



非構造化データ分析

---

**SAP Sybase IQ 16.0 SP03**

ドキュメント ID：DC00285-01-1603-01

改訂：2013 年 12 月

Copyright © 2013 by SAP AG or an SAP affiliate company. All rights reserved.

このマニュアルの内容を SAP AG による明示的な許可なく複製または転載することは、形態や目的を問わず禁じられています。ここに記載された情報は事前の通知なしに変更されることがあります。

SAP AG およびディストリビュータが販売しているソフトウェア製品には、他のソフトウェアベンダ独自のソフトウェアコンポーネントが含まれているものがあります。国内製品の仕様は変わることがあります。

これらの資料は SAP AG および関連会社 (SAP グループ) が情報のみを目的として提供するものであり、いかなる種類の表明または保証も行わないものではなく、SAP グループはこの資料に関する誤りまたは脱落について責任を負わないものとします。SAP グループの製品およびサービスに関する保証は、かかる製品およびサービスに付属している明確な保証文書がある場合、そこで明記されている保証に限定されます。ここに記載されているいかなる内容も、追加保証を構成するものとして解釈されるものではありません。

ここに記載された SAP および他の SAP 製品とサービス、ならびに対応するロゴは、ドイツおよび他の国における SAP AG の商標または登録商標です。その他の商標に関する情報および通知については、<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> を参照してください。

# 目次

非構造化データ分析の概要 .....	1
対象読者 .....	1
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>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 (Interactive SQL) .....	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 .....	5
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 (Interactive SQL) .....	6
TEXT インデックスの編集 (Interactive SQL) .....	7
TEXT インデックスのロケーションの変更 (Interactive SQL) .....	7
TEXT インデックスの削除 (Interactive SQL) .....	7
TEXT インデックスの更新 .....	8
TEXT_DELETE_METHOD データベースオプション .....	8
NGRAM TEXT インデックス .....	9
テキスト設定オブジェクト .....	9
デフォルトのテキスト設定オブジェクト .....	10
テキスト設定の作成 (Interactive SQL) .....	11
テキスト設定オブジェクトの設定 .....	11

テキスト設定のリストの表示 (Interactive SQL)	15
テキスト設定の変更 (Interactive SQL)	15
ストップリストの変更 (Interactive SQL)	15
テキスト設定の削除 (Interactive SQL)	16
テキスト設定オブジェクトの例	16
MAX_PREFIX_PER_CONTAINS_PHRASE	
データベースオプション	19
<b>外部ライブラリ</b>	<b>21</b>
プレフィルタと単語分割の外部ライブラリ	21
外部ライブラリの制限	22
マルチプレックスサーバでの外部ライブラリ	22
起動時の外部ライブラリの有効化と無効化	22
外部ライブラリのアンロード	23
<b>非構造化データのクエリ</b>	<b>25</b>
全文検索	25
全文検索のタイプ	25
NGRAM TEXT インデックス検索	49
TEXT インデックス上のあいまい検索	50
TEXT インデックス上の非あいまい検索	51
LONG BINARY カラムに対するクエリ	52
LONG VARCHAR カラムに対するクエリ	53
CONTAINS 述部のサポート	53
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	54
<b>ストアドプロシージャのサポート</b>	<b>55</b>
TEXT インデックス内の単語の管理	55
sa_char_terms システムプロシージャ	55
sa_nchar_terms システムプロシージャ	56
sa_text_index_stats システムプロシージャ	57
sa_text_index_vocab システムプロシージャ	58
外部ライブラリの確認	59

sa_external_library_unload システムプロシ ージャ .....	60
sa_list_external_library システムプロシ ージャ ...	60
ラージオブジェクトデータの圧縮 .....	61
sp_iqsetcompression プロシ ージャ .....	61
sp_iqshowcompression プロシ ージャ .....	62
ラージオブジェクトカラムについての情報 .....	63
LONG BINARY カラムのサイズ .....	63
LONG VARCHAR カラムのサイズ .....	64
<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>65</b>
ラージオブジェクトデータのエクスポート .....	65
BFILE 関数 .....	65
ラージオブジェクトデータのロード .....	67
拡張 LOAD TABLE 構文 .....	67
ラージオブジェクトデータのロード例 .....	68
ロードエラーの制御 .....	69
後続ブランクを含むラージオブジェクトデー タのロード .....	69
引用符を含むラージオブジェクトデータの ロード .....	70
部分的なマルチバイト文字データのトラン ケート .....	70
ラージオブジェクト変数のロードサポート .....	71
<b>ラージオブジェクトデータ型 .....</b>	<b>73</b>
ラージオブジェクトデータ型、LONG BINARY と BLOB .....	73
LONG BINARY データ型の変換 .....	74
ラージオブジェクトデータ型、LONG VARCHAR と CLOB .....	74
LONG VARCHAR データ型の変換 .....	75
ラージオブジェクト変数 .....	76
ラージオブジェクト変数のデータ型変換 .....	77

ラジオオブジェクトカラムのインデックスのサポート .....	78
ラジオオブジェクトカラムのTEXT インデックスのサポート .....	78
LONG VARCHAR (CLOB) カラムのWD インデックスのサポート .....	78
<b>SQL 文のサポート .....</b>	<b>79</b>
ALTER TEXT CONFIGURATION 文 .....	79
ALTER TEXT INDEX 文 .....	82
CREATE TEXT CONFIGURATION 文 .....	84
CREATE TEXT INDEX 文 .....	85
DROP TEXT CONFIGURATION 文 .....	87
DROP TEXT INDEX 文 .....	88
<b>関数のサポート .....</b>	<b>91</b>
ラジオオブジェクトデータをサポートする関数の概要 .....	91
BIT_LENGTH 関数 .....	92
BYTE_LENGTH 関数 [文字列] .....	92
BYTE_LENGTH64 関数 [文字列] .....	93
BYTE_SUBSTR64 関数と BYTE_SUBSTR 関数 [文字列] .....	94
CHAR_LENGTH 関数 [文字列] .....	95
CHAR_LENGTH64 関数 [文字列] .....	95
CHARINDEX 関数 [文字列] .....	96
LOCATE 関数 [文字列] .....	97
OCTET_LENGTH 関数 [文字列] .....	98
PATINDEX 関数 [文字列] .....	98
SUBSTRING 関数 [文字列] .....	99
ANSI_SUBSTRING オプション [TSQL] .....	100
SUBSTRING64 関数 [文字列] .....	101
ラジオオブジェクトカラムの集合関数のサポート .....	102

ラジオオブジェクトカラムのユーザ定義関数のサ ポート .....	102
<b>索引</b> .....	<b>103</b>

# 目次



# 非構造化データ分析の概要

SAP® Sybase® IQ の非構造化データ分析について紹介し、ラージオブジェクトデータの互換性と準拠について説明します。

## 対象読者

---

このマニュアルは、SAP 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ユーザを対象としています。

非構造化データ分析機能に関連する、利用可能な構文、パラメータ、関数、ストアプロシージャ、インデックス、オプションについて説明します。このマニュアルをリファレンスとして使用し、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。

## 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によって SAP Sybase IQ の機能が拡張され、データベース内のバイナリラージオブジェクト (BLOB) とキャラクターラージオブジェクト (CLOB) の格納、取得、および全文検索を行うことができます。

**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の格納および抽出のみを行います。
- 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) を提供します。

一般的な LOB データソースとして、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。SAP 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、SAP Sybase IQ を使用することで、同じアプリケーションとインタ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。SAP Sybase IQ の全文検索機能

## 非構造化データ分析の概要

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と語句を検索します。

**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

---

SAP Sybase SQL Anywhere® (SA) と SAP Adaptive Server® Enterprise (ASE) は、テキストラージオブジェクトとバイナリラージオブジェクトを格納します。

SQL Anywhere では、ラージオブジェクト (最大長 2GB) を LONG VARCHAR 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 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 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。

プロキシテーブルの LONG BINARY カラムは、Microsoft SQL Server テーブルの VARBINARY (max) カラムにマップされます。

## 標準への準拠

---

SAP Sybase IQ の 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

**TEXT** インデックスとテキスト設定オブジェクトの使用方法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。

## 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ャ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 のカラムの場合、**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

構文と機能の観点での WD インデックスと TEXT インデックスの比較

表 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、構文例： tbl.col CONTAINS ('great', 'white', 'whale')	あり、構文例： CONTAINS (tbl.col, 'great white whale')
一般的なブール式	あり、構文例： tbl.col CONTAINS ('great') AND ( tbl.col CONTAINS ('white') OR tbl.col CONTAINS ('whale') AND NOT tbl.col CONTAINS ('ship'))	あり、構文例： 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	なし	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%'	あり (TEXT インデックスが外部ドキュメントプレフィルタなしの NGRAM TEXT インデックスの場合)、構文例： c LIKE '%apple%fruit'
近接する単語の検索	なし	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	なし	あり

TEXT インデックスでは、プレフィ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 (Interactive SQL)

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。

1. **CREATE ANY INDEX** または **CREATE ANY OBJECT** システム権限を持つユーザとして、あるいは基本となるテーブルの所有者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。

次の例は、**TEXT** インデックス `myTxtIdx` を、`iqdemo` データベースの `Customers` テーブルの `CompanyName` カラムに作成します。`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

次の式を使用して、**TEXT** インデックスのメインストアのサイズを見積もります。

$$\text{Number of bytes} = (15+L)*U + U*\text{PAGESIZE} * R + T$$

各オブジェクトの意味は、次のとおりです。

- **L** = ボキャブラリの単語の平均長
- **U** = ボキャブラリの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** = テキスト設定に対する単語の最大長 (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

テキスト設定オブジェクトと **TEXT** インデックスには、設計上の制限があります。

- SAP Sybase IQ 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ていません。
- **TEXT** インデックスの手動更新オプションや自動更新オプションはサポートされていません。
- **sp\_iqrebuildindex** を使用して **TEXT** インデックスを構築することはできません。
- **BEGIN PARALLEL IQ...END PARALLEL IQ** 内で **TEXT** インデックスを作成することはできません。
- **NGRAM** 単語分割は **TEXT** インデックスに構築されます。そのため、テキスト設定オブジェクトの設定を使用して、**NGRAM** と **GENERIC** のどちらの **TEXT** インデックスを使用するかを定義します。
- **NGRAM TEXT** インデックス検索は、主に、単語に入力ミスが多い場合に役立ちます。SAP Sybase IQ では、同意語や反意語のような検索はサポートされていません。

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

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。

1. 次のいずれかのシステム権限を持つユーザとしてデータベースに接続します。

- CREATE ANY INDEX
- ALTER ANY INDEX
- DROP ANY INDEX
- CREATE ANY OBJECT
- ALTER ANY OBJECT
- DROP ANY OBJECT
- MANAGE ANY DBSPACE

2. **SELECT** 文を実行します。

すべての **TEXT** インデックスをリストするには、次の構文を使用します。

```
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 (Interactive SQL)

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。

1. ALTER ANY INDEX または ALTER ANY OBJECT システム権限を持つユーザとして、あるいは基本となるテーブルの所有者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。

TEXT インデックス myTxtIdx の名前を MyTextIndex に変更するには、次の構文を使用します。

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

## TEXT インデックスのロケーションの変更 (Interactive SQL)

TEXT インデックスが格納される DB 領域を変更します。

1. MANAGE ANY DBSPACE システム権限を持つユーザとしてデータベースに接続します。
2. MOVE TO 句を指定して ALTER TEXT INDEX 文を実行します。

TEXT インデックス MyTextIndex を tispac という名前の DB 領域に移動するには、次の構文を使用します。

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

## TEXT インデックスの削除 (Interactive SQL)

データベースから TEXT インデックスを削除します。

1. DROP ANY INDEX または DROP ANY OBJECT システム権限を持つユーザとして、あるいは基本となるテーブルの所有者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。

MyTextIndex TEXT インデックスを削除するには、次の構文を使用します。

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

## TEXT インデックスの更新

SAP 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、基本と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。

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せん。インデックスは作成時に値が設定されます。また、**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せん。初期更新の間、テーブルに排他ロックが保持されます。

## TEXT\_DELETE\_METHOD データベースオプション

**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。

*デフォルト値*

0

*スコープ*

**PUBLIC** または他のユーザ/ロールに対してこのオプションを設定するには **SET ANY PUBLIC OPTION** システム権限が必要です。**PUBLIC** ロールに対してのみ一時的に設定できます。すぐに有効になります。

*説明*

**TEXT\_DELETE\_METHOD** では、**TEXT** インデックスの削除処理で使用するアルゴリズムを指定します。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

次の文では、**TEXT** インデックスからのラージデリートが強制されます。

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

## NGRAM TEXT インデックス

NGRAM TEXT インデックスには、カラムのテキストをテキスト値 N の n-gram に分割したテキストが格納されています。ここで、N はユーザ指定の値です。

クエリの **CONTAINS** 句で指定したテキスト値の n-gram をインデックスに格納されている n-gram に一致させることによって、**NGRAMTEXT** インデックス上で検索を実行できます。

**NGRAM TEXT** インデックスは、ヨーロッパ言語と非ヨーロッパ言語の両方のテキストに対するあいまい検索機能を備えています。

**注意：** **NGRAM TEXT** インデックス検索は、主に、単語に入力ミスが多い場合に役立ちます。SAP Sybase IQ では、同意語や反意語のような検索はサポートされていません。

**NGRAM** 単語分割は **TEXT** インデックスに構築されます。そのため、**NGRAM** インデックスまたは **GENERICTEXT** インデックスを使用するかどうかを定義するには、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

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** インデックスの作成時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。SAP 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。

## TEXT インデックスと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、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、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します。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ます。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ます。
- 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単語、または指定された規則に従って区切られた単語に分割します。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ます。
- 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します。

### デフォルトのテキスト設定オブジェクト

SAP Sybase IQ には、デフォルトのテキスト設定オブジェクトが用意されています。

デフォルトのテキスト設定オブジェクト `default_char` は、非 NCHAR データとともに使用されます。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きに作成されます。

テキスト設定オブジェクト `default_nchar` 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。テーブルの **TEXT** インデックスで `default_nchar` テキスト設定を使用することはできません。

表「デフォルトのテキスト設定オブジェクトの設定」は、ほとんどの文字ベースの言語に最適な、`default_char` と `default_nchar` のデフォルト設定を示します。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：デフォルトのテキスト設定オブジェクトの設定

設定	インストールされる値
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 (Interactive SQL)

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。

1. **CREATE TEXT CONFIGURATION**、**CREATE ANY TEXT CONFIGURATION**、または **CREATE ANY OBJECT** システム権限を持つ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。

`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` という名前のテキスト設定オブジェクトを作成するには、次の構文を使用します。

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

## テキスト設定オブジェクトの設定

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、また全文検索クエリがどのように解釈されるかについて説明します。

### 参照：

- テキスト設定オブジェクトの設定の解釈 (16 ページ)

## 単語分割アルゴリズム (TERM BREAKER)

**TERM BREAKER** 設定は、文字列を単語に分割するために使用されるアルゴリズムを指定します。

SAP Sybase IQ では、単語の格納に関して **GENERIC** (デフォルト) または **NGRAM** を選択できます。

---

**注意：** **NGRAM** 単語分割には、 $n$ -gram が格納されます。 $n$ -gram は、長さ  $n$  の文字のグループです。 $n$  は、**MAXIMUM TERM LENGTH** の値です。

---

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せん。

表 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
<ul style="list-style-type: none"> <li>• <b>GENERIC TEXT</b> インデックス – <b>GENERIC TEXT</b> インデックス (デフォルト) を作成すると、データベースサーバは、英数字以外の文字に挟まれた英数字の文字グループを単語として処理する。単語が定義されると、単語長の設定値を超える単語やストップリストに含まれる単語は、カウントはされるが、<b>TEXT</b> インデックスには挿入されない。</li> </ul> <p><b>GENERIC TEXT</b> インデックスのパフォーマンスは、<b>NGRAM TEXT</b> インデックスより高速になる可能性がある。ただし、<b>GENERIC TEXT</b> インデックスではあいまい検索は実行できない。</p>	<ul style="list-style-type: none"> <li>• <b>GENERIC TEXT</b> インデックス – <b>GENERIC TEXT</b> 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る。クエリ単語と <b>TEXT</b> 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る。</li> </ul>
<ul style="list-style-type: none"> <li>• <b>NGRAM TEXT</b> インデックス – <b>NGRAM TEXT</b> インデックスを作成すると、データベースサーバは、英数字以外の文字に挟まれた英数字の文字グループをすべて単語として処理する。単語が定義されると、データベースサーバは単語を N-gram に分割する。これにより、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 は破棄される。</li> </ul> <p>たとえば、<b>MAXIMUM TERM LENGTH</b> が 3 の <b>NGRAM TEXT</b> インデックスの場合、文字列 'my red table' は、<b>TEXT</b> インデックスで red tab abl ble という N-gram として表される。</p>	<ul style="list-style-type: none"> <li>• <b>NGRAM TEXT</b> インデックス – <b>NGRAM TEXT</b> 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る。クエリ単語の N-gram とインデックス付けされた単語の N-gram を比較して照合が実行される。</li> </ul>

**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 より大きくする必要があります。**MAXIMUM TERM LENGTH** より大きい値に設定すると、**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 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
<ul style="list-style-type: none"> <li>• <b>GENERIC TEXT</b> インデックス – <b>GENERIC TEXT</b> インデックスの場合、<b>TEXT</b> インデックスには、<b>MINIMUM TERM LENGTH</b> よりも短い単語は格納されない。</li> </ul>	<ul style="list-style-type: none"> <li>• <b>GENERIC TEXT</b> インデックス – <b>GENERIC TEXT</b> インデックスに対してクエリする場合、<b>MINIMUM TERM LENGTH</b> よりも短いクエリ単語は <b>TEXT</b> インデックスに存在できないため、無視される。</li> </ul>
<ul style="list-style-type: none"> <li>• <b>NGRAM TEXT</b> インデックス – <b>NGRAM TEXT</b> インデックスの場合、この設定は無視される。</li> </ul>	<ul style="list-style-type: none"> <li>• <b>NGRAM TEXT</b> インデックス – <b>MINIMUM TERM LENGTH</b> 設定は、<b>NGRAM TEXT</b> インデックスに対する全文クエリには影響しない。</li> </ul>

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

**MAXIMUM 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大長 (文字数) を指定します。

**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 以下にする必要があります。**MAXIMUM TERM LENGTH** を **MINIMUM TERM LENGTH** よりも小さい値に設定すると、**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 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
<ul style="list-style-type: none"> <li>• <b>GENERIC TEXT</b> インデックス – <b>GENERIC TEXT</b> インデックスの場合、<b>MAXIMUM TERM LENGTH</b> は、<b>TEXT</b> インデックスに挿入される単語の最大長 (文字数) を指定する。</li> </ul>	<ul style="list-style-type: none"> <li>• <b>GENERIC TEXT</b> インデックス – <b>GENERIC TEXT</b> インデックスの場合、<b>MAXIMUM TERM LENGTH</b> よりも長いクエリ単語は <b>TEXT</b> インデックスに存在できないため、無視される。</li> </ul>

TEXT インデックスに対して	クエリ単語に対して
<ul style="list-style-type: none"> <li>• <b>NGRAM TEXT</b> インデックス – <b>NGRAM TEXT</b> インデックスの場合、<b>MAXIMUM TERM LENGTH</b> によって、単語が分割される N-gram の長さが決まる。<b>MAXIMUM TERM LENGTH</b> の適切な長さは、言語によって異なる。一般的な値は、英語の場合は 4 文字または 5 文字、中国語の場合は 2 文字または 3 文字である。</li> </ul>	<ul style="list-style-type: none"> <li>• <b>NGRAM TEXT</b> インデックス – <b>NGRAM TEXT</b> インデックスの場合、クエリ単語は長さ n の N-gram に分割される。n は <b>MAXIMUM TERM LENGTH</b> と同じ。データベースサーバは、N-gram を使用して <b>TEXT</b> インデックスを検索する。<b>MAXIMUM TERM LENGTH</b> よりも短い単語は <b>TEXT</b> インデックス内の N-gram と一致しないため、無視される。</li> </ul>

### 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。

ストップリスト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 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
<ul style="list-style-type: none"> <li>• <b>GENERIC TEXT</b> インデックス – <b>GENERIC TEXT</b> インデックスの場合、ストップリストに含まれる単語は <b>TEXT</b> インデックスに挿入されない。</li> </ul>	<ul style="list-style-type: none"> <li>• <b>GENERIC TEXT</b> インデックス – <b>GENERIC TEXT</b> 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 <b>TEXT</b> インデックスに存在できないため、無視される。</li> </ul>
<ul style="list-style-type: none"> <li>• <b>NGRAM TEXT</b> インデックス – <b>NGRAM TEXT</b> インデックスの場合、<b>TEXT</b> インデックスには、ストップリストに含まれる単語から形成された N-gram は格納されない。</li> </ul>	<ul style="list-style-type: none"> <li>• <b>NGRAM TEXT</b> インデックス – ストップリストに含まれる単語は N-gram に分割され、N-gram はストップリストで使用される。同様に、クエリ単語も N-gram に分割され、ストップリストに含まれる N-gram と一致するものは <b>TEXT</b> インデックスに存在できないため、削除される。</li> </ul>

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は、実際に指定したストップリストの単語ではなく、実際には N-gram の形式で格納されるため、予期しない結果が生

じる場合があります。たとえば、**MAXIMUM TERM LENGTH** が 3 の **NGRAM TEXT** インデックスでは、**STOPLIST** 'there' を指定した場合、the her ere という 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 を含むすべての単語に対するクエリに影響します。

## テキスト設定のリストの表示 (Interactive SQL)

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。

1. **CREATE TEXT CONFIGURATION**、**CREATE ANY TEXT CONFIGURATION**、または **CREATE ANY OBJECT** システム権限を持つ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。

すべてのテキスト設定オブジェクトをリストするには、次の構文を使用します。

```
SELECT * FROM SYSTEXTCONFIG;
```

## テキスト設定の変更 (Interactive SQL)

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。

1. **ALTER ANY TEXT CONFIGURATION** または **ALTER ANY OBJECT** システム権限を持つユーザとして、あるい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。

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 (Interactive SQL)

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が格納されているストップリストを変更します。

**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。

## TEXT インデックスとテキスト設定オブジェクト

1. ALTER ANY TEXT CONFIGURATION または ALTER ANY OBJECT システム権限を持つユーザとして、あるい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. STOPLIST 句を指定して ALTER TEXT CONFIGURATION 文を実行します。

ストップリストを myTxtConfig 設定オブジェクトに追加するには、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 (Interactive SQL)

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。

1. DROP ANY TEXT CONFIGURATION または DROP ANY OBJECT システム権限を持つ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。

テキスト設定オブジェクト myTxtConfig 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

以下の例では、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、全文検索クエリ文字列がどのように解釈されるかを示します。

すべての例で、文字列 'I'm not sure I understand' を使用しています。



表 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

以下の例では、テキスト設定オブジェクトの文字列の設定が CONTAINS クエリでどのように解釈されるかを示します。

表「CONTAINS 文字列の解釈」の「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。この数値は、説明のためにマニュアルに記載しています。格納される実際の単語にはカッコ内の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 8 : CONTAINS 文字列の解釈

設定	文字列	文字列の解釈
TERM BREAKER : GENERIC MINIMUM TERM LENGTH : 3	'w*'	'"w*(1) "'
	'we*'	'"we*(1) "'

TEXT インデックスとテキスト設定オブジェクト

設定	文字列	文字列の解釈
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) weath- er(2) "'
	'"wonderlandwonderlandwonderland* wonderland"'	'"wonderland(1) "'
	'"wonderlandwonderlandwonderland* weather"'	'"weather(1) "'
	'"the_wonderlandwonderlandwonderland* weather"'	'"the(1) weath- er(3) "'
	'the_wonderlandwonderlandwonderland* weather'	'"the(1) " & "weath- er(1) "'
	'"light_a* the end" & tunnel'	'"light(1) the(3) end(4) " & "tun- nel(1) "'
	light_b* the end" & tunnel'	'"light(1) the(3) end(4) " & "tun- nel(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) " & "jour- ney(1) "'	

## MAX\_PREFIX\_PER\_CONTAINS\_PHRASE データベースオプション

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)

### デフォルト値

1

### スコープ

このオプションを設定するには、**SET ANY PUBLIC OPTION** システム権限が必要です。個々の接続または **PUBLIC** ロール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

**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過している場合、SAP Sybase IQ はエラーを検出して報告します。

### 例

デフォルトの **MAX\_PREFIX\_PER\_CONTAINS\_PHRASE** 設定を使用する例：

```
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** 設定の 1 を使用した場合、次の **CONTAINS** 句は構文エラーを返します。

```
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、上記の **CONTAINS** 句は有効です。

## TEXT インデックスとテキスト設定オブジェクト

# 外部ライブラリ

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と単語分割を提供する方法について説明します。

## プレフィルタと単語分割の外部ライブラリ

---

SAP 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、SAP 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 サイトにアクセスし、認定レポートをフィルタ処理して、SAP Sybase IQ に関する認定を表示してください。

---

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と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** インデックス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** 呼び出しが実行されてもロードされません。また、**DROP TEXT CONFIGURATION** 呼び出しが実行されても、自動的にアンロードされません。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して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、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。

## 外部ライブラリ

ISYSTEXTCONFIG システムビューには、テキスト設定オブジェクトに関連する、外部ライブラリについての情報が格納されています。

### 参照：

- 起動時の外部ライブラリの有効化と無効化 (22 ページ)

## 外部ライブラリの制限

外部ライブラリを使用するテキスト設定オブジェクトと **TEXT** インデックスには、設計上の制限があります。

- 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ります。SAP 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ません。
- 外部の単語分割を使用してドキュメントをトークン化する場合、N-gram ベースのテキスト検索はサポートされません。

## マルチプレックスサーバでの外部ライブラリ

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

SAP Sybase IQ には、サードパーティ製の外部ライブラリのロードを有効または無効にする **-sf** 起動スイッチが用意されています。

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、次の構文を使用します。

```
-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、次の構文を使用します。

```
-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

---

外部ライブラリが不要になったときにライブラリをアンロードするには、システムプロシージャ **dbo.sa\_external\_library\_unload** を使用します。

**dbo.sa\_external\_library\_unload** は、LONG VARCHAR 型のオプションパラメータを 1 つ取ります。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。

外部ライブラリをアンロードするには、次の構文を使用します。

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





# 非構造化データのクエリ

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。

## 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 (ワード) のすべてのインスタンスをすばやく見つけます。

**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。

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せん。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。

参照：

- **CONTAINS** 条件 (41 ページ)

## 全文検索のタイプ

全文検索を使用して、単語、プレフィクス、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** 句のいずれかで **CONTAINS** 句を使用して、全文検索を実行します。

### 単語とフレーズの検索

単語のリストの全文検索を実行する場合は、フレーズ内の単語でないかぎり、単語の順序は重要ではありません。単語がフレーズ内にある場合、データベースサーバは、出現する順序と相対位置が指定されたものと厳密に一致する単語を検索します。

単語検索やフレーズ検索を実行する際、単語長の設定を超過するため、またはストップリストに含まれているために単語がクエリから削除された場合、予期した

## 非構造化データのクエリ

い数のローが返される可能性があります。これは、単語をクエリから削除することは探索条件を変更することと同じであるためです。たとえば、'"grown cotton"' というフレーズを検索し、grown がストップリストに入っている場合、cotton を含むインデックス付けされたローがすべて返されます。

CONTAINS 句の文法でキーワードと見なされる単語は、フレーズに含まれている場合のみ検索できます。

### 単語検索

サンプルデータベースでは、MarketingInformation テーブルの Description カラムに MarketingTextIndex というテキストインデックスが作成されています。次の文は、MarketingInformation テーブルの Description カラムを問い合わせ、Description カラムの値に cotton という単語が含まれるローを返します。

```
SELECT ID, Description
FROM MarketingInformation
WHERE CONTAINS ( Description, 'cotton' );
```

ID	Description
906	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Visor</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>Lightweight 100% organically grown <b>cotton</b> construction. Shields against sun and precipitation.cotton Metallic ions in the fibers inhibit bacterial growth, and help neutralize odor.</span></p></body></html>
908	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Sweatshirt</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>Lightweight 100% organically grown <b>cotton</b> hooded sweatshirt with taped neck seams. Comes pre-washed for softness and to lessen shrinkage.</span></p></body></html>

ID	Description
909	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Sweatshirt</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>Top-notch construction includes durable topstitched seams for strength with low-bulk, resilient rib-knit collar, cuffs and bottom. An 80% <b>cotton</b> /20% polyester blend makes it easy to keep them clean.</span></p></body></html>
910	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Shorts</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>These quick-drying <b>cotton</b> shorts provide all day comfort on or off the trails. Now with a more comfortable and stretchy fabric and an adjustable drawstring waist.</span></p></body></html>

次の例は、MarketingInformation テーブルを問い合わせ、各ローについて、Description カラムの値に cotton という単語が含まれるかどうかを示す1つの値を返します。

```
SELECT ID, IF CONTAINS ( Description, 'cotton' )
      THEN 1
      ELSE 0
      ENDIF AS Results
FROM MarketingInformation;
```

ID	Results
901	0
902	0
903	0
904	0
905	0
906	1
907	0
908	1
909	1

ID	Results
910	1

次の例は、MarketingInformation テーブルを問い合わせ、Description カラムに cotton という単語が含まれる項目を検索し、一致ごとのスコアを表示します。

```
SELECT ID, ct.score, Description
FROM MarketingInformation CONTAINS
( MarketingInformation.Description, 'cotton' ) as ct
ORDER BY ct.score DESC;
```

ID	score	Description
908	0.9461597363521859	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"> <title>Sweatshirt</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>Lightweight 100% organically grown <b>cotton</b> hooded sweatshirt with taped neck seams. Comes pre-washed for softness and to lessen shrinkage.</span></p></body></html>
910	0.9244136988525732	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"> <title>Shorts</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>These quick-drying <b>cotton</b> shorts provide all day comfort on or off the trails. Now with a more comfortable and stretchy fabric and an adjustable drawstring waist.</span></p></body></html>

ID	score	Description
906	0.9134171046194403	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"> <title>Visor</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>Lightweight 100% organically grown <b>cotton</b> construction. Shields against sun and precipitation. Metallic ions in the fibers inhibit bacterial growth, and help neutralize odor.</span></p></body></html>
909	0.8856420222728282	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"> <title>Sweatshirt</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>Top-notch construction includes durable top-stitched seams for strength with low-bulk, resilient rib-knit collar, cuffs and bottom. An 80% <b>cotton</b> /20% polyester blend makes it easy to keep them clean.</span></p></body></html>

### フレーズ検索

フレーズの全文検索を実行する場合、フレーズを二重引用符で囲みます。指定の順序と相対位置の単語が含まれる場合、カラムは一致します。

単語がフレーズ内に含まれている場合を除き、AND や FUZZY などの CONTAINS キーワードを、検索する単語として指定することはできません (単一単語のフレーズは検索可能)。たとえば、NOT は CONTAINS キーワードですが、次の文は実行できます。

```
SELECT * FROM table-name CONTAINS ( Remarks, '"NOT"' );
```

フレーズの中で使用される特殊文字は、特殊文字として解釈されません (アスタリスクを除く)。

フレーズは、近接検索の引数としては使用できません。

次の文は、MarketingInformation テーブルの Description カラムを問い合わせ、"grown cotton" というフレーズを検索し、一致ごとのスコアを表示します。

## 非構造化データのクエリ

```
SELECT ID, ct.score, Description
FROM MarketingInformation CONTAINS
( MarketingInformation.Description, '"grown cotton"' ) as ct
ORDER BY ct.score DESC;
```

ID	score	Description
908	1.6619019465461564	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"> <title>Sweatshirt</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family: Arial'>Lightweight 100% organically grown <b>cotton</b> hooded sweatshirt with taped neck seams. Comes pre-washed for softness and to lessen shrinkage.</span></p></body></html>
906	1.6043904700786786	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"> <title>Visor</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family: Arial'>Lightweight 100% organically grown <b>cotton</b> construction. Shields against sun and precipitation. Metallic ions in the fibers inhibit bacterial growth, and help neutralize odor.</span></p></body></html>

### プレフィクス検索

全文検索では、単語の先頭部分を検索できる機能があります。これは**プレフィクス検索**と呼ばれます。プレフィクス検索を実行するには、検索するプレフィクスを指定した後に、アスタリスクを付けます。これは**プレフィクス単語**と呼ばれません。

CONTAINS 句のキーワードは、フレーズに含まれていないかぎり、プレフィクス検索では使用できません。

クエリ文字列に複数のプレフィクス単語(フレーズ内のプレフィクス単語も含む)を指定することもできます(たとえば、'"shi\* fab"' など)。

次の例は、MarketingInformation テーブルを問い合わせ、プレフィクス shi で始まる項目を検索します。

```
SELECT ID, ct.score, Description
FROM MarketingInformation CONTAINS
```

```
( MarketingInformation.Description, 'shi*' ) AS ct
ORDER BY ct.score DESC;
```

ID	score	Description
906	2.295363835537917	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"> <title>Visor</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>Lightweight 100% organically grown cotton construction. <b>Shields</b> against sun and precipitation. Metallic ions in the fibers inhibit bacterial growth, and help neutralize odor.</span></p></body></html>
901	1.6883275743936228	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Tee <b>Shirt</b> </title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>We've improved the design of this perennial favorite. A sleek and technical <b>shirt</b> built for the trail, track, or sidewalk. UPF rating of 50+.</span></p></body></html>
903	1.6336529491832605	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Tee <b>Shirt</b> </title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>A sporty, casual <b>shirt</b> made of recycled water bottles. It will serve you equally well on trails or around town. The fabric has a wicking finish to pull perspiration away from your skin.</span></p></body></html>

ID	score	Description
902	1.6181703448678983	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Tee <b>Shirt</b> </title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>This simple, sleek, and lightweight technical <b>shirt</b> is designed for high-intensity workouts in hot and humid weather. The recycled polyester fabric is gentle on the earth and soft against your skin.</span></p></body></html>

テキストインデックスで **shield** という単語が出現する頻度は **shirt** よりも低いいため、ID 906 のスコアが最も高くなります。

#### GENERIC テキストインデックスへのプレフィクス検索

GENERIC テキストインデックスでは、プレフィクス検索は次のように動作します。

- プレフィクス単語が MAXIMUM TERM LENGTH よりも長い場合、MAXIMUM TERM LENGTH よりも長い単語はテキストインデックスには存在できないため、クエリ文字列から削除されます。したがって、MAXIMUM TERM LENGTH が 3 のテキストインデックスで 'red appl\*' を検索することは、'red' を検索することと同義になります。
- プレフィクス単語が MINIMUM TERM LENGTH よりも短く、フレーズ検索の一部でもない場合、プレフィクス検索は通常通り処理されます。したがって、MINIMUM TERM LENGTH が 5 の GENERIC テキストインデックスで、'macintosh a\*' を検索すると、macintosh と a で始まる 5 文字以上の任意の単語を含むインデックス付けされたローが返されます。
- プレフィクス単語が MINIMUM TERM LENGTH よりも短いものの、フレーズ検索の一部になっている場合は、プレフィクス単語はクエリから削除されます。したがって、MINIMUM TERM LENGTH が 5 の GENERIC テキストインデックスで、'"macintosh appl\* turnover"' を検索することは、「macintosh 任意の単語 turnover」を検索することと同義になります。"macintosh turnover" を含むローは検索されません。macintosh と turnover の間に単語が必要になります。

#### NGRAM テキストインデックスへのプレフィクス検索

NGRAM テキストインデックスの場合、NGRAM テキストインデックスには N-gram しか含まれておらず、単語の先頭に関する情報は含まれていないため、プレ



フィクス検索を実行すると予期しない結果が返される可能性があります。また、クエリ単語は N-gram に分割され、検索はクエリ単語ではなく N-gram を使用して実行されます。そのため、次のような動作に注意する必要があります。

- プレフィクス単語が N-gram の長さ (MAXIMUM TERM LENGTH) よりも短い場合、クエリはそのプレフィクス単語で始まる N-gram を含むすべてのインデックス付けされたローを返します。たとえば、3-gram のテキストインデックスの場合、'ea\*' を検索すると、ea で始まる N-gram を含むすべてのインデックス付けされたローが返されます。したがって、weather と fear という単語がインデックス付けされている場合、これらの N-gram にはそれぞれ eat と ear が含まれているため、ローは一致すると見なされます。
- プレフィクス単語が N-gram の長さよりも長く、フレーズの一部ではなく、近接検索の引数ではない場合、プレフィクス単語は N-gram フレーズに変換され、アスタリスクは削除されます。たとえば、3-gram のテキストインデックスの場合、'purple blac\*' を検索することは、'"pur urp rpl ple" AND "bla lac"' を検索することと同義になります。
- フレーズの場合は、次のような動作も発生します。
  - フレーズ内にプレフィクス単語以外の単語がない場合、そのプレフィクス単語は N-gram フレーズに変換され、アスタリスクは削除されます。たとえば、3-gram のテキストインデックスの場合、'"purpl\*"' を検索することは、'"pur urp rpl"' を検索することと同義になります。
  - プレフィクス単語がフレーズの末尾に位置する場合は、アスタリスクは削除され、すべての単語は N-gram のフレーズに変換されます。たとえば、3-gram のテキストインデックスの場合、'"purple blac\*"' を検索することは、'"pur urp rpl ple bla lac"' を検索することと同義になります。
  - プレフィクス単語がフレーズの末尾に位置していない場合、フレーズは複数のフレーズに分割され、AND で結合されます。たとえば、3-gram のテキストインデックスの場合、'"purp\* blac\*"' を検索することは、'"pur urp" AND "bla lac"' を検索することと同義になります。
- プレフィクス単語が近接検索の引数の場合、近接検索は AND に変換されます。たとえば、3-gram のテキストインデックスの場合、'red NEAR[1] appl\*' を検索することは、'red AND "app ppl"' を検索することと同義になります。

## 近接検索

全文検索では、単一のカラム内で互いに近接する単語を検索できる機能があります。これは**近接検索**と呼ばれます。近接検索を実行するには、2つの単語を指定して、間に NEAR または BEFORE キーワードかチルダ (~) を挿入します。

proximity-expression を使用して、互いの距離が近い単語を検索することができます。たとえば、`b NEAR [2,5] c` は、互いの距離が 5 語以下 2 語以上の `b` と `c` の出現を探します。単語の順序に意味はなく、`b NEAR c` は `c NEAR b` と同義です。NEAR を distance なしで指定すると、デフォルトの 10 語が適用されます。NEAR の代わりとしてチルダ (~) を指定することもできます。これは、距離を指定せずに NEAR を指定することと同義で、デフォルトの 10 語が適用されます。`a NEAR [1] b NEAR [1] c` のように、NEAR 式を連ねて使用することはできません。

BEFORE は単語の順序に意味があること以外は NEAR と同様です。`b BEFORE c` は `c BEFORE b` と同義ではありません。前者では単語 `b` は `c` の前でなければなりません。後者では単語 `b` は `c` の後でなければなりません。BEFORE では最小距離と最大距離の両方を指定できます (NEAR と同様)。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、それは最大距離以下でなければなりません。そうでない場合はエラーが返されます。

距離を指定しない場合、データベースサーバはデフォルトの距離として 10 を使用します。

NEAR キーワードの代わりに、チルダ (~) を指定することもできます。たとえば `'term1 ~ term2'` のように指定します。ただし、チルダ形式を使用する場合は距離を指定できず、デフォルトの 10 が適用されます。

近接検索の引数としてフレーズを指定することはできません。

NGRAM テキストインデックスを使用した近接検索でプレフィクス単語を引数として指定すると、近接検索は AND 式に変換されます。たとえば、3-gram のテキストインデックスの場合、`'red NEAR [1] appl*'` を検索することは、`'red AND "app ppl"'` を検索することと同義になります。これは近接検索ではなくなったため、CONTAINS 句で複数のカラムが指定されている場合に、検索が 1 つのカラムに制限されることはありません。

## 例

MarketingInformation テーブルの Description カラムで、単語 skin から 10 語以内の距離にある単語 fabric を検索するとします。この場合、次の文を実行できます。

```
SELECT ID, "contains".score, Description
FROM MarketingInformation CONTAINS ( Description, 'fabric ~
skin' );
```

ID	score	Description
902	1.5572371866083279	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Tee Shirt</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>This simple, sleek, and lightweight technical shirt is designed for high-intensity workouts in hot and humid weather. The recycled polyester <b>fabric</b> is gentle on the earth and soft against your <b>skin</b> .</span></p></body></html>

デフォルトの距離が 10 語なので、距離を指定する必要はありませんでした。しかし、距離を 1 語拡張することで、次のような別のローが返されます。

```
SELECT ID, "contains".score, Description
  FROM MarketingInformation CONTAINS ( Description, 'fabric NEAR[11]
skin' );
```

ID	score	Description
903	1.5787803210404958	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Tee Shirt</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>A sporty, casual shirt made of recycled water bottles. It will serve you equally well on trails or around town. The <b>fabric</b> has a wicking finish to pull perspiration away from your <b>skin</b> .</span></p></body></html>

ID	score	Description
902	2.163125855043747	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Tee Shirt</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>This simple, sleek, and lightweight technical shirt is designed for high-intensity workouts in hot and humid weather. The recycled polyester <b>fabric</b> is gentle on the earth and soft against your <b>skin</b> .</span></p></body></html>

単語間の距離が近い場合、ID 903 のスコアの方が高くなります。

### ブール検索

全文検索を実行する場合に、複数の単語をブール演算子で区切って指定できます。SAP Sybase IQ では、全文検索の実行時にブール演算子 AND、OR、AND NOT がサポートされます。

#### 全文検索での AND 演算子の使用

AND 演算子では、AND の両側に指定された単語を両方とも含むローが一致となります。AND 演算子としてアンパサンド (&) を使用することもできます。複数の単語を、間に演算子を入れないで指定した場合、AND を暗黙で指定したことになります。

単語がリストされる順番は重要ではありません。

たとえば、次の文はすべて、MarketingInformation テーブルの Description カラムで fabric という単語と ski で始まる単語を含むローを検索します。

```
SELECT *
  FROM MarketingInformation
 WHERE CONTAINS ( MarketingInformation.Description, 'ski* AND
fabric' );
SELECT *
  FROM MarketingInformation
 WHERE CONTAINS ( MarketingInformation.Description, 'fabric &
ski*' );
SELECT *
  FROM MarketingInformation
 WHERE CONTAINS ( MarketingInformation.Description, 'ski*
fabric' );
```

### 全文検索での OR 演算子の使用

OR 演算子では、OR の両側に指定された検索語のうちの少なくとも1つを含むローが一致となります。OR 演算子としてパイプ記号 (|) を使用することもできます。この2つは同等です。

単語がリストされる順番は重要ではありません。

たとえば、次の文はいずれも、MarketingInformation テーブルの Description カラムで fabric という単語または ski で始まる単語を含むローを返します。

```
SELECT *
  FROM MarketingInformation
 WHERE CONTAINS ( MarketingInformation.Description, 'ski* OR
fabric' );
SELECT *
  FROM MarketingInformation
 WHERE CONTAINS ( MarketingInformation.Description, 'fabric |
ski*' );
```

### 全文検索での AND NOT 演算子の使用

AND NOT 演算子は、左の引数に一致し、右の引数に一致しない結果を検索します。AND NOT 演算子としてハイフン (-) を使用することもできます。この2つは同等です。

たとえば、次の文は同義であり、いずれも fabric という単語を含み、ski で始まる単語は含まないローを返します。

```
SELECT *
  FROM MarketingInformation
 WHERE CONTAINS ( MarketingInformation.Description, 'fabric AND NOT
ski*' );
SELECT *
  FROM MarketingInformation
 WHERE CONTAINS ( MarketingInformation.Description, 'fabric -
ski*' );
SELECT *
  FROM MarketingInformation
 WHERE CONTAINS ( MarketingInformation.Description, 'fabric & -
ski*' );
```

### 異なるブール演算子の組み合わせ

ブール演算子は、クエリ文字列で組み合わせて使用できます。たとえば、次の文は同義であり、いずれも MarketingInformation テーブルの Description カラムで fabric と skin を含み、cotton は含まない項目を検索します。

```
SELECT *
  FROM MarketingInformation
 WHERE CONTAINS ( MarketingInformation.Description, 'skin fabric -
cotton' );
SELECT *
```

## 非構造化データのクエリ

```
FROM MarketingInformation
WHERE CONTAINS ( MarketingInformation.Description, 'fabric -cotton
AND skin' );
```

次の文は同義であり、いずれも MarketingInformation テーブルの Description カラムで fabric を含むか、または cotton と skin の両方を含む項目を検索します。

```
SELECT *
FROM MarketingInformation
WHERE CONTAINS ( MarketingInformation.Description, 'fabric |
cotton AND skin' );
SELECT *
FROM MarketingInformation
WHERE CONTAINS ( MarketingInformation.Description, 'cotton skin OR
fabric' );
```

### 単語とフレーズのグループ化

単語と式はカッコを使用してグループ化できます。たとえば、次の文は MarketingInformation テーブルの Description カラムで cotton または fabric を含み、ski で始まる単語を含む項目を検索します。

```
SELECT ID, Description FROM MarketingInformation
WHERE CONTAINS( MarketingInformation.Description, '( cotton OR
fabric ) AND shi*' );
```

ID	Description
902	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Tee Shirt</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>This simple, sleek, and light-weight technical <b>shirt</b> is designed for high-intensity workouts in hot and humid weather. The recycled polyester <b>fabric</b> is gentle on the earth and soft against your skin.</span></p></body></html>
903	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Tee Shirt</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>A sporty, casual <b>shirt</b> made of recycled water bottles. It will serve you equally well on trails or around town. The <b>fabric</b> has a wicking finish to pull perspiration away from your skin.</span></p></body></html>

ID	Description
906	<html><head><meta http-equiv=Content-Type content="text/html; charset=windows-1252"><title>Visor</title></head><body lang=EN-US><p><span style='font-size:10.0pt;font-family:Arial'>Lightweight 100% organically grown <b>cotton</b> construction. <b>Shields</b> against sun and precipitation. Metallic ions in the fibers inhibit bacterial growth, and help neutralize odor.</span></p></body></html>

### 複数カラムに渡る検索

すべてのカラムが同じテキストインデックスの一部である場合は、1つのクエリで複数のカラムに渡る全文検索を実行できます。

```
SELECT *
  FROM t
 WHERE CONTAINS ( t.c1, t.c2, 'term1|term2' );
```

```
SELECT *
  FROM t
 WHERE CONTAINS( t.c1, 'term1' )
 OR CONTAINS( t.c2, 'term2' );
```

*t1.c1* に *term1* が含まれている、または *t1.c2* に *term2* が含まれている場合、1つ目のクエリは一致します。

*t1.c1* または *t1.c2* に、*term1* または *term2* が含まれている場合、2つ目のクエリは一致します。この方法で `contains` を使用すると、一致のスコアも返されます。

### あいまい検索

あいまい検索を使用して、スペルミスや単語の変形を検索できます。あいまい検索を実行するには、FUZZY 演算子の後に二重引用符で囲まれた文字列を付けて、文字列の近似一致を検索します。たとえば、CONTAINS

( Products.Description, 'FUZZY "cotton"' ) では、cotton および coton や cotten などのスペルミスが返されます。

---

**注意：** あいまい検索は、NGRAM 単語区切りを使用して作成されたテキストインデックスにしか実行できません。

---

FUZZY 演算子を使用することは、文字列を長さ *n* の部分文字列に手動で分割し、それらを OR 演算子で区切ることに同じです。たとえば、NGRAM 単語区切りと MAXIMUM TERM LENGTH 3 を指定して設定されたテキストインデックスがあるとし、'FUZZY "500 main street"' を指定することは、'500 OR mai OR ain OR str OR tre OR ree OR eet' を指定することと同義になります。

FUZZY 演算子は、スコアを返す全文検索で便利です。これは、多くの近似一致が返されても、通常は最高スコアを持つ一致だけが意味のある一致だからです。

### ビュー検索

ビューまたは派生テーブルで全文検索を使用するには、全文検索を実行したいベーステーブルのカラムにテキストインデックスを構築する必要があります。次の文では、サンプルデータベースの MarketingInformation テーブルにビューを作成します。ビューにはすでにテキストインデックス名があるため、そのビューで全文検索を実行できます。

MarketingInformation ベーステーブルにビューを作成するには、次の文を実行します。

```
CREATE VIEW MarketingInfoView AS
SELECT MI.ProductID AS ProdID,
       MI."Description" AS "Desc"
FROM GROUPO.MarketingInformation AS MI
WHERE MI."ID" > 3
```

次の文を使用すると、基本となるテーブルのテキストインデックスを使用してビューを問い合わせることができます。

```
SELECT *
FROM MarketingInfoView
WHERE CONTAINS ( "Desc", 'Cap OR Tee*' )
```

また、次の文を実行すると、基本テーブルのテキストインデックスを使用して派生テーブルにクエリを実行できます。

```
SELECT *
FROM (
    SELECT MI.ProductID, MI."Description"
    FROM MarketingInformation AS MI
    WHERE MI."ID" > 4 ) AS dt ( P_ID, "Desc" )
WHERE CONTAINS ( "Desc", 'Base*' )
```

---

**注意：**全文検索を実行するカラムが、ビューまたは派生テーブルの SELECT リストに含まれている必要があります。

---

基本となるテーブルのテキストインデックスを使用してビューを検索する場合、次のような制限事項があります。

- ビューには、TOP 句、FIRST 句、DISTINCT 句、GROUP BY 句、ORDER BY 句、UNION 句、INTERSECT 句、EXCEPT 句、または Window 関数を含めることはできない。
- ビューに集合関数を含めることはできない。
- CONTAINS クエリはビュー内のベーステーブルは参照できるが、別のビュー内にあるビュー内のベーステーブルは参照できない。



FROM 句

**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* を使用して参照できるスコアカラムとともに返されます (*score-correlation-name* を指定している場合)。

*score-correlation-name* を指定しない場合は、デフォルトの相関名 *contains* によってスコア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** インデックスが設定されている必要があります。

## 参照：

- **CONTAINS** 条件 (41 ページ)

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
```

## 非構造化データのクエリ

```
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** – 検索に使用される単一のインデックス単語を表す、スペースと特殊文字で区切られた文字列。
- **distance** – 正の整数。
- **and-expression** – *primary-expression* と *simple-expression* の両方が **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。デフォルトでは、単語または式の間には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。アンパサンド (&) は AND の代わりとして使用でき、式や単語のどちら側にも配置できます ('a & b' など)。

- **and-not-expression** – *primary-expression* は **TEXT** インデックスに存在する必要があるが、*basic-expression* は **TEXT** インデックスに存在してはならないことを指定します。これは否定とも呼ばれます。否定を示すの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、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' と同義ですが、'a - b' ではハイフンが無視され、この文字列は 'a AND b' と同義になります。'a-b' はフレーズ '"a b"' と同義です。
- **or-expression** – 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** – 指定した単語と似ている単語を検索します。あいまい一致は、**NGRAM TEXT** インデックスでのみサポートされています。
- **proximity-expression** – 互いの距離が近い単語を検索します。たとえば、'b NEAR[2,5] c' は、互いに単語 2～5 個分離れている 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なく、'b NEAR c' は 'c NEAR b' と同義です。*distance* 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** の代わりとしてチルダ (~) を指定することもできます。これは、距離を指定せずに **NEAR** を指定するのと同じなので、デフォルトの単語 10 個分が適用されます。'a NEAR[1] b NEAR[1] c' のように **NEAR** 式をつなげることはできません。

**BEFORE** は **NEAR** に似ていますが、単語の順序が重要です。'b BEFORE c' は 'c BEFORE b' と同義ではありません。前者の場合、単語 'b' が 'c' より前にある必要があり、後者の場合、単語 'b' が 'c' より後にある必要があります。**BEFORE** では、**NEAR** と同様に最小距離と最大距離の両方を指定できます。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** – 指定されたプレフィ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、プレフィクス検索とも呼ばれます。プレフィクス検索では、単語のアスタリスクより左の部分について照合が実行されます。

### 使用法

**CONTAINS** 検索条件は、引数としてカラムリストと *contains-query-string* を取りま

ず。  
**CONTAINS** 検索条件は、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ます。この検索条件は、TRUE または FALSE を返します。 *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、それらすべてが単一のバ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、複数のバーステーブルにまたがることはできません。バーステーブルは、**FROM** 句で直接参照するか、ビューまたは抽出テーブルで参照できます。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**、またはロー制限が使用されていない場合にかぎります。

ANSI ジョイン構文 (**FULL OUTER JOIN**、**RIGHT OUTER JOIN**、**LEFT OUTER JOIN**) を使用するクエリはサポートされますが、最良のパフォーマンスが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ます。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、SAP Sybase IQ テーブルおよび 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。次に例を示します。

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

クエリ文字列に英数字以外の文字を使用する場合は、次の点に注意が必要です。

- アスタリスクが単語内にあると、エラーが返されます。
- **fuzzy-expression** で英数字以外の文字 (特殊文字を含む) を使用しないでください。このような文字はスペースとして扱われ、単語の分割記号として機能します。
- 可能であれば、特殊文字ではない英数字以外の文字をクエリ文字列で使用しないでください。特殊文字ではない英数字以外の文字があると、それを含む単語がフレーズとして扱われ、その文字の位置で単語が分割されます。たとえば、**'things we've done'** は、**'things "we ve" done'** と解釈されます。

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す。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します。

*contains-query-string* の解釈は、主に次の 2 つのステップで行われます。

- ステップ1：演算子と優先度の解釈。このステップでは、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ます。
- ステップ2：テキスト設定オブジェクトの設定の適用。このステップでは、テキスト設定オブジェクトの設定が単語に適用されます。単語の長さの設定を超えているクエリ単語、およびストップリストに含まれているクエリ単語は削除されます。

**参照：**

- TEXT インデックス上のあいまい検索 (50 ページ)

**CONTAINS 検索条件での演算子の優先度**

クエリの評価中、式は、優先順位を使用して評価されます。

クエリ式を評価する場合の優先順位は、次のとおりです。

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、クエリでのプレフィ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、アンパサンド、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「アスタリスクの解釈」は、アスタリスクの許可される使用法を示しています。

**表 9：アスタリスクの解釈**

クエリ文字列	同義の文字列	解釈
'th*&best'	'th* AND best' および 'th* best'	th で始まる単語と、単語 best を検索する。
'th* best'	'th* OR best'	th で始まる単語、または単語 best を検索する。
'very&(best th*)'	'very AND (best OR th*)'	単語 very と、単語 best または th で始まる単語を検索する。
'"fast auto*"'		auto で始まる単語が直後に続く、単語 fast を検索する。

クエリ文字列	同義の文字列	解釈
'"auto* price"'		単語 price が直後に続く、auto で始まる単語を検索する。

**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、単語の一部または式の反転としてクエリ内で使用でき、NOT と同じ機能を提供します。

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。

あいまい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「ハイフンの解釈」は、ハイフンの許可される構文を示しています。

表 10 : 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。
'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検索する。
'wild - west'	'wild west' と 'wild AND west'	単語 wild と単語 west の両方が含まれるものを検索する。

## 特殊文字の許可される構文

表「特殊文字の解釈」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。

アスタリスクとハイフンの文字は、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せん。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せん。

表 11：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 <b>AND</b> と同義で、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 <b>OR</b> と同義で、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 ("")	二重引用符は、順序と相対的な距離が重要な一連の単語を囲むために使用する。たとえば、クエリ文字列 'learn "full text search"' では、"full text search" はフレーズである。この例では、learn はこのフレーズの前後どちらにあっても、別のカラムにあってもかまわないが ( <b>TEXT</b> インデックスが複数カラムに対して構築されている場合)、完全に一致するフレーズが1つのカラムに存在している必要がある。
丸カッコ ()	丸カッコは、式の評価の順序がデフォルトの順序と異なる場合にその順序を指定するために使用する。たとえば、'a AND (b c)' は「a と、b または c」と解釈される。
チルダ (~)	チルダは <b>NEAR</b> と同義で、特別な構文規則はない。クエリ文字列 'full~text' は 'full NEAR text' と同義で、「単語 text から 10 語以内の距離にある単語 full」と解釈される。

文字または構文	使用例および備考
角カッコ[]	角カッコは、 <i>distance</i> を指定するためにキーワード <b>NEAR</b> と組み合わせて使用する。角カッコをそれ以外の方法で使用すると、エラーが返される。

### 削除した単語の影響

**TEXT** インデックスでは、特定の条件を満たす単語が除外されることがあります。

**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も適用されます。削除される単語は、フレーズの末尾または先頭の 0 個以上の単語と一致させることができます。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。

- 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'、'the NEAR apple' を検索することは、'apple' を検索することと同じです。
- 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。
- 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)' と同じです。**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' と比較してみてください。この式は、'orange and not apple' を検索することと同じです。
- 単語がフレーズ内にある場合、そのフレーズは、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' の検索は、'feed the dog'、'feed my dog'、'feed any dog' などに一致します。

---

**注意：** 検索対象のすべての単語が削除される場合、SAP 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。

---



### クエリ一致スコア

一致度を示すスコアを使用して、クエリ結果をソートできます。

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。このスコアは一致がどれだけ近いかを示しており、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、スコアは高くなります。
- **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、スコアは低くなります。

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、検索語にどれだけ近接しているか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。`"contains".score` は、**SELECT** リスト、**ORDER BY** 句、またはクエリの他の部分で参照できます。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、**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"` 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;
```

## NGRAM TEXT インデックス検索

**TEXT** インデックス上のあいまい検索と非あいまい検索は、タイプ **NGRAM** の **TEXT** インデックス上で可能です。

## TEXT インデックス上のあいまい検索

**TEXT** インデックス上のあいまい検索は、**TEXT** インデックスがタイプ **NGRAM** の場合のみ可能です。**GENERICTEXT** インデックスでは、あいまい検索は処理できません。

あいまい検索は、単語の入力ミスまたは変形を含む検索に使用できます。これを行うには、**FUZZY** 演算子を使用し、その後、近接一致を見つけるための文字列を二重引用符で囲んで指定します。

**FUZZY** 演算子を使用することは、文字列を長さ  $n$  の部分文字列に手動で分割し、それらを **OR** 演算子で区切ることと同じです。たとえば、**NGRAM** 単語分割と **MAXIMUM TERM LENGTH 3** を指定して設定されたテキストインデックスがある場合、`'FUZZY "500 main street" '` を指定することは、`'500 OR mai OR ain OR str OR tre OR ree OR eet'` を指定することと同じです。

**FUZZY** 演算子は、スコアを返す全文検索で役立ちます。多くの近接一致が返される可能性があります。通常、最高のスコアを持つ一致文字列のみが意味を持ちます。

---

**注意：** あいまい検索では、プレフィクス検索とサフィックス検索はサポートされません。たとえば、検索句に `"v*"` または `"*vis"` は指定できません。

---

### 例 1： NGRAM TEXT インデックス上のあいまい検索

テーブルと **NGRAMTEXT** インデックスを作成します。

```
CREATE TEXT CONFIGURATION NGRAMTtxtcfg
  FROM default_char;
ALTER TEXT CONFIGURATION NGRAMTtxtcfg TERM BREAKER      NGRAM;
ALTER TEXT CONFIGURATION NGRAMTtxtcfg maximum term      length 3;
CREATE TABLE t_iq(a int, b varchar(100));
CREATE TEXT INDEX TXT_IQ on t_iq(b) CONFIGURATION        NGRAMTtxtcfg
```

次のデータをテーブルに挿入します。

```
INSERT INTO t_iq values (1,'hello this is hira ');
INSERT INTO t_iq values(2, ' book he ookw worm okw
kwor');
INSERT INTO t_iq values(3,'Michael is a good person');
INSERT INTO t_iq values(4,'hello this is evaa');
INSERT INTO t_iq values(5,'he is a bookworm');
INSERT INTO t_iq values (6,'boo ook okw kwo wor orm');
```

データを挿入したら、次のクエリを実行して、**NGRAMTEXT** インデックス上であいまい検索を行います。

```
SELECT * FROM t_iq WHERE CONTAINS (b,'FUZZY "bookerm"');
```

このクエリの結果を次に示します。

```
a      b
2      book he ookw worm okwo kwor
5      he is a bookworm
6      boo ook okw kwo wor orm
```

### 例 2：あいまい検索句での余分な文字

次のクエリは、あいまい検索句に余分な文字を指定する例を示します。

```
SELECT * FROM t_iq WHERE CONTAINS (b, 'FUZZY "hellow"');
```

このクエリの結果を次に示します。

```
a      b
1      hello this is hira
4      hello this is evaa
```

### 例 3：あいまい検索句から文字の削除

次のクエリでは、あいまい検索句に 1 文字を削除して指定します。

```
SELECT * FROM t_iq WHERE CONTAINS (b, 'FUZZY "hlllo"');
```

このクエリの結果を次に示します。

```
a      b
1      hello this is hira
4      hello this is evaa
```

## TEXT インデックス上の非あいまい検索

**NGRAM** 上の非あいまい検索は、単語を対応する n-gram に分割し、n-gram を **NGRAMTEXT** インデックス内で検索します。

クエリ `CONTAINS ( M.Description, 'ams' ) ct;` は **2GRAM** インデックス上の非あいまい **NGRAM** 検索を示しており、セマンティック上、検索クエリ `CONTAINS ( M.Description, '"am ms"' ) ct;` と同じです。

**2GRAM** インデックス上で 'v\*' 単語を検索すると、v で始まり、その後任意のアルファベット文字が続く文字列が、検索単語に一致する **2GRAM** と見なされ、結果として出力されます。

クエリ `CONTAINS ( M.Description, 'white whale' ) ct;` は **3GRAM** インデックス上の非あいまい **NGRAM** 検索を示しており、セマンティック上、検索クエリ `CONTAINS ( M.Description, '"whi hit ite wha hal ale"' );` と同じです。

**NGRAM** のあいまい検索と非あいまい検索の違いは、あいまい検索は、個々の **GRAM** の分離であるということです。非あいまい検索は、個々の **GRAM** の結合です。**GENERIC** と **NGRAM** の **TEXT** のインデックスを同じカラムに作成すると、

**GENERICTEXT** インデックスが非あいまい検索を伴うクエリに使用され、**NGRAMTEXT** インデックスがあいまい検索に使用されます。

**例 1：同じカラムに **GENERICTEXT** インデックスを作成した後の非あいまい検索**  
次のクエリは、同じカラムに **GENERICTEXT** インデックスを作成した後の非あいまい検索を示しています。

```
SELECT * FROM t_iq WHERE CONTAINS (b, 'bookworm');
```

このクエリの結果を次に示します。

a	b
5	he is a bookworm

**例 2：同じカラムに **NGRAM** と **GENERIC** の両方の **TEXT** インデックスがある場合のあいまい検索**

次のクエリは、同じカラムに **NGRAM** と **GENERIC** の両方の **TEXT** インデックスがある場合のあいまい検索を示しています。

```
SELECT * FROM t_iq  
WHERE CONTAINS (b, 'FUZZY "bookworm"');
```

このクエリの結果を次に示します。

a	b
2	book he ookw worm okwo kwor
5	he is a bookworm
6	boo ook okw kwo wor orm

**例 3：非あいまい検索句でのあいまい検索語句**

次のクエリは、非あいまい検索句でのあいまい検索語句の動作を示しています。

```
SELECT * FROM t_iq WHERE CONTAINS (b, 'bookwerm');
```

このクエリでは、結果は何も返されません。

## LONG BINARY カラムに対するクエリ

**SELECT** 文の **WHERE** 句では、**BYTE\_LENGTH64**、**BYTE\_SUBSTR64**、**BYTE\_SUBSTR**、**BIT\_LENGTH**、**OCTET\_LENGTH**、**CHARINDEX**、**LOCATE** の各関数に加えて、**IS NULL** 式と **IS NOT NULL** 式のみで **LONG BINARY** カラムを使用できます。

**SELECT** 文の **ORDER BY** 句、**GROUP BY** 句、**HAVING** 句で **LONG BINARY** カラムを使用することはできません。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。

SAP Sybase IQ は、**LONG BINARY** (BLOB) カラムまたは変数では **LIKE** 述部をサポートしていません。**LIKE** 述部を使用して **LONG BINARY** カラムのパターン検索を行うと、エラー「Invalid data type comparison in predicate」が返されます。

参照：

- 関数のサポート (91 ページ)

## LONG VARCHAR カラムに対するクエリ

**SELECT** 文の **WHERE** 句では、**LONG VARCHAR** カラムは、**BIT\_LENGTH**、**CHAR\_LENGTH**、**CHAR\_LENGTH64**、**CHARINDEX**、**LOCATE**、**OCTET\_LENGTH**、**PATINDEX**、**SUBSTRING64**、**SUBSTRING** の各関数、**IS NULL** 式と **IS NOT NULL** 式のみで使用できます。

**LIKE** 述部を使用して、**LONG VARCHAR** カラムに対してパターン検索を実行できます。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は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せん。現在、**SQL** 変数で保持できる最大の長さは 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。**DISTINCT** キーワード (**SELECT DISTINCT** と **COUNT DISTINCT**) と一緒に使用することもできません。

参照：

- 関数のサポート (91 ページ)

## 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ていません。**CONTAINS** 述部を使用して、**WD** インデックスを使用している **LONG BINARY** カラムで文字列を検索しようとする、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 **CONTAINS** をサポートしています。

## **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**

---

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。

# ストアドプロシージャのサポート

LONG BINARY (BLOB) データ型と LONG VARCHAR (CLOB) データ型のカラムと全文検索に対するストアドプロシージャのサポートについて説明します。

## TEXT インデックス内の単語の管理

ストアドプロシージャを使用して、文字列の単語への分割、**TEXT** インデックス内の単語の個数とそれぞれの位置の情報の取得、**TEXT** インデックスについての統計情報の表示を実行できます。

### sa\_char\_terms システムプロシージャ

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

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

#### パラメータ

- **char-string** – 解析する CHAR 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\_char' です。
- **owner** – 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。

#### 権限

なし

#### 備考

**sa\_char\_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。

#### 例

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 12 : 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

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' ] ] )
```

#### パラメータ

- **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。
- **owner** – 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。

#### 権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。

#### 備考

**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。インデックス処理で、またはクエリ文字列から、どの単語が削除されるかを確認する場合に便利です。

**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せん。



**注意：** NCHAR データ型は、**IN SYSTEM** テーブルについてのみサポートされています。

## sa\_text\_index\_stats システムプロシージャ

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(
```

### 権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- MANAGE ANY STATISTICS
- CREATE ANY INDEX
- ALTER ANY INDEX
- DROP ANY INDEX
- CREATE ANY OBJECT
- ALTER ANY OBJECT
- DROP ANY OBJECT

### 備考

**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。

**表 13 : sa\_text\_index\_stats によって返される TEXT インデックスの統計情報**

カラム名	データ型	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

カラム名	データ型	説明
doc_count	UNSIGNED BIGINT	TEXT インデックスに含まれるインデックス付けされたカラム値の総数
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	保留中の変更の合計長
deleted_length	UNSIGNED BIGINT	保留中の削除の合計長
last_refresh	TIMESTAMP	最終更新日時

IMMEDIATE REFRESH TEXT インデックスの場合、pending\_length、deleted\_length、および last\_refresh の値は NULL です。

*例*

データベース内の各 TEXT インデックスに関する統計情報を返します。

```
CALL sa_text_index_stats( );
```

### sa\_text\_index\_vocab システムプロシージャ

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のリストを返します。

*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。

*権限*

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。

- SELECT ANY TABLE システム権限

- インデックステーブルに対する SELECT 権限

#### 備考

**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、合計出現回数未満となります)。

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。引数の *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。

#### 例

**sa\_text\_index\_vocab** を実行して、GROUPO によって所有されるテーブル Customers の TEXT インデックス MyTextIndex に含まれるすべての単語を返します。

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

表 14 : 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

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。ライブラリを指定しない場合は、使用中でない外部ライブラリがすべてアンロードされます。

### 権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。また、次のものも必要です。MANAGE ANY EXTERNAL OBJECT システム権限

### 備考

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。

### 例

myextlib.dll という名前の外部ライブラリをアンロードします。

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

現在使用されていないすべてのライブラリをアンロードします。

```
CALL sa_external_library_unload ();
```

## sa\_list\_external\_library システムプロシージャ

現在サーバにロードされている外部ライブラリをリストします。

### 構文

```
sa_list_external_library ( )
```

### 権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。また、次のものも必要です。MANAGE ANY EXTERNAL OBJECT システム権限

### 備考

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。

例

外部ライブラリとその参照カウントを一覧表示します。

```
CALL sa_list_external_library()
```

## ラジオオブジェクトデータの圧縮

---

ラジオオブジェクトカラムの圧縮の制御は、**sp\_iqsetcompression** ストアドプロシージャで行います。

**sp\_iqsetcompression** は、データベースバッファをディスクに書き込むときに実行される、LONG BINARY データ型と 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。

ラジオ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

LONG BINARY (BLOB) データ型と LONG VARCHAR (CLOB) データ型のカラムのデータの圧縮を設定します。

構文

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

パーミッション

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- ALTER ANY TABLE
- ALTER ANY OBJECT

備考

**sp\_iqsetcompression** 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します。圧縮設定は、ベーステーブルにのみ適用されます。

**sp\_iqsetcompression** 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 15 : **sp\_iqsetcompression** 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
<i>on_off_flag</i>	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

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。

```
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

LONG BINARY (BLOB) データ型と LONG VARCHAR (CLOB) データ型のカラムの圧縮設定を表示します。

### 構文

```
sp_iqshowcompression ( owner, table, column )
```

### 権限

そのシステムプロシージャに対する EXECUTE 権限が必要です。さらに、次のいずれかが必要です。システム権限：

- ALTER ANY TABLE
- ALTER ANY OBJECT

### 備考

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 16 : 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

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。

```
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

ストアードプロシージャ **sp\_iqindexsize** では、LONG BINARY と LONG VARCHAR の個々の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 **sp\_iqindexsize** 出力は、約 42GB のデータが格納されている LONG BINARY カラムを示しています。

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

## ストアドプロシージャのサポート

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 `sp_iqindexsize` 出力は、約 42GB のデータが格納されている LONG VARCHAR カラムを示しています。

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

ラージオブジェクトデータのエクスポートとロードを行う方法について説明します。

## ラージオブジェクトデータのエクスポート

---

SAP Sybase 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** 関数が用意されています。

**BFILE** は、データ抽出機能と一緒に使用できるほか、単独でも使用できます。

### **BFILE** 関数

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、ファイルシステムでサポートされている有効なパス名を指定する必要があります。
- **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)取得される **LONG BINARY** セルと **LONG VARCHAR** セルの最大長は、**2GB** 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 VARCHAR** 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では、**2GB** を超えるセルを1つの **SELECT** で返すことはできません。

**BFILE** は、データ抽出機能と一緒に使用できるほか、単独でも使用できます。

### **BFILE 関数の例**

**BFILE** を使用して、LOB データの抽出と再ロードを行います。

LobA テーブルを作成します。

```
create table LobA
  (rowid  int primary key,
   col1   clob null,
   col2   blob null)
```

LobA に2つのデータローがあるとしします。

非LOB データとファイルへのパスを、LOB データの抽出先に抽出します。

```
BEGIN
  SET TEMPORARY OPTION
    Temp_Extract_Name1 = LobA_data.txt';
  SELECT rowid,
    'row' + string(rowid) + '.' + 'col1',
    'row' + string(rowid) + '.' + 'col2'
  FROM LobA;
END
```

ファイル LobA\_data.txt が作成され、この非LOB データとこれらのファイル名が格納されます。

```
1,row1.col1,row1.col2,
2,row2.col1,row2.col2,
```

LOB データの抽出を実行します。

```
SELECT
  BFILE('row' + string(rowid) + '.' + 'col1',col1),
  BFILE('row' + string(rowid) + '.' + 'col2',col2)
FROM LobA;
```

抽出が終了すると、抽出された LOB データのセルごとにファイルが作成されます。たとえば、テーブル LobA に、rowid 値として 1 と 2 を持つ 2 つのデータローが含まれている場合、次のファイルが作成されます。

- row1.col1
- row1.col2
- row2.col1
- row2.col2

抽出されたデータを再ロードします。

```
LOAD TABLE LobA
(rowid,
 col1 ASCII FILE ('') NULL('NULL'),
 col2 BINARY FILE ('') NULL('NULL'))
FROM LobA_data.txt'
DELIMITED BY ','
ROW DELIMITED BY '¥n'
ESCAPES OFF;
```

## 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 TABLE** には、ラージオブジェクトデータをロードするための拡張構文が用意されています。

```
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:
...
```

## ラージオブジェクトデータのロードとアンロード

```
| { BINARY | ASCII } FILE ( integer )  
| { BINARY | ASCII } FILE ( 'string' )
```

キーワード **BINARY FILE** (LONG BINARY 用) または **ASCII FILE** (LONG VARCHAR 用) は、カラムのプライマリ入力ファイルには、LONG BINARY データ自体または LONG VARCHAR データ自体ではなく、LONG BINARY セルまたは LONG VARCHAR セルの値を含むセカンダリファイルのパスが含まれていることをロード処理に指示します。セカンダリファイルのパス名は、完全修飾パスでも相対パスでもかまいません。セカンダリファイルのパス名が完全修飾パスでない場合、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。

SAP 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、SAP 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、「カラムで奇数の長さのバイナリデータ値が検出されました」というエラーが報告されます。LONG BINARY をロードする入力ファイルには、必ず偶数個の 16 進数字が含まれている必要があります。

SAP 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。

**LOAD TABLE FORMAT BCP** では、ロードの指定にカラム名、**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、セカンダリファイルは使用できません。

## ラージオブジェクトデータのロード例

LONG BINARY データを使用するテーブルを作成し、ロードします。

```
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 データがテーブル tab にロードされると、1 番目と 2 番目のローの lobcol カラムには、それぞれ boston.jpg ファイルと concord.bmp ファイルの内容が入ります。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実行する、ロードのアクションを指定できます。

SECONDARY\_FILE\_ERROR を ON 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 OFF (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 セルは次のいずれかの値となります。

- NULL (カラムが NULL を許可している場合)
- 長さ 0 の値 (カラムが NULL を許可していない場合)

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することができます。また、このオプションの設定はすぐに有効になります。

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 カラムについて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)

## 後続ブランクを含むラージオブジェクトデータのロード

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 ON でも、LONG VARCHAR データの後続ブランクは削除されません。

## 引用符を含むラージオブジェクトデータのロード

**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)** データまたは **LONG VARCHAR (CLOB)** データのロードに適用されません。

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、**QUOTESON** 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。

## 部分的なマルチバイト文字データのトランケート

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。

- **TRIM\_PARTIAL\_MBC** を **ON** 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。
- **TRIM\_PARTIAL\_MBC** を **OFF** 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。

表「**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」では、**TRIM\_PARTIAL\_MBC** と **SECONDARY\_FILE\_ERROR** の値に応じて後続マルチバイト文字がどのようにロードされるかを説明します。

表 17 : **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

<b>TRIM_PARTIAL_MBC</b>	<b>SECONDARY_FILE_ERROR</b>	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)

## ラージオブジェクト変数のロードサポート

ラージオブジェクト変数は、**LOAD TABLE**、**INSERT...VALUES**、**INSERT...SELECT**、**INSERT...LOCATION**、**SELECT...INTO**、**UPDATE** の各 SQL 文でサポートされています。

### 参照：

- ラージオブジェクトデータ型 (73 ページ)
- ラージオブジェクト変数 (76 ページ)

## ラージオブジェクトデータのロードとアンロード



# ラージオブジェクトデータ型

ラージオブジェクト LONG BINARY データ型カラムとラージオブジェクト LONG VARCHAR データ型カラムの特性、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。

## ラージオブジェクトデータ型、LONG BINARY と BLOB

---

SAP 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 データ値の長さは、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 0 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 インデックスを LONG BINARY カラムに作成することはできません。

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 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。

## LONG BINARY データ型の変換

LONG BINARY データ型と非 LONG BINARY データ型の間で、限定された暗黙的なデータ型変換が行われます。

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が行われます。BIT、REAL、DOUBLE、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** 文と **UPDATE** 文の暗黙的な変換の入力として受け入れられます。

LONG BINARY データ型は、明示的に BINARY または VARBINARY に変換できません。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 (**CAST** 関数や **CONVERT** 関数を使用した変換など) 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 データと VARBINARY データのトランケーションが処理されるのと同じ方法で処理されます。STRING\_RTRUNCATION オプションが ON の場合、バイナリカラムの **INSERT** または **UPDATE** で (スペース以外の文字だけではなくすべての値の) 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発生し、トランザクションがロールバックされます。

### 参照：

- 関数のサポート (91 ページ)

## ラージオブジェクトデータ型、LONG VARCHAR と CLOB

SAP 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 データ値の長さは、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。

SAP Sybase IQ は、シングルバイトとマルチバイトの両方の 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 0 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、カラム定義で NOT NULL を明示的に指定します。

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

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

```
ALTER TABLE lvtab ADD c4 CLOB;
```

LONG VARCHAR カラムに **WORD (WD)** インデックスを作成できます。その他の非 **FP** インデックスとジョインインデックスを LONG VARCHAR カラムに作成することはできません。

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 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。挿入されたすべての TEXT 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。

## LONG VARCHAR データ型の変換

LONG VARCHAR データ型と非 LONG VARCHAR データ型の間で、限定された暗黙的なデータ型変換が行われます。

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 データ型と VARCHAR データ型から LONG VARCHAR データ型への暗黙的なデータ型変換が行われます。BIT、REAL、DOUBLE、NUMERIC、TINYINT、SMALLINT、INT、UNSIGNED INT、BIGINT、UNSIGNED BIGINT、BINARY、VARBINARY、LONG BINARY データ型から LONG VARCHAR データ型へ

## ラージオブジェクトデータ型

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 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 文と UPDATE 文の暗黙的な変換の入力として受け入れられます。

LONG VARCHAR データ型は、明示的に CHAR と VARCHAR 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その他の明示的なデータ型変換 (CAST 関数や CONVERT 関数を使用した変換など) 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が処理されるのと同じ方法で処理されます。STRING\_RTRUNCATION オプションが ON の場合、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。後続マルチバイト文字の一部は、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が処理されるのと同じ方法で処理されます。STRING\_RTRUNCATION オプションが ON の場合、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。後続マルチバイト文字の一部は、変換時にトランケートされます。

### 参照：

- 関数のサポート (91 ページ)

## ラージオブジェクト変数

---

SAP Sybase IQ はラージオブジェクト変数をサポートしています。

インバウンドの LONG BINARY 変数と 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 変数と 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

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

*デフォルト値*

OFF

*スコープ*

オプションは、データベース (PUBLIC) レベルまたは ユーザレベルで設定できます。データベースレベルで設定した場合、値は新しいユーザのデフォルト値になりますが、既存のユーザには影響を与えません。ユーザレベルで設定した場合は、そのユーザの PUBLIC 値のみが上書きされます。自分のオプションを設定する場合は、システム権限は必要ありません。自分以外のユーザのオプションをデータベースレベルまたはユーザレベルで設定する場合は、システム権限が必要です。

このオプションを設定するには、**SET ANY PUBLIC OPTION** システム権限が必要です。個々の接続または PUBLIC ロールに一時的に設定できます。すぐに有効になります。

*備考*

**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** 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レポートされます。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、どのサイズのラージオブジェクト変数も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

SAP Sybase IQ は、LONG BINARY カラムと LONG VARCHAR カラムの **TEXT** インデックス、LONG VARCHAR カラムの **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

**TEXT** インデックスは、LONG BINARY カラムと LONG VARCHAR カラムをサポートします。

### 参照：

- SQL 文のサポート (79 ページ)
- TEXT インデックスとテキスト設定オブジェクト (3 ページ)

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

SAP Sybase IQ は、LONG VARCHAR (CLOB) カラムの **WORD (WD)** インデックスを限定的にサポートします。

- **WD** インデックスでサポートされる最大幅のカラムは、LOB カラムの最大幅です (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)。SAP Sybase IQ でサポートされる最大ワード長は 255 バイトです。
- CHAR カラムと VARCHAR カラムの **WD** インデックスの **sp\_iqcheckdb** オプションはすべて、allocation、check、verify の各モードを含め、LONG VARCHAR (CLOB) カラムでもサポートされます。
- **sp\_iqrebuildindex** を使用して、LONG VARCHAR (CLOB) カラムの **WD** インデックスを再構築できます。

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。

# SQL 文のサポート

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。

## ALTER TEXT CONFIGURATION 文

テキスト設定オブジェクトを変更します。

**注意：** この文は、非構造化データ分析 (IQ\_UDA) ライセンスを必要とします。

クイックリンク：

「パラメータ」 (79 ページ)

「例」 (81 ページ)

「使用法」 (81 ページ)

「パーミッション」 (82 ページ)

### 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
  | DROP PREFILTER
```

### パラメータ

(先頭に戻る) (79 ページ)

- **stoplist** – 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、置き換えたりする場合に使用する文字列式。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。ストップリストの単語はスペースで区切ります。

ストップリストの単語にスペースを含めることはできません。また、英数字以外の文字を使用しないでください。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。ストップリストの単語の最大数は 7999 です。

- **DROP STOPLIST** – テキスト設定オブジェクトのストップリストを削除する場合に使用します。
- **MINIMUM TERM LENGTH** – 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、MINIMUM TERM LENGTH 句に指定された値は無視されます。

この設定より短い単語は、TEXT インデックスの構築時または更新時に無視されます。このオプションの値は 0 より大きくする必要があります。このオプションを MAXIMUM TERM LENGTH より大きい値に設定すると、MAXIMUM TERM LENGTH の値が自動的に調整され、MINIMUM TERM LENGTH の新しい値と同じになります。

- **MAXIMUM TERM LENGTH** – GENERIC TEXT インデックスを使用する場合、TEXT インデックスに含める単語の最大長 (文字数) を指定します。この設定より長い単語は、TEXT インデックスの構築時または更新時に無視されます。

MAXIMUM TERM LENGTH の値は 60 以下にする必要があります。このオプションを MINIMUM TERM LENGTH より小さい値に設定すると、MINIMUM TERM LENGTH の値が自動的に調整され、MAXIMUM TERM LENGTH の新しい値と同じになります。

- **TERM BREAKER** – 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を指定します。IN SYSTEM テーブルでの選択肢は、GENERIC (デフォルト) または NGRAM です。GENERIC アルゴリズムは、英数字以外の文字で区切られた 1 つまたは複数の英数字の文字列を単語として扱います。

NGRAM アルゴリズムは文字列を N-gram に分割します。N-gram は、ある文字列中の n 文字分の部分文字列です。NGRAM 単語分割は、ファジー (近似) 一致、または単語の区切りにスペースや英数字以外の文字を使用しないドキュメントに必要です。NGRAM は IN SYSTEM テーブルでサポートされます。

NGRAM 単語分割は TEXT インデックスに構築されます。そのため、テキスト設定オブジェクトの設定を使用して、NGRAM と GENERIC のどちらの TEXT インデックスを使用するかを定義します。

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。

- **library-and-entry-point-name-string** – *[operating-system:]function-name@library*



- **PREFILTER EXTERNAL NAME** – 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 (`entry_point`) とライブラリ名を指定します。
- **DROP PREFILTER** – 外部のプレフィルタを削除し、`ISYTEXTCONFIG` テーブルのプレフィルタカラムに `NULL` を設定します。

## 例

(先頭に戻る) (79 ページ)

- **例 1** – テキスト設定オブジェクト `maxTerm16` を作成し、単語の最大長を 16 に変更します。

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

- **例 2** – ストップリストの単語を `maxTerm16` 設定オブジェクトに追加します。

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

- **例 3** – 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` を更新します。

```
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** – 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` を更新します。

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

## 使用法

(先頭に戻る) (79 ページ)

TEXT インデックスは、テキスト設定オブジェクトに依存しています。SAP Sybase IQ の 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際は、その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。

関連する動作：

## SQL 文のサポート

- オートコミット。

### パーミッション

(先頭に戻る) (79 ページ)

**TERM BREAKER** 句または **PREFILTER EXTERNAL NAME** 句 - CREATE ANY EXTERNAL REFERENCE システム権限に加え、次のいずれかが必要です。

- ALTER ANY TEXT CONFIGURATION システム権限
- ALTER ANY OBJECT システム権限
- テキスト設定オブジェクトを所有している

その他のすべての句 - ユーザがテキスト設定オブジェクトの所有者であるかどうかに関係なく、ALTER ANY TEXT CONFIGURATION システム権限が必要です。

## ALTER TEXT INDEX 文

TEXT インデックスの名前の変更、移動、または定義の変更を行います。

---

**注意：** この文は、非構造化データ分析 (IQ\_UDA) ライセンスを必要とします。

---

クイックリンク：

「パラメータ」 (82 ページ)

「例」 (83 ページ)

「使用法」 (83 ページ)

「パーミッション」 (83 ページ)

### 構文

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

**alter-clause** - (構文に戻る)

**rename-object** | **move-object**

**rename-object** - (back to alter-clause)

**RENAME { AS | TO } *new-name***

**move-object** - (back to alter-clause)

**MOVE TO *dbspace-name***

### パラメータ

(先頭に戻る) (82 ページ)

- **RENAME** – TEXT インデックスの名前を変更します。
- **MOVE** – 指定した DB 領域に TEXT インデックスを移動します。

## 例

(先頭に戻る) (82 ページ)

- **例** – TEXT インデックス `MyTextIndex` を作成し、**IMMEDIATE REFRESH** として定義します。TEXT インデックスの名前を `Text_index_daily` に変更し、この TEXT インデックスを `tispace` という名前の DB 領域に移動します。

```
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

(先頭に戻る) (82 ページ)

関連する動作：

- オートコミット

## パーミッション

(先頭に戻る) (82 ページ)

**move-object** 句 - 次のいずれかが必要です。

- ALTER ANY INDEX システム権限
- ALTER ANY OBJECT システム権限
- 基本となるテーブルに対する REFERENCES 権限
- 基本となるテーブルを所有している

**rename-object** 句 - 次のいずれかが必要です。

- ALTER ANY INDEX システム権限
- ALTER ANY OBJECT システム権限
- MANAGE ANY DBSPACE システム権限
- 次のいずれかが必要です。
  - インデックスの基本となるテーブルを所有している
  - そのテーブルに対する REFERENCES 権限に加え、さらに次のいずれかが必要です。

- CREATE ANY OBJECT システム権限
- ターゲット DB 領域に対する CREATE 権限

## CREATE TEXT CONFIGURATION 文

---

テキスト設定オブジェクトを作成します。

**注意：**この文は、非構造化データ分析 (IQ\_UDA) ライセンスを必要とします。。

クイックリンク：

「パラメータ」 (84 ページ)

「例」 (84 ページ)

「使用法」 (85 ページ)

「パーミッション」 (85 ページ)

### 構文

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

### パラメータ

(先頭に戻る) (84 ページ)

- **FROM** – 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 SAP Sybase IQ テーブルのみでサポートされており、DEFAULT\_NCHAR は SQL Anywhere テーブルのみでサポートされています。

### 例

(先頭に戻る) (84 ページ)

- **例 1** – 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 を作成し、**ALTER TEXT CONFIGURATION** を使用して max\_term\_sixteen の単語の最大長を 16 に変更します。

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

## 使用法

(先頭に戻る) (84 ページ)

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。

関連する動作：

- オートコミット。

## パーミッション

(先頭に戻る) (84 ページ)

自分が所有するテキスト設定オブジェクトの場合、次のものがが必要です。

- CREATE TEXT CONFIGURATION システム権限

任意のユーザが所有するテキスト設定オブジェクトの場合、次のいずれかが必要です。

- CREATE ANY TEXT CONFIGURATION システム権限
- CREATE ANY OBJECT システム権限

すべてのテキスト設定オブジェクトに PUBLIC アクセス権があります。TEXT インデックスを作成する権限を持つユーザは、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**

---

TEXT インデックスを作成して、使用するテキスト設定オブジェクトを指定します。

---

**注意：** この文は、非構造化データ分析 (IQ\_UDA) ライセンスを必要とします。。

---

クイックリンク：

「パラメータ」 (86 ページ)

「例」 (86 ページ)

「使用法」 (86 ページ)

「パーミッション」 (87 ページ)

### 構文

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

### パラメータ

(先頭に戻る) (85 ページ)

- **ON** - TEXT インデックスを構築するテーブルとカラムを指定します。
- **IN** - TEXT インデックスが格納される 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、基本となるテーブルと同じ DB 領域に作成されます。
- **CONFIGURATION** - 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。
- **IMMEDIATE REFRESH** - (デフォルト) 基本となるテーブルでの変更によって TEXT インデックスのデータが影響を受けるたびに TEXT インデックスを更新します。SAP Sybase IQ メインストアのテーブルでは、この値のみが許可されます。作成後に IMMEDIATE REFRESH 句を変更することはできません。

### 例

(先頭に戻る) (85 ページ)

- **例 1** - max\_term\_sixteen テキスト設定オブジェクトを使用して、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 を作成します。

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

### 使用法

(先頭に戻る) (85 ページ)

ビューやテンポラリテーブル、および 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。BEGIN PARALLEL IQ...END PARALLEL IQ 文では CREATE TEXT INDEX はサポートされていません。

関連する動作：

- オートコミット。

### パーミッション

(先頭に戻る) (85 ページ)

次のいずれかが必要です。

- CREATE ANY INDEX システム権限に加え、インデックスが作成される DB 領域に対する CREATE 権限
- CREATE ANY OBJECT システム権限

## DROP TEXT CONFIGURATION 文

テキスト設定オブジェクトを削除します。

**注意：**この文は、非構造化データ分析 (IQ\_UDA) ライセンスを必要とします。。

クイックリンク：

「例」 (87 ページ)

「使用法」 (87 ページ)

「パーミッション」 (88 ページ)

### 構文

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

### 例

(先頭に戻る) (87 ページ)

- **例 1** – mytextconfig テキスト設定オブジェクトを作成して、削除します。

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

### 使用法

(先頭に戻る) (87 ページ)

テキスト設定オブジェクトを削除するには、**DROP TEXT CONFIGURATION** を使用します。

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。

## SQL 文のサポート

テキスト設定オブジェクトは、ISYTEXTCONFIG システムテーブルに格納されません。

関連する動作：

- オートコミット。

### パーミッション

(先頭に戻る) (87 ページ)

自分が所有するテキスト設定オブジェクトの場合、何も必要ありません。

任意のユーザが所有するテキスト設定オブジェクトの場合、次のいずれかが必要です。

- DROP ANY TEXT CONFIGURATION システム権限
- DROP ANY OBJECT システム権限

## DROP TEXT INDEX 文

---

データベースから TEXT インデックスを削除します。

---

**注意：** この文は、非構造化データ分析 (IQ\_UDA) ライセンスを必要とします。。

---

クイックリンク：

「パラメータ」 (88 ページ)

「例」 (88 ページ)

「使用法」 (89 ページ)

「パーミッション」 (89 ページ)

### 構文

```
DROP TEXT INDEX text-index-name  
ON [ owner ] table-name
```

### パラメータ

(先頭に戻る) (88 ページ)

- **ON** - 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。

### 例

(先頭に戻る) (88 ページ)



- **例 1** – TextIdx TEXT インデックスを作成して、削除します。

```
CREATE TEXT INDEX TextIdx ON Customers ( Street );  
DROP TEXT INDEX TextIdx ON Customers;
```

## 使用法

(先頭に戻る) (88 ページ)

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。

関連する動作：

- オートコミット。

## パーミッション

(先頭に戻る) (88 ページ)

次のいずれかが必要です。

- DROP ANY INDEX システム権限
- DROP ANY OBJECT システム権限
- インデックス付きのテーブルに対する REFERENCES 権限
- 基本となるテーブルを所有している

## SQL 文のサポート

## 関数のサポート

LONG BINARY データ型と LONG VARCHAR データ型をサポートする関数について説明します。

### ラージオブジェクトデータをサポートする関数の概要

次の表は、LONG BINARY (BLOB) データ型、LONG VARCHAR (CLOB) データ型、LONG BINARY 変数、LONG VARCHAR 変数に対する関数のサポートをまとめたものです。

この表に記載されている関数の他に、**BFILE** 関数を使用して LOB データを抽出できます。

ユーザ定義のスカラー関数および集合関数は、入力パラメータとしてラージオブジェクトデータ型をサポートしています。

表 18 : LOB データ型と変数に対する関数のサポート

関数	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

関数	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
SUBSTRING64()	あり	あり	あり	あり

\***BYTE\_LENGTH** 関数が **LONG BINARY** および **LONG VARCHAR** の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。

#### 参照：

- ラージオブジェクトカラムのユーザ定義関数のサポート (102 ページ)
- ラージオブジェクトデータのエクスポート (65 ページ)

## BIT\_LENGTH 関数

---

**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、あるいは **LONG BINARY** カラムまたは変数の名前です。

#### 使用法

**BIT\_LENGTH** は、すべての SAP Sybase IQ データ型、および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。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 [文字列]

---

文字列のバイト数を返します。

#### 構文

```
BYTE_LENGTH ( string-expression )
```

## パラメータ

パラメータ	説明
string-expression	長さが計算される文字列。

戻り値  
INT

## 備考

末尾にあるスペース文字を含めた長さが返されます。

NULL 文字を指定すると、NULL 値が返ります。

マルチバイト文字セットの文字列の場合、**BYTE\_LENGTH** の値は **CHAR\_LENGTH** で返される文字数と異なります。

非構造化データ分析機能の使用ライセンスを取得している場合は、この関数でラージオブジェクトデータを使用できます。

- **BYTE\_LENGTH** 関数が LONG BINARY および LONG VARCHAR の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 以上の場合は、**BYTE\_LENGTH** によって、**BYTE\_LENGTH64** 関数を使用する必要があります。あることを示すエラーが返されます。

## 標準と互換性

- SQL — ISO/ANSI SQL 文法のベンダ拡張。
- SAP Sybase — Adaptive Server では、サポートされていません。

## 例

次の文は、値 12 を返します。

```
SELECT BYTE_LENGTH( 'Test Message' ) FROM iq_dummy
```

## BYTE\_LENGTH64 関数 [文字列]

**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、あるいは LONG BINARY カラムまたは変数の名前です。

### 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 [文字列]**

**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、あるいは LONG BINARY カラムまたは変数の名前です。

*start* - 部分文字列の始まりを表す整数式です。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 であることを示します。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です。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** 関数、**BYTE\_SUBSTR64** 関数、**BYTE\_SUBSTR** 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および変数はサポートされません。
- **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。

## CHAR\_LENGTH 関数 [文字列]

---

**CHAR\_LENGTH** 関数は、LONG VARCHAR カラムまたは変数の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 変数の名前です。

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 [文字列]

---

**CHAR\_LENGTH64** 関数は、LONG VARCHAR カラムまたは変数の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

```
CHAR_LENGTH64 ( long-varchar-object )
```

パラメータ

*long-varchar-object* - テーブル内の LONG VARCHAR カラムまたは LONG VARCHAR 変数の名前です。

使用法

- **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\_LENGTH64** は NULL を返します。

## CHARINDEX 関数 [文字列]

---

**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表す 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

```
CHARINDEX ( string-expression, large-object-column )
```

### パラメータ

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

*large-object-column* – LONG VARCHAR カラムまたは変数、あるいは LONG BINARY カラムまたは変数の名前です。

### 使用法

- **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ります。
- 検索対象のラージオブジェクトセルに *string-expression* 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します。
- カラムに文字列が含まれない場合、**CHARINDEX** 関数はゼロ (0) を返します。
- 長さが 255 バイトを超える文字列を検索すると、NULL が返されます。
- 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 になります。
- **CHARINDEX** は、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 変数の検索をサポートします。現時点では、SQL 変数で保持できる最大長は 2GB - 1 です。



## LOCATE 関数 [文字列]

---

**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指定された文字列の位置を表す 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* - 検索対象の LONG VARCHAR カラムまたは変数、あるいは LONG BINARY カラムまたは変数の名前です。

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

*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、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、最初に一致した文字列ではなく、最後に一致した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、 $(-1 * \text{オフセット}) - 1$  で計算されます。

### 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ります。
- 検索対象の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ます。
  - *numeric-expression* を指定した場合、**LOCATE** は文字列のオフセットから検索を開始します。
  - *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します。
- カラムに文字列が含まれない場合、**LOCATE** はゼロ (0) を返します。
- 長さが 255 バイトを超える文字列を検索すると、NULL が返されます。
- 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 になります。

- **LOCATE** は、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 変数の検索をサポートします。現時点では、SQL 変数で保持できる最大長は 2GB - 1 です。

## OCTET\_LENGTH 関数 [文字列]

---

**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、あるいは LONG BINARY カラムまたは変数の名前です。

### 使用法

- 引数が NULL の場合、**OCTET\_LENGTH** は NULL を返します。
- **OCTET\_LENGTH** は、すべての SAP Sybase IQ データ型、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。

## PATINDEX 関数 [文字列]

---

**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表す 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た 126 バイトまでの文字列に制限されます。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 を返し、それ以外の場合は 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です。

### 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ります。
- 検索対象の LONG VARCHAR セルに文字列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します。
- カラムにパターンが含まれない場合、**PATINDEX** はゼロ (0) を返します。
- 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 が返されます。
- 引数のどれか1つでも NULL の場合、結果はゼロ (0) となります。
- **PATINDEX** 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。**PATINDEX** は、LONG BINARY 変数、および LONG BINARY カラムの検索をサポートしません。

## SUBSTRING 関数 [文字列]

**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。引数のいずれか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 ] )
```

### パラメータ

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

*start* - 部分文字列の始まりを表す整数式です。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であることを示します。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です。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、および `LONG BINARY` カラムの検索をサポートしません。

**ansi\_substring** データベースオプションを `ON` (デフォルト) に設定すると、負の値は無効となります。

## ANSI\_SUBSTRING オプション [TSQL]

`start` または `length` パラメータに負の値が指定されている場合に、**SUBSTRING** (**SUBSTR**) 関数の動作を制御します。

### 指定できる値

`ON`、`OFF`

### デフォルト値

`ON`

### スコープ

オプションは、データベース (`PUBLIC`) レベルまたはユーザレベルで設定できます。データベースレベルで設定した場合、値は新しいユーザのデフォルト値になりますが、既存のユーザには影響を与えません。ユーザレベルで設定した場合は、そのユーザの `PUBLIC` 値のみが上書きされます。自分のオプションを設定する場合は、システム権限は必要ありません。自分以外のユーザのオプションをデータベースレベルまたはユーザレベルで設定する場合は、システム権限が必要です。

このオプションを設定するには、`SET ANY PUBLIC OPTION` システム権限が必要です。個々の接続または `PUBLIC` ロールに一時的に設定できます。すぐに有効になります。

### 備考

**ANSI\_SUBSTRING** オプションを `ON` に設定した場合、**SUBSTRING** 関数は `ANSI/ISO SQL/2003` と同じ動作をします。開始オフセットが負またはゼロの場合は、文字列の左側に非文字が埋め込まれているものと解釈され、負の長さが指定された場合はエラーが発生します。

このオプションを `OFF` に設定した場合、**SUBSTRING** 関数の動作は以前のバージョンの `SAP Sybase IQ` と同じになります。負の開始オフセットは、文字列の末尾からのオフセットになり、負の長さは、開始オフセットから左に向かって指定の部分文字列の終端までの長さになります。開始オフセット `0` を使用した場合、開始オフセット `1` と同じ結果となります。

**SUBSTRING** 関数では、正以外の開始オフセットおよび負の長さを使用しないでください。可能な限り、**SUBSTRING** 関数の代わりに **LEFT** 関数または **RIGHT** 関数を使用してください。

#### 例

次の例は、**ANSI\_SUBSTRING** オプションの設定に応じて **SUBSTRING** 関数によって返される値の違いを示しています。

```
SUBSTRING( 'abcdefgh', -2, 4 );
ansi_substring = Off ==> 'gh'
// substring starts at second-last character
ansi_substring = On ==> 'gh'
// takes the first 4 characters of
// ???abcdefgh and discards all ?
```

```
SUBSTRING( 'abcdefgh', 4, -2 );
ansi_substring = Off ==> 'cd'
ansi_substring = On ==> value -2 out of range
for destination
```

```
SUBSTRING( 'abcdefgh', 0, 4 );
ansi_substring = Off ==> 'abcd'
ansi_substring = On ==> 'abcd'
```

## SUBSTRING64 関数 [文字列]

**SUBSTRING64** 関数は、ラージオブジェクトカラムまたは変数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、あるいは LONG BINARY カラムまたは変数の名前です。

*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

col1 という名前のカラムに文字列 ("ABCDEFGF") が格納されている場合、**SUBSTRING64** は次の値を返します。

## 関数のサポート

`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

- 引数のいずれか1つでも NULL の場合、**SUBSTRING64** は NULL を返します。
- **SUBSTRING64** 関数、**SUBSTRING** 関数、**SUBSTR** 関数、**BYTE\_SUBSTR** 関数、**BYTE\_SUBSTR64** 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および変数はサポートされません。
- **SUBSTRING64** は、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 変数の検索をサポートします。現時点では、SQL 変数で保持できる最大長は 2GB - 1 です。

## ラージオブジェクトカラムの集合関数のサポート

---

LONG BINARY カラムと LONG VARCHAR カラムでサポートされている唯一の集合関数は、**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。

## ラージオブジェクトカラムのユーザ定義関数のサポート

---

ユーザ定義のスカラ関数と集合関数は、入力パラメータとして 4GB (ギガバイト) までのラージオブジェクトデータ型 LONG VARCHAR (CLOB) と LONG BINARY (BLOB) をサポートしています。LOB データ型は、出力パラメータとしてはサポートされていません。

ユーザ定義関数のサポート機能を使用するには、正規のライセンスを取得しておく必要があります。

## 索引

## A

## Adaptive Server

IMAGE データの挿入 73

TEXT データの挿入 75

## ALTER TEXT CONFIGURATION 15

構文 79

## ALTER TEXT INDEX 7

構文 82

## ANSI\_SUBSTRING オプション 100

## B

## BEGIN PARALLEL IQ 文 85

## BFILE 関数 65

構文 65

抽出の例 66

抽出機能 65

例 66

## BIT\_LENGTH 関数

構文 92

説明 92

## BLOB

BIT\_LENGTH 関数 92

BYTE\_LENGTH64 関数 93

BYTE\_SUBSTR 関数 94

BYTE\_SUBSTR64 関数 94

CHARINDEX 関数 96

IMAGE データの挿入 73

LOCATE 関数 97

LONG BINARY 73

OCTET\_LENGTH 関数 98

sp\_iqindexsize 63

SUBSTRING64 関数 101

TEXT インデックス 78

インデックス 73, 78

インデックスサポート 78

カラム 73

クエリ 25, 52

サイズ 73

ストアドプロシージャのサポート 55

データのエクスポート 65

データのロード 67

データの更新 73

データの挿入 73

データ型 73

データ型変換 74

バイナリラージオブジェクト 73

パフォーマンスのモニタリング 54

プリフェッチ 73

ユーザ定義関数のサポート 102

関数のサポート 91

集合関数のサポート 102

説明 73

変更 73

変数 76

変数の関数のサポート 91

## BLOB 変数

データ型変換 77

## BYTE\_LENGTH 関数 92

## BYTE\_LENGTH64 関数

構文 93

説明 93

## BYTE\_SUBSTR 関数

構文 94

説明 94

## BYTE\_SUBSTR64 関数

構文 94

説明 94

## C

## char

単語への分割 55

## CHAR\_LENGTH 関数

構文 95

説明 95

## CHAR\_LENGTH64 関数

構文 95

説明 95

## CHARINDEX 関数

構文 96

説明 96

## 索引

### CLOB

- BIT\_LENGTH 関数 92
  - BYTE\_LENGTH64 関数 93
  - BYTE\_SUBSTR 関数 94
  - BYTE\_SUBSTR64 関数 94
  - CHAR\_LENGTH 関数 95
  - CHAR\_LENGTH64 関数 95
  - CHARINDEX 関数 96
  - LOCATE 関数 97
  - LONG VARCHAR 74
  - OCTET\_LENGTH 関数 98
  - PATINDEX 関数 98
  - sp\_iqindexsize 64
  - SUBSTRING 関数 99
  - SUBSTRING64 関数 101
  - TEXT インデックス 78
  - TEXT データの挿入 75
  - WD インデックス 75, 78
  - WORD インデックス 75, 78
  - インデックス 75, 78
  - インデックスサポート 73, 78
  - カラム 75
  - キャラクターラージオブジェクト 74
  - クエリ 25, 53
  - サイズ 74
  - ストアドプロシージャのサポート 55
  - データ型変換 75
  - データのエクスポート 65
  - データのロード 67
  - データの更新 75
  - データの挿入 75
  - データ型 74
  - ユーザ定義関数のサポート 102
  - 関数のサポート 91
  - 集合関数のサポート 102
  - 説明 73
  - 変更 75
  - 変数 76
  - 変数の関数のサポート 91
- CLOB 変数
- データ型変換 77
- CONTAINS
- テーブル式 41, 49

### CONTAINS 句

- 近似一致の決定 49
- CONTAINS の例
- テキスト設定オブジェクト 17
- CONTAINS 条件
- TEXT インデックス 41
- contains-expression
- FROM 句 41
- CREATE TEXT CONFIGURATION 11
- 構文 84
- CREATE TEXT INDEX 5
- 構文 85

## D

### DB 領域

- TEXT インデックス 7
  - 変更 7
- default\_char 10
- default\_nchar 10
- DROP TEXT CONFIGURATION 16
- 構文 87
- DROP TEXT INDEX 7
- 構文 88

## E

- ENABLE\_LOB\_VARIABLES オプション 77
- END PARALLEL IQ
- CREATE TEXT INDEX 85

## F

### FROM 句

- CONTAINS 41, 49
- contains-expression 41
- 構文 41

## I

### IMAGE データ

- ASE からの挿入 73
- LONG BINARY への挿入 73



## L

## LOAD TABLE

- セカンダリロードファイル 67
- プライマリロードファイル 67
- 拡張構文 67
- 例 68

## LOB

- インデックスサポート 78
- データのエキスポート 65
- データのロード 67
- データの更新 73, 75
- データの挿入 73, 75
- ユーザ定義関数のサポート 102
- 一般的なデータソース 1
- 概要 1
- 関数のサポート 91

## LOB データの圧縮 61

- 設定の表示 62
- 設定の変更 61

## LOB のデータ圧縮 61

- 設定の表示 62
- 設定の変更 61

## LOB 圧縮

- 設定の表示 62
- 設定の変更 61
- 無効化 61
- 有効化 61

## LOB 変数

- データ型変換 77

## LOCATE 関数

- 構文 97
- 説明 97

## LONG BINARY

- BIT\_LENGTH 関数 92
- BLOB 73
- BYTE\_LENGTH64 関数 93
- BYTE\_SUBSTR 関数 94
- BYTE\_SUBSTR64 関数 94
- CHARINDEX 関数 96
- DELETE 73
- IMAGE データの挿入 73
- INSERT 73
- LOAD TABLE 73
- LOCATE 関数 97

OCTET\_LENGTH 関数 98

SELECT...INTO 73

sp\_iqindexsize 63

SUBSTRING64 関数 101

TEXT インデックス 78

TRUNCATE 73

UPDATE 73

インデックス 73, 78

インデックスサポート 78

カラム 73

クエリ 25, 52

サイズ 73

ストアードプロシージャのサポート 55

データのエキスポート 65

データのロード 67

データの更新 73

データの挿入 73

データ型変換 74

バイナリラージオブジェクト 73

パフォーマンスのモニタリング 54

ユーザ定義関数のサポート 102

集合関数のサポート 102

変更 73

変数 76

## LONG BINARY 変数

データ型変換 77

## LONG VARCHAR

BIT\_LENGTH 関数 92

BYTE\_LENGTH64 関数 93

BYTE\_SUBSTR 関数 94

BYTE\_SUBSTR64 関数 94

CHAR\_LENGTH 関数 95

CHAR\_LENGTH64 関数 95

CHARINDEX 関数 96

CLOB 74

DELETE 75

INSERT 75

LOAD TABLE 75

LOCATE 関数 97

OCTET\_LENGTH 関数 98

PATINDEX 関数 98

SELECT...INTO 75

sp\_iqindexsize 64

SUBSTRING 関数 99

SUBSTRING64 関数 101

## 索引

TEXT インデックス 78  
TEXT データの挿入 75  
TRUNCATE 75  
UPDATE 75  
WD インデックス 75, 78  
WORD インデックス 75, 78  
インデックス 75, 78  
インデックスサポート 78  
カラム 75  
キャラクターラージオブジェクト 74  
クエリ 25, 53  
サイズ 74  
ストアドプロシージャのサポート 55  
データ型変換 75  
データのエクスポート 65  
データのロード 67  
データの更新 75  
データの挿入 75  
ユーザ定義関数のサポート 102  
集合関数のサポート 102  
変更 75  
変数 76  
LONG VARCHAR のエクスポート 65  
LONG VARCHAR 変数  
データ型変換 77

## M

MAX\_PREFIX\_PER\_CONTAINS\_PHRASE オプション 19

## N

nchar  
単語への分割 56  
NGRAM  
TEXT インデックス検索 49–51  
NGRAM TEXT インデックス 9  
あいまい検索 9

## O

OCTET\_LENGTH 関数  
構文 98  
説明 98

## P

PATINDEX 関数  
構文 98  
説明 98

## S

sa\_char\_terms ストアドプロシージャ 55  
sa\_external\_library\_unload ストアドプロシージャ 23, 60  
sa\_list\_external\_library ストアドプロシージャ 22, 60  
sa\_nchar\_terms ストアドプロシージャ 56  
sa\_text\_index\_stats ストアドプロシージャ 57  
sa\_text\_index\_vocab ストアドプロシージャ 58  
SECONDARY\_FILE\_ERROR オプション 69  
SELECT 文  
FROM 句の構文 41  
sp\_iqindexsize  
BLOB 63  
CLOB 64  
LONG BINARY 63  
LONG VARCHAR 64  
キャラクターラージオブジェクト 64  
バイナリラージオブジェクト 63  
sp\_iqindexsize ストアドプロシージャ 63  
sp\_iqsetcompression ストアドプロシージャ 61  
sp\_iqshowcompression ストアドプロシージャ 62  
STRING\_RTRUNCATION オプション 74, 75  
SUBSTRING 関数  
構文 99  
説明 99  
SUBSTRING64 関数  
構文 101  
説明 101

## T

TEXT インデックス 3, 78  
CONTAINS 条件 41  
DB 領域の変更 7  
NGRAM 9  
WD インデックスとの比較 4

あいまい検索 49, 50  
 テキスト設定オブジェクト 9, 10  
 リスト 6  
 ローの削除 8  
 更新 8  
 作成 5, 85  
 削除 7, 88  
 制限 6  
 単語 55  
 統計 57, 58  
 非あいまい検索 49, 51  
 変更 82  
 編集 7  
 TEXT データ  
   ASE からの挿入 75  
   LONG VARCHAR への挿入 75  
 TEXT\_DELETE\_METHOD オプション 8  
 TRIM\_PARTIAL\_MBC オプション 70

## W

WD インデックス 78  
   TEXT インデックスとの比較 4

## あ

あいまい検索 49, 50  
 アップグレード  
   LONG BINARY 73  
   既存の LONG BINARY カラム 73  
 アンロード  
   外部ライブラリ 23, 60

## い

インスタンス  
   外部ライブラリ 60  
 インデックス  
   BLOB 73, 78  
   CLOB 75, 78  
   LOB 78  
   LONG BINARY 73, 78  
   LONG VARCHAR 75, 78  
   TEXT 3, 78  
   WD 75, 78  
   WORD 75, 78

キャラクタラージオブジェクト 75, 78  
 バイナリラージオブジェクト 73, 78  
 ラージオブジェクトデータ 78  
 リスト 6  
 更新 8  
 全文検索 2, 25  
 包含 75, 78

## え

エクスポート  
   BFILE の例 66  
   BFILE 関数 65  
   BLOB 65  
   CLOB 65  
   LOB 65  
   LONG BINARY 65  
   ラージオブジェクトデータ 65

## お

オプション  
   ENABLE\_LOB\_VARIABLES 77  
   MAX\_PREFIX\_PER\_CONTAINS\_PHRASE  
     19  
   TEXT\_DELETE\_METHOD 8  
   非構造化データ分析 1

## き

キャラクタラージオブジェクト  
   BIT\_LENGTH 関数 92  
   BYTE\_LENGTH64 関数 93  
   BYTE\_SUBSTR 関数 94  
   BYTE\_SUBSTR64 関数 94  
   CHAR\_LENGTH 関数 95  
   CHAR\_LENGTH64 関数 95  
   CHARINDEX 関数 96  
   CLOB 74  
   LOCATE 関数 97  
   LONG VARCHAR 74  
   OCTET\_LENGTH 関数 98  
   PATINDEX 関数 98  
   sp\_iqindexsize 64  
   SUBSTRING 関数 99  
   SUBSTRING64 関数 101

## 索引

TEXT インデックス 78  
TEXT データの挿入 75  
WD インデックス 75, 78  
WORD インデックス 75, 78  
インデックス 75, 78  
インデックスサポート 73, 78  
カラム 75  
クエリ 25, 53  
サイズ 74  
ストアドプロシージャのサポート 55  
データ型変換 75  
データ型 74  
集合関数のサポート 102  
説明 73  
変更 75  
変数 76  
キャラクターラージオブジェクト変数  
データ型変換 77

## く

クエリ  
BLOB 25, 52  
CLOB 25, 53  
LONG BINARY 25, 52  
LONG VARCHAR 25, 53  
キャラクターラージオブジェクト 25, 53  
バイナリラージオブジェクト 25, 52  
クエリ結果のソート 49

## す

ストアドプロシージャ  
BLOB 55  
CLOB 55  
LONG BINARY 55  
LONG VARCHAR 55  
sa\_char\_terms 55  
sa\_external\_library\_unload 60  
sa\_list\_external\_library 60  
sa\_nchar\_terms 56  
sa\_text\_index\_stats 57  
sa\_text\_index\_vocab 58  
sp\_iqindexsize 63  
sp\_iqsetcompression 61  
sp\_iqshowcompression 62  
キャラクターラージオブジェクト 55

バイナリラージオブジェクト 55  
ストップリスト 9, 10, 14  
変更 15

## て

データベースオプション  
ENABLE\_LOB\_VARIABLES 77  
MAX\_PREFIX\_PER\_CONTAINS\_PHRASE  
19  
TEXT\_DELETE\_METHOD 8  
データ型  
BLOB 73  
CLOB 74  
LONG BINARY 73  
LONG VARCHAR 74  
データ型変換  
LONG BINARY から BINARY へ 74  
LONG BINARY から VARBINARY へ 74  
LONG BINARY 変数 77  
LONG VARCHAR から CHAR 75  
LONG VARCHAR から VARCHAR 75

テキスト検索  
FROM contains-expression 41  
テキスト設定オブジェクト 9  
CONTAINS の例 17  
デフォルト 10  
リスト 15  
作成 11, 84  
削除 16, 87  
設定 11-14  
単語分割 11  
変更 15, 79  
例 16, 17

## は

バイナリラージオブジェクト  
BIT\_LENGTH 関数 92  
BLOB 73  
BYTE\_LENGTH64 関数 93  
BYTE\_SUBSTR 関数 94  
BYTE\_SUBSTR64 関数 94  
IMAGE データの挿入 73  
LONG BINARY 73  
OCTET\_LENGTH 関数 98

- sp\_iqindexsize 63
  - SUBSTRING64 関数 101
  - TEXT インデックス 78
  - インデックス 73, 78
  - インデックスサポート 78
  - カラム 73
  - クエリ 25, 52
  - サイズ 73
  - ストアドプロシージャのサポート 55
  - データ型 73
  - データ型変換 74
  - パフォーマンスのモニタリング 54
  - ユーザ定義関数のサポート 102
  - 集合関数のサポート 102
  - 説明 73
  - 変更 73
  - 変数 76
  - バイナリラージオブジェクト変数
    - データ型変換 77
  - パフォーマンスモニタ
    - BLOB 54
    - LONG BINARY 54
    - バイナリラージオブジェクト 54
- ひ**
- 非あいまい検索 49, 51
- ふ**
- プリフェッチ 73
  - プレフィクス
    - 単語の制限 19
  - プレフィルタライブラリ 9-11, 21
    - 制限 22
    - 単語の最小長 12
    - 単語の最大長 13
- ま**
- マルチバイト文字
    - TRIM\_PARTIAL\_MBC オプション 70
  - ロード時のトランケート 70
    - 部分的なマルチバイト文字の削除 70
  - マルチプレックスサーバ
    - 外部ライブラリ 22
- ら**
- ラージオブジェクトデータ
    - インデックスサポート 78
    - エクスポート 65
    - ロード 67
    - 更新 73, 75
    - 挿入 73, 75
  - ライセンス 1
  - ライブラリ、外部 9, 10
- り**
- リスト
    - TEXT インデックス 6
    - テキスト設定オブジェクト 15
    - 外部ライブラリ 60
- ろ**
- ロード
    - BLOB 67
    - CLOB 67
    - LOAD TABLE の例 68
    - LOB 67
    - LONG BINARY 67
    - LONG VARCHAR 67
    - SECONDARY\_FILE\_ERROR オプション 69
    - TRIM\_PARTIAL\_MBC オプション 70
    - エラーの制御 69
    - ラージオブジェクトデータ 67
    - 後続ブランクの削除 69
    - 文字データのトランケート 70

