構文 71  
CHAR\_LENGTH 関数  
  構文 70  
  説明 70

# 索引

## CHARINDEX 関数

構文 71  
説明 71

## CLOB

BIT\_LENGTH 関数 69  
BYTE\_LENGTH64 関数 69  
BYTE\_LENGTH 関数 69  
BYTE\_SUBSTR64 関数 70  
BYTE\_SUBSTR 関数 70  
CHAR\_LENGTH64 関数 71  
CHAR\_LENGTH 関数 70  
CHARINDEX 関数 71  
LOCATE 関数 72  
LONG VARCHAR 54  
OCTET\_LENGTH 関数 73  
PATINDEX 関数 73  
sp\_iqindexsize 46  
SUBSTRING64 関数 75  
SUBSTRING 関数 74  
TEXT インデックス 57  
TEXT データの挿入 55  
WD インデックス 55, 58  
WORD インデックス 55, 58  
インデックス 55, 57, 58  
インデックスのサポート 53, 57, 58  
カラム 54  
関数のサポート 68  
キャラクタ・ラージ・オブジェクト 54  
クエリ 27, 36  
サイズ 54  
集合関数のサポート 75  
ストアド・プロシージャのサポート 39  
説明 53  
データ型 54  
データ型変換 55  
データのエクスポート 47  
データの更新 55  
データの挿入 55  
データのロード 48  
変更 55  
変数 56  
変数の関数のサポート 68

## 関数

CLOB 変数 68  
CLOB 変数  
データ型変換 57

## CONTAINS

テーブル式 28, 35  
contains-expression  
FROM 句 28  
CONTAINS 条件  
TEXT インデックス 28  
CREATE TEXT CONFIGURATION 12  
構文 62  
CREATE TEXT INDEX 4, 5, 6  
構文 63

## D

DB 領域  
TEXT インデックス 8  
変更 8  
default\_char 11  
default\_nchar 11  
DROP TEXT CONFIGURATION 18  
構文 64  
DROP TEXT INDEX 9  
構文 65

## E

ENABLE\_LOB\_VARIABLES オプション 57  
END PARALLEL IQ  
CREATE TEXT INDEX 63

## F

FROM 句  
CONTAINS 28, 35  
contains-expression 28  
構文 28

## I

IMAGE データ  
ASE からの挿入 54  
LONG BINARY への挿入 54  
index  
リスト 7

## L

- list
  - TEXT インデックス 7
  - 外部ライブラリ 43
  - テキスト設定オブジェクト 16
- LOAD TABLE
  - 拡張構文 49
  - セカンダリ・ロード・ファイル 49
  - の例 50
  - プライマリ・ロード・ファイル 49
- LOB
  - 一般的データ・ソース 1
  - インデックスのサポート 57
  - 概要 1
  - 関数のサポート 67
  - データのエクスポート 47
  - データの更新 53, 55
  - データの挿入 53, 55
  - データのロード 48
- LOB 変数
  - データ型変換 57
- LOB データの圧縮 44
  - 設定の表示 45
  - 設定の変更 44
- LOB の圧縮
  - 設定の表示 45
  - 設定の変更 44
  - 無効化 44
  - 有効化 44
- LOB のデータ圧縮 44
  - 設定の表示 45
  - 設定の変更 44
- LOCATE 関数
  - 構文 72
  - 説明 72
- LONG BINARY
  - BYTE\_SUBSTR 関数 70
  - BYTE\_SUBSTR64 関数 70
  - BIT\_LENGTH 関数 69
  - BLOB 53
  - BYTE\_LENGTH64 関数 69
  - BYTE\_LENGTH 関数 69
  - CHARINDEX 関数 71
  - DELETE 53
  - IMAGE データの挿入 54
  - INSERT 53
  - LOAD TABLE 53
- LOCATE 関数 72
- OCTET\_LENGTH 関数 73
- SELECT...INTO 53
- sp\_iqindexsize 46
- SUBSTRING64 関数 75
- TEXT インデックス 57
- TRUNCATE 53
- UPDATE 53
- インデックス 53, 57
- インデックスのサポート 57
- カラム 53
- クエリ 27, 36
- サイズ 53
- 集合関数のサポート 75
- ストアド・プロシージャのサポート 39
- データ型変換 54
- データのエクスポート 47
- データの更新 53
- データの挿入 53
- データのロード 48
- バイナリ・ラージ・オブジェクト 53
- パフォーマンスのモニタリング 37
- 変更 53
- 変数 56
- LONG BINARY 変数
  - データ型変換 57
- LONG VARCHAR
  - BYTE\_LENGTH 関数 69
  - BYTE\_LENGTH64 関数 69
  - BYTE\_SUBSTR 関数 70
  - BYTE\_SUBSTR64 関数 70
  - CHARINDEX 関数 71
  - LOCATE 関数 72
  - PATINDEX 関数 73
  - SUBSTRING 関数 74
  - BIT\_LENGTH 関数 69
  - CHAR\_LENGTH64 関数 71
  - CHAR\_LENGTH 関数 70
  - CLOB 54
  - DELETE 55
  - INSERT 55
  - LOAD TABLE 55
  - OCTET\_LENGTH 関数 73
  - SELECT...INTO 55
  - sp\_iqindexsize 46
  - SUBSTRING64 関数 75
  - TEXT インデックス 57

## 索引

TEXT データの挿入 55  
TRUNCATE 55  
UPDATE 55  
WD インデックス 55, 58  
WORD インデックス 55, 58  
インデックス 55, 57, 58  
インデックスのサポート 57, 58  
カラム 54  
キャラクタ・ラージ・オブジェクト 54  
クエリ 27, 36  
サイズ 54  
集合関数のサポート 75  
ストアド・プロシージャのサポート 39  
データ型変換 55  
データのエクスポート 47  
データの更新 55  
データの挿入 55  
データのロード 48  
変更 55  
変数 56  
LONG VARCHAR のエクスポート 47  
LONG VARCHAR 変数  
データ型変換 57

## M

MAX\_PREFIX\_PER\_CONTAINS\_PHRASE オプション 21

## N

nchar  
単語への分割 40

## O

OCTET\_LENGTH 関数  
構文 73  
説明 73

## P

PATINDEX 関数  
構文 73  
説明 73

## S

sa\_char\_terms ストアド・プロシージャ 39  
sa\_external\_library\_unload ストアド・プロシージャ 25, 43  
sa\_list\_external\_library ストアド・プロシージャ 25, 43  
sa\_nchar\_terms ストアド・プロシージャ 40  
sa\_text\_index\_stats ストアド・プロシージャ 41  
sa\_text\_index\_vocab ストアド・プロシージャ 42  
SECONDARY\_FILE\_ERROR オプション 50  
SELECT 文  
  FROM 句の構文 28  
sp\_iqindexsize  
  BLOB 46  
  CLOB 46  
  LONG BINARY 46  
  LONG VARCHAR 46  
  キャラクタ・ラージ・オブジェクト 46  
  バイナリ・ラージ・オブジェクト 46  
sp\_iqindexsize ストアド・プロシージャ 46  
sp\_iqsetcompression ストアド・プロシージャ 44  
sp\_iqshowcompression ストアド・プロシージャ 45  
STRING\_RTRUNCATION オプション 54, 56  
SUBSTRING64 関数  
  構文 75  
  説明 75  
SUBSTRING 関数  
  構文 74  
  説明 74

## T

TEXT インデックス 3, 57  
CONTAINS 条件 28  
DB 領域の変更 8  
削除 9, 65  
作成 4, 5, 6, 63  
制限 6  
単語 39  
テキスト設定オブジェクト 11  
統計 41, 42  
変更 61  
編集 7  
リスト 7  
リフレッシュ 9  
ローの削除 9, 10  
TEXT\_DELETE\_METHOD オプション 9, 10

TEXT データ  
ASE からの挿入 55  
LONG VARCHAR への挿入 55  
TRIM\_PARTIAL\_MBC オプション 51

## W

WD インデックス 58

## あ

アップグレード  
LONG BINARY 53  
既存の LONG BINARY カラム 53  
アンロード  
外部ライブラリ 25, 43

## い

インスタンス  
外部ライブラリ 43  
インデックス  
BLOB 53, 57  
CLOB 55, 57, 58  
LOB 57  
LONG BINARY 53, 57  
LONG VARCHAR 55, 57, 58  
TEXT 3, 57  
WD 55, 58  
WORD 55, 58  
キャラクタ・ラージ・オブジェクト 55, 57, 58  
全文検索 2, 27  
バイナリ・ラージ・オブジェクト 53, 57  
包含 55, 58  
ラージ・オブジェクト・データ 57  
リフレッシュ 9

## え

エクスポート  
BFILE 関数 47  
BFILE の例 48  
BLOB 47  
CLOB 47  
LOB 47

LONG BINARY 47  
ラージ・オブジェクト・データ 47  
エラー・メッセージ 77

## お

オプション  
ENABLE\_LOB\_VARIABLES 57  
MAX\_PREFIX\_PER\_CONTAINS\_PHRASE 21  
TEXT\_DELETE\_METHOD 10  
非構造化データ分析 1

## か

外部ライブラリ 23  
アンロード 25, 43  
識別 43  
制限 24  
マルチプレックス・サーバ 25  
無効化 25  
有効化 25  
リスト 43  
関数  
BYTE\_SUBSTR 70  
BYTE\_SUBSTR64 70  
BFILE 47  
BFILE の例 48  
BIT\_LENGTH 69  
BLOB 用 68  
BYTE\_LENGTH 69  
BYTE\_LENGTH64 69  
BYTE\_SUBSTR 70  
BYTE\_SUBSTR64 70  
CHAR\_LENGTH 70  
CHAR\_LENGTH64 71  
CHARINDEX 71  
CLOB 用 68  
LOB 用関数 67  
LOCATE 72  
LONG BINARY 集合関数のサポート 75  
LONG VARCHAR 集合関数のサポート 75  
OCTET\_LENGTH 73  
PATINDEX 73  
SUBSTRING 74  
SUBSTRING64 75

## き

キャラクタ・ラージ・オブジェクト  
CHAR\_LENGTH64 関数 71  
BIT\_LENGTH 関数 69  
BYTE\_LENGTH 関数 69  
BYTE\_LENGTH64 関数 69  
BYTE\_SUBSTR 関数 70  
BYTE\_SUBSTR64 関数 70  
CHAR\_LENGTH 関数 70  
CHARINDEX 関数 71  
LOCATE 関数 72  
OCTET\_LENGTH 関数 73  
PATINDEX 関数 73  
SUBSTRING 関数 74  
SUBSTRING64 関数 75  
CLOB 54  
LONG VARCHAR 54  
sp\_iqindexsize 46  
TEXT インデックス 57  
TEXT データの挿入 55  
WD インデックス 55, 58  
WORD インデックス 55, 58  
インデックス 55, 57, 58  
インデックスのサポート 53, 57, 58  
カラム 54  
クエリ 27, 36  
サイズ 54  
集合関数のサポート 75  
ストアド・プロシージャのサポート 39  
説明 53  
データ型 54  
データ型変換 55  
変更 55  
変数 56  
キャラクタ・ラージ・オブジェクト変数  
データ型変換 57

## く

クエリ  
BLOB 27, 36  
CLOB 27, 36  
LONG BINARY 27, 36  
LONG VARCHAR 27, 36  
キャラクタ・ラージ・オブジェクト 27, 36  
バイナリ・ラージ・オブジェクト 27, 36

## け

警告メッセージ 77  
検索  
CONTAINS 句 35  
CONTAINS 条件 28  
TEXT インデックス 3  
全文 2, 27  
プレフィックス単語の制限 21  
検索条件  
CONTAINS 句 35  
CONTAINS 条件 28

## こ

更新  
BLOB 53  
CLOB 55  
immediate 41, 42  
LOB 53, 55  
LONG BINARY 53  
LONG VARCHAR 55  
ラージ・オブジェクト・データ 53, 55  
互換性  
Adaptive Server Enterprise との互換性 2  
ASE との互換性 2  
SA との互換性 2  
SQL Anywhere との互換性 2

## さ

削除  
TEXT インデックス 9, 65  
テキスト設定オブジェクト 18, 64  
作成  
TEXT インデックス 4, 5, 6, 63  
テキスト設定オブジェクト 12, 62

## し

識別  
外部ライブラリ 43

## す

ストアド・プロシージャ  
 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 ストアド・プロシージャ 41  
 sa\_text\_index\_vocab 42  
 sp\_iqindexsize 46  
 sp\_iqsetcompression 44  
 sp\_iqshowcompression 45  
 キャラクタ・ラージ・オブジェクト 39  
 バイナリ・ラージ・オブジェクト 39  
 ストップリスト 11, 16  
 変更 18

## せ

制限  
 TEXT インデックス 6  
 外部ライブラリ 24  
 全文検索 2, 3, 27  
 タイプ 27

## そ

挿入  
 BLOB 53  
 CLOB 55  
 LOB 53, 55  
 LONG BINARY 53  
 LONG VARCHAR 55  
 ラージ・オブジェクト・データ 53, 55  
 即時更新 9, 41, 42

## た

単語  
 TEXT インデックス 39  
 最小の長さ 14  
 最大長 15  
 ストップリスト 18

全文検索 27  
 ブレーク 39, 40  
 無視 16  
 ロー位置 39, 40  
 単語分割ライブラリ 11, 13, 14, 15, 23  
 制限 24

## ち

抽出機能  
 BFILE 関数 47

## つ

追加  
 TEXT インデックス 4, 5, 6  
 テキスト設定オブジェクト 12

## て

データ型  
 BLOB 53  
 CLOB 54  
 LONG BINARY 53  
 LONG VARCHAR 54  
 データ型変換  
 LONG BINARY 変数 57  
 LONG BINARY から BINARY に変換 54  
 LONG BINARY から VARBINARY に変換 54  
 LONG VARCHAR から CHAR に変換 55  
 LONG VARCHAR から VARCHAR に変換 55  
 データベース・オプション  
 ENABLE\_LOB\_VARIABLES 57  
 MAX\_PREFIX\_PER\_CONTAINS\_PHRASE 21  
 TEXT\_DELETE\_METHOD 10  
 テキスト検索  
 FROM contains-expression 28  
 テキスト設定オブジェクト 11  
 削除 18, 64  
 作成 12, 62  
 設定 13, 14, 15, 16  
 デフォルト 11  
 変更 17, 59  
 リスト 16  
 例 19, 20

と

統計  
TEXT インデックス 41, 42

は

バイナリ・ラージ・オブジェクト  
BYTE\_LENGTH64 関数 69  
BYTE\_SUBSTR 関数 70  
BYTE\_SUBSTR64 関数 70  
BIT\_LENGTH 関数 69  
BLOB 53  
BYTE\_LENGTH 関数 69  
IMAGE データの挿入 54  
LONG BINARY 53  
OCTET\_LENGTH 関数 73  
sp\_iqindexsize 46  
SUBSTRING64 関数 75  
TEXT インデックス 57  
インデックス 53, 57  
インデックスのサポート 57  
カラム 53  
クエリ 27, 36  
サイズ 53  
集合関数のサポート 75  
ストアド・プロシージャのサポート 39  
説明 53  
データ型 53  
データ型変換 54  
パフォーマンスのモニタリング 37  
変更 53  
変数 56  
バイナリ・ラージ・オブジェクト変数  
データ型変換 57  
パフォーマンスのモニタリング  
BLOB 37  
LONG BINARY 37  
バイナリ・ラージ・オブジェクト 37

ひ

非構造化データ分析オプション 1  
ライセンス 1  
標準 2

ふ

プリフェッチ 53  
ブレーク  
単語 39, 40  
プレフィックス  
単語の制限 21  
プレフィルタ・ライブラリ 11, 13, 23  
制限 24  
単語の最小長 14  
単語の最大長 15  
分析  
非構造化データ分析オプション 1

へ

変更  
DB 領域 8  
TEXT インデックス 7, 61  
ストップリスト 18  
テキスト設定オブジェクト 17, 59  
編集  
TEXT インデックス 7  
テキスト設定オブジェクト 17  
変数  
BLOB 56  
BLOB に対する関数のサポート 68  
BLOB の変換 57  
CLOB 56  
CLOB に対する関数のサポート 68  
CLOB の変換 57  
LONG BINARY 56  
LONG BINARY の変換 57  
LONG VARCHAR 56  
LONG VARCHAR の変換 57  
キャラクタ・ラージ・オブジェクト 56  
キャラクタ・ラージ・オブジェクトの変換 57  
バイナリ・ラージ・オブジェクト 56  
バイナリ・ラージ・オブジェクトの変換 57

**ま**

- マニュアル
  - SQL Anywhere ix
  - Sybase IQ vii
- マルチバイト文字
  - TRIM\_PARTIAL\_MBC オプション 51
  - 一部分の削除 51
  - ロード時のトランケート 51
- マルチプレックス・サーバ
  - 外部ライブラリ 25

**む**

- 無効化
  - compression 44
  - 外部ライブラリ 25

**め**

- メッセージ
  - エラー 77
  - 警告 77

**め**

- 有効化
  - compression 44
  - 外部ライブラリ 25

**ら**

- ラージ・オブジェクト・データ
  - インデックスのサポート 57
  - エクスポート 47
  - 更新 53, 55
  - 挿入 53, 55
  - ロード 48
  - ライセンス 1
  - ライブラリ、外部 11

**り**

- リフレッシュ
  - immediate 9
  - TEXT インデックス 9

**れ**

- 例
  - テキスト設定オブジェクト 19, 20

**ろ**

- ロード
  - BLOB 48
  - CLOB 48
  - LOAD TABLE の例 50
  - LOB 48
  - LONG BINARY 48
  - LONG VARCHAR 48
  - SECONDARY\_FILE\_ERROR オプション 50
  - TRIM\_PARTIAL\_MBC オプション 51
  - エラーの制御 50
  - 後続ブランクの削除 51
  - 文字データのトランケート 51
  - ラージ・オブジェクト・データ 48

