Oracle Text

リファレンス

リリース 9.2

2002 年 7 月
部品番号 : J06273-01

ORACLE®

Oracle Text リファレンス , リリース 9.2

部品番号：J06273-01

原本名：Oracle Text Reference, Release 9.2

原本部品番号：A96518-01

原本著者：Colin McGregor

原本協力者：Omar Alonso, Shamim Alpha, Steve Buxton, Chung-Ho Chen, Jack Chen, Yun Cheng, Michele Cyran, Paul Dixon, Mohammad Faisal, Elena Huang, Garrett Kaminaga, Ji Sun Kang, Bryn Llewellyn, Wesley Lin, Yasuhiro Matsuda, Gerda Shank, and Steve Yang.

Printed in Japan

目次

はじめに xiii

Oracle Text の新機能 xxi

1 SQL 文と演算子

ALTER INDEX 1-2
ALTER TABLE: サポートされるパーティション化文 1-12
CATSEARCH 1-16
CONTAINS 1-23
CREATE INDEX 1-28
DROP INDEX 1-45
MATCHES 1-46
SCORE 1-47

2 索引付け

概要 2-2
プリファレンスの作成 2-2
データストア型 2-3
DIRECT_DATASTORE 2-3
MULTI_COLUMN_DATASTORE 2-4
DETAIL_DATASTORE 2-7
FILE_DATASTORE 2-10
URL_DATASTORE 2-11
USER_DATASTORE 2-15
NESTED_DATASTORE 2-18

フィルタ型 2-21
CHARSET_FILTER 2-22
INSO_FILTER 2-24
NULL_FILTER 2-28
USER_FILTER 2-28
PROCEDURE_FILTER 2-30
レクサー型 2-35
BASIC_LEXER 2-35
MULTI_LEXER 2-43
CHINESE_VGRAM_LEXER 2-45
CHINESE_LEXER 2-46
JAPANESE_VGRAM_LEXER 2-46
JAPANESE_LEXER 2-47
KOREAN_LEXER 2-48
KOREAN_MORPH_LEXER 2-49
USER_LEXER 2-53
ワードリスト型 2-69
BASIC_WORDLIST 2-69
BASIC_WORDLIST の例 2-74
記憶域型 2-76
BASIC_STORAGE 2-76
セクション・グループ型 2-78
セクション・グループの例 2-80
分類型 2-81
RULE_CLASSIFIER 2-81
ストップリスト 2-82
マルチ言語のストップリスト 2-82
ストップリストの作成 2-83
デフォルトのストップリストの変更 2-83
システム定義プリファレンス 2-84
データ記憶域 2-84
フィルタ 2-85
レクサー 2-85
セクション・グループ 2-86
ストップリスト 2-87
記憶域 2-87
ワードリスト 2-87

システム・パラメータ 2-88
汎用システム・パラメータ 2-88
デフォルトの索引付けパラメータ 2-89

3 CONTAINS 問合せ演算子

演算子の優先順位 3-3
グループ 1 演算子 3-3
グループ 2 の演算子および文字 3-3
プロシージャ型演算子 3-3
優先順位の例 3-4
優先順位の変更 3-4
ABOUT 3-5
ACCUMulate（,） 3-9
AND（&） 3-11
BROADER TERM（BT、BTG、BTP、BTI） 3-12
EQUIValence（=） 3-15
FUZZY 3-16
HASPATH 3-18
INPATH 3-20
MINUS（-） 3-28
NARROWER TERM（NT、NTG、NTP、NTI） 3-29
NEAR（;） 3-32
NOT（~） 3-36
OR（|） 3-37
PREFERRED TERM（PT） 3-38
RELATED TERM（RT） 3-39
SOUNDEX（!） 3-40
STEM（$） 3-41
ストアド・クエリー式（SQE） 3-42
SYNONYM（SYN） 3-43
THRESHOLD（>） 3-45
TRANSLATION TERM（TR） 3-46
TRANSLATION TERM SYNONYM（TRSYN） 3-48
TOP TERM（TT） 3-50
WEIGHT（*） 3-51

ワイルド・カード（% _） 3-53
右側切捨て問合せ 3-53
左側切捨て問合せおよび左右切捨て問合せ 3-53
ワイルド・カード問合せのパフォーマンス改善 3-54
WITHIN 3-55

4 問合せの特殊文字

グループ化文字 4-2
エスケープ文字 4-2
エスケープ文字の問合せ 4-3
予約語 4-3

5 CTX_ADM パッケージ

RECOVER 5-2
SET_PARAMETER 5-3

6 CTX_CLS パッケージ

TRAIN 6-2

7 CTX_DDL パッケージ

ADD_ATTR_SECTION 7-3
ADD_FIELD_SECTION 7-5
ADD_INDEX 7-9
ADD_SPECIAL_SECTION 7-11
ADD_STOPCLASS 7-13
ADD_STOP_SECTION 7-14
ADD_STOPTHEME 7-16
ADD_STOPWORD 7-17
ADD_SUB_LEXER 7-19
ADD_ZONE_SECTION 7-22
CREATE_INDEX_SET 7-26
CREATE_POLICY 7-27
CREATE_PREFERENCE 7-30
CREATE_SECTION_GROUP 7-33
CREATE_STOPLIST 7-36

DROP_INDEX_SET 7-38
DROP_POLICY 7-39
DROP_PREFERENCE 7-40
DROP_SECTION_GROUP 7-41
DROP_STOPLIST 7-42
OPTIMIZE_INDEX 7-43
REMOVE_INDEX 7-46
REMOVE_SECTION 7-47
REMOVE_STOPCLASS 7-49
REMOVE_STOPTHEME 7-50
REMOVE_STOPWORD 7-51
SET_ATTRIBUTE 7-52
SYNC_INDEX 7-53
UNSET_ATTRIBUTE 7-55
UPDATE_POLICY 7-56

8 CTX_DOC パッケージ

FILTER 8-2
GIST 8-5
HIGHLIGHT 8-10
IFILTER 8-13
MARKUP 8-14
PKENCODE 8-19
SET_KEY_TYPE 8-20
THEMES 8-21
TOKENS 8-24

9 CTX_OUTPUT パッケージ

ADD_EVENT 9-2
END_LOG 9-3
LOGFILENAME 9-4
REMOVE_EVENT 9-5
START_LOG 9-6

10 CTX_QUERY パッケージ

BROWSE_WORDS 10-2
COUNT_HITS 10-5
EXPLAIN 10-6
HFEEDBACK 10-10
REMOVE_SQE 10-14
STORE_SQE 10-15

11 CTX_REPORT パッケージ

CTX_REPORT のプロシージャ 11-2
複数バージョンのファンクションの使用方法 11-2
DESCRIBE_INDEX 11-3
DESCRIBE_POLICY 11-4
CREATE_INDEX_SCRIPT 11-5
CREATE_POLICY_SCRIPT 11-6
INDEX_SIZE 11-7
INDEX_STATS 11-8
TOKEN_INFO 11-12
TOKEN_TYPE 11-14

12 CTX_THES パッケージ

ALTER_PHRASE 12-3
ALTER_THESAURUS 12-5
BT 12-7
BTG 12-10
BTI 12-12
BTP 12-14
CREATE_PHRASE 12-16
CREATE_RELATION 12-18
CREATE_THESAURUS 12-20
CREATE_TRANSLATION 12-21
DROP_PHRASE 12-22
DROP_RELATION 12-24
DROP_THESAURUS 12-27
DROP_TRANSLATION 12-28
HAS_RELATION 12-29

NT 12-30
NTG 12-33
NTI 12-35
NTP 12-37
OUTPUT_STYLE 12-39
PT 12-40
RT 12-42
SN 12-44
SYN 12-45
THES_TT 12-48
TR 12-49
TRSYN 12-51
TT 12-53
UPDATE_TRANSLATION 12-55

13 CTX_ULEXER パッケージ

WILDCARD_TAB 13-2

14 実行可能ファイル

シソーラス・ローダー（ctxload） 14-2
テキストのロード 14-2
ctxload 構文 14-2
ctxload の例 14-5
ナレッジ・ベース拡張コンパイラ（ctxkbtc） 14-6
ctxkbtc 構文 14-7
ctxkbtc の使用上の注意 14-8
ctxkbtc の制限事項 14-8
シソーラス語句に対する ctxkbtc の制約 14-8
シソーラス・リレーションに対する ctxkbtc の制約 14-9
ナレッジ・ベースの拡張 14-10
言語固有のナレッジ・ベースの追加 14-11
複数のシソーラスの優先順位 14-12
拡張ナレッジ・ベースのサイズ制限 14-12

A 結果表

CTX_QUERY 結果表 A-2
実行計画表 A-2
HFEEDBACK 表 A-4
CTX_DOC 結果表 A-7
フィルタ表 A-8
要点表 A-8
ハイライト表 A-9
マークアップ表 A-9
テーマ表 A-10
トークン表 A-10
CTX_THES 結果表およびデータ型 A-11
EXP_TAB 表型 A-11

B サポートされているドキュメント形式

ドキュメント・フィルタ処理テクノロジ B-2
サポートされているプラットフォーム B-2
環境変数 B-3
UNIX プラットフォームに対する要件 B-3
OLE2 オブジェクトのサポート B-4
サポートされているドキュメント形式 B-4
ワード処理 - 共通 B-5
ワード処理 - DOS B-5
ワード処理 - インターナショナル B-6
ワード処理 - Windows B-7
ワード処理 - Macintosh B-8
ワード処理 - Unix B-8
デスクトップ・パブリッシング B-8
スプレッドシート形式 B-8
データベース形式 B-10
表示形式 B-10
プレゼンテーション形式 B-11
標準グラフィック形式 B-12
その他 B-14
サポートされていない形式 B-14

C ロード例

SQL の INSERT 例 C-2
SQL*Loader 例 C-2
表の作成 C-2
SQL*Loader コマンドの発行 C-3
`ctxload` シソーラス・インポート・ファイルの構造 C-4
代替階層構造 C-7
インポート・ファイル内の語句の使用上の注意 C-8
インポート・ファイル内の関連語句の使用上の注意 C-9
インポート・ファイル例 C-10

D 提供されるストップリスト

英語のデフォルト・ストップリスト D-2
中国語（繁体字）のストップリスト D-3
中国語（簡体字）のストップリスト D-4
デンマーク語（dk）のデフォルト・ストップリスト D-5
オランダ語（nl）のデフォルト・ストップリスト D-5
フィンランド語（sf）のデフォルト・ストップリスト D-7
フランス語（f）のデフォルト・ストップリスト D-8
ドイツ語（d）のデフォルト・ストップリスト D-9
イタリア語（i）のデフォルト・ストップリスト D-10
ポルトガル語（pt）のデフォルト・ストップリスト D-11
スペイン語（e）のデフォルト・ストップリスト D-12
スウェーデン語（s）のデフォルト・ストップリスト D-13

E 代替スペルの規則

概要 E-2
代替スペルの使用可能 E-2
代替スペルの使用禁止 E-2
ドイツ語の代替スペル E-3
デンマーク語の代替スペル E-3
スウェーデン語の代替スペル E-4

F スコア付けのアルゴリズム

ワード問合せのスコア付けのアルゴリズム F-2
例 F-3
DML およびスコア付け F-3

G ビュー

CTX_CLASSES G-3
CTX_INDEXES G-3
CTX_INDEX_ERRORS G-4
CTX_INDEX_OBJECTS G-4
CTX_INDEX_PARTITIONS G-5
CTX_INDEX_SETS G-5
CTX_INDEX_SET_INDEXES G-6
CTX_INDEX_SUB_LEXERS G-6
CTX_INDEX_SUB_LEXER_VALUES G-7
CTX_INDEX_VALUES G-7
CTX_OBJECTS G-8
CTX_OBJECT_ATTRIBUTES G-8
CTX_OBJECT_ATTRIBUTE_LOV G-9
CTX_PARAMETERS G-9
CTX_PENDING G-11
CTX_PREFERENCES G-11
CTX_PREFERENCE_VALUES G-12
CTX_SECTIONS G-12
CTX_SECTION_GROUPS G-13
CTX_SQES G-13
CTX_STOPLISTS G-13
CTX_STOPWORDS G-14
CTX_SUB_LEXERS G-14
CTX_THESAURI G-14
CTX_THES_PHRASES G-15
CTX_USER_INDEXES G-15
CTX_USER_INDEX_ERRORS G-16
CTX_USER_INDEX_OBJECTS G-16
CTX_USER_INDEX_PARTITIONS G-16
CTX_USER_INDEX_SETS G-17

CTX_USER_INDEX_SET_INDEXES G-17
CTX_USER_INDEX_SUB_LEXERS G-18
CTX_USER_INDEX_SUB_LEXER_VALS G-18
CTX_USER_INDEX_VALUES G-18
CTX_USER_PENDING G-19
CTX_USER_PREFERENCES G-19
CTX_USER_PREFERENCE_VALUES G-19
CTX_USER_SECTIONS G-20
CTX_USER_SECTION_GROUPS G-20
CTX_USER_SQES G-21
CTX_USER_STOPLISTS G-21
CTX_USER_STOPWORDS G-21
CTX_USER_SUB_LEXERS G-22
CTX_USER_THESAURI G-22
CTX_USER_THES_PHRASES G-22
CTX_VERSION G-23

H ストップワード変換

ストップワード変換の理解 H-2
ワード変換 H-3
AND 変換 H-3
OR 変換 H-4
ACCUMulate 変換 H-4
MINUS 変換 H-5
NOT 変換 H-5
EQUIValence 変換 H-6
NEAR 変換 H-6
WEIGHT 変換 H-7
THRESHOLD 変換 H-7
WITHIN 変換 H-7

I　英語のナレッジ・ベースのカテゴリ階層

ブランチ 1: science and technology I-2
ブランチ 2: business and economics I-9
ブランチ 3: government and military I-10
ブランチ 4: social environment I-11
ブランチ 5: geography I-14
ブランチ 6: abstract ideas and concepts I-18

索引

はじめに

このマニュアルでは、Oracle Text のリファレンス情報を記載しています。Oracle Text の索引の作成、Oracle Text 問合せの発行、ドキュメントの表示および Oracle Text PL/SQL パッケージの使用に関するリファレンスとして使用できます。

この章の内容は次のとおりです。

- 対象読者
- このマニュアルの構成
- 関連文書
- 表記規則

対象読者

このマニュアルは、Oracle Text アプリケーション開発者または Oracle Text システムのメンテナンスを行うシステム管理者を対象としています。

このマニュアルを使用するには、Oracle リレーショナル・データベース管理システム（RDBMS）、SQL、SQL*Plus および PL/SQL での作業経験が必要です。詳細は、使用しているハードウェアおよびソフトウェアのドキュメントを参照してください。

Oracle RDBMS および関連する Oracle のツール製品に慣れていない場合は、『Oracle9*i* データベース概要』を参照してください。Oracle ドキュメント全体で使用される概念および用語が広く紹介されています。

このマニュアルの構成

このマニュアルの構成は、次のとおりです。

第 1 章「SQL 文と演算子」
この章では、Oracle Text で使用可能な SQL 文と演算子について説明します。

第 2 章「索引付け」
この章では、Oracle Text の索引の作成に使用できる索引タイプについて説明します。

第 3 章「CONTAINS 問合せ演算子」
この章では、CONTAINS 問合せに使用できる演算子について説明します。

第 4 章「問合せの特殊文字」
この章では、CONTAINS 問合せに使用できる特殊文字について説明します。

第 5 章「CTX_ADM パッケージ」
この章では、PL/SQL パッケージ CTX_ADM のプロシージャについて説明します。

第 6 章「CTX_CLS パッケージ」
この章では、PL/SQL パッケージ CTX_CLS のプロシージャについて説明します。

第 7 章「CTX_DDL パッケージ」
この章では、PL/SQL パッケージ CTX_DDL のプロシージャについて説明します。索引のメンテナンスには、このパッケージを使用します。

第 8 章「CTX_DOC パッケージ」
この章では、PL/SQL パッケージ CTX_DOC のプロシージャについて説明します。ドキュメントの表示などのドキュメント・サービスには、このパッケージを使用します。

第 9 章「CTX_OUTPUT パッケージ」
この章では、PL/SQL パッケージ CTX_OUTPUT のプロシージャについて説明します。索引エラー・ログ・ファイルの管理には、このパッケージを使用します。

第 10 章「CTX_QUERY パッケージ」
この章では、PL/SQL パッケージ CTX_QUERY のプロシージャについて説明します。ヒット数のカウント、問合せ実行計画情報の生成などの問合せの管理には、このパッケージを使用します。

第 11 章「CTX_REPORT パッケージ」
この章では、PL/SQL パッケージ CTX_REPORT のプロシージャについて説明します。様々な索引レポートの作成には、このパッケージを使用します。

第 12 章「CTX_THES パッケージ」
この章では、PL/SQL パッケージ CTX_THES のプロシージャについて説明します。シソーラスの管理には、このパッケージを使用します。

第 13 章「CTX_ULEXER パッケージ」
この章では、PL/SQL パッケージ CTX_ULEXER のデータ型について説明します。このパッケージは、ユーザー定義のレクサーとともに使用します。

第 14 章「実行可能ファイル」
この章では、シソーラス・ロード・プログラムの ctxload やナレッジ・ベース・コンパイラの ctxkbtc など、Oracle Text で提供される実行可能プログラムについて説明します。

付録 A「結果表」
この付録では、CTX_DOC、CTX_QUERY および CTX_THES パッケージのプロシージャの結果表について説明します。

付録 B「サポートされているドキュメント形式」
この付録では、索引付け用の Inso フィルタによるフィルタ処理がサポートされているドキュメント形式について説明します。

付録 C「ロード例」
この付録では、テキスト表の移入についての基本的な例を示します。

付録 D「提供されるストップリスト」
この付録では、サポート対象言語ごとに提供されるストップリストについて説明します。

付録 E「代替スペルの規則」
この付録では、ドイツ語、デンマーク語およびスウェーデン語で使用する代替スペルの規則について説明します。

付録 F「スコア付けのアルゴリズム」
この付録では、ワード問合せに使用するスコア付けのアルゴリズムについて説明します。

付録 G「ビュー」
この付録では、Oracle Text のビューについて説明します。

付録 H「ストップワード変換」
この付録では、ストップワード変換について説明します。

付録 I「英語のナレッジ・ベースのカテゴリ階層」
この付録では、提供されるナレッジ・ベースについて説明します。

関連文書

詳細は、次の Oracle リソースを参照してください。

Oracle Text の詳細は、次のマニュアルを参照してください。

- 『Oracle Text アプリケーション開発者ガイド』

Oracle9*i* の詳細は、次のマニュアルを参照してください。

- 『Oracle9*i* データベース概要』
- 『Oracle9*i* データベース管理者ガイド』
- 『Oracle9*i* データベース・ユーティリティ』
- 『Oracle9*i* データベース・パフォーマンス・チューニング・ガイドおよびリファレンス』
- 『Oracle9*i* SQL リファレンス』
- 『Oracle9*i* データベース・リファレンス』
- 『Oracle9*i* アプリケーション開発者ガイド - 基礎編』
- 『Oracle9*i* アプリケーション開発者ガイド - XML』

PL/SQL の詳細は、次のマニュアルを参照してください。

- 『PL/SQL ユーザーズ・ガイドおよびリファレンス』

リリース・ノート、インストレーション・マニュアル、ホワイト・ペーパーまたはその他の関連文書は、OTN-J（Oracle Technology Network Japan）に接続すれば、無償でダウンロードできます。OTN-J を使用するには、オンラインでの登録が必要です。次の URL で登録できます。

`http://otn.oracle.co.jp/membership/`

OTN-J のユーザー名とパスワードを取得済みであれば、次の OTN-J Web サイトの文書セクションに直接接続できます。

`http://otn.oracle.co.jp/document/`

表記規則

このマニュアル・セットの本文とコード例に使用されている表記規則について説明します。

- 本文の表記規則
- コード例の表記規則

本文の表記規則

本文中には、特別な用語が一目でわかるように様々な表記規則が使用されています。次の表は、本文の表記規則と使用例を示しています。

規則	意味	例
太字	太字は、本文中に定義されている用語または用語集に含まれている用語、あるいはその両方を示します。	**ub4**、**sword**、**OCINumber** などの C のデータ型は有効です。 この句を指定する場合は、**索引構成表**を作成します。
イタリック	イタリックは、問合せ語句、構文の句またはプレースホルダを示します。	次の問合せでは、*oracle* を検索します。 *parallel_clause* を指定できます。 U*old_release*`.SQL` を実行します。*old_release* は、アップグレード前にインストールしたリリースを示します。

規則	意味	例
固定幅フォントの大文字	固定幅フォントの大文字は、システムにより指定される要素を示します。この要素には、パラメータ、権限、データ型、Recovery Manager キーワード、SQL キーワード、SQL*Plus またはユーティリティ・コマンド、パッケージとメソッドの他、システム指定の列名、データベース・オブジェクトと構造体、ユーザー名、およびロールがあります。	この句は、NUMBER 列に対してのみ指定できます。 BACKUP コマンドを使用すると、データベースのバックアップを作成できます。 USER_TABLES データ・ディクショナリ・ビューの TABLE_NAME 列を問い合せます。 ROLLBACK_SEGMENTS パラメータを指定します。 DBMS_STATS.GENERATE_STATS プロシージャを使用します。
固定幅フォントの小文字	固定幅フォントの小文字は、実行可能ファイルとサンプルのユーザー指定要素を示します。この要素には、コンピュータ名とデータベース名、ネット・サービス名、接続識別子の他、ユーザー指定のデータベース・オブジェクトと構造体、列名、パッケージとクラス、ユーザー名とロール、プログラム・ユニット、およびパラメータ値があります。	sqlplus と入力して SQL*Plus をオープンします。 department_id、department_name および location_id の各列は、hr.departments 表にあります。 初期化パラメータ QUERY_REWRITE_ENABLED を true に設定します。 oe ユーザーで接続します。

コード例の表記規則

コード例は、SQL、PL/SQL、SQL*Plus またはその他のコマンドラインを示します。次のように、固定幅フォントで、通常の本文とは区別して記載されています。

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

次の表は、コード例の記載上の表記規則とその使用例を示しています。

規則	意味	例
[ ]	大カッコで囲まれている項目は、1 つ以上のオプション項目を示します。大カッコ自体は入力しないでください。	DECIMAL (digits [, precision])
{ }	中カッコで囲まれている項目は、そのうちの 1 つのみが必要であることを示します。中カッコ自体は入力しないでください。	{ENABLE | DISABLE}
|	縦線は、大カッコまたは中カッコ内の複数の選択肢を区切るために使用します。オプションのうち 1 つを入力します。縦線自体は入力しないでください。	{ENABLE | DISABLE} [COMPRESS | NOCOMPRESS]

規則	意味	例
. . .	水平の省略記号は、次のいずれかを示します。	
	■ 例に直接関係のないコード部分が省略されていること。	`CREATE TABLE ...AS subquery;`
	■ コードの一部が繰返し可能なこと。	`SELECT col1, col2, ...,coln FROM employees;`
. . .	垂直の省略記号は、例に直接関係のない数行のコードが省略されていることを示します。	
その他の表記	大カッコ、中カッコ、縦線および省略記号以外の記号は、表示されているとおりに入力してください。	`acctbal NUMBER(11,2);` `acct CONSTANT NUMBER(4) := 3;`
イタリック	イタリックの文字は、特定の値を指定する必要のある変数を示します。	`CONNECT SYSTEM/`*`system_password`*

大文字	大文字は、システムにより指定される要素を示します。これらの用語は、ユーザー定義の用語と区別するために大文字で記載されています。大カッコで囲まれている場合を除き、記載されているとおりの順序とスペルで入力してください。ただし、この種の用語は大 / 小文字区別がないため、小文字でも入力できます。	`SELECT last_name, employee_id FROM employees;`
		`SELECT * FROM USER_TABLES;`
		`DROP TABLE hr.employees;`

小文字	小文字は、ユーザー指定のプログラム要素を示します。たとえば、表名、列名またはファイル名を示します。	`SELECT last_name, employee_id FROM employees;` `sqlplus hr/hr`

Oracle Text の新機能

この章では、Oracle Text（以前の Oracle8*i* *inter*Media Text）の新機能を説明し、追加情報の参照先を示します。この章の内容は次のとおりです。

- Oracle Text リリース 9.2 の新機能
- Oracle Text リリース 9.0.1 の新機能

Oracle Text リリース 9.2 の新機能

このリリースでの新機能は次のとおりです。

- **ドキュメントの分類**

  新規 CTX_CLS.TRAIN プロシージャを使用して、ドキュメントを様々なカテゴリにルーティングするルールを生成できます。

  関連項目： 第 6 章「CTX_CLS パッケージ」の「TRAIN」

- **ユーザー定義レクサー**

  ユーザー定義のレクサーを使用して、アラビア語などの Oracle Text でサポートされていない言語の索引付けや問合せ用のレクサー処理ソリューションを作成できます。

  関連項目： 第 2 章「索引付け」の「USER_LEXER」

- **問合せテンプレートの作成**

  CONTAINS と CATSEARCH の文法が、それぞれ CONTEXT と CTXCAT に制限されなくなりました。問合せテンプレートの作成によって、CATSEARCH 問合せで CONTEXT 文法と関連する演算子を使用したり、CONTAINS 問合せで CTXCAT 文法と関連する演算子を使用できます。

  関連項目： 第 1 章「SQL 文と演算子」の「CATSEARCH」

- **CREATE INDEX ONLINE のサポート**

  CONTEXT 索引の作成時に、元表に対する挿入、更新および削除を実行できます。

  関連項目： 第 1 章「SQL 文と演算子」の「CREATE INDEX」

- **パラレル索引付けの拡張機能**

  非パーティション表に対するパラレル索引付けがサポートされるようになりました。`replace`、`resume` および `sync` の各パラメータを指定した `CREATE INDEX` と `ALTER INDEX` で、並列度を使用できます。`CTX_DDL.SYNC_INDEX` および `CTX_DDL.OPTIMIZE_INDEX` に並列度を指定して実行することもできます。

  関連項目：

  第 1 章「SQL 文と演算子」の「CREATE INDEX」

  第 7 章「CTX_DDL パッケージ」の「SYNC_INDEX」および「OPTIMIZE_INDEX」

- **語幹索引付け**

  語幹索引付けを使用して、基本形に加えて語幹形を索引付けすることで、ステミング（$）問合せのパフォーマンスを改善できます。

  関連項目： 第 2 章「索引付け」の「BASIC_LEXER」

- **中国語レクサー**

  新規 CHINESE_LEXER を使用して、中国語（繁体字および簡体字）テキストをさらに効率的に索引付けできます。

  関連項目： 第 2 章「索引付け」の「CHINESE_LEXER」

- **URIType 索引付け**

  URIType 列に CONTEXT 索引を作成できます。

  関連項目： 第 1 章「SQL 文と演算子」の「CREATE INDEX」

- **CTXXPATH**

  CTXXPATH 索引タイプを使用して、XMLType 列に対する existsNode() 問合せを高速化できます。

  関連項目： 第 1 章「SQL 文と演算子」の「CTXXPATH 索引タイプの構文」

  『Oracle9*i* アプリケーション開発者ガイド - XML』

- **existsNode() での ora:contains のサポート**

  existsNode() 問合せ内で、CONTAINS ファンクションをテキスト索引なしでコールできます。

  関連項目：

  『Oracle9*i* アプリケーション開発者ガイド - XML』

  第 7 章「CTX_DDL パッケージ」の「CREATE_POLICY」

Oracle Text リリース 9.0.1 の新機能

次の各項目で、このリリースでの新機能の概要を示します。

- **ドキュメントの分類**

  ドキュメント分類アプリケーションは、ドキュメントの内容に基づいてドキュメントの着信ストリームを分類するアプリケーションです。このアプリケーションは、ドキュメント・ルーティング・アプリケーションまたはドキュメント・フィルタ処理アプリケーションとも呼ばれています。たとえば、オンラインの報道機関では、記事の着信時にその着信ストリームを政治、犯罪、スポーツなどのカテゴリに分類する必要がある場合があります。

  Oracle Text では、新しい CTXRULE 索引タイプを使用して、このような分類アプリケーションを構築できます。この索引タイプによって、分類またはルーティング基準を定義するルール（問合せ）を索引付けします。ドキュメントの着信時に、新しい MATCHES 演算子によって、各ドキュメントを分類しルーティングできます。

  注意： Oracle Text では、プレーン・テキスト、XML および HTML の各ドキュメントに対してのみドキュメント分類をサポートしています。

  関連項目： 第 1 章「SQL 文と演算子」の「CREATE INDEX」および「MATCHES」

  ドキュメント分類の詳細は、『Oracle Text アプリケーション開発者ガイド』を参照してください。

- **ローカル・パーティション索引のサポート**

  パーティション化されたテキスト表に対してローカル・パーティション索引を作成できます。ローカル・パーティション索引を作成するには、CREATE INDEX に LOCAL PARTITION 句を指定します。パーティション索引は、ALTER INDEX を使用して再構築することもできます。

  関連項目： 第 1 章「SQL 文と演算子」の「CREATE INDEX」および「ALTER INDEX」

- **IGNORE フォーマット列の値**

  テキスト表のフォーマット列を使用すると、テキスト列にバイナリ・データとテキスト・データのどちらを格納するかを指定できます。

  IGNORE という新しいフォーマット列の値が用意されました。CREATE INDEX 文を発行してフォーマット列を指定する場合、フォーマット列が IGNORE に設定されている行は、索引付けの実行時に無視されます。この機能は、テキストの索引付けと互換性のないイメージや RAW バイナリ・データなどのデータが含まれたテキスト列の索引付けを行う場合に便利です。

  関連項目： 第 1 章「SQL 文と演算子」の「CREATE INDEX」

- **USER_DATASTORE の拡張機能**

  USER_DATASTORE にユーザー・プロシージャを指定するときに、IN/OUT パラメータに対して永続 BLOB ロケータおよび CLOB ロケータを戻すことができます。

  関連項目： 第 2 章「索引付け」の「USER_DATASTORE」

- **新しい韓国語レクサー**

  このリリースの Oracle Text では、新しい韓国語レクサー KOREAN_MORPH_LEXER によって、韓国語テキストの索引付けおよび問合せを引き続きサポートしています。KOREAN_MORPH_LEXER レクサーは、KOREAN_LEXER と比較して次の利点があります。

  - 韓国語テキストの語形分析機能の向上
  - 索引付けの高速化
  - 索引サイズの小型化
  - 問合せ検索精度の向上

  関連項目： 第 2 章「索引付け」の「KOREAN_MORPH_LEXER」

- **新しい日本語レクサー**

  このリリースの Oracle Text では、新しい日本語レクサー JAPANESE_LEXER によって、日本語テキストの索引付けおよび問合せを引き続きサポートしています。新しい日本語レクサーは、JAPANESE_VGRAM_LEXER と比較して次の利点があります。

  - 小型の索引の生成
  - 問合せ応答時間の短縮
  - 実ワード・トークンの生成による問合せ精度の向上

  関連項目： 第 2 章「索引付け」の「JAPANESE_LEXER」

- **XMLType の索引付け**

  Oracle Text では、XMLType 型のテキスト列の索引付けをサポートしています。

  注意： XMLType の索引付けがサポートされているのは、CONTEXT 索引タイプのみです。

  関連項目： XMLType の索引付けの詳細は、『Oracle Text アプリケーション開発者ガイド』を参照してください。

- **全言語のストップワード**

  複数の言語で停止対象となるワードを格納する MULTI_STOPLIST 型のストップリストを作成できます。新しいストップワードの型は、ALL です。たとえば、英語のフラグメントを含む国際文書の索引付けが必要な場合は、ALL ストップワードを使用します。

  関連項目： 第 7 章「CTX_DDL パッケージ」の「ADD_STOPWORD」

- **UTF-16 自動識別**

  Oracle Text では、charset および Inso の各フィルタによって、データベース・キャラクタ・セットの UTF-16 変換をサポートしています。これらのフィルタは、UTF-16 の big-endian（AL16UTF16）または little-endian（AL16UTF16LE）方式のドキュメントを変換できます。

  Oracle Text では、キャラクタ・セット列または charset フィルタが UTF16AUTO に設定されている場合は、endian 自動識別もサポートします。

  関連項目： 第 2 章「索引付け」の「CHARSET_FILTER」

- **INSO_FILTER タイムアウト属性**

  INSO_FILTER ドキュメント・フィルタには、新しいタイムアウト属性が追加されました。この属性を使用すると、索引付けの実行時にドキュメントのフィルタ処理を待機する最大時間を指定できます。このメカニズムを使用すると、索引操作中の停止を回避できます。

  関連項目： 第 2 章「索引付け」の「INSO_FILTER」

- **XML パス検索**

  XML ドキュメントには、次のような親子タグの構造を設定できます。

  ```
  <A> <B> <C> dog </C> </B </A>
  ```

  この例のタグ C は、タグ A の子であるタグ B の子です。

  Oracle Text では、新しい PATH_SECTION_GROUP によってパス検索ができます。このセクション・グループを使用すると、問合せ内に直接の親子関係を指定できます。たとえば、要素 B の子である要素 C にある用語 *dog* を含む全ドキュメントを検索できます。

  また、タグ属性値の検索および属性の等価性の検査も実行できます。

  この新機能に関連する新しい演算子は、次のとおりです。

  - INPATH
  - HASPATH

  関連項目： 第 3 章「CONTAINS 問合せ演算子」の「INPATH」および「HASPATH」

  XML ドキュメントでのパス・セクション検索の詳細は、『Oracle Text アプリケーション開発者ガイド』を参照してください。

- **CTX_DDL の更新されたプロシージャ**

  PL/SQL パッケージ CTX_DDL では、次のプロシージャが更新されました。

  - CTX_DDL.SYNC_INDEX

  このプロシージャには、メモリー・サイズとパーティション名を指定するための 2 つの新しいパラメータが追加されました。

  - CTX_DDL.SET_ATTRIBUTE

  このプロシージャでは、TRUE、T、FALSE、F、YES、Y、NO および N に加えて、ON/OFF ブール属性を指定できます。

  関連項目： 第 7 章「CTX_DDL パッケージ」

- **CTX_DOC の新しいプロシージャ**

  - CTX_DOC.IFILTER

  USER_DATASTORE プロシージャを使用して連結前にバイナリ・データをテキスト・データにフィルタ処理する必要がある場合は、このプロシージャを使用します。

  関連項目： 第 8 章「CTX_DOC パッケージ」

- **CTX_OUTPUT の新しいプロシージャ**

  CTX_OUTPUT パッケージには、次の新しいプロシージャが追加されました。

  - CTX_OUTPUT.ADD_EVENT
  - CTX_OUTPUT.REMOVE_EVENT

  ROWID 情報で索引ログ・ファイルを補強するには、最初のプロシージャを使用します。これは、索引操作のデバッグに便利です。

  関連項目： 第 9 章「CTX_OUTPUT パッケージ」

- **新規ビューおよび更新されたビュー**

  このリリースでは、次のビューが更新されました。

  - CTX_VERSION
  - CTX_PENDING
  - CTX_USER_PENDING

  CTX_VERSION ビューには新しい列 VER_CODE が追加されました。この列は、Oracle シャドウ・プロセスにリンクする Oracle Text コードのバージョン・ナンバーです。パッチ・リリースを検出および確認するには、この列を使用します。

次のビューは新しいビューです。最初の 4 つのビューは、マルチレクサー・プリファレンスを含むサブレクサーに関する情報の問合せに使用します。

- CTX_INDEX_SUB_LEXERS
- CTX_USER_INDEX_SUB_LEXERS
- CTX_INDEX_SUB_LEXER_VALUES
- CTX_USER_INDEX_SUB_LEXER_VALS
- CTX_INDEX_PARTITIONS
- CTX_USER_INDEX_PARTITIONS
- CTX_INDEX_SETS
- CTX_USER_INDEX_SETS
- CTX_INDEX_SET_INDEXES
- CTX_USER_INDEX_SET_INDEXES
- CTX_THES_PHRASES
- CTX_USER_THES_PHRASES

関連項目： 付録 G「ビュー」

1

SQL 文と演算子

この章では、テキスト索引の作成と管理およびテキスト問合せの実行に使用する SQL 文と Oracle Text 演算子について説明します。

この章で説明する文は、次のとおりです。

- ALTER INDEX
- ALTER TABLE: サポートされるパーティション化文
- CATSEARCH
- CONTAINS
- CREATE INDEX
- CATSEARCH
- MATCHES
- SCORE

ALTER INDEX

注意： この項では、テキスト・ドメイン索引の管理に関連する ALTER INDEX 文について説明します。

ALTER INDEX 文の詳細は、『Oracle9*i* SQL リファレンス』を参照してください。

用途

ALTER INDEX を使用して、CONTEXT、CTXCAT または CTXRULE 索引に対して次のメンテナンス作業を実行します。

すべての索引タイプ

- 索引または索引パーティションの改名。RENAME 構文を参照してください。
- 異なるプリファレンスを使用した索引の再構築。一部の制限事項が CTXCAT 索引タイプに適用されます。REBUILD 構文を参照してください。
- 索引へのストップワードの追加。REBUILD 構文を参照してください。

CONTEXT および CTXRULE 索引タイプ

- 失敗した索引処理（作成 / 最適化）の再開。REBUILD 構文を参照してください。
- バッチでの DML 処理（同期化）。REBUILD 構文を参照してください。
- 索引の完全またはトークン別の最適化。REBUILD 構文を参照してください。
- セクションおよび停止セクションの索引への追加。REBUILD 構文を参照してください。

RENAME 構文

次の構文は、索引または索引パーティションの改名に使用します。

```
ALTER INDEX [schema.]index_name RENAME TO new_index_name;

ALTER INDEX [schema.]index_name RENAME PARTITION part_name TO new_part_name;
```

[schema.]index_name
改名する索引名を指定します。

new_index_name
schema.index の新しい名前を指定します。new_index_name パラメータは、25 バイト以内で設定します。26 バイト以上の名前を指定するとエラーが戻り、改名した索引は無効になります。

注意： new_index_name が 26 バイト以上 30 バイト未満の場合は、システムがエラーを戻しても、索引は改名されます。索引およびそれに関連する表を削除するには、DROP INDEX 文で new_index_name を削除した後、index_name を再作成し、削除してください。

---

part_name
改名する索引パーティション名を指定します。

new_part_name
パーティションに新しい名前を指定します。

REBUILD 構文

次の構文を使用して、索引および索引パーティションの再構築、失敗した処理の再開、バッチ DML の実行、ストップワードの索引への追加、セクションおよび停止セクションの索引への追加、索引の最適化を実行します。

```
ALTER INDEX [schema.]index REBUILD [PARTITION partname] [ONLINE] [PARAMETERS
(paramstring)][PARALLEL N];
```

PARTITION *partname*
索引パーティション partname を再構築します。一度に構築できる索引パーティションは 1 つのみです。

パーティションの再構築時に paramstring に指定できるのは、sync、optimize full/fast、resume または replace のみです。この操作は、指定した partname に対してのみ実行されます。パーティションまたはパーティション索引の再構築時に、resume は指定できません。

パーティションの追加 元表にパーティションを追加するには、ALTER TABLE 文を使用します。索引表にパーティションを追加すると、新しい索引パーティションのメタデータが自動的に作成されます。新しい索引パーティションの名前と新しい表パーティションの名前は同じです。索引パーティション名は、ALTER INDEX RENAME で変更できます。新しい索引パーティションを移入する場合は、ALTER INDEX REBUILD で再構築する必要があります。

パーティションの分割とマージ ALTER TABLE によって表パーティションを分割またはマージすると、その索引パーティションは無効になります。したがって、ALTER INDEX REBUILD を使用してその索引パーティションを再構築する必要があります。

[ONLINE]
オプションで ONLINE パラメータを非ブロック操作に指定すると、ALTER INDEX の同期化操作時または最適化操作時に索引を問い合せることができます。

ONLINE を指定する場合は、PARALLEL を使用できません。

> **注意：** 索引を ONLINE で再構築する場合は、replace または resume を指定できますが、索引パーティションを ONLINE で再構築する場合は、指定できません。

PARALLEL n
オプションで、パラレル索引付けに対する並列度を n で指定できます。このパラメータがサポートされるのは、sync、replace および resume を paramstring に使用した場合のみです。実際の並列度は、リソースによっては多少低くなります。

パラレル索引付けによって、索引付けするデータが大量にある場合およびオペレーティング・システムが複数の CPU をサポートしている場合は、索引付けを高速化できます。

ONLINE を指定する場合は、PARALLEL を使用できません。

PARAMETERS（paramstring）
オプションで paramstring を指定します。paramstring を指定しない場合、既存のプリファレンス設定で索引が再構築されます。

paramstring の構文は、次のとおりです。

```
paramstring =
    'REPLACE
    [datastore datastore_pref]
    [filter filter_pref]
    [lexer lexer_pref]
    [wordlist wordlist_pref]
    [storage storage_pref]
    [stoplist stoplist]
    [section group section_group]
    [memory memsize]
    [index set index_set]

    |    resume [memory memsize]
    |    optimize [token index_token | fast | full [maxtime (time | unlimited)]
    |    sync [memory memsize]
    |    add stopword word [language language]
    |    add zone section section_name tag tag
    |    add field section section_name tag tag [(VISIBLE | INVISIBLE)]
    |    add attr section section_name tag tag@attr
    |    add stop section tag'
```

replace [*optional_preference_list*]
索引を再構築します。オプションで、ユーザー独自またはシステム定義のプリファレンスを指定できます。

置換できるプリファレンスは、その索引タイプのサポート対象となっているプリファレンスのみです。たとえば、CONTEXT 索引または CTXRULE 索引の場合、索引セットは置換できません。同様に、CTXCAT 索引タイプの場合は、レクサー、ワードリスト、記憶域索引セットおよびメモリー・プリファレンスのみ置換できます。

> **関連項目：** システム定義のプリファレンスの詳細、およびプリファレンスの作成と設定の詳細は、第 2 章「索引付け」を参照してください。

resume [memory *memsize*]
失敗した索引操作を再開します。オプションで、使用するメモリー量を memsize で指定できます。

> **注意：** この ALTER INDEX 操作が適用されるのは、CONTEXT 索引と CTXRULE 索引のみです。CTXCAT 索引には適用されません。

optimize [token index_token | fast | full [maxtime (time | unlimited)]

> **注意：** この ALTER INDEX 操作は、将来のリリースではサポートされません。
>
> 索引を最適化するには、CTX_DDL.OPTIMIZE_INDEX を使用してください。

索引を最適化します。token（トークン）、fast（高速）または full（完全）の最適化を指定します。通常は、索引を同期化した後で最適化を行います。

token モードで最適化を行うと、索引トークン index_token のみが token モードで最適化されます。この最適化方法を使用すると、特定ワードの索引情報が迅速に最適化されます。

fast モードで最適化を行うと、索引全体が最適化され、断片化した行が圧縮されます。ただし、fast モードでは、古いデータは削除されません。

full モードで最適化を行うと、索引全体または索引の一部を最適化できます。この方法では行が圧縮され、古いデータ（削除済みの行）が削除されます。

注意： full モードでの最適化は、削除済みドキュメントの行が存在しない場合でも実行されます。この方法は、maxtime パラメータで制限時間が設定されているバッチの最適化が必要な場合に便利です。

---

maxtime パラメータを使用して、最適化操作に費やす時間を分単位で指定します。最適化は、実行された場所から始まり、操作が完了するまで、または制限時間に達するまでのいずれか早い方まで行われます。制限時間を指定すると、索引最適化を自動化し、定期的に指定した時間、索引を最適化するように設定できます。

maxtime unlimited を指定すると、索引全体が最適化されます。これはデフォルトです。maxtime を 0（ゼロ）に指定すると、最小限の最適化が行われます。

注意： この ALTER INDEX 操作が適用されるのは、CONTEXT 索引と CTXRULE 索引のみです。CTXCAT 索引には適用されません。

---

sync [memory *memsize*]

注意： この ALTER INDEX 操作は、将来のリリースではサポートされません。

索引を同期化するには、CTX_DDL.SYNC_INDEX を使用してください。

索引を同期化します。オプションで、使用するランタイム・メモリー量を memsize で指定できます。元表で DML 操作を実行すると索引が同期化されます。

注意： この ALTER INDEX 操作が適用されるのは、CONTEXT 索引と CTXRULE 索引のみです。CTXCAT 索引には適用されません。

メモリーに関する考慮点 memory パラメータ memsize は、索引をディスクにフラッシュする前に、ALTER INDEX 操作に使用するメモリー量を指定します。大きいメモリー量を指定すると、I/O が削減されるため索引付けのパフォーマンスが向上します。また、断片化も削減されるため問合せのパフォーマンスおよびメンテナンスが向上します。

小さいメモリー量を指定すると、ディスク I/O および索引の断片化が増加しますが、索引付けの処理過程を追跡する場合、またはランタイム・メモリーが不足している場合に有効なことがあります。

add stopword *word* [language *language*]
索引にストップワードの word を動的に追加します。

使用しているストップリストがマルチ言語のストップリストの場合は、language を指定する必要があります。

この文によって索引が再構築されることはありません。

add zone section *section_name* tag *tag*
tag で識別されるゾーン・セクション section_name を、既存の索引に動的に追加します。

追加されたセクション section_name は、この操作の後に索引付けされるドキュメントのみに適用されます。変更を有効にするためには、このタグを含むすべての既存のドキュメントを、手動で再索引付けする必要があります。

この文によって索引が再構築されることはありません。

注意： この ALTER INDEX 操作が適用されるのは、CONTEXT 索引と CTXRULE 索引のみです。CTXCAT 索引には適用されません。

関連項目： 1-9 ページ「セクション追加時の制約」

add field section *section_name* tag *tag* [(VISIBLE | INVISIBLE)]
tag で識別されるフィールド・セクション section_name を、既存の索引に動的に追加します。

オプションで VISIBLE を指定すると、フィールド・セクションを参照できます。デフォルトは INVISIBLE です。

関連項目： 参照できるフィールド・セクションおよび表示されないフィールド・セクションの詳細は、「CTX_DDL.ADD_FIELD_SECTION」を参照してください。

追加されたセクション section_name は、この操作の後に索引付けされるドキュメントのみに適用されます。すでに索引付けされているドキュメントに対してこの変更を有効にするためには、このタグを含むすべてのドキュメントを、明示的に再索引付けする必要があります。

この文によって索引が再構築されることはありません。

注意： この ALTER INDEX 操作が適用されるのは、CONTEXT 索引と CTXRULE 索引のみです。CTXCAT 索引には適用されません。

関連項目： 1-9 ページ「セクション追加時の制約」

add attr section *section_name* tag *tag@attr*

属性セクション section_name を、既存の索引に動的に追加します。XML タグおよび属性を、tag@attr という形式で指定する必要があります。属性セクションは、XML セクション・グループにのみ追加できます。

追加されたセクション section_name は、この操作の後に索引付けされるドキュメントのみに適用されます。変更を有効にするためには、このタグを含むすべての既存のドキュメントを、手動で再索引付けする必要があります。

この文によって索引が再構築されることはありません。

注意： この ALTER INDEX 操作が適用されるのは、CONTEXT 索引のみです。CTXCAT 索引には適用されません。

関連項目： 「セクション追加時の制約」を参照してください。

add stop section *tag*

tag で識別される停止セクションを、既存の索引に動的に追加します。停止セクションは XML ドキュメントの自動セクション処理のみに適用されるため、索引には AUTO_SECTION_GROUP セクション・グループを使用する必要があります。指定するタグは、自動セクション・グループ内で一意で、大 / 小文字を区別する必要があります。これに従わない場合、ALTER INDEX はエラーを戻します。

追加された停止セクション tag は、この操作の後に索引付けされるドキュメントのみに適用されます。すでに索引付けされているドキュメントに対してこの変更を有効にするためには、このタグを含むすべてのドキュメントを、明示的に再索引付けする必要があります。

停止セクション内のテキストは常に検索可能です。

追加できる停止セクションの数は無制限です。

この文によって索引が再構築されることはありません。

注意： この ALTER INDEX 操作が適用されるのは、CONTEXT 索引のみです。CTXCAT 索引には適用されません。

セクション追加時の制約

索引セクション情報を変更する前に、すべての妥当性制約が満たされていることを確認するために、新しいセクションが既存のセクションに対してチェックされます。PL/SQL パッケージ CTX_DDL を使用してセクションをセクション・グループに追加するときにも同じ制約が適用されます。制約は次のとおりです。

- ゾーン、フィールドまたは停止セクションを NULL_SECTION_GROUP に追加することはできません。
- ゾーン、フィールドまたは属性セクションを自動セクション・グループに追加することはできません。
- 属性セクションを XML セクション・グループ以外に追加することはできません。
- 2 つの異なるセクションに同じタグを含めることはできません。
- ゾーン、フィールドおよび属性セクションのセクション名に、同じ名前を付けることはできません。
- フィールド・セクションの数は、64 を超えることはできません。
- 停止セクションを基本、HTML、XML または新しいセクション・グループに追加することはできません。
- SENTENCE および PARAGRAPH は、予約セクション名です。

例

失敗した索引付けの再開

次の文は、2MB のメモリーで newsindex の索引付け操作を再開します。

```
ALTER INDEX newsindex REBUILD PARAMETERS('resume memory 2M');
```

索引の再構築

次の文は、索引を再構築し、ストップリスト・プリファレンスを new_stop で置換します。

```
ALTER INDEX newsindex REBUILD PARAMETERS('replace stoplist new_stop');
```

パーティション索引の再構築

次の例では、パーティション化されたテキスト表を作成し、それを移入してパーティション索引を作成します。次に、その表に新しいパーティションを追加し、ALTER INDEX を使用して索引を再構築します。

```
PROMPT create partitioned table and populate it

create table part_tab (a int, b varchar2(40)) partition by range(a)
(partition p_tab1 values less than (10),
 partition p_tab2 values less than (20),
 partition p_tab3 values less than (30));

insert into part_tab values (1,'Actinidia deliciosa');
insert into part_tab values (8,'Distictis buccinatoria');
insert into part_tab values (12,'Actinidia quinata');
insert into part_tab values (18,'Distictis Rivers');
insert into part_tab values (21,'pandorea jasminoides Lady Di');
insert into part_tab values (28,'pandorea rosea');

commit;

PROMPT create partitioned index
create index part_idx on part_tab(b) indextype is ctxsys.context
local (partition p_idx1, partition p_idx2, partition p_idx3);


PROMPT add a partition and populate it
alter table part_tab add partition p_tab4 values less than (40);
insert into part_tab values (32, 'passiflora citrina');
insert into part_tab values (33, 'passiflora alatocaerulea');
commit;
```

次の文は、新しく移入されたパーティションに索引を再構築します。通常、新しく追加されたパーティションの索引パーティション名は、その名前がすでに使用済みでないかぎり、表パーティションの名前と同じになります。使用済みの場合は、新しい名前が生成されます。

```
alter index part_idx rebuild partition p_tab4;
```

次の文は、表を問い合せ、新しく追加されたパーティション内の 2 つのヒットを取り出します。

```
select * from part_tab where contains(b,'passiflora') >0;
```

次の文は、新しく追加されたパーティションを直接問い合せます。

```
select * from part_tab partition (p_tab4) where contains(b,'passiflora') >0;
```

索引の最適化

ALTER INDEX を使用した索引の最適化は将来のリリースではサポートされません。索引を最適化するには、CTX_DDL.OPTIMIZE_INDEX を使用してください。

索引の同期化

ALTER INDEX を使用した索引の同期化は将来のリリースではサポートされません。索引を同期化するには、CTX_DDL.SYNC_INDEX を使用してください。

ゾーン・セクションの追加

タグ <author> で識別されるゾーン・セクション author を索引に追加するには、次の文を発行します。

```
ALTER INDEX myindex REBUILD PARAMETERS('add zone section author tag author');
```

停止セクションの追加

タグ <fluff> で識別される停止セクションを、AUTO_SECTION_GROUP を使用する索引に追加するには、次の文を発行します。

```
ALTER INDEX myindex REBUILD PARAMETERS('add stop section fluff');
```

属性セクションの追加

次のテキストが XML ドキュメントで使用されているとします。

```
<book title="Tale of Two Cities">It was the best of times.</book>
```

タイトル属性ごとに別のセクションを作成し、その新しい属性セクションに booktitle という名前を付けるとします。そのためには、次の文を発行します。

```
ALTER INDEX myindex REBUILD PARAMETERS('add attr section booktitle tag title@book');
```

関連項目

第 7 章「CTX_DDL パッケージ」の「SYNC_INDEX」、「OPTIMIZE_INDEX」および「CREATE INDEX」

ALTER TABLE: サポートされるパーティション化文

注意： この項では、CONTEXT ドメイン索引を持つパーティション化されたテキスト表の追加と変更に関連する ALTER TABLE 文について説明します。

ALTER TABLE 文の詳細は、『Oracle9*i* SQL リファレンス』を参照してください。

用途

ALTER TABLE を使用して、CONTEXT ドメイン索引を持つパーティション化されたテキスト表を追加、変更、分割、マージ、交換または削除できます。次の項では、発行できる ALTER TABLE 操作をいくつか説明します。

MODIFY PARTITION 構文

UNUSABLE LOCAL INDEXES

```
ALTER TABLE  [schema.]table MODIFY PARTITION partition UNUSABLE LOCAL INDEXES
```

指定した表パーティションに対応する索引パーティションに UNUSABLE のマークを付けます。

REBUILD UNUSABLE LOCAL INDEXES

```
ALTER TABLE  [schema.]table MODIFY PARTITION partition REBUILD UNUSABLE LOCAL
INDEXES
```

指定した UNUSABLE ステータスの表パーティションに対応する索引パーティションを再構築します。

注意： 索引パーティションのステータスが、このコマンドの発行前にすでに VALID になっている場合、その索引パーティションは再構築されません。このコマンドは、ステータスが UNUSABLE でない索引パーティションの再構築には使用できません。

ADD PARTITION 構文

```
ALTER TABLE [schema.]table ADD PARTITION [partition]
VALUES LESS THAN (value_list) [partition_description]
```

新しいパーティションをレンジ・パーティション化表の最後に追加します。

表の先頭または中間にパーティションを追加する場合は、ALTER TABLE SPLIT PARTITION を使用します。

新しく追加された表パーティションは常に空で、このパーティションの CONTEXT ドメイン索引（ある場合）のステータスは、常に VALID になります。DML の実行後に、新しく追加された索引パーティションを同期化または最適化する場合は、索引パーティション名を検索して、ALTER INDEX REBUILD PARTITION コマンドを発行する必要があります。通常、新しく追加されたパーティションの索引パーティション名は、表パーティション名と同じになります。ただし、その表パーティション名がすでに別の索引パーティションの名前に使用されている場合は、SYS_Pn の形式で名前が割り当てられます。

USER_IND_PARTITIONS ビューを問い合せて HIGH_VALUE フィールドを照合することで、新しく追加されたパーティションの索引パーティション名を確認できます。

MERGE PARTITION 構文

```
ALTER TABLE [schema.]table
MERGE PARTITIONS partition1, partition2
[INTO PARTITION [new_partition] [partition_description]]
```

レンジ・パーティションのみに適用されます。このコマンドは、隣接する 2 つのパーティションの内容を新しいパーティションにマージした後、元の 2 つのパーティションを削除します。生成されたパーティションが空でない場合は、対応するローカル・ドメイン索引パーティションに UNUSABLE のマークが付けられます。ALTER TABLE MODIFY PARTITION を使用して、パーティション索引を再構築できます。

生成された索引パーティションのネーミング規則は、ALTER TABLE ADD PARTITION のネーミング規則と同じです。

SPLIT PARTITION 構文

```
ALTER TABLE [schema.]table
SPLIT PARTITION partition_name_old
AT (value_list)
[into (partition_description, partition_description)]
[prallel_clause]
```

レンジ・パーティションのみに適用されます。このコマンドは、1 つの表パーティションを 2 つのパーティションに分割することによって、新しいパーティションを表に追加します。対応する表パーティションが空でない場合は、対応するローカル索引パーティションに UNUSABLE のマークが付けられます。ALTER TABLE MODIFY PARTITION を使用して、パーティション索引を再構築できます。

生成された 2 つの索引パーティションのネーミング規則は、ALTER TABLE ADD PARTITION のネーミング規則と同じです。

EXCHANGE PARTITION 構文

```
ALTER TABLE [schema.]table EXCHANGE PARTITION partition WITH TABLE table
[INCLUDING|EXCLUDING INDEXES}
[WITH|WITHOUT VALIDATION]
[EXCEPTIONS INTO [schema.]table]
```

それぞれのデータ・セグメントを交換することによって、パーティションを非パーティション表に、および表をパーティション表のパーティションに変換します。ROWID は保持されます。

EXCLUDING INDEXES が指定されている場合は、パーティションに対応するすべての CONTEXT 索引と、交換された表のすべての索引に UNUSABLE のマークが付けられます。新しい索引パーティションの再構築には、ALTER TABLE MODIFY PARTITION を発行できます。

INCLUDING INDEXES が指定されている場合は、パーティション表の各ローカル・ドメイン索引に対して、非パーティション表の非パーティション・ドメイン索引が存在している必要があります。ローカル索引パーティションは、対応する通常の索引と交換されます。

フィールド・セクション

非パーティション索引とローカル索引で、同じフィールド・セクションに対して異なるセクション ID が使用されている場合、フィールド・セクション問合せは同じように機能しません。

記憶域
記憶域は変更されません。したがって、非パーティション表の $I 表の索引が表領域 XYZ に存在していた場合、その索引は EXCHANGE PARTITION の実行後も表領域 XYZ に存在しますが、今度は索引パーティションに対する $I 表になります。

記憶域プリファレンスの切替えはありません。このため、切り替えた後で索引を再構築しても、異なる場所に表が作成されます。

制限事項
両方の索引が等価であることが必要です。索引に同じオブジェクトが使用され、各オブジェクトに同一の設定が使用されている必要があります。索引に同一のオブジェクトが使用されていることのみがチェックされますが、その他すべてについても正確に同じである必要があります。

索引オブジェクトはパーティション化できません。つまり、ユーザーが $I 表または $N 表のパーティション化に、記憶域オブジェクトを使用した場合です。

索引または索引パーティションが、これらいずれかの制限事項に違反している場合はすべてエラーが発生し、その索引および索引パーティションは無効になります。ALTER INDEX ... REBUILD を 2 回使用して、索引と索引パーティションをそれぞれ手動で再構築する必要があります。

TRUNCATE PARTITION 構文

```
ALTER TABLE [schema.]table TRUNCATE PARTITION [DROP|REUSE STORAGE]
```

表内のパーティションからすべての行を削除します。対応する CONTEXT 索引パーティションも削除されます。

CATSEARCH

CTXCAT 索引を検索するには、CATSEARCH 演算子を使用します。この演算子は、SELECT 文の WHERE 句で使用します。

この演算子の文法は、CTXCAT と呼ばれます。シソーラス、ファジー・マッチング、近接検索またはステミングなど、検索基準に特殊な機能が必要な場合は、CONTEXT 文法を使用することもできます。CONTEXT 文法を利用する場合は、この項で説明する text_query パラメータに問合せテンプレートを指定します。

パフォーマンス

CATSEARCH 演算子を CTXCAT 索引で使用すると、主に複合問合せのパフォーマンスを向上させることができます。テキスト問合せ条件の指定には text_query を使用し、構造化条件の指定には structured_query を使用します。

内部的には、テキスト列および構造化列に対して結合 B ツリー索引が使用され、問合せを満たす結果が迅速に生成されます。

制限事項

この演算子では、機能の起動はサポートされません。

構文

```
CATSEARCH(
    [schema.]column,
    text_query       VARCHAR2,
    structured_query VARCHAR2,
RETURN NUMBER;
```

[schema.]column
検索するテキスト列を指定します。検索する列には、対応付けられた CTXCAT 索引が必要です。

text_query
次のいずれかを指定して、column の検索を定義します。

- CATSEARCH 問合せ演算子
- 問合せテンプレート（CONTEXT 文法を使用する場合）

CATSEARCH 問合せ演算子 CATSEARCH 演算子では、次の問合せ操作のみがサポートされます。

- 論理 AND
- 論理 OR（|）
- 論理 NOT（-）
- " "（引用符で囲んだ句）
- ワイルド・カード

これらの演算子の構文は、次のとおりです。

操作	構文	操作の説明
論理 AND	a b c	a、b および c を含む行を戻します。
論理 OR（|）	a | b | c	a、b または c を含む行を戻します。
論理 NOT（-）	a - b	a を含み、b を含まない行を戻します。
空白なしのハイフン	a-b	通常の文字として処理されるハイフンです。 たとえば、ハイフンが skipjoin として定義されている場合、*web-site* などのワードは、単一の問合せ語句 *website* として処理されます。 同様に、ハイフンが printjoin として定義されている場合、*web-site* などのワードは、CTXCAT 問合せ言語では *web site* として処理されます。
" "	"a b c"	句 "a b c" を含む行を戻します。 たとえば、"Sony CD Player" と入力すると、この一連のワードを含むすべての行を戻します。
( )	(A B) | C	グループ設定をカッコで囲みます。この問合せは、CONTAINS 問合せ (A &B) | C に相当します。

操作	構文	操作の説明
ワイルド・カード （右側切捨ておよび左右切捨て）	term* a*b	ワイルド・カード文字は、0（ゼロ）個以上の文字と一致します。 たとえば、*do** は、*dog* に一致し、*gl*s* は、*glass* に一致します。 左側切捨ては、サポートされていません。 **注意：**アプリケーションでワイルド・カード検索を使用する場合は、プリフィックス索引を作成することをお薦めします。プリフィックス索引付けは、BASIC_WORDLIST プリファレンスを使用して設定します。

問合せテンプレート CONTEXT 文法に基づいた問合せを指定するマークアップ文字列を指定します。次のタグと属性値を使用します（大 / 小文字が区別されます）。

タグ	説明	有効な値
<query> </query>	この問合せを問合せテンプレートとして解釈することを示します。	
<textquery> </textquery>	問合せ文字列を指定します。	
grammar=	問合せの文法を指定します。	CONTEXT CTXCAT
<score></score>	スコア・プリファレンスを指定します。	
datatype=	スコアとして戻す数値の型を指定します。	INTEGER FLOAT

structured_query
構造化条件と ORDER BY 句を指定します。指定するすべての列に索引が存在している必要があります。たとえば、'category_id=1 order by bid_close' と指定するとします。CTX_DDL.ADD_INDEX で指定した場合と同様に、'category_id, bid_close' に索引が存在している必要があります。

structured_query では、標準 SQL 構文と併用できるのは次の演算子のみです。

- =
- <=
- >=
- >
- <
- IN
- BETWEEN

> **注意：** structured_query パラメータにカッコ '()' は使用できません。

例

表の作成 次の文では、索引付けする表を作成します。

```
CREATE TABLE auction (category_id number primary key, title varchar2(20),
bid_close date);
```

次の表では、値を表に挿入します。

```
INSERT INTO auction values(1, 'Sony CD Player', '20-FEB-2000');
INSERT INTO auction values(2, 'Sony CD Player', '24-FEB-2000');
INSERT INTO auction values(3, 'Pioneer DVD Player', '25-FEB-2000');
INSERT INTO auction values(4, 'Sony CD Player', '25-FEB-2000');
INSERT INTO auction values(5, 'Bose Speaker', '22-FEB-2000');
INSERT INTO auction values(6, 'Tascam CD Burner', '25-FEB-2000');
INSERT INTO auction values(7, 'Nikon digital camera', '22-FEB-2000');
INSERT INTO auction values(8, 'Canon digital camera', '26-FEB-2000');
```

CTXCAT 索引の作成 次の文では、CTXCAT 索引を作成します。

```
begin
    ctx_ddl.create_index_set('auction_iset');
    ctx_ddl.add_index('auction_iset','bid_close');
end;

CREATE INDEX auction_titlex ON auction(title) INDEXTYPE IS CTXCAT PARAMETERS ('index
set auction_iset');
```

表の問合せ CATSEARCH による一般的な問合せには、次のように、bid_close によって順序付けられたワード *camera* を含むすべての行を検索する構造化句が組み込まれています。

```
SELECT * FROM auction WHERE CATSEARCH(title, 'camera', 'order by bid_close desc')>
0;

CATEGORY_ID TITLE                BID_CLOSE
----------- -------------------- ---------
          8 Canon digital camera 26-FEB-00
          7 Nikon digital camera 22-FEB-00
```

次の問合せでは、語句 *Sony CD Player* を含み、入札終了日が 2000 年 2 月 20 日のすべての行を検索します。

```
SELECT * FROM auction WHERE CATSEARCH(title, '"Sony CD Player"', 'bid_
close=''20-FEB-00''')> 0;

CATEGORY_ID TITLE                BID_CLOSE
----------- -------------------- ---------
          1 Sony CD Player       20-FEB-00
```

次の問合せでは、語句 *Sony*、*CD* および *Player* を含むすべての行を検索します。

```
SELECT * FROM auction WHERE CATSEARCH(title, 'Sony CD Player', 'order by bid_close
desc')> 0;
CATEGORY_ID TITLE                BID_CLOSE
----------- -------------------- ---------
          4 Sony CD Player       25-FEB-00
          2 Sony CD Player       24-FEB-00
          1 Sony CD Player       20-FEB-00
```

次の問合せでは、語句 *CD* は含まれているが *Player* は含まれていないすべての行を検索します。

```
SELECT * FROM auction WHERE CATSEARCH(title, 'CD - Player', 'order by bid_close
desc')> 0;

CATEGORY_ID TITLE                BID_CLOSE
----------- -------------------- ---------
          6 Tascam CD Burner     25-FEB-00
```

次の問合せでは、語句 *CD* または *DVD* または *Speaker* を含むすべての行を検索します。

```
SELECT * FROM auction WHERE CATSEARCH(title, 'CD | DVD | Speaker', 'order by bid_
close desc')> 0;

CATEGORY_ID TITLE                BID_CLOSE
----------- -------------------- ---------
          3 Pioneer DVD Player   25-FEB-00
          4 Sony CD Player       25-FEB-00
          6 Tascam CD Burner     25-FEB-00
          2 Sony CD Player       24-FEB-00
          5 Bose Speaker         22-FEB-00
          1 Sony CD Player       20-FEB-00
```

次の問合せは、*audio equipment* に関連するすべての行を検索します。

```
SELECT * FROM auction WHERE CATSEARCH(title, 'ABOUT(audio equipment)', NULL)> 0;
```

CONTEXT 問合せ文法の例

```
PROMPT
PROMPT fuzzy: query = ?test
PROMPT should match all fuzzy variations of test (e.g. text)
select pk||' ==> '||text from test
where catsearch(text,
'<query>
  <textquery grammar="context">
     ?test
  </textquery>
  <score datatype="integer"/>
</query>','')>0
order by pk;

PROMPT
PROMPT fuzzy: query = !sail
PROMPT should match all soundex variations of bot (e.g. sell)
select pk||' ==> '||text from test
where catsearch(text,
```

```
'<query>
  <textquery grammar="context">
     !sail
  </textquery>
  <score datatype="integer"/>
</query>','')>0
order by pk;

PROMPT
PROMPT theme (ABOUT) query
PROMPT query: about(California)
select pk||' ==> '||text from test
where catsearch(text,
'<query>
  <textquery grammar="context">
     about(California)
  </textquery>
  <score datatype="integer"/>
</query>','')>0
order by pk;
```

関連項目

この章の「CTXCAT 索引タイプの構文」

『Oracle Text アプリケーション開発者ガイド』

CONTAINS

SELECT 文の WHERE 句で CONTAINS 演算子を使用して、テキスト問合せの問合せ式を指定します。

CONTAINS は、選択されたすべての行に対して関連性スコアを戻します。関連性スコアは、SCORE 演算子で取得します。

この演算子の文法は、CONTEXT と呼ばれます。単純な構文のほうが効率的に動作するアプリケーションの場合は、CTXCAT 文法を使用することもできます。CTXCAT 文法を利用する場合は、この項で説明する text_query パラメータに問合せテンプレートを指定します。

構文

```
CONTAINS(
         [schema.]column,
         text_query    VARCHAR2
         [,label       NUMBER])
RETURN NUMBER;
```

[schema.]column

検索するテキスト列を指定します。検索する列は、対応付けられたテキスト索引を持っている必要があります。

text_query

次のいずれかを指定します。

- **列**の検索を定義する問合せ式
- CTXCAT 文法に基づいた問合せを指定するマークアップ文字列 この問合せ文字列には、次のタグを使用します。

タグ	説明	有効な値
<query> </query>	この問合せを問合せテンプレートとして解釈することを示します。	
<textquery> </textquery>	問合せ文字列を指定します。	
grammar=	問合せの文法を指定します。	CONTEXT CTXCAT
<score></score>	スコア・プリファレンスを指定します。	
datatype=	スコアとして戻す数値の型を指定します。	INTEGER FLOAT

タグと属性値はすべて、大 / 小文字が区別されます。

関連項目： 問合せ式で使用できる演算子の詳細は、第 3 章「CONTAINS 問合せ演算子」を参照してください。

label

オプションで、CONTAINS 演算子で生成されたスコアを識別するラベルを指定します。

戻り値

CONTAINS は、選択された各行に対して 0 ～ 100 の数を戻し、そのドキュメントの行が問合せに対してどれだけ適合しているかを示します。0（ゼロ）は、その行に一致する検索結果がないことを示します。

注意： この数を取得するには、label を付けて SCORE 演算子を使用してください。

例

次の例では、*oracle* というワードを含むテキスト列の、すべてのドキュメントを検索します。各行のスコアは、ラベル 1 を使用して SCORE 演算子で選択されます。

```
SELECT SCORE(1), title from newsindex
    WHERE CONTAINS(text, 'oracle', 1) > 0;
```

CONTAINS 演算子の後には、必ず > 0 という構文を付けてください。これは、CONTAINS 演算子で計算されたスコア値が、選択された行では 0（ゼロ）より大きくなる必要があることを指定します。

SCORE 演算子が（SELECT 句などで）コールされる場合、次のように、CONTAINS 句が SCORE ラベル値を参照するように指定してください。

```
SELECT SCORE(1), title from newsindex
     WHERE CONTAINS(text, 'oracle', 1) > 0 ORDER BY SCORE(1) DESC;
```

次の例では、CATSEARCH 文法を使用して解析する問合せを指定します。

```
SELECT id FROM test WHERE CONTAINS (text,
'<query>
<textquery lang="ENGLISH" grammar="CATSEARCH">
cheap pokemon
</textquery>
<score datatype="INTEGER"/>
</query>') 0;
```

注意

マルチ言語表の問合せ

マルチレクサー・プリファレンスを使用すると、マルチ言語表から索引を作成できます。

問合せ時に、マルチレクサー・プリファレンスはセッションの言語設定を調べ、その言語に対してサブレクサー・プリファレンスを使用して問合せを解析します。言語設定がマップされていない場合は、デフォルト・レクサーが使用されます。

言語設定がマップされている場合は、問合せが解析され、通常に実行されます。索引には複数言語からのトークンが含まれているため、このような問合せはドキュメントを複数の言語で戻すことができます。

指定した言語のドキュメントのみを戻すように問合せを制限する場合は、言語列に構造化句を使用します。

パーティション索引による問合せパフォーマンスの制限

Oracle Text では、パーティション化されたテキスト表の CONTEXT 索引付けと問合せをサポートしています。

ただし、ORDER BY SCORE 句を使用してパーティション表を問い合せる場合は、問合せパフォーマンスを最適化するために、パーティションを問い合せます。表全体を問い合せるために ORDER BY SCORE 句を使用する場合は、問合せを 1 つのパーティションに制限できる範囲述語を組み込まないかぎり、最適な問合せは実行されません。

たとえば、次の文はパーティション p_tab4 を直接問い合せます。

```
select * from part_tab partition (p_tab4) where contains(b,'oracle') >0 ORDER BY
SCORE;
```

CTXCAT 問合せ文法の例

次の例は、CONTAINS 問合せでの CTXCAT 文法の使用方法を示しています。この例では、CTXCAT 索引と CONTEXT 索引を同じ表に作成し、問合せ結果を比較しています。

```
PROMPT
PROMPT create context and ctxcat indexes both with theme indexing on
PROMPT
create index tdrbqcq101x on test(text) indextype is ctxsys.context
parameters ('lexer theme_lexer');

create index tdrbqcq101cx on test(text) indextype is ctxsys.ctxcat
parameters ('lexer theme_lexer');

PROMPT
PROMPT *****  San Diego             ***********
PROMPT *****  CONTEXT grammar       ***********
PROMPT ** should be interpreted as phrase query **
select pk||' ==> '||text from test
where contains(text,'San Diego')>0
order by pk;

PROMPT
PROMPT *****  San Diego      ***********
PROMPT *****  CTXCAT grammar ***********
PROMPT ** should be interpreted as AND query  ***
select pk||' ==> '||text from test
where contains(text,
'<query>
  <textquery grammar="CTXCAT">San Diego</textquery>
  <score datatype="integer"/>
</query>')>0
order by pk;
```

```
PROMPT
PROMPT *****  Hitlist from CTXCAT index ***********
select pk||' ==> '||text from test
where catsearch(text,'San Diego','')>0
order by pk;
```

関連項目

この章の「CONTEXT 索引タイプの構文」および「SCORE」

第 3 章「CONTAINS 問合せ演算子」

『Oracle Text アプリケーション開発者ガイド』

CREATE INDEX

注意： この項では、テキスト・ドメイン索引の作成に関連する CREATE INDEX 文について説明します。

CREATE INDEX 文の詳細は、『Oracle9*i* SQL リファレンス』を参照してください。

用途

CREATE INDEX を使用して、Oracle Text の索引を作成します。Oracle Text の索引は、CONTEXT、CTXCAT、CTXRULE または CTXXPATH 型の Oracle ドメイン索引です。

CONTAINS、CATSEARCH または MATCHES 問合せを発行するには、適切な Oracle Text の索引を作成する必要があります。

次の型の Oracle Text の索引を作成できます。

- CONTEXT 索引。1 つのテキスト列に対する索引です。この索引は、SELECT 文の WHERE 句で CONTAINS 演算子を使用して問い合せます。この索引は、DML 操作後に手動で同期化する必要があります。CONTEXT 索引タイプの構文を参照してください。
- CTXCAT 索引。1 つのテキスト列と 1 つ以上の他の列に対する結合された索引です。この索引は、SELECT 文の WHERE 句で CATSEARCH 演算子を使用して問い合せます。この型の索引は、複合問合せのために最適化されます。この索引はトランザクション型です。元表に対する DML 操作によって、索引自体が自動的に更新されます。CTXCAT 索引タイプの構文を参照してください。
- CTXRULE 索引。問合せの集合を含む列に対する索引です。この索引は、SELECT 文の WHERE 句で MATCHES 演算子を使用して問い合せます。CTXRULE 索引タイプの構文を参照してください。
- CTXXPATH 索引。XMLType 列に対する existsNode() 問合せの高速化が必要な場合は、この索引を作成します。CTXXPATH 索引タイプの構文を参照してください。

必要な権限

Oracle Text の索引の作成には、CTXAPP ロールは不要です。テキスト列に B ツリー索引を作成する Oracle 権限を持っていれば、そのユーザーにはテキスト索引を作成する権限があります。発行所有者、表所有者および索引所有者がすべて異なるユーザーである場合もあります。これは、通常の B ツリー索引の作成時の Oracle 標準と一貫性があります。

一時表領域の要件

Oracle Text 索引の作成には、CTXSYS ユーザーに属する一時表領域が必要です。表領域が不足していると、ORA-01652 エラーになります。これを解決するには、発行ユーザーの表領域ではなく、CTXSYS 表領域を拡張します。通常、必要な一時表領域のサイズは、使用するデータの 50 ～ 200 パーセントです。

CONTEXT 索引タイプの構文

この索引タイプを使用して、テキスト列に索引を作成します。この索引は、SELECT 文の WHERE 句で CONTAINS 演算子を使用して問い合せます。この索引は、DML 操作後に手動で同期化する必要があります。

```
CREATE INDEX [schema.]index on [schema.]table(column) INDEXTYPE IS
    ctxsys.context [ONLINE]
    LOCAL [(PARTITION [partition] [PARAMETERS('paramstring')]
        [, PARTITION [partition] [PARAMETERS('paramstring')])])]
    [PARAMETERS(paramstring)] [PARALLEL n] [UNUSABLE];
```

[*schema*.]*index*
作成するテキスト索引の名前を指定します。

[*schema*.]*table*(*column*)
索引付けする表および列の名前を指定します。

CTX_DOC のプロシージャを使用して行を識別する場合は、必要に応じてその表に主キーを格納できます。表に主キーがない場合、ドキュメント・サービスでは、ROWID によってドキュメントを識別します。

指定する列は、CHAR、VARCHAR、VARCHAR2、BLOB、CLOB、BFILE、XMLType または URIType のうちいずれかの型である必要があります。

DATE 列、NUMBER 列および NESTED TABLE 列は、索引付けできません。また、Object 列にも索引付けできませんが、属性が基本データ型であれば、Object 列の属性に索引付けできます。

CONTEXT 索引タイプでは、複数列に対する索引はサポートされていません。列リスト内の 1 つの列のみを指定する必要があります。

注意： CTXCAT 索引タイプを使用すると、テキスト列と構造化列の索引を作成できます。この章の CTXCAT 索引タイプの構文を参照してください。

ONLINE
索引の作成時に、元表に対する挿入 / 更新 / 削除（DML）を実行できます。

索引付け中、DML 要求は保留キューにエンキューされます。索引の作成が完了すると、元表はロックされます。その間、DML はブロックされています。

制限事項 ONLINE を使用する場合は、次の制限事項が適用されます。

- そのプロセスの最初または最後に、DML が失敗する可能性があります。
- ONLINE を使用して、ローカル・パーティション索引をオンラインで作成することはできません。
- ONLINE がサポートされているのは、CONTEXT 索引のみです。
- ONLINE には PARALLEL を使用できません。

LOCAL [(PARTITION [*partition*] [PARAMETERS('*paramstring*')]
パーティション表にローカル・パーティション CONTEXT 索引を作成する場合は、LOCAL を指定します。パーティション表は、レンジ・パーティション化する必要があります。ハッシュ・パーティション、コンポジット・パーティションおよびリスト・パーティションは、サポートされていません。

索引パーティション名のリストは、*partition* で指定できます。パーティション名を指定しない場合は、システムによって割り当てられます。索引パーティション・リストの順序は、表パーティションの順序に対応している必要があります。

各パーティションに関連付けられた PARAMETERS 句によって、そのパーティション固有のパラメータ文字列が指定されます。各索引パーティションの *memory* と *storage* のみを指定することもできます。

索引パーティション名や索引パーティションの状態などの索引パーティション情報を検索するには、ビュー CTX_INDEX_PARTITIONS または CTX_USER_INDEX_PARTITIONS を問い合せます。

この操作に ONLINE パラメータは使用できません。

> **関連項目：** 「ローカル・パーティション索引の作成」

パーティション索引による問合せパフォーマンスの制限 ORDER BY SCORE 句を使用してパーティション索引を問い合せる場合は、問合せパフォーマンスを最適化するために、パーティションを問い合せます。表全体を問い合せるために ORDER BY SCORE 句を使用する場合は、問合せを 1 つのパーティションに制限できる範囲述語を組み込まないかぎり、最適な問合せは実行されません。

> **関連項目：** 「CONTAINS」の「パーティション索引による問合せパフォーマンスの制限」を参照してください。

PARALLEL n
オプションで、パラレル索引付けに対する並列度を n で指定できます。実際の並列度は、リソースによっては多少低くなります。

このパラメータは、非パーティション表に使用できます。パラレルで作成された非パーティション索引の場合、パラレル問合せ処理は実行されません。

このパラメータをパーティション表に使用すると、シリアル索引付けになります。パーティション表をパラレルで索引付けするには、UNUSABLE の索引を作成してから、DBMS_PCLXUTIL.BUILD_PART_INDEX ユーティリティを実行する必要があります。

関連項目：

「パラレル索引付け」

「ローカル・パーティション索引のパラレルでの作成」

『Oracle Text アプリケーション開発者ガイド』の第 5 章

パフォーマンス パラレル索引付けによって、索引付けするデータが大量にある場合およびオペレーティング・システムが複数の CPU をサポートしている場合は、索引付けを高速化できます。

注意： ローカル索引に対して PARALLEL を使用すると、索引がシリアルで作成された場合でも、パラレル問合せが実行されます （パラレルで作成された非パーティション索引の場合、パラレル問合せ処理は実行されません）。

パラレル問合せは、特に負荷の大きいシステムに対して実行すると、問合せのスループットが低下します。このため、ローカル索引の作成後に、パラレル問合せを使用禁止にすることをお薦めします。使用禁止にするには、ALTER INDEX NOPARALLEL を使用します。

パラレル問合せの詳細は、『Oracle Text アプリケーション開発者ガイド』のパフォーマンスのチューニングに関する章を参照してください。

制限事項 PARALLEL を使用する場合は、次の制限事項が適用されます。

- パラレル索引付けがサポートされているのは、CONTEXT 索引のみです。
- ローカル索引をパラレルで作成するように指定すると、パラレル句は無視され、索引はシリアルで作成されます。この場合は、パラレル問合せが使用可能になります。

  関連項目： ローカル・パーティション索引に実際の並列度を指定して作成する方法は、この項にある「ローカル・パーティション索引のパラレルでの作成」を参照してください。

- PARALLEL には ONLINE を使用できません。

UNUSABLE

UNUSABLE の索引を作成します。このパラメータによって、索引メタデータのみが作成され、即時に終了します。

UNUSABLE の索引は、ローカル・パーティション索引をパラレルで作成する必要がある場合に作成します。

関連項目： 「ローカル・パーティション索引のパラレルでの作成」

PARAMETERS(*paramstring*)

オプションで、paramstring に索引付けパラメータを指定します。user.preference 表記法を使用して、他のユーザーが所有するプリファレンスを指定できます。

paramstring の構文は、次のとおりです。

```
paramstring =
    '[datastore datastore_pref]
    [filter filter_pref]
    [charset column charset_column_name]
    [format column format_column_name]

    [lexer lexer_pref]
    [language column language_column_name]

    [wordlist wordlist_pref]
    [storage storage_pref]
    [stoplist stoplist]
    [section group section_group]
    [memory memsize]
    [populate | nopopulate]'
```

CTX_DDL.CREATE_PREFERENCE で、データストア、フィルタ、レクサー、ワードリストおよび記憶域プリファレンスを作成します。作成後、それらを paramstring に指定します。

注意： paramstring を指定しない場合は、システムのデフォルト値が使用されます。

これらのデフォルトの詳細は、第 2 章の「デフォルトの索引付けパラメータ」を参照してください。

---

datastore *datastore_pref*

データストア・プリファレンスの名前を指定します。データストア・プリファレンスを使用して、テキストの格納場所を指定します。第 2 章「索引付け」の「データストア型」を参照してください。

filter *filter_pref*
フィルタ・プリファレンスの名前を指定します。フィルタ・プリファレンスを使用して、書式化されたドキュメントのフィルタ処理方法をプレーン・テキストまたは HTML に指定します。第 2 章「索引付け」のフィルタ型を参照してください。

charset column *charset_column_name*
キャラクタ・セット列の名前を指定します。キャラクタ・セット列は、テキスト列と同じ表内にある必要があり、CHAR、VARCHAR または VARCHAR2 型である必要があります。この列を使用して、データベースのキャラクタ・セットに変換するドキュメントのキャラクタ・セットを指定します。この値には大 / 小文字区別はありません。JA16EUC などの NLS キャラクタ・セット文字列を指定する必要があります。

ドキュメントがプレーン・テキストまたは HTML である場合、INSO_FILTER および CHARSET フィルタはこの列を使用して、索引付けのためにドキュメントのキャラクタ・セットをデータベースのキャラクタ・セットに変換します。

プレーン・テキストまたは HTML ドキュメントが、異なる複数のキャラクタ・セットを持つ、またはデータベースのキャラクタ・セットとは異なるキャラクタ・セットである場合は、この列を使用してください。

> **注意：** キャラクタ・セット列のみが変更された場合、ドキュメントは再索引付け用にマークされません。索引付けされた列が再索引付けにフラグされるには、その列を更新する必要があります。

format column *format_column_name*
フォーマット列の名前を指定します。フォーマット列は、テキスト列と同じ表内にある必要があり、CHAR、VARCHAR または VARCHAR2 型である必要があります。

INSO_FILTER は、ドキュメントのフィルタ時にフォーマット列を使用します。プレーン・テキストまたは HTML ドキュメントに対する Inso フィルタ処理をオプションでバイパスするために、この列を異機種間ドキュメント・セットとともに使用してください。

フォーマット列には、次のいずれかを指定できます。

- TEXT
- BINARY
- IGNORE

TEXT は、ドキュメントがプレーン・テキストまたは HTML のいずれかであることを示します。TEXT を指定すると、ドキュメントはフィルタされませんが、キャラクタ・セットが変換される場合があります。

BINARY は、ドキュメントが PDF のような、プレーン・テキストまたは HTML 以外の INSO_FILTER オブジェクトがサポートする形式であることを示します。フォーマット列のエントリをマップできない場合は、BINARY がデフォルトです。

IGNORE は、索引付けの実行時に行が無視されることを示します。テキストの索引付けと互換性のないイメージ・データなどのデータが含まれた行をバイパスする必要がある場合は、この値を使用します。

注意： フォーマット列のみが変更された場合、ドキュメントは再索引付け用にマークされません。索引付けされた列が再索引付けにフラグされるには、その列を更新する必要があります。

---

lexer *lexer_pref*
レクサー・プリファレンスまたはマルチレクサー・プリファレンスの名前を指定します。レクサー・プリファレンスを使用して、テキストの言語およびテキストを索引付け用にトークン化する方法を識別します。第 2 章「索引付け」の「レクサー型」を参照してください。

language column *language_column_name*
マルチレクサー・プリファレンスを使用する場合は、言語列の名前を指定します。第 2 章「索引付け」の MULTI_LEXER を参照してください。

この列は、元表内にある必要があります。この列は、索引付けされた列とは別の列である必要があります。言語列の最初の 30 バイトのみが、言語を識別するために調べられます。

注意： 言語列のみが変更された場合、ドキュメントは再索引付け用にマークされません。索引付けされた列が再索引付けにフラグされるには、その列を更新する必要があります。

---

wordlist *wordlist_pref*
ワードリスト・プリファレンスの名前を指定します。ワードリスト・プリファレンスを使用して、ファジー、ステミングおよびプリフィックス索引付けなどの機能を有効化すると、ワイルド・カード検索のパフォーマンスが向上します。第 2 章「索引付け」のワードリスト型を参照してください。

storage *storage_pref*
テキスト索引用の記憶域プリファレンスの名前を指定します。記憶域プリファレンスを使用して、索引表の格納方法を指定します。第 2 章「索引付け」の「記憶域型」を参照してください。

stoplist *stoplist*
ストップリストの名前を指定します。ストップリストを使用して、索引付けの対象でないワードを識別します。第 7 章「CTX_DDL パッケージ」の「CREATE_STOPLIST」を参照してください。

section group *section_group*
セクション・グループの名前を指定します。セクション・グループを使用して、構造化ドキュメントに検索可能なセクションを作成します。第 7 章「CTX_DDL パッケージ」の「CREATE_SECTION_GROUP」を参照してください。

memory *memsize*
索引付けに使用するランタイム・メモリー量を指定します。memsize の構文は、次のとおりです。

```
memsize = number[M|G|K]
```

M は MB（メガバイト）、G は GB（ギガバイト）、K は KB（キロバイト）を表します。

memsize の値は、1MB から CTX_PARAMETERS ビューの MAX_INDEX_MEMORY の値までの間で指定する必要があります。MAX_INDEX_MEMORY の値以上のメモリー・サイズを指定する場合は、CTX_ADM.SET_PARAMETER によってこのパラメータを memsize 以上に再設定する必要があります。

デフォルトは、CTX_PARAMETERS の DEFAULT_INDEX_MEMORY に指定された値です。

memsize パラメータは、索引をディスクにフラッシュする前に、索引付けに使用するメモリー量を指定します。大きいメモリー量を指定すると、I/O が削減されるため索引付けのパフォーマンスが向上します。また、断片化も削減されるため問合せのパフォーマンスおよびメンテナンスが向上します。

小さいメモリー量を指定すると、ディスク I/O および索引の断片化が増加しますが、ランタイム・メモリーが不足している場合に有効なことがあります。

populate | nopopulate
空の索引を作成するには、nopopulate を指定します。デフォルトは populate です。

> **注意：** このオプションのみ、デフォルト値を CTX_ADM.SET_PARAMETER で設定できません。

空の索引は、元表を更新または元表に挿入することで移入されます。索引を増分的に作成したり、元表内のドキュメントを選択的に索引付けするときに、空の索引の作成が必要な場合があります。空の索引は、ドキュメント・セットからテーマおよび要点の出力のみを必要とする場合にも作成します。

CONTEXT 索引の例

次の項では、CONTEXT 索引の作成例を示します。

デフォルトのプリファレンスを使用した CONTEXT 索引の作成

次の例では、mytable 内の docs 列に、myindex という CONTEXT 索引を作成します。デフォルトのプリファレンスが使用されます。

```
CREATE INDEX myindex ON mytable(docs) INDEXTYPE IS ctxsys.context;
```

> **関連項目：** デフォルト設定の詳細は、第 2 章の「デフォルトの索引付けパラメータ」を参照してください。
>
> また、『Oracle Text アプリケーション開発者ガイド』も参照してください。

カスタム・プリファレンスを使用した CONTEXT 索引の作成

次の例では、mytable 内の docs 列に、myindex という CONTEXT 索引を作成します。索引は、my_lexer というカスタム・レクサー・プリファレンスおよび my_stop というカスタム・ストップリストで作成されます。

また、この例では、これらのプリファレンスは、my_lexer が CTX_DDL.CREATE_PREFERENCE で、my_stop が CTX_DDL.CREATE_STOPLIST で作成されていることを前提としています。未指定のプリファレンスに対しては、デフォルトのプリファレンスが使用されます。

```
CREATE INDEX myindex ON mytable(docs) INDEXTYPE IS ctxsys.context
  PARAMETERS('LEXER my_lexer STOPLIST my_stop');
```

すべてのユーザーが、あらゆるプリファレンスを使用できます。他のユーザーのスキーマにあるプリファレンスを指定するには、そのユーザー名をプリファレンス名に追加します。次の例では、プリファレンス my_lexer および my_stop が、ユーザー kenny のスキーマ内にあることを前提としています。

```
CREATE INDEX myindex ON mytable(docs) INDEXTYPE IS ctxsys.context
  PARAMETERS('LEXER kenny.my_lexer STOPLIST kenny.my_stop');
```

マルチレクサー・プリファレンスを使用した CONTEXT 索引の作成

マルチレクサーは、各行にどのレクサーを使用するかを、言語列に基づいて決定します。これは、テキスト列内のドキュメントの言語識別名を格納するキャラクタ列です。たとえば、異なる複数の言語のドキュメントを持つ globaldoc 表を、次のように作成するとします。

```
CREATE TABLE globaldoc (
   doc_id NUMBER PRIMARY KEY,
   lang VARCHAR2(10),
   text CLOB
);
```

作成したマルチレクサー・プリファレンスが、global_lexer であるとします。global_doc 表を索引付けするには、マルチレクサー・プリファレンスおよび言語列の名前を次のように指定します。

```
CREATE INDEX globalx ON globaldoc(text) INDEXTYPE IS ctxsys.context PARAMETERS
('LEXER global_lexer LANGUAGE COLUMN lang');
```

> **関連項目：** マルチレクサー・プリファレンスの作成の詳細は、第 2 章の「MULTI_LEXER」を参照してください。

パラレル索引付け

次の例では、パラレル索引付けの設定方法を示します。次の手順に従ってください。

並列度を使用して索引を作成します。この例では、並列度 3 を使用します。

```
CREATE INDEX myindex ON mytab(pk) INDEXTYPE IS ctxsys.context PARALLEL 3;
```

ローカル・パーティション索引の作成

次の例では、3 つにパーティション化されたテキスト表を作成し、それを移入した後、パーティション索引を作成します。

```
PROMPT create partitioned table and populate it

CREATE TABLE part_tab (a int, b varchar2(40)) PARTITION BY RANGE(a)
    (partition p_tab1 values less than (10),
     partition p_tab2 values less than (20),
     partition p_tab3 values less than (30));

PROMPT create partitioned index
CREATE INDEX part_idx on part_tab(b) INDEXTYPE IS CTXSYS.CONTEXT
    LOCAL (partition p_idx1, partition p_idx2, partition p_idx3);
```

ローカル・パーティション索引のパラレルでの作成

ローカル索引をパラレルで作成するには、UNUSABLE の索引を作成してから、DBMS_PCLXUTIL.BUILD_PART_INDEX ユーティリティを実行します。

この例の元表には 3 つのパーティションがあります。最初にローカル・パーティションの UNUSABLE の索引を作成してから、DBMS_PCLUTIL.BUILD_PART_INDEX を実行すると、3 つのパーティションがパラレルに作成されます（パーティション間並列性）。また、各パーティション内の索引も、並列度 2 でパラレルに作成されます（パーティション内並列性）。

```
create index tdrbip02bx on tdrbip02b(text)
indextype is ctxsys.context local (partition tdrbip02bx1,
                                   partition tdrbip02bx2,
                                   partition tdrbip02bx3)
unusable;

exec dbms_pclxutil.build_part_index(3,2,'TDRBIP02B','TDRBIP02BX',TRUE);
```

索引エラーの表示

CREATE INDEX 操作または ALTER INDEX 操作が完了した後、Oracle Text のビューに索引エラーを表示できます。索引エラーを表示するには、CTX_USER_INDEX_ERRORS ビューを問い合せます。全索引のエラーを CTXSYS として表示するには、CTX_INDEX_ERRORS ビューを問い合せます。

たとえば、索引に最後に発生したエラーを表示するには、次の問合せを発行します。

```
SELECT err_timestamp, err_text FROM ctx_user_index_errors ORDER BY err_timestamp
DESC;
```

索引エラーの削除

索引エラーのビューを消去するには、次の問合せを発行します。

```
DELETE FROM ctx_user_index_errors;
```

CTXCAT 索引タイプの構文

CTXCAT 索引は、1 つのテキスト列と 1 つ以上の他の列に対する結合された索引です。この索引は、SELECT 文の WHERE 句で CATSEARCH 演算子を使用して問い合せます。この型の索引は、複合問合せのために最適化されます。この索引はトランザクション型です。元表に対する DML 操作によって、索引自体が自動的に更新されます。

```
CREATE INDEX [schema.]index on [schema.]table(column) INDEXTYPE IS ctxsys.ctxcat
    [PARAMETERS
        ('[index set index_set]
        [lexer lexer_pref]
        [storage storage_pref]
        [stoplist stoplist]
        [wordlist wordlist_pref]
        [memory memsize]');
```

[*schema.*]*table*(*column*)
索引付けする表および列の名前を指定します。

CTXCAT 索引の作成時に指定する列は、CHAR 型または VARCHAR2 型である必要があります。CTXCAT では、それ以外の型はサポートされていません。

サポートされているプリファレンス

index set *index_set*
CTXCAT 索引を作成するには、索引セット・プリファレンスを指定します。「CTXCAT 索引の作成」を参照してください。

索引パフォーマンスとサイズに関する考慮点 CTXCAT 索引には問合せパフォーマンスの向上という利点がありますが、索引の作成にコストがかかります。CTXCAT 索引の作成に要する時間は、その合計サイズによって異なります。CTXCAT 索引の合計サイズに直接関係する要因は、次のとおりです。

- 索引付けするテキストの合計数
- 索引セットに含まれるコンポーネント索引の数
- コンポーネント索引を構成する元表の列数

索引セットに多数のコンポーネント索引がある場合、更新が必要な索引が増えるため、DML のパフォーマンスは低下します。

CTXCAT 索引の作成には追加コストが必要になるため、索引セットにコンポーネント索引を追加する前に、それぞれのコンポーネント索引によりアプリケーションに提供される問合せパフォーマンス上の利点を慎重に考慮する必要があります。

> **関連項目：** CTXCAT 索引の作成とその利点に関する詳細は、『Oracle Text アプリケーション開発者ガイド』を参照してください。

その他のプリファレンス
CTXCAT 型の索引を作成する場合、parameters 文字列で使用できるのは次の索引プリファレンスのみです。

表 1-1

プリファレンス・クラス	サポートされている型
データストア	ありません。
フィルタ	ありません。
レクサー	BASIC_LEXER（index_themes 属性はサポートされていません。）
	CHINESE_VGRAM_LEXER
	JAPANESE_VGRAM_LEXER
	KOREAN_LEXER
	KOREAN_MORPH_LEXER
ワードリスト	BASIC_WORDLIST
記憶域	BASIC_STORAGE

表 1-1（続き）

プリファレンス・クラス	サポートされている型
ストップリスト	1 つの言語のストップリストのみがサポートされます（BASIC_STOPLIST 型）。
セクション・グループ	ありません。

サポートされないプリファレンスとパラメータ

CTXCAT 索引を作成する場合、データストア、フィルタおよびセクション・グループの各プリファレンスは指定できません。CONTEXT 索引の場合と同様、言語列、フォーマット列およびキャラクタ・セット列は指定できません。

CTXCAT 索引の作成

この項では、CTXCAT 索引の作成の簡単な例を示します。詳細な例は、『Oracle Text アプリケーション開発者ガイド』を参照してください。

次のスキーマを使用して AUCTION という表について考えます。

```
create table auction(
    item_id number,
    title varchar2(100),
    category_id number,
    price number,
    bid_close date);
```

表の問合せに、category_id に対する必須のテキスト問合せ句とオプションの構造化条件が含まれているとします。結果は、bid_close と category_id に基づいてソートする必要があります。

ユーザーが入力する可能性がある様々なタイプの構造化問合せをサポートするために、カタログ索引を作成できます。構造化問合せの場合、CONTEXT 索引に比べ、CTXCAT 索引のほうが問合せパフォーマンスが向上します。

索引を作成する場合は、最初に索引セット・プリファレンスを作成した後、そのプリファレンスに必要な索引を追加します。

```
begin
    ctx_ddl.create_index_set('auction_iset');
    ctx_ddl.add_index('auction_iset','bid_close');
    ctx_ddl.add_index('auction_iset','price, bid_close');
end;
```

CTXCAT 索引は、CREATE INDEX を使用して次のように作成します。

```
create index auction_titlex on AUCTION(title) indextype is CTXSYS.CTXCAT parameters
('index set auction_iset');
```

CTXCAT 索引の問合せ

ワード *pokemon* の title 列を問い合せるには、次のように、通常の複合問合せを発行できます。

```
select * from AUCTION where CATSEARCH(title, 'pokemon',NULL)> 0;
select * from AUCTION where CATSEARCH(title, 'pokemon', 'price < 50 order by bid_
close desc')> 0;
```

> **関連項目：** CTXCAT の例の詳細は、『Oracle Text アプリケーション開発者ガイド』を参照してください。

CTXRULE 索引タイプの構文

問合せの集合を含む列に対する索引です。この索引は、SELECT 文の WHERE 句で MATCHES 演算子を使用して問い合せます。

```
CREATE INDEX [schema.]index on [schema.]table(column) INDEXTYPE IS
    ctxsys.ctxrule
    [PARAMETERS ('[lexer lexer_pref] [storage storage_pref]
        [section group section_pref] [wordlist wordlist_pref]');
    [PARALLEL n];
```

[*schema.*]*table*(*column*)

索引付けする表および列の名前を指定します。

CTXRULE 索引の作成時に指定する列は、VARCHAR2 型または CLOB 型である必要があります。CTXRULE では、それ以外の型はサポートされていません。

lexer_pref

MATCHES 関数を使用して、問合せおよび分類するドキュメントの処理に使用するレクサー・プリファレンスを指定します。現在は、BASIC_LEXER レクサー型のみがサポートされています。

問合せ処理の場合、このレクサーがサポートしている演算子は、ABOUT、STEM、AND、NEAR、NOT、OR および WITHIN です。

シソーラスを使用した演算子（BT*、NT*、PT、RT、SYN、TR、TRSYS、TT など）がサポートされています。ただし、これらの演算子は、MATCHES 関数の発行時ではなく、索引付け時にシソーラスのスナップショットを使用して拡張されます。したがって、索引付けの後にシソーラスを変更する場合は、問合せセットの再索引付けが必要になります。

ACCUM、EQIUV、MINUS、WEIGHT、THRESHOLD、WILDCARD、FUZZY および SOUNDEX の演算子はサポートされていません。

storage_pref
問合せに対して索引の記憶域プリファレンスを指定します。この記憶域プリファレンスを使用して索引表の格納方法を指定します。第 2 章「索引付け」の「記憶域型」を参照してください。

section group
セクション・グループを指定します。このパラメータは問合せには影響を与えません。分類するドキュメントのセクションに適用されます。CTXRULE 索引タイプに対して、次のセクション・グループがサポートされています。

- BASIC_SECTION_GROUP
- HTML_SECTION_GROUP
- XML_SECTION_GROUP
- AUTO_SECTION_GROUP

第 2 章「索引付け」のセクション・グループ型を参照してください。

CTXRULE では、特殊セクションはサポートされません。

wordlist_pref
ワードリスト・プリファレンスを指定します。このプリファレンスを使用すると、問合せ語句でステミング操作を有効化できます。第 2 章「索引付け」の「ワードリスト型」を参照してください。

CTXRULE 索引の作成例

ドキュメント・ルーティング・アプリケーションでの CTXRULE 索引タイプの使用方法の詳細は、『Oracle Text アプリケーション開発者ガイド』を参照してください。

CTXXPATH 索引タイプの構文

XMLType 列に対する existsNode() 問合せの高速化が必要な場合は、この索引を作成します。

```
CREATE INDEX [schema.]index on [schema.]table(XMLType column) INDEXTYPE IS
ctxsys.CTXXPATH
[PARAMETERS ('[storage storage_pref]
              [memory memsize]
              [populate | nopopulate]')];
```

[*schema.*]*table*(*column*)
索引付けする表および列の名前を指定します。

CTXXPATH 索引の作成時に指定する列は、XMLType 型である必要があります。CTXXPATH では、それ以外の型はサポートされていません。

storage_pref
問合せに対して索引の記憶域プリファレンスを指定します。この記憶域プリファレンスを使用して索引表の格納方法を指定します。第 2 章「索引付け」の「記憶域型」を参照してください。

memory *memsize*
索引付けに使用するランタイム・メモリー量を指定します。memsize の構文は、次のとおりです。

```
memsize = number[M|G|K]
```

M は MB（メガバイト）、G は GB（ギガバイト）、K は KB（キロバイト）を表します。

memsize の値は、1MB から CTX_PARAMETERS ビューの MAX_INDEX_MEMORY の値までの間で指定する必要があります。MAX_INDEX_MEMORY の値以上のメモリー・サイズを指定する場合は、CTX_ADM.SET_PARAMETER によってこのパラメータを memsize 以上に再設定する必要があります。

デフォルトは、CTX_PARAMETERS の DEFAULT_INDEX_MEMORY に指定された値です。

populate | nopopulate
空の索引を作成するには、nopopulate を指定します。デフォルトは populate です。

> **注意：** このオプションのみ、デフォルト値を CTX_ADM.SET_PARAMETER で設定できません。

空の索引は、元表を更新または元表に挿入することで移入されます。索引を増分的に作成したり、元表内のドキュメントを選択的に索引付けするときに、空の索引の作成が必要な場合があります。

CTXXPATH の例

XMLType 列に対する索引の作成

```
CREATE INDEX xml_index ON xml_tab(col_xml) indextype is ctxsys.CTXXPATH;
```

または

```
CREATE INDEX xml_index ON xml_tab(col_xml) indextype is ctxsys.CTXXPATH
 PARAMETERS('storage my_storage memory 40M');
```

existsNode を使用した表の問合せ

```
select xml_id from xml_tab x where x.col_
xml.existsnode('/book/chapter[@title="XML"]') > 0;
```

関連項目： CTXXPATH 索引タイプの使用方法の詳細は、『Oracle9*i* アプリケーション開発者ガイド - XML』を参照してください。

関連項目

第 7 章「CTX_DDL パッケージ」の「CREATE_PREFERENCE」、
「CREATE_STOPLIST」および
「CREATE_SECTION_GROUP」
「ALTER INDEX」
「CATSEARCH」

DROP INDEX

注意： この項では、テキスト・ドメイン索引の削除に関連する DROP INDEX 文について説明します。

DROP INDEX 文の詳細は、『Oracle9*i* SQL リファレンス』を参照してください。

用途

DROP INDEX を使用して、指定したテキスト索引を削除します。

構文

```
DROP INDEX [schema.]index [force];
```

[force]
オプションで、索引を強制的に削除します。

例

次の例では、カレント・ユーザーのデータベース・スキーマ内にある doc_index という名前の索引を削除します。

```
DROP INDEX doc_index;
```

注意

force オプションは、索引付け操作がクラッシュしたときなど、索引の状態を判定できない場合に使用してください。

関連項目

「ALTER INDEX」

「CREATE INDEX」

MATCHES

この演算子を使用して、指定したドキュメントに一致している問合せ表のすべての行を検索します。ドキュメントは、プレーン・テキスト、HTML または XML のドキュメントである必要があります。

この演算子では、問合せセットに CTXRULE 索引が必要です。

MATCHES は、1 つ以上の一致の場合は 1 を、一致なしの場合は 0（ゼロ）を戻します。

制限事項

MATCHES では、機能の起動はサポートされません。

構文

```
MATCHES(
    [schema.]column,
    document VARCHAR2 or CLOB)
RETURN NUMBER;
```

column
索引付けられた問合せセットを含む列を指定します。

document
分類するドキュメントを指定します。ドキュメントは、プレーン・テキスト、HTML または XML ドキュメントである必要があります。バイナリ・フォーマットはサポートされていません。

例

表 querytable に CTXRULE 索引が関連付けられている場合、次の問合せを発行して分類するドキュメント文字列を渡すことができます。SELECT 文は、着信ドキュメントが適合する行（問合せ）をすべて戻します。

```
SELECT classification FROM querytable WHERE MATCHES(text, 'Smith is a common name in
the United States') > 0;
```

関連項目

この章の「CTXRULE 索引タイプの構文」

『Oracle Text アプリケーション開発者ガイド』

SCORE

SELECT 文で SCORE 演算子を使用して、CONTAINS 問合せによって生成されたスコア値を戻します。

構文

```
SCORE(label NUMBER)
```

label
問合せで生成されたスコアを識別するための番号を指定します。この番号を使用して、CONTAINS 句内のスコアを識別します。

注意

SCORE 演算子は、SELECT、ORDER BY または GROUP BY 句で使用できます。

例

1 つの CONTAINS

SCORE 演算子が（SELECT 句などで）コールされる場合、次のように、CONTAINS 句が SCORE ラベル値を参照するように指定してください。

```
SELECT SCORE(1), title from newsindex
          WHERE CONTAINS(text, 'oracle', 1) > 0 ORDER BY SCORE(1) DESC;
```

複数の CONTAINS

news データベースでは、新しい記事のタイトルおよび本文が別々に格納および索引付けされるとします。次の問合せは、タイトルに *Oracle* というワードおよび本文に *java* というワードを含むすべてのドキュメントを戻します。戻された記事は、最初の CONTAINS（*Oracle*）に対するスコア順にソートされた後、次の CONTAINS（*java*）のスコア順にソートされます。

```
SELECT title, body, SCORE(10), SCORE(20)
    FROM news
    WHERE CONTAINS (news.title, 'Oracle', 10) > 0 OR
        CONTAINS (news.body, 'java', 20) > 0
        ORDER BY NVL(SCORE(10),0), NVL(SCORE(20),0);
```

関連項目

「CONTAINS」

付録 F「スコア付けのアルゴリズム」

2

索引付け

この章では、Oracle Text の索引の作成に使用できる様々な要素について説明します。

この章の内容は次のとおりです。

- 概要
- データストア型
- フィルタ型
- レクサー型
- ワードリスト型
- 記憶域型
- セクション・グループ型
- 分類型
- ストップリスト
- システム定義プリファレンス
- システム・パラメータ

概要

索引の作成に CREATE INDEX を使用する場合、または索引の管理に ALTER INDEX を使用する場合は、オプションでパラメータ文字列に索引付けプリファレンス、ストップリストおよびセクション・グループを指定できます。プリファレンス、ストップリストまたはセクション・グループを指定することによって、Oracle がテキストを索引付けする方法を 1 つずつ設定できます。

プリファレンス・クラス	説明
データストア	ドキュメントの保存方法
フィルタ	ドキュメントのプレーン・テキストへの変換方法
レクサー	索引付けされる言語
ワードリスト	ステミング問合せおよびファジー問合せの拡張方法
記憶域	索引表の格納方法
ストップリスト	索引付けしないワードまたはテーマ
セクション・グループ	セクション内の問合せの使用可能化、およびドキュメント・セクションの定義方法

この章では、各プリファレンスの設定方法について説明します。オプションを使用可能にするには、この章で説明する型の 1 つを使用してプリファレンスを作成します。

たとえば、ドキュメントを外部ファイルに格納するように指定するには、FILE_DATASTORE 型を使用して mydatastore というデータストア・プリファレンスを作成できます。mydatastore をデータストア・プリファレンスとして CREATE INDEX の PARAMETERS 句に指定します。

プリファレンスの作成

データストア、レクサー、フィルタ、ワードリストまたは記憶域プリファレンスを作成するには、CTX_DDL.CREATE_PREFERENCE プロシージャを使用し、この章で説明する型の 1 つを指定します。いくつかの型には、CTX_DDL.SET_ATTRIBUTE プロシージャで属性も設定できます。

ストップリストを作成するには、CTX_DDL.CREATE_STOPLIST を使用します。CTX_DDL.ADD_STOPWORD を使用すると、ストップリストにストップワードを追加できます。

セクション・グループを作成するには、CTX_DDL.CREATE_SECTION_GROUP を使用し、セクション・グループのタイプを指定します。CTX_DDL.ADD_ZONE_SECTION または CTX_DDL.ADD_FIELD_SECTION を使用すると、セクション・グループにセクションを追加できます。

データストア型

データストア型を使用して、テキストの格納方法を指定します。データストア・プリファレンスを作成するには、次のデータストア型の 1 つを使用する必要があります。

データストア型	使用する場合
DIRECT_DATASTORE	データをテキスト列に内部的に格納する場合。各列が単一のドキュメントとして索引付けされます。
MULTI_COLUMN_DATASTORE	データをテキスト表の複数の列に格納する場合。列が連結され、各行に 1 つずつ仮想ドキュメントが作成されます。
DETAIL_DATASTORE	データをテキスト列に内部的に格納する場合。ドキュメントがディテール表のテキスト列にある 1 つ以上の行で構成され、ヘッダー情報はマスター表に格納されます。
FILE_DATASTORE	データをオペレーティング・システム・ファイルに外部的に格納する場合。ファイル名が、テキスト列の各行に 1 つずつ格納されます。
NESTED_DATASTORE	データを、NESTED TABLE に格納する場合。
URL_DATASTORE	データをインターネットまたはイントラネット上にあるファイルに外部的に格納する場合。Uniform Resource Locator（URL）がテキスト列に格納されます。
USER_DATASTORE	ドキュメントが、索引付け時にユーザー定義ストアド・プロシージャによって合成されます。

DIRECT_DATASTORE

テキスト列の各行にドキュメントが 1 つずつ直接格納されるテキストには、DIRECT_DATASTORE 型を使用します。DIRECT_DATASTORE には、属性はありません。

CHAR、VARCHAR、VARCHAR2、BLOB、CLOB、BFILE または XMLType の列型がサポートされています。

注意： 列が BFILE の場合、BFILE によって使用されるすべてのディレクトリに対して CTXSYS の読取り権限を付与する必要があります。

DIRECT_DATASTORE CLOB の例

次の例では、テキスト・データを格納する CLOB 列がある表を作成します。表の作成後、2 つの行にテキスト・データを移入し、システム定義プリファレンス CTXSYS.DEFAULT_DATASTORE を使用して表を索引付けします。

```
create table mytable(id number primary key, docs clob);

insert into mytable values(111555,'this text will be indexed');
insert into mytable values(111556,'this is a direct_datastore example');
commit;

create index myindex on mytable(docs)
  indextype is ctxsys.context
  parameters ('DATASTORE CTXSYS.DEFAULT_DATASTORE');
```

MULTI_COLUMN_DATASTORE

テキストが複数の列に格納されている場合は、このデータストアを使用します。索引付け時、システムはテキスト列を連結し、テキストを単一ドキュメントとして索引付けします。

MULTI_COLUMN_DATASTORE には、次の属性があります。

属性	属性値
columns	索引付け時に連結する列のカンマで区切られたリストを指定します。元表に関する SELECT 文の列リストに使用可能な式も指定できます。この中には、各種の式、PL/SQL ファンクション、列別名などが含まれます。 NUMBER 列型および DATE 列型はサポートされています。これらの型は、索引付け前にデフォルトの書式マスクを使用してテキストに変換されます。書式設定には、TO_CHAR ファンクションを列リストで使用できます。 RAW 列および BLOB 列は、バイナリ・データとして直接連結されます。 LONG、LONG RAW、NCHAR、NCLOB、NESTED TABLE の各列およびコレクションはサポートされていません。 列リストは 500 バイトに制限されています。

索引付けと DML

索引付けには、CREATE INDEX 文で指定するダミー列を作成する必要があります。ダミー列の内容は、その列名が列属性で指定されていないかぎり、仮想ドキュメントの一部にはなりません。

索引は、ダミー列の更新時にのみ同期化されます。必要に応じて、変更内容を伝播するためにトリガーを作成できます。

MULTI_COLUMN_DATASTORE のセキュリティ

MULTI_COLUMN_DATASTORE 型のプリファレンスを作成できるのは、CTXSYS のみです。それ以外のユーザーが MULTI_COLUMN_DATASTORE プリファレンスを作成しようとすると、エラーが戻されます。

これは、列属性にファンクション・コールが含まれている場合、そのコールの実行を CTXSYS スキーマに制限するためです。また、不適切な CTXAPP ユーザーが、実行権限のないファンクションを許可なく実行する可能性があるためです。

この制限が厳しすぎる場合、CTXSYS でストアド・プロシージャを作成し、MULTI_COLUMN_DATASTORE プリファレンスを作成できます。このプリファレンスを作成し所有する有効なユーザーは、CTXSYS です。ただし、このプロシージャは、任意のスキーマから CTXSYS でコールできます。

MULTI_COLUMN_DATASTORE の例

次の例では、3 つのテキスト列を持つ my_multi という複数列のデータストア・プリファレンスを作成します。

```
begin
    ctx_ddl.create_preference('my_multi', 'MULTI_COLUMN_DATASTORE');
    ctx_ddl.set_attribute('my_multi', 'columns', 'column1, column2, column3');
end;
```

タグ付けの動作

索引付け時、列ごとに仮想ドキュメントが作成されます。仮想ドキュメントは、リスト順に連結され自動的に列名タグが追加された列の内容で構成されています。次に例を示します。

```
create table mc(id number primary key, name varchar2(10), address varchar2(80));
insert into mc values(1, 'John Smith', '123 Main Street');

exec ctx_ddl.create_preference('mymds', 'MULTI_COLUMN_DATASTORE');
exec ctx_ddl.set_attibute('mymds', 'columns', 'name, address');
```

これによって、次の仮想テキストが索引付けのために作成されます。

```
<NAME>
John Smith
</NAME>
<ADDRESS>
123 Main Street
</ADDRESS>
```

システムでは、タグ自体は無視して、タグで囲まれたテキストを索引付けします。

セクションとしての列の索引付け

これらのタグをセクションとして索引付けするには、オプションで、BASIC_SECTION_GROUP を使用してフィールド・セクションを作成できます。

> **注意：** MULTI_COLUMN_DATASTORE を使用した場合、セクション・グループは作成されません。これらのタグに対してセクションを作成するには、セクション・グループを作成する必要があります。

式またはファンクションを使用する場合、列別名が使用されていないかぎり、タグは使用する式の最初の 30 文字で構成されます。

たとえば、次の式を使用するとします。

```
exec ctx_ddl.set_attibute('mymds', 'columns', '4 + 17');
```

次の仮想テキストが生成されます。

```
<4 + 17>
21
</4 + 17>
```

次の式を使用するとします。

```
exec ctx_ddl.set_attibute('mymds', 'columns', '4 + 17 col1');
```

次の仮想テキストが生成されます。

```
<col1>
21
<col1>
```

列名または列別名が小文字で二重引用符で囲まれていないかぎり、タグは大文字になります。次に例を示します。

```
exec ctx_ddl.set_attibute('mymds', 'COLUMNS', 'foo');
```

次の仮想テキストが生成されます。

```
<FOO>
content of foo
</FOO>
```

小文字のタグを生成するには、次のようにします。

```
exec ctx_ddl.set_attibute('mymds', 'COLUMNS', 'foo "foo"');
```

この式によって、次のタグが生成されます。

```
<foo>
content of foo
</foo>
```

DETAIL_DATASTORE

データベースのディテール表に直接格納されているテキストには、DETAIL_DATASTORE 型を使用します。この場合、マスター表に索引付けされたテキスト列があります。

DETAIL_DATASTORE には、次の属性があります。

属性	属性値
binary	TRUE を指定すると、各ディテール表の行の後ろに改行文字は追加されません。 FALSE を指定すると、各ディテール表の行の後ろに改行文字（\n）が自動的に追加されます。
detail_table	ディテール表の名前（必要な場合は OWNER.TABLE）を指定します。
detail_key	ディテール表の外部キー列の名前を指定します。
detail_lineno	ディテール表の順序列の名前を指定します。
detail_text	ディテール表のテキスト列の名前を指定します。

マスター表 / ディテール表の索引の同期化

ディテール表が変更された場合、索引の同期化時に再索引付け操作はトリガーされません。マスター表の索引付けされた列が変更された場合のみ、索引の同期化時に再索引付け操作がトリガーされます。

ディテール表にトリガーを作成すると、マスター表の行の索引付けされた列に変更内容を伝播できます。

マスター表 / ディテール表の例

この例では、マスター表およびディテール表が互いにどのように関連しているかを示します。

マスター表の例 . マスター表は、マスター / ディテールの関係にあるドキュメントを定義します。各ドキュメントに識別番号を割り当てます。次の表は、my_master というマスター表の例です。

列名	列型	説明
article_id	NUMBER	各ドキュメントに一意のドキュメント ID（主キー）
author	VARCHAR2(30)	ドキュメントの作成者
title	VARCHAR2(50)	ドキュメントのタイトル
body	CHAR(1)	CREATE INDEX 内で指定するダミー列

注意： DETAIL_DATASTORE 型を使用する場合は、マスター表に主キー列を組み込む必要があります。

ディテール表の例 ディテール表にはドキュメントのテキストがあり、通常、その内容はいくつかの行に分散して格納されます。次のディテール表 my_detail は、article_id 列でマスター表 my_master と関連しています。この列によって、各ディテール表の行（サブドキュメント）が属するマスター・ドキュメントを識別します。

列名	列型	説明
article_id	NUMBER	マスター表に関連するドキュメント ID
seq	NUMBER	article_id によって定義されるマスター・ドキュメントのドキュメント順序
text	VARCHAR2	ドキュメント・テキスト

ディテール表の例の属性 この例で、DETAIL_DATASTORE 属性には次の値が指定されます。

属性	属性値
binary	TRUE
detail_table	my_detail
detail_key	article_id
detail_lineno	seq
detail_text	text

CTX_DDL.CREATE_PREFERENCE を使用して、DETAIL_DATASTORE を持つプリファレンスを作成します。CTX_DDL.SET_ATTRIBUTE を使用して、前述のプリファレンスに属性を設定します。次に、これを行う方法を示します。

```
begin
    ctx_ddl.create_preference('my_detail_pref', 'DETAIL_DATASTORE');
    ctx_ddl.set_attribute('my_detail_pref', 'binary', 'true');
    ctx_ddl.set_attribute('my_detail_pref', 'detail_table', 'my_detail');
    ctx_ddl.set_attribute('my_detail_pref', 'detail_key', 'article_id');
    ctx_ddl.set_attribute('my_detail_pref', 'detail_lineno', 'seq');
    ctx_ddl.set_attribute('my_detail_pref', 'detail_text', 'text');
end;
```

マスター / ディテールの索引の例 このマスター / ディテールの関係で定義されたドキュメントを索引付けするには、CREATE INDEX でマスター表の列を指定します。指定する列は、許容される型のいずれかである必要があります。

この例では body 列を使用します。この列には、マスター / ディテールの索引の作成を可能にし、コードの可読性を向上させる機能があります。my_detail_pref プリファレンスは、必要な属性で DETAIL_DATASTORE に設定されます。

```
CREATE INDEX myindex on my_master(body) indextype is ctxsys.context
parameters('datastore my_detail_pref');
```

この例では、title または author 列を指定して索引を作成することもできます。ただし、これを指定すると、これらの列が変更された場合に、索引の再作成操作がトリガーされます。

FILE_DATASTORE

FILE_DATASTORE 型は、ローカル・ファイル・システム上のファイルに格納されているテキストに使用します。

FILE_DATASTORE には、次の属性があります。

属性	属性値
path	*path1*:*path2*::*pathn*

path
ファイル・システムに外部的に格納されたファイルのフル・ディレクトリ・パス名を指定します。このようにフル・ディレクトリ・パスを指定する場合、テキスト列に組み込む必要があるのはファイル名のみです。

path には複数のパスを指定できます（パスはコロン（:）で区切ります）。ファイル名は、テキスト表のテキスト列に格納されます。

この属性を使用して外部ファイルのパスを指定しない場合は、テキスト列にあるファイル名にパスを組み込む必要があります。

FILE_DATASTORE の例

この例では、パス /mydocs を持つ COMMON_DIR というファイル・データストア・プリファレンスを作成します。

```
begin
 ctx_ddl.create_preference('COMMON_DIR','FILE_DATASTORE');
 ctx_ddl.set_attribute('COMMON_DIR','PATH','/mydocs');
end;
```

mytable 表にデータを移入する場合、テキスト列にはファイル名の挿入のみが必要となります。path 属性によって、索引付け操作中に検索する場所がシステムに示されます。

```
create table mytable(id number primary key, docs varchar2(2000));
insert into mytable values(111555,'first.txt');
insert into mytable values(111556,'second.txt');
commit;
```

次のように索引を作成します。

```
create index myindex on mytable(docs)
  indextype is ctxsys.context
  parameters ('datastore COMMON_DIR');
```

URL_DATASTORE

次のファイルに格納されるテキストには、URL_DATASTORE 型を使用します。

- HTTP または FTP を使用してアクセスする World Wide Web 上のファイル
- ファイル・プロトコルを使用してアクセスするローカル・ファイル・システム内のファイル

各 URL を単一のテキスト・フィールドに格納します。

URL の構文

テキスト・フィールドに格納する URL の構文は、次のとおりです（カッコ内は、オプションのパラメータです）。

```
[URL:]<access_scheme>://<host_name>[:<port_number>]/[<url_path>]
```

access_scheme 文字列は、*ftp*、*http* または *file* のいずれかです。たとえば、次のようにします。

```
http://mymachine.us.oracle.com/home.html
```

この構文は RFC1738 仕様に部分的に適合しているため、次の制限が URL の構文に適用されます。

- URL には、印字可能な ASCII 文字のみが含まれる必要があります。印字できない ASCII 文字およびマルチバイト文字は、%*xx* 表記法でエスケープする必要があります。この *xx* は特殊文字の 16 進コードを表します。

> **注意：** login:password@ 構文が URL 内でサポートされるのは、FTP アクセス・スキームの場合のみです。

URL_DATASTORE 属性

URL_DATASTORE には、次の属性があります。

属性	属性値
timeout	タイムアウトを秒単位で指定します。有効値の範囲は 15 ～ 3600 秒です。デフォルトは 30 です。
maxthreads	同時に実行できるスレッドの最大数を指定します。1 ～ 1024 の数を使用します。デフォルトは 8 です。
urlsize	URL 文字列の最大長をバイト単位で指定します。32 ～ 65535 の数を指定します。デフォルトは 256 です。
maxurls	URL バッファの最大サイズを指定します。32 ～ 65535 の数を指定します。デフォルトは 256 です。
maxdocsize	ドキュメントの最大サイズを指定します。256 ～ 2,147,483,647 バイト（2GB）の数を指定します。デフォルトは 2,000,000 です。
http_proxy	HTTP プロキシ・サーバーのホスト名を指定します。オプションで、hostname:port という形式でコロンを使用してポート番号を指定します。
ftp_proxy	FTP プロキシ・サーバーのホスト名を指定します。オプションで、hostname:port という形式でコロンを使用してポート番号を指定します。
no_proxy	非プロキシ・サーバーのドメインを指定します。カンマで区切られた文字列を使用して、最大 16 個のドメイン名が指定できます。

timeout

接続や読込みなどのネットワーク処理がタイムアウトになるまでの時間、およびアプリケーションにタイムアウト・エラーを戻すまでの時間の長さを秒単位で指定します。timeout の有効値の範囲は 15 ～ 3600 で、デフォルトは 30 です。

> **注意：** タイムアウトはネットワーク処理を基準にしています。タイムアウトの合計時間は timeout に指定された時間より長くなることがあります。

maxthreads

同時に実行できるスレッドの最大数を指定します。maxthreads の有効値の範囲は 1 ～ 1024 で、デフォルトは 8 です。

urlsize
データベースに格納されている URL に対して、URL データストアがサポートする最大長をバイト単位で指定します。URL が最大長より長い場合はエラーが戻ります。urlsize の有効値の範囲は 32 ～ 65535 で、デフォルトは 256 です。

注意： maxurls と urlsize に指定した値の積が 5,000,000 を超えないようにしてください。

URL で使用できるメモリー・バッファの最大サイズ（maxurls × urlsize）は約 5MB です。

---

maxurls
内部バッファが、テキスト表から取り出された HTML ドキュメント（行）に保持できる行の最大数を指定します。maxurls の有効値の範囲は 32 ～ 65535 で、デフォルトは 256 です。

注意： maxurls と urlsize に指定した値の積が 5,000,000 を超えないようにしてください。

URL で使用できるメモリー・バッファの最大サイズ（maxurls × urlsize）は約 5MB です。

URL データストアが、データベースに格納されている URL を持つ HTML ドキュメントにアクセスする場合にサポートする最大サイズを、バイト単位で指定します。maxdocsize の有効値の範囲は 1 ～ 2,147,483,647（2GB）で、デフォルトは 2,000,000 です。

http_proxy
Oracle Text がインストールされているマシンの HTTP プロキシ（ゲートウェイ）として機能するホスト・マシンの完全修飾名を指定します。オプションで、hostname:port という形式でコロンを使用してポート番号を指定できます。

この属性の設定が必要なのは、ファイアウォールの外側にある Web ファイルにアクセスするためにプロキシ・サーバーによる認証が必要なイントラネット上にマシンがある場合です。

ftp_proxy
Oracle Text がインストールされているマシンの FTP プロキシ（ゲートウェイ）として機能するホスト・マシンの完全修飾名を指定します。オプションで、hostname:port という形式でコロンを使用してポート番号を指定できます。

この属性の設定が必要なのは、ファイアウォールの外側にある Web ファイルにアクセスするためにプロキシ・サーバーによる認証が必要なイントラネット上にマシンがある場合です。

no_proxy
イントラネット上のマシンのほとんどすべてにあるドメイン文字列（16 個以内、カンマで区切られている）を指定します。ドメインの 1 つがホスト名内で検出されると、ftp_proxy および http_proxy に指定されたマシンには要求が送られません。かわりに、要求は、URL 内で指定されたホスト・マシンによって直接処理されます。

たとえば、no_proxy に *us.oracle.com, uk.oracle.com* という文字列を指定した場合、ホスト名にこのドメインのいずれかが含まれているマシンへのすべての URL 要求は、プロキシ・サーバーでは処理されません。

URL_DATASTORE の例

この例では、URL_PREF という URL_DATASTORE プリファレンスを作成します。このプリファレンスには、http_proxy、no_proxy および timeout 属性を設定します。設定されない属性には、デフォルトが使用されます。

```
begin
 ctx_ddl.create_preference('URL_PREF','URL_DATASTORE');
 ctx_ddl.set_attribute('URL_PREF','HTTP_PROXY','www-proxy.us.oracle.com');
 ctx_ddl.set_attribute('URL_PREF','NO_PROXY','us.oracle.com');
 ctx_ddl.set_attribute('URL_PREF','Timeout','300');
end;
```

表を作成し、その表に値を挿入します。

```
create table urls(id number primary key, docs varchar2(2000));
insert into urls values(111555,'http://context.us.oracle.com');
insert into urls values(111556,'http://www.sun.com');
commit;
```

索引を作成するには、データストアとして URL_PREF を指定します。

```
create index datastores_text on urls ( docs )
  indextype is ctxsys.context
  parameters ( 'Datastore URL_PREF' );
```

USER_DATASTORE

USER_DATASTORE 型を使用して、索引付け時にドキュメントを合成するストアド・プロシージャを定義します。たとえば、ユーザー・プロシージャによって、1 つのドキュメント内に作成者、日付およびテキスト列を合成することで、作成者と日付の情報を索引付けされるドキュメントの一部にすることもできます。

USER_DATASTORE には、次の属性があります。

属性	属性値
procedure	索引付けされるドキュメントを合成するプロシージャを指定します。 このプロシージャは、CTXSYS が所有し、索引所有者が実行できる必要があります。
output_type	procedure への 2 つ目の引数のデータ型を指定します。有効な値は、CLOB、BLOB、CLOB_LOC、BLOB_LOC または VARCHAR2 です。デフォルトは CLOB です。 CLOB_LOC または BLOB_LOC を指定することは、一時 CLOB または一時 BLOB が不要であることを示します。これは、プロシージャによってロケータが IN/OUT の第 2 パラメータにコピーされるためです。

procedure
索引付けされるドキュメントを合成するプロシージャ名を指定します。この指定は、PROCEDURENAME または PACKAGENAME.PROCEDURENAME の形式である必要があります。スキーマの所有者名は CTXSYS に制限されているため、所有者名の指定は必要ありません。

指定するプロシージャには、次のように定義された 2 つの引数が必要です。

```
procedure (r IN ROWID, c IN OUT NOCOPY <output_type>)
```

1 つ目の引数 r は、ROWID 型にしてください。2 つ目の引数 c は、output_type 型である必要があります。NOCOPY は、可能な場合は参照によってパラメータ c を渡すように Oracle に指示する、コンパイラ・ヒントです。

注意： プロシージャ名およびその引数には、どのような名前でも付けることができます。この例では、説明を単純にするために、r および c という引数を使用しています。

ストアド・**プロシージャ**は、索引付けされる行ごとに 1 回コールされます。現在行の ROWID が引数として渡されるたびに、この**プロシージャ**は、ドキュメントのテキストを 2 つ目の引数（output_type 型で指定）に書き込む必要があります。

制約 次の制約が、procedure に適用されます。

- procedure の所有者は、CTXSYS である必要があります。
- procedure は、索引所有者が実行できる必要があります。
- procedure は、COMMIT のようなトランザクション制御文または DDL を発行できません。

索引付け後のプロシージャの編集 ストアド・プロシージャを変更または編集する場合、そのプロシージャをベースにしている索引には通知されないため、このような索引を手動で再作成する必要があります。ストアド・プロシージャが他の列を使用し、その列の値が変更された場合、行は再索引付けされません。行が再索引付けされるのは、索引列が変更されたときのみです。

output_type
procedure への 2 つ目の引数のデータ型を指定します。CLOB、BLOB、CLOB_LOC、BLOB_LOC または VARCHAR2 のいずれかを指定できます。

CLOB を含む USER_DATASTORE の例

次のように定義された articles 表のように、作成者、タイトルおよびテキスト・フィールドが分割されている表を考えます。

```
create table articles(
    id       number,
    author   varchar2(80),
    title    varchar2(120),
    text     clob );
```

作成者フィールドおよびタイトル・フィールドは、索引付けドキュメントのテキストの一部になります。ユーザー appowner が、テキスト、作成者およびタイトル・フィールドからドキュメントを合成するストアド・プロシージャを作成するとします。

```
create procedure myproc(rid in rowid, tlob in out clob) is
  begin
      for c1 in (select author, title, text from articles
                  where rowid = rid)
      loop
       dbms_lob.writeappend(tlob, length(c1.title), c1.title);
       dbms_lob.writeappend(tlob, length(c1.author), c1.author);
       dbms_lob.writeappend(tlob, length(c1.text), c1.text);
       end loop;
    end;
```

このプロシージャは、ROWID および一時 CLOB ロケータを取得し、すべての列を一時 CLOB に連結します。for ループは 1 回のみ実行されます。

CTXSYS が所有するプロシージャのみがユーザー・データストアに使用可能であるため、CTXSYS は、（appowner が所有する）ユーザー・プロシージャを、CTXSYS が所有するプロシージャで次のようにラップする必要があります。

```
create procedure s_myproc(rid in rowid, tlob in out clob) is
 begin
      appowner.myproc(rid, tlob);
 end;
```

CTXSYS ユーザーは、次のように実行権限を付与することによって、索引所有者がスタブ・プロシージャを実行できることを確認する必要があります。

```
grant execute on s_myproc to appowner ;
```

ユーザー appowner は、procedure 属性を次のように ctxsys スタブ・プロシージャの名前に設定して、プリファレンスを作成します。

```
begin
    ctx_ddl.create_preference('myud', 'user_datastore');
    ctx_ddl.set_attribute('myud', 'procedure', 's_myproc');
    ctx_ddl.set_attribute('myud', 'output_type', 'CLOB');
end;
```

appowner が、このプリファレンスを使用して索引を articles(text) に作成する場合、索引付け操作では、ドキュメント・テキストにある作成者およびタイトルが参照されます。

BLOB_LOC を含む USER_DATASTORE の例

次のプロシージャは、OUTPUT_TYPE BLOB_LOC で使用できます。

```
    procedure myds(rid in rowid, dataout in out nocopy blob)
    is
      l_dtype varchar2(10);
      l_pk    number;
    begin
      select dtype, pk into l_dtype, l_pk from mytable where rowid = rid;
      if (l_dtype = 'MOVIE') then
        select movie_data into dataout from movietab where fk = l_pk;
      elsif (l_dtype = 'SOUND') then
        select sound_data into dataout from soundtab where fk = l_pk;
      end if;
    end;
```

CTXSYS が所有するプロシージャのみがユーザー・データストアに使用可能であるため、CTXSYS は、（appowner が所有する）ユーザー・プロシージャを、CTXSYS が所有するプロシージャで次のようにラップする必要があります。

```
create procedure s_myproc(rid in rowid, tlob in out blob) is
 begin
      appowner.myds(rid, tlob);
 end;
```

CTXSYS ユーザーは、次のように実行権限を付与することによって、索引所有者がスタブ・プロシージャを実行できることを確認する必要があります。

```
grant execute on s_myproc to appowner ;
```

ユーザー appowner は、procedure および output_type 属性を、次のように ctxsys スタブ・プロシージャに対応するように設定して、プリファレンスを作成します。

```
begin
    ctx_ddl.create_preference('myud', 'user_datastore');
    ctx_ddl.set_attribute('myud', 'procedure', 's_myproc');
    ctx_ddl.set_attribute('myud', 'output_type', 'blob_loc');
end;
```

NESTED_DATASTORE

NESTED_DATASTORE 型は、NESTED TABLE に行として格納されているドキュメントを索引付けするために使用します。

属性	属性値
nested_column	NESTED TABLE 列の名前を指定します。この属性は必須です。列名のみを指定します。スキーマ所有者または格納表の名前は指定しないでください。
nested_type	NESTED TABLE の型を指定します。この属性は必須です。所有者名および型を指定する必要があります。
nested_lineno	行を順序付けする、NESTED TABLE 内の属性名を指定します。これは、ディテール・データストアの DETAIL_LINENO に似ています。この属性は必須です。

属性	属性値
nested_text	行のテキストを含む NESTED TABLE 型内の列名を指定します。これは、ディテール・データストアの DETAIL_TEXT に似ています。この属性は必須です。LONG 列型は、NESTED TABLE テキスト列としてはサポートされていません。
binary	FALSE を指定すると、ドキュメント・テキストの合成時に、改行が行間に自動的に挿入されます。TRUE を指定すると、これは行われません。この属性は必須ではありません。デフォルトは FALSE です。

NESTED TABLE データストアを使用する場合、拡張索引作成機能フレームワークでは NESTED TABLE 列の索引付けが禁止されているため、ダミー列を索引付けする必要があります。例を参照してください。

NESTED TABLE の DML は、索引付けに使用されたダミー列には自動的に伝播されません。NESTED TABLE の DML をダミー列に伝播するには、アプリケーション・コードまたはトリガーによって、ダミー列を明示的に更新する必要があります。

索引に対するフィルタのデフォルトは、nested_text 列の型によって異なります。

妥当性チェックの間に、その型が存在するかどうか、および nested_lineno と nested_text に対して指定した属性が NESTED TABLE 型に存在するかどうかがチェックされます。指定された NESTED TABLE 列が表に存在するかどうかはチェックされません。

NESTED_DATASTORE の例

NESTED TABLE の作成 次のコードは、NESTED TABLE および NESTED TABLE の記憶表 mytab を作成します。

```
create type nt_rec as object (
  lno number, -- line number
  ltxt varchar2(80) -- text of line
);

create type nt_tab as table of nt_rec;
create table mytab (
   id number primary key, -- primary key
   dummy char(1), -- dummy column for indexing
   doc nt_tab -- nested table
)
nested table doc store as myntab;
```

NESTED TABLE への値の挿入 次のコードは、NESTED TABLE の親行で、ID が 1 であるものに値を挿入します。

```
insert into mytab values (1, null, nt_tab());
insert into table(select doc from mytab where id=1) values (1, 'the dog');
insert into table(select doc from mytab where id=1) values (2, 'sat on mat ');
commit;
```

NESTED TABLE プリファレンスの作成 次のコードは、NESTED TABLE 型 nt_tab および親表 mytab の定義に従って、NESTED_DATASTORE にプリファレンスおよび属性を設定します。

```
begin
-- create nested datastore pref
ctx_ddl.create_preference('ntds','nested_datastore');

-- nest tab column in main table
ctx_ddl.set_attribute('ntds','nested_column', 'doc');

-- nested table type
ctx_ddl.set_attribute('ntds','nested_type', 'scott.nt_tab');

-- lineno column in nested table
ctx_ddl.set_attribute('ntds','nested_lineno','lno');

--text column in nested table
ctx_ddl.set_attribute('ntds','nested_text', 'ltxt');
end;
```

NESTED TABLE への索引の作成 次のコードは、NESTED TABLE データストアを使用して索引を作成します。

```
create index myidx on mytab(dummy) -- index dummy column, not nest table
indextype is ctxsys.context parameters ('datastore ntds');
```

ネストしたデータストアの問合せ 次の SELECT 文は、NESTED TABLE から作成された索引を問い合せます。

```
select * from mytab where contains(dummy, 'dog and mat')>0;
-- returns document 1, since it has dog in line 1 and mat in line 2.
```

フィルタ型

フィルタ型を使用して、ドキュメントをフィルタ処理して索引付けテキスト文字列を取得するためのプリファレンスを作成します。フィルタ処理によって、プレーン・テキスト、HTML および XML ドキュメントの他に、ワード・プロセッサ形式のドキュメントおよび形式設定されたドキュメントの索引付けができるようになります。

形式設定されたドキュメントについては、ドキュメントを固有の形式で格納した後、フィルタを使用してそのドキュメントの一時的なプレーン・テキストまたは HTML 形式を作成します。プレーン・テキストまたは HTML で形式設定されたドキュメントから導出されたワードが、索引付けされます。

フィルタ・プリファレンスを作成するには、次の型のうちいずれか 1 つを使用する必要があります。

フィルタ・プリファレンス型	説明
CHARSET_FILTER	キャラクタ・セット変換フィルタです。
INSO_FILTER	形式設定されたドキュメントのフィルタ処理用 Inso フィルタです。
NULL_FILTER	フィルタ処理は必要ありません。プレーン・テキスト、HTML または XML ドキュメントの索引付けに使用します。
USER_FILTER	カスタム・フィルタ処理に使用するユーザー定義外部フィルタです。
PROCEDURE_FILTER	カスタム・フィルタ処理に使用するユーザー定義ストアド・プロシージャ・フィルタです。

CHARSET_FILTER

CHARSET_FILTER を使用して、非データベース・キャラクタ・セットからデータベース・キャラクタ・セットにドキュメントを変換します。

CHARSET_FILTER には、次の属性があります。

属性	属性値
charset	変換元キャラクタ・セットのグローバリゼーション・サポート名を指定します。 UTF16AUTO を設定すると、このフィルタは、キャラクタ・セットの設定が UTF16 big-endian であるか、little-endian であるかを自動的に識別します。 日本語キャラクタ・セットの自動識別には JAAUTO を指定します。このフィルタは、JA16EUC または JA16SJIS のカスタム文字仕様を自動的に識別し、データベース・キャラクタ・セットに変換します。このフィルタは、データ・ファイルにキャラクタ・セットが混在する場合の日本語に有効です。

関連項目： サポートされているグローバリゼーション・サポート・キャラクタ・セットの詳細は、『Oracle9*i* Database グローバリゼーション・サポート・ガイド』を参照してください。

UTF16 big-endian および little-endian の識別

キャラクタ・セットが UTF-16 の場合、UTF16AUTO を指定すると、big-endian または little-endian を自動的に識別できます。Oracle では、ドキュメント行の最初の 2 バイトを調べて、自動識別を実行します。

最初の 2 バイトが 0xFE、0xFF の場合、そのドキュメントは little-endian と認識され、この 2 バイトを除いた残りのドキュメントが索引付けのために渡されます。

最初の 2 バイトが 0xFF、0xFE の場合、そのドキュメントは big-endian と認識され、この 2 バイトを除いた残りのドキュメントが索引付けのために渡されます。

最初の 2 バイトがそれ以外の場合、そのドキュメントは big-endian とみなされ、最初の 2 バイトを含むドキュメント全体が索引付けのために渡されます。

複合キャラクタ・セット列の索引付け

複合キャラクタ・セット列は、異なるキャラクタ・セットを持つドキュメントを格納する列です。たとえば、1つのテキスト表に、WE8ISO8859P1 のドキュメントと UTF8 のドキュメントが格納されていることがあります。

異なるキャラクタ・セットのドキュメントを持つ表を索引付けするには、キャラクタ・セット列を持つ元表を作成する必要があります。この列では、ドキュメントのキャラクタ・セットを行ごとに指定します。ドキュメントは、データベース・キャラクタ・セットに変換され、索引付けされます。

キャラクタ・セット変換は、CHARSET_FILTER で機能します。キャラクタ・セット列が NULL、または認識されない場合、変換元キャラクタ・セットは、charset 属性に指定されたものの1つとみなされます。

> **注意：** ドキュメント・フォーマット列が TEXT に設定されている場合、キャラクタ・セット変換は、INSO_FILTER でも機能します。

複合キャラクタ・セットの索引付けの例 キャラクタ・セット列を持つ表を、次のように作成するとします。

```
create table hdocs (
     id number primary key,
     fmt varchar2(10),
     cset varchar2(20),
     text varchar2(80)
);
```

プレーン・テキストのドキュメントを挿入し、キャラクタ・セット名を指定します。

```
insert into hdocs values(1, 'text', 'WE8ISO8859P1', '/docs/iso.txt');
insert in hdocs values (2, 'text', 'UTF8', '/docs/utf8.txt');
commit;
```

索引を作成し、キャラクタ・セット列を指定します。

```
create index hdocsx on hdocs(text) indextype is ctxsys.context
  parameters ('datastore ctxsys.file_datastore
  filter ctxsys.charset_filter
  format column fmt
  charset column cset');
```

INSO_FILTER

INSO_FILTER は、ほとんどのドキュメント形式をフィルタ処理する汎用フィルタです。このフィルタ処理テクノロジは、Stellent Chicago, Inc. からライセンス許可を受けています。

単一書式または複合書式の列の索引付けに使用します。

> **関連項目：** INSO_FILTER がサポートしている形式のリストおよびこのフィルタを使用する環境の設定方法は、付録 B「サポートされているドキュメント形式」を参照してください。

INSO_FILTER には、次の属性があります。

属性	属性値
timeout	INSO_FILTER タイムアウトを秒単位で指定します。0 ～ 42,949,672 の数を使用します。デフォルトは 120 です。この値を 0（ゼロ）に設定すると、この機能は使用禁止になります。 この待機時間がどのように使用されるかは、timeout_type の設定方法によって異なります。 この機能は、行に対応しているキャラクタ・セット列やフォーマット列によって INSO_FILTER がバイパスする行には使用できません。たとえば、その形式が TEXT にマークされている場合です。 この機能を使用して、Oracle の索引付け操作が停止中の Inso フィルタ操作を無制限に待機するのを回避します。
timeout_type	HEURISTIC または FIXED を指定します。デフォルトは HEURISTIC です。 INSO_FILTER による出力が増加した場合に、すべてのタイムアウト秒をチェックする場合は、HEURISTIC を指定します。出力が増加していない場合、そのドキュメントに対する操作は終了します。エラーは CTX_USER_INDEX_ERRORS ビューに記録され、操作は次の索引付け対象のドキュメント行に移動します。 フィルタ処理が正常に進行中か停止中かに関係なく、タイムアウト秒後に INSO_FILTER 処理を終了するには、FIXED を指定します。この値は、大規模なドキュメントのフィルタ処理を正常に行うために時間をかけるより、索引付けのスループットが重要な場合に便利です。

形式設定されたドキュメントの索引付け

Microsoft Word などの形式設定されたドキュメントを含むテキスト列を索引付けするには、INSO_FILTER を使用します。このフィルタは、ドキュメント形式を自動的に検出します。CTXSYS.INSO_FILTER システム定義プリファレンスを、PARAMETERS 句で次のように使用できます。

```
create index hdocsx on hdocs(text) indextype is ctxsys.context
  parameters ('datastore ctxsys.file_datastore
  filter ctxsys.inso_filter');
```

複合フォーマット列のプレーン・テキストまたは HTML のバイパス

複合フォーマット列は、複数のドキュメント形式を含むテキスト列です。これには、Microsoft Word、PDF、プレーン・テキストおよび HTML ドキュメントを含む列などがあります。

INSO_FILTER によって、複合フォーマット列を索引付けできます。ただし、Inso フィルタに、プレーン・テキストまたは HTML ドキュメントのフィルタ処理をバイパスさせた方が効率的です。INSO_FILTER によるプレーン・テキストまたは HTML のフィルタ処理は、冗長です。

元表内のフォーマット列では、テキスト列に含まれるドキュメントのタイプを指定できます。指定できるタイプは、TEXT および BINARY の 2 つのみです。索引付け中に、INSO_FILTER は、TEXT に指定されたすべてのドキュメントを無視します（キャラクタ・セット列が指定されていない場合）。

INSO_FILTER のバイパス・メカニズムを設定するには、元表にフォーマット列を作成する必要があります。

たとえば、次のように作成します。

```
create table hdocs (
     id number primary key,
     fmt varchar2(10),
     text varchar2(80)
);
```

主に Word ドキュメントを索引付けすると想定すると、Word ドキュメントをフィルタするために、フォーマット列に BINARY を指定します。一方、INSO_FILTER に HTML ドキュメントを無視させるには、フォーマット列に TEXT を指定します。

たとえば、次の文によってテキスト表に 2 つのドキュメントを追加し、1 つのドキュメントの形式に BINARY を、もう 1 つのドキュメントの形式に TEXT を割り当てるとします。

```
insert into hdocs values(1, 'binary', '/docs/myword.doc');
insert in hdocs values (2, 'text', '/docs/index.html');
commit;
```

索引を作成するには、CREATE INDEX を使用して、パラメータ文字列にフォーマット列名を指定します。

```
create index hdocsx on hdocs(text) indextype is ctxsys.context
  parameters ('datastore ctxsys.file_datastore
  filter ctxsys.inso_filter
  format column fmt');
```

フォーマット列に TEXT または BINARY を指定しない場合は、BINARY が使用されます。

注意： INSO_FILTER を使用している場合は、CREATE INDEX にフォーマット列を指定する必要はありません。

Inso によるキャラクタ・セット変換

ドキュメント形式列が TEXT に設定されている場合、INSO_FILTER はドキュメントをデータベースのキャラクタ・セットに変換します。この場合、INSO_FILTER はキャラクタ・セット列を検索して、ドキュメントのキャラクタ・セットを判断します。

キャラクタ・セット列値が Oracle キャラクタ・セット名でない場合、ドキュメントはキャラクタ・セット変換なしで渡されます。

注意： INSO_FILTER を使用している場合は、キャラクタ・セット列を指定する必要はありません。

キャラクタ・セット列を指定して、フォーマット列を指定しない場合、INSO_FILTER は CHARSET_FILTER のように機能します。ただし、この場合、日本語キャラクタ・セットの自動識別は行われません。

関連項目： 2-28 ページ「CHARSET_FILTER」

プレーン・テキストの索引付けおよび INSO_FILTER

プレーン・テキスト・ドキュメントの索引付けに INSO_FILTER を使用することは、お薦めできません。

表にテキスト・ドキュメントのみが含まれている場合は、NULL_FILTER または USER_FILTER を使用します。

表に、形式設定されたドキュメントとテキスト・ドキュメントが混在する場合、フォーマット列を作成し、テキスト・ドキュメントを TEXT にマークして、INSO_FILTER をバイパスすることをお薦めします。この場合、キャラクタ・セット列を作成してドキュメントのキャラクタ・セットを識別することもお薦めします。

ただし、INSO_FILTER を使用して非バイナリ・ドキュメント（テキスト）を索引付けし、フォーマット列およびキャラクタ・セット列を指定しない場合、INSO_FILTER はドキュメントを処理します。このため、索引付け処理に、Inso テクノロジのキャラクタ・セット制限が適用されます。特に、使用しているアプリケーションで、次の条件のいずれかに該当することを確認する必要があります。

- ドキュメントのキャラクタ・セットがデータベースのキャラクタ・セットと同じであり、データベースのキャラクタ・セットが次のいずれかであること。
  - US7ASCII
  - WE8ISO8859P1
  - JA16SJIS
  - KO16KSC5601
  - ZHS16CGB231280
  - ZHT16BIG5
- データベースのキャラクタ・セットが前述のいずれでもなく、ドキュメントのキャラクタ・セットが WE8ISO8859P1 であること。

NULL_FILTER

NULL_FILTER 型は、プレーン・テキストまたは HTML が索引付けされ、フィルタ処理を行う必要がない場合に使用します。NULL_FILTER には属性がありません。

HTML ドキュメントの索引付け

ドキュメント・セット全体が HTML である場合は、フィルタ・プリファレンスに NULL_FILTER を使用することをお薦めします。

たとえば、HTML ドキュメント・セットを索引付けするには、NULL_FILTER および HTML_SECTION_GROUP に対するシステム定義プリファレンスを次のように指定できます。

```
create index myindex on docs(htmlfile) indextype is ctxsys.context
  parameters('filter ctxsys.null_filter
  section group ctxsys.html_section_group');
```

> **関連項目：** セクション・グループと HTML ドキュメントの索引付けの詳細は、「セクション・グループ型」を参照してください。

USER_FILTER

USER_FILTER 型を使用して、列内のドキュメントをフィルタ処理するための外部フィルタを指定します。USER_FILTER には、次の属性があります。

属性	属性値
command	実行可能なフィルタの名前を指定します。

command

列に格納されているすべてのテキストをフィルタ処理するときに使用する、単一外部フィルタの実行可能ファイルを指定します。複数のドキュメント形式が列に格納されている場合、command に指定された外部フィルタはそれらの形式のすべてを認識し、処理する必要があります。

指定する実行可能ファイルは、$ORACLE_HOME/ctx/bin ディレクトリに存在する必要があります。次の 2 つのパラメータで、ユーザー・フィルタ実行可能ファイルを作成します。1 つ目のパラメータは読み込まれる入力ファイルの名前で、2 つ目のパラメータは書き込まれる出力ファイルの名前です。

すべてのドキュメント形式が INSO_FILTER でサポートされている場合は、フィルタ処理以外にドキュメントに対する追加処理が必要でないかぎり、USER_FILTER のかわりに INSO_FILTER を使用します。

ユーザー・フィルタの例

次に、ユーザー・フィルタとして使用される perl スクリプトの例を示します。このスクリプトは、1 つ目の引数に指定された入力テキスト・ファイルを大文字に変換し、出力を 2 つ目の引数に指定された場所に書き込みます。

```
#!/usr/local/bin/perl

open(IN, $ARGV[0]);
open(OUT, ">".$ARGV[1]);

while (<IN>)
{
  tr/a-z/A-Z/;
  print OUT;
}

close (IN);
close (OUT);
```

このファイルの名前が upcase.pl である場合、次のようにフィルタ・プリファレンスを作成します。

```
begin
  ctx_ddl.create_preference
    (
      preference_name => 'USER_FILTER_PREF',
      object_name     => 'USER_FILTER'
    );
  ctx_ddl.set_attribute
    ('USER_FILTER_PREF','COMMAND','upcase.pl');
end;
```

次のように SQL*Plus で索引を作成します。

```
create index user_filter_idx on user_filter ( docs )
  indextype is ctxsys.context
  parameters ('FILTER USER_FILTER_PREF');
```

PROCEDURE_FILTER

ドキュメントをストアド・プロシージャでフィルタ処理するには、PROCEDURE_FILTER 型を使用します。ストアド・プロシージャは、ドキュメントのフィルタ処理が必要になるたびにコールされます。

この型の属性は、次のとおりです。

属性	用途	使用可能な値
procedure	フィルタのストアド・プロシージャの名前。	CTXSYS 所有のプロシージャ。プロシージャには、PL/SQL ストアド・プロシージャを指定できます。
input_type	ストアド・プロシージャの入力引数の型。	VARCHAR2、BLOB、CLOB、FILE
output_type	ストアド・プロシージャの出力引数の型。	VARCHAR2、CLOB、FILE
rowid_parameter	ROWID パラメータを含めるか。	TRUE/FALSE
format_parameter	format パラメータを含めるか。	TRUE/FALSE
charset_parameter	charset パラメータを含めるか。	TRUE/FALSE

procedure
フィルタ処理に使用するストアド・プロシージャの名前を指定します。プロシージャには、PL/SQL ストアド・プロシージャを指定できます。プロシージャは、セーフ・コールアウトになることも、セーフ・コールアウトをコールすることもできます。

プロシージャは、CTXSYS が所有し、次のシグネチャのいずれかを所有している必要があります。

```
PROCEDURE(IN BLOB, IN OUT NOCOPY CLOB)
PROCEDURE(IN CLOB, IN OUT NOCOPY CLOB)
PROCEDURE(IN VARCHAR, IN OUT NOCOPY CLOB)
PROCEDURE(IN BLOB, IN OUT NOCOPY VARCHAR2)
PROCEDURE(IN CLOB, IN OUT NOCOPY VARCHAR2)
PROCEDURE(IN VARCHAR2, IN OUT NOCOPY VARCHAR2)
PROCEDURE(IN BLOB, IN VARCHAR2)
PROCEDURE(IN CLOB, IN VARCHAR2)
PROCEDURE(IN VARCHAR2, IN VARCHAR2)
```

1 つ目の引数は、データストアによって渡されたままのフィルタ処理されていない行の内容です。2 つ目の引数は、フィルタ処理済みのドキュメント・テキストを戻すプロシージャ用の引数です。

プロシージャの属性は必須です。したがって、デフォルトはありません。

input_type
フィルタ・プロシージャの入力引数の型を指定します。次のうちのいずれかを指定できます。

型	説明
BLOB	入力引数は、BLOB 型です。渡された BLOB には、フィルタ処理されていないドキュメントが格納されます。
CLOB	入力引数は、CLOB 型です。渡された CLOB には、フィルタ処理されていないドキュメントが格納されます。 フィルタ処理もキャラクタ・セット変換も実行されません。データストアがバイナリ・データを出力すると、そのバイナリ・データは直接 CLOB に書き込まれます。このとき、グローバリゼーション・サポートによって文字データへの暗黙的なマッピングが最大限実行されます。
VARCHAR2	入力引数は、VARCHAR2 型です。渡された VARCHAR2 には、フィルタ処理されていないドキュメントが格納されます。 ドキュメントのデータは、最大 32767 バイトです。フィルタ処理されていないドキュメントがこの長さより長い場合は、ドキュメントに対してエラーが戻され、フィルタ・プロシージャはコールされません。
FILE	入力引数は、VARCHAR2 型です。フィルタ処理されていないドキュメントの内容は、ファイル・システム内の一時ファイルに格納され、そのファイル名が渡された VARCHAR2 に格納されます。 たとえば、渡された VARCHAR2 の値が tmp/mydoc.tmp の場合、ドキュメントの内容はファイル /tmp/mydoc.tmp に格納されます。 この FILE 入力型は、プロシージャがファイルの読込みができるセーフ・コールアウトの場合のみ有効です。

input_type 属性は必須ではありません。指定しない場合は、BLOB がデフォルトになります。

output_type
フィルタ・プロシージャの出力引数の型を指定します。次のいずれかの型を指定できます。

型	説明
CLOB	出力引数は、IN OUT NOCOPY CLOB です。プロシージャでは、渡された CLOB にフィルタ処理済みの内容を書き込む必要があります。
VARCHAR2	出力引数は、IN OUT NOCOPY VARCHAR2 です。プロシージャでは、渡された VARCHAR2 変数にフィルタ処理済みの内容を書き込む必要があります。
FILE	出力引数は、IN VARCHAR2 である必要があります。フィルタ・プロシージャの入力時、出力引数は一時ファイルの名前です。フィルタ・プロシージャでは、この名前が指定されたファイルにフィルタ処理済みの内容を書き込む必要があります。 FILE 出力型の使用は、プロシージャがファイルへの書込みができるセーフ・コールアウトの場合のみ有効です。

output_type 属性は必須ではありません。指定しない場合は、CLOB がデフォルトになります。

rowid_ parameter
TRUE を指定するとフィルタ処理を行うドキュメントの ROWID が、入力および出力パラメータの前に、1 つ目のパラメータとして渡されます。

たとえば、INPUT_TYPE BLOB、OUTPUT_TYPE CLOB および ROWID_PARAMETER TRUE を使用したとします。フィルタ・プロシージャには次のようなシグネチャが必要です。

```
procedure(in rowid, in blob, in out nocopy clob)
```

この属性は、プロシージャが他の列または表のデータを要求する場合に有効です。この属性は必須ではありません。デフォルトは FALSE です。

format_parameter
TRUE を指定すると、フィルタ処理を行うドキュメントのフォーマット列の値が入力および出力パラメータの前にフィルタ・プロシージャに渡されます。ただし、ROWID パラメータが有効な場合は、その後に渡されます。

索引付け時にフォーマット列の名前をパラメータ文字列に指定します。このとき、キーワード 'format column <columnname>' を使用します。パラメータ・タイプは、IN VARCHAR2 である必要があります。

フォーマット列の値は、ROWID パラメータを介して読み込むことができますが、この属性では、単一フィルタが複数の表構造体で機能できます。これは、format 属性が抽象化されており、表またはフォーマット列の名前を認識する必要がないためです。

FORMAT_PARAMETER は必須ではありません。デフォルトは FALSE です。

charset_parameter
TRUE を指定すると、フィルタ処理を行うドキュメントのキャラクタ・セット列の値が入力および出力パラメータの前にフィルタ・プロシージャに渡されます。だたし、ROWID および format パラメータが有効な場合は、その後に渡されます。

索引付け時にキャラクタ・セット列の名前をパラメータ文字列に指定します。このとき、キーワード 'charset column <columnname>' を使用します。パラメータ・タイプは、IN VARCHAR2 である必要があります。

CHARSET_PARAMETER 属性は必須ではありません。デフォルトは FALSE です。

パラメータの順序

ROWID_PARAMETER、FORMAT_PARAMETER および CHARSET_PARAMETER は、すべて非依存型です。順序は rowid、format、次に charset となります。ただし、フィルタ・プロシージャには、必要最小数のパラメータのみが渡されます。

たとえば、INPUT_TYPE が BLOB で、OUTPUT_TYPE が CLOB であるとします。フィルタ・プロシージャにすべてのパラメータが必要な場合、そのプロシージャのシグネチャは、次のように指定する必要があります。

```
(id IN ROWID, format IN VARCHAR2, charset IN VARCHAR2, input IN BLOB, output IN OUT
NOCOPY CLOB)
```

プロシージャで必要なパラメータが ROWID のみの場合、プロシージャのシグネチャは、次のように指定する必要があります。

```
(id IN ROWID,input IN BLOB, ouput IN OUT NOCOPY CLOB)
```

索引作成上の要件

PROCEDURE_FILTER プリファレンスを使用して索引を作成するには、索引所有者が、プロシージャに対する実行権限を所有している必要があります。これは、索引付け時にチェックされます。USER_DATASTORE に対するセキュリティ保護装置と類似しています。

エラー処理

フィルタ・プロシージャでは、通常の PL/SQL raise_application_error 機能を使用して、必要に応じてエラーを戻すことができます。戻されたエラーは、フィルタの起動方法に応じて、CTX_USER_INDEX_ERRORS ビューに伝播されるか、あるいはユーザーにレポートされます。

プロシージャ・フィルタ・プリファレンスの例

次のシグネチャを使用して定義するフィルタ・プロシージャ CTXSYS.NORMALIZE を考えてみます。

```
PROCEDURE NORMALIZE(id IN ROWID, charset IN VARCHAR2, input IN CLOB,
output IN OUT NOCOPY VARCHAR2);
```

このプロシージャをフィルタとして使用するには、フィルタ・プリファレンスを次のように設定します。

```
begin
ctx_ddl.create_preference('myfilt', 'procedure_filter');
ctx_ddl.set_attribute('myfilt', 'procedure', 'normalize');
ctx_ddl.set_attribute('myfilt', 'input_type', 'clob');
ctx_ddl.set_attribute('myfilt', 'output_type', 'varchar2');
ctx_ddl.set_attribute('myfilt', 'rowid_parameter', 'TRUE');
ctx_ddl.set_attribute('myfilt', 'charset_parameter', 'TRUE');
end;
```

レクサー型

レクサー・プリファレンスを使用して、索引付けするテキストの言語を指定します。レクサー・プリファレンスを作成するには、次のレクサー型のうちいずれか 1 つを使用する必要があります。

型	説明
BASIC_LEXER	空白のデリミタ付きワードを使用する英語およびほとんどのヨーロッパ言語で、テキストからのトークンの抽出に使用するレクサー
MULTI_LEXER	異なる複数言語のドキュメントを含む表の索引付けに使用するレクサー
CHINESE_VGRAM_LEXER	中国語テキストからのトークンの抽出に使用するレクサー
CHINESE_LEXER	中国語テキストからのトークンの抽出に使用するレクサー
JAPANESE_VGRAM_LEXER	日本語テキストからのトークンの抽出に使用するレクサー
JAPANESE_LEXER	日本語テキストからのトークンの抽出に使用するレクサー
KOREAN_LEXER	韓国語テキストからのトークンの抽出に使用するレクサー
KOREAN_MORPH_LEXER	韓国語テキストからのトークンの抽出に使用するレクサー（推奨）
USER_LEXER	特定言語を索引付けするために作成するレクサー

BASIC_LEXER

BASIC_LEXER 型を使用して、英語およびサポートされているその他すべての空白のデリミタ付き言語のテキスト索引を作成するためにトークンを識別します。

また、BASIC_LEXER を使用して、拡張キャラクタ・セットを持つ空白のデリミタ付き言語に対する基本文字変換、コンポジット・ワードの索引付け、大 / 小文字を区別した索引付けおよび代替スペルを使用可能にします。

英語およびフランス語では、BASIC_LEXER を使用してテーマの索引付けを使用可能にできます。

注意： 索引付けの前にレクサーがトークンに対して実行した処理（文字の削除や基本文字変換など）は、問合せ時に問合せ語句に対しても実行されます。これによって、問合せ語句は、テキスト索引内のトークンのフォームと確実に一致します。

BASIC_LEXER は、任意のデータベース・キャラクタ・セットをサポートしています。

BASIC_LEXER には、次の属性があります。

属性	属性値
continuation	文字
numgroup	文字
numjoin	文字
printjoin	文字
punctuation	文字
skipjoin	文字
startjoin	トークンの先頭にある英数字以外の文字（文字列）
endjoin	トークンの末尾にある英数字以外の文字（文字列）
whitespace	文字（文字列）
newline	NEWLINE (\n) CARRIAGE_RETURN (\r)
base_letter	NO（使用禁止） YES（使用可能）
mixed_case	NO（使用禁止） YES（使用可能）
composite	DEFAULT（コンポジット・ワードの索引付けなし、デフォルト） GERMAN（ドイツ語のコンポジット・ワード索引付け） DUTCH（オランダ語のコンポジット・ワードの索引付け）
index_stems	0 NONE 1 ENGLISH 2 DERIVATIONAL 3 DUTCH 4 FRENCH 5 GERMAN 6 ITALIAN 7 SPANISH

属性	属性値
index_themes	YES（使用可能）
	NO（使用禁止、デフォルト）
index_text	YES（使用可能、デフォルト）
	NO（使用禁止）
prove_themes	YES（使用可能、デフォルト）
	NO（使用禁止）
theme_language	AUTO（デフォルト）
	（任意のグローバリゼーション・サポート対象言語）
alternate_spelling	GERMAN（ドイツ語の代替スペル）
	DANISH（デンマーク語の代替スペル）
	SWEDISH（スウェーデン語の代替スペル）
	NONE（代替スペルなし、デフォルト）

continuation
ワードが次の行に続き、そのワードを 1 つのトークンとして索引付けする必要があることを示す文字を指定します。最も一般的な連結文字はハイフン '-' およびバックスラッシュ '\' です。

numgroup
数字列の中で使用する場合に、1 つの大きな数字の集まりをいくつかの桁にグループ分けすることを示す文字を指定します。

たとえば、カンマ ',' は、数字列の中で使用されている場合、千の位のグループ分けを示すことがあるため、numgroup 文字として定義されることがあります。

numjoin
数字列の中で使用する場合に、数字の文字列を 1 つの単位またはワードとみなして索引付けするように文字を指定します。

たとえば、ピリオド '.' は、数字列の中で使用される場合、小数点を示すことがあるため、numjoin 文字として定義されることがあります。

注意： numjoin および numgroup のデフォルト値は、データベースに指定されたグローバリゼーション・サポート初期化パラメータによって決定されます。

通常、BASIC_LEXER のレクサー・プリファレンスを作成する場合、numjoin または numgroup のいずれにも値を指定する必要はありません。

---

printjoin
英数字以外の文字を指定します。この文字は、ワード内（先頭、中程または末尾）にあれば英数字として処理され、テキスト索引にトークンとともに組み込まれます。連続している printjoin も同様に処理されます。

たとえば、ハイフン '-' およびアンダースコア '_' 文字が printjoin として定義されている場合、*pseudo-intellectual* や *_file_* などの語句は、*pseudo-intellectual* および *_file_* としてテキスト索引に格納されます。

注意： printjoin 文字が同時に punctuation 文字としても定義されている場合は、その文字の直後の文字が標準の英数字であるか、printjoin 文字または skipjoin 文字として定義されていれば、その文字は英数字としてのみ処理されます。

---

punctuation
ワードの末尾に使用される場合に、文の終わりを示す英数字以外の文字を指定します。デフォルトは、ピリオド '.'、疑問符 '?' および感嘆符 '!' です。

punctuation として定義されている文字は、テキストの索引付け前に削除されます。ただし、punctuation 文字が printjoin 文字としても定義されている場合は、その文字がトークンの最後の文字であり、直前に同じ文字がある場合にのみ削除されます。

たとえば、ピリオド（.）が printjoin 文字と punctuation 文字の両方に定義されている場合は、索引付けおよび問合せのときに次のように変換されます。

トークン	索引付けされたトークン
.doc	.doc
dog.doc	dog.doc
dog..doc	dog..doc
dog.	dog
dog...	dog..

また、BASIC_LEXER は、punctuation 文字を newline 文字および whitespace 文字と組み合せて使用し、文 / 段落検索用の文デリミタおよび段落デリミタを決定します。

skipjoin
英数字以外の文字を指定します。この文字がワード内で使用されている場合、そのワードを単一のトークンとして識別します。ただし、その文字はテキスト索引内にトークンとともに格納されません。

たとえば、ハイフン文字 '-' が skipjoin として定義されている場合、ワード *pseudo-intellectual* は、テキスト索引に *pseudointellectual* として格納されます。

注意： printjoin および skipjoin は相互に排他的です。同じ文字を両方の属性に指定できません。

---

startjoin/endjoin
startjoin は、トークンの最初の文字として検出された場合、明示的にトークンの始まりを識別する文字を指定します。その文字は、そのすぐ後に続く他の startjoin 文字と同様に、トークンに対するテキスト索引内のエントリに含まれます。また、startjoin 文字列の最初の startjoin 文字は、暗黙的に前のトークンを終了します。

endjoin は、トークンの最後の文字として検出された場合、明示的にトークンの終わりを識別する文字を指定します。その文字は、そのすぐ後に続く他の *startjoin* 文字と同様に、トークンに対するテキスト索引内のエントリに含まれます。

次のルールが startjoin と endjoin の両方に適用されます。

- startjoin/endjoin に指定された文字は、BASIC_LEXER のその他の属性には指定できません。
- startjoin/endjoin 文字は、トークンの最初または最後にのみ使用できます。

whitespace
トークン間で空白として扱われる文字を指定します。BASIC_LEXER は、文および段落検索用の文デリミタとして機能する文字列を識別するため、whitespace 文字を punctuation 文字および newline 文字と組み合せて使用します。

whitespace の事前定義のデフォルト値は、'space' と 'tab' です。これらの値は変更できません。whitespace として文字を指定すると、これらのデフォルト値に追加されます。

newline
テキストの行の終わりを示す文字を指定します。BASIC_LEXER は、文および段落検索用の段落デリミタとして機能する文字列を識別するため、newline 文字を punctuation 文字および whitespace 文字と組み合せて使用します。

newline に対する有効値は、NEWLINE および CARRIAGE_RETURN（改行）のみです。デフォルトは NEWLINE です。

base_letter
発音区別記号（ウムラウト、セディーユ、揚音アクセントなど）を持つ文字を、テキスト索引に格納する前に、基本形に変換するかどうかを指定します。デフォルトは NO（基本文字変換は使用禁止）です。

mixed_case
レクサーがトークンをテキストに表示されたままにするか、あるいはすべて大文字に変換するかを指定します。デフォルトでは NO（トークンはすべて大文字に変換される）です。

注意： Oracle では、ワード問合せを、問合せされる索引の大 / 小文字の区別に確実に一致させます。そのため、テキスト索引に対して大 / 小文字の区別を有効にすると、その索引に対する問合せでは常に大文字と小文字が区別されます。

---

composite
ドイツ語またはオランダ語のテキストに対して、コンポジット・ワードの索引付けを使用可能または使用禁止にするかどうかを指定します。デフォルトは NO（コンポジット・ワードの索引付けは使用禁止）です。

ドイツ語ディクショナリで通常 1 エントリであるワードは、複合語幹に分割されませんが、ディクショナリのエントリでないワードは、複合語幹に分割されます。

索引付けされた複合語幹を取得するには、ステミング問合せ（例 : *$bahnhof*）を発行する必要があります。ワードリスト・ステマーの言語は、複合語幹の言語と一致している必要があります。

ステミング・ユーザー・ディクショナリ 使用している言語で、ユーザー・ディクショナリを作成し、ワードの分解方法をカスタマイズできます。

ユーザー・ディクショナリは、$ORACLE_HOME/ctx/data/< 言語 > ディレクトリに作成します。ユーザー・ディクショナリには、拡張子 .dct を付ける必要があります。

たとえば、ドイツ語に対するユーザー・ディクショナリ・ファイルを次のように指定します。

```
$ORACLE_HOME/ctx/data/del/drde.dct
```

このユーザー・ディクショナリの書式は、次のとおりです。

```
input term <tab> output term
```

分解されたワードの個々の部分は、# 文字で区切る必要があります。ドイツ語のワード *Hauptbahnhof* に対するエントリの例を次に示します。

```
Hauptbahnhof<tab>Haupt#Bahnhof
Hauptbahnhofes<tab>Haupt#Bahnhof
Hauptbahnhof<tab>Haupt#Bahnhof
Hauptbahnhoefe<tab>Haupt#Bahnhof
```

index_themes
英語またはフランス語のテーマ情報の索引付けには、YES を指定します。これによって ABOUT 問合せがより正確になります。index_themes 属性および index_text 属性の両方に NO を指定することはできません。

BASIC_LEXER を使用し、index_themes に値を指定しない場合、この属性はデフォルトの NO に設定されます。

このパラメータは、CTXCAT も含めたすべての索引タイプに対して TRUE に設定できます。CATSEARCH で ABOUT 問合せを発行する場合は、CONTEXT 文法による問合せテンプレートを使用してください。

prove_themes
テーマを検証するには、YES を指定します。テーマの検証では、ドキュメント内の関連テーマが検索されます。関連テーマが検索されない場合、親テーマはそのドキュメントから排除されます。

テーマの検証は大規模なドキュメントに対して使用可能です。ワード数の少ない短い記述のテキストで親テーマが検証されることはほとんどなく、結果的に ABOUT 問合せによる再コールでのパフォーマンスが低下します。

テーマの検証によって、ABOUT 問合せの精度が向上し、再コール数が減少（戻される行が減少）します。ABOUT 問合せでの再コールが増加し、検索精度が低下する可能性がある場合は、テーマの検証を使用禁止にできます。デフォルトは YES です。

prove_themes 属性は、CONTEXT 索引および CTXRULE 索引に対してサポートされています。

theme_language
index_themes が YES に設定されている場合は、テーマ生成にどのナレッジ・ベースを使用するかを指定します。index_themes が NO に設定されている場合は、このパラメータを設定しても、何も影響はありません。

任意のグローバリゼーション・サポート対象言語または AUTO を指定できます。指定する言語のナレッジ・ベースが必要です。このリリースで提供されているのは、英語とフランス語のナレッジ・ベースのみです。それ以外の言語では、独自のナレッジ・ベースを作成できます。

> **関連項目：** 第 14 章「実行可能ファイル」の「言語固有のナレッジ・ベースの追加」

デフォルトは AUTO です。AUTO に設定すると、環境の言語に従って、システムがこのパラメータを設定します。

index_stems
語幹索引付けに使用するステマーを指定します。索引付け時に、トークンが通常の形式に加えて、単一の基本形にステミングされます。語幹索引付けによって、ステミング（$）問合せ（例 : *$computed*）の問合せパフォーマンスが向上します。

index_text
ワード情報の索引付けには、YES を指定します。index_themes 属性および index_text 属性の両方に NO を指定することはできません。

デフォルトは NO です。

alternate_spelling
ドイツ語、デンマーク語またはスウェーデン語のいずれかを指定して、これらの言語のうちの 1 つの代替スペルを使用可能にします。代替スペルを使用可能にすると、任意の代替形式でワードを問合せできます。

デフォルトでは、これら 3 言語すべてにおいて代替スペルが使用可能です。代替スペルがない場合は、NONE を指定します。

> **関連項目：** Oracle が使用する代替スペルの規則については、付録 E「代替スペルの規則」を参照してください。

BASIC_LEXER の例

次の例では、printjoin 文字を設定し、BASIC_LEXER を使用してテーマの索引付けを使用禁止にします。

```
begin
ctx_ddl.create_preference('mylex', 'BASIC_LEXER');
ctx_ddl.set_attribute('mylex', 'printjoins', '_-');
ctx_ddl.set_attribute ( 'mylex', 'index_themes', 'NO');
ctx_ddl.set_attribute ( 'mylex', 'index_text', 'YES');
end;
```

テーマを索引付けせず、前述の printjoin キャラクタ・セットを使用して索引を作成するには、次の文を発行します。

```
create index myindex on mytable ( docs )
  indextype is ctxsys.context
  parameters ( 'LEXER mylex' );
```

MULTI_LEXER

MULTI_LEXER は、異なる複数の言語のドキュメントを含むテキスト列の索引付けに使用します。たとえば、このレクサーを使用して、英語、ドイツ語および日本語のドキュメントを格納するテキスト列を索引付けできます。

このレクサーには属性がありません。

元表に言語列が存在する必要があります。マルチ言語表に索引付けするには、索引の作成時に言語列を指定します。

CTX_DDL.CREATE_PREFERENCE を使用して、マルチレクサー・プリファレンスを作成します。CTX_DDL.ADD_SUB_LEXER プロシージャを使用して、そのマルチレクサー・プリファレンスに言語固有のレクサーを追加します。

索引付け時に、MULTI_LEXER は、各行の言語列の値を調べ、言語固有のレクサーに切り替えた後、ドキュメントを処理します。

マルチ言語のストップリスト

MULTI_LEXER を使用するときに、索引付け用のマルチ言語ストップリストも使用できます。

関連項目： この章の「マルチ言語のストップリスト」

MULTI_LEXER の例

次のように、主キー、テキスト列および言語列を持つマルチ言語表を作成します。

```
create table globaldoc (
   doc_id number primary key,
   lang varchar2(3),
   text clob
);
```

保持するドキュメントのほとんどが英語で、ドイツ語または日本語のドキュメントが少しある表を考えてみます。3 つの言語を処理するには、英語、ドイツ語および日本語に対して 1 つずつの、3 つのサブレクサーを作成する必要があります。

```
ctx_ddl.create_preference('english_lexer','basic_lexer');
ctx_ddl.set_attribute('english_lexer','index_themes','yes');
ctx_ddl.set_attribute('english_lexer','theme_language','english');

ctx_ddl.create_preference('german_lexer','basic_lexer');
ctx_ddl.set_attribute('german_lexer','composite','german');
ctx_ddl.set_attribute('german_lexer','mixed_case','yes');
ctx_ddl.set_attribute('german_lexer','alternate_spelling','german');

ctx_ddl.create_preference('japanese_lexer','japanese_vgram_lexer');
```

マルチレクサー・プリファレンスを作成します。

```
ctx_ddl.create_preference('global_lexer', 'multi_lexer');
```

格納されているドキュメントのほとんどが英語であるため、CTX_DDL.ADD_SUB_LEXER を使用して、英語のレクサーをデフォルトにします。

```
ctx_ddl.add_sub_lexer('global_lexer','default','english_lexer');
```

ここで、CTX_DDL.ADD_SUB_LEXER プロシージャを使用して、ドイツ語および日本語のレクサーをそれぞれの言語に追加します。また、言語列が ISO 標準言語コード 639-2 で表現されている場合には、これらを代替値として追加します。

```
ctx_ddl.add_sub_lexer('global_lexer','german','german_lexer','ger');
ctx_ddl.add_sub_lexer('global_lexer','japanese','japanese_lexer','jpn');
```

次のように、PARAMETERS 句にマルチレクサー・プリファレンスおよび言語列を指定して、索引 globalx を作成します。

```
create index globalx on globaldoc(text) indextype is ctxsys.context
parameters ('lexer global_lexer language column lang');
```

マルチ言語表の問合せ

問合せ時に、マルチレクサー・プリファレンスはセッションの言語設定を調べ、その言語に対してサブレクサー・プリファレンスを使用して問合せを解析します。言語が設定されていない場合は、デフォルト・レクサーが使用されます。

言語が設定されている場合は、問合せが解析され、通常に実行されます。索引には複数言語からのトークンが含まれているため、このような問合せはドキュメントを複数の言語で戻すことができます。問合せを特定の言語に制限する場合は、言語列に構造化句を使用します。

CHINESE_VGRAM_LEXER

CHINESE_VGRAM_LEXER 型は、テキスト索引作成用の中国語テキストのトークンを識別します。属性はありません。

データベース・キャラクタ・セットが次のいずれかである場合は、このレクサーを使用できます。

- ZHS16CGB231280
- ZHS16GBK
- ZHT32EUC
- ZHT16BIG5
- ZHT32TRIS
- ZHT16MSWIN950
- ZHT16HKSCS
- UTF8

CHINESE_LEXER

CHINESE_LEXER 型は、テキスト索引作成用の中国語（繁体字および簡体字）テキストのトークンを識別します。属性はありません。

このレクサーは、CHINESE_VGRAM_LEXER と比較して次の利点があります。

- 小型の索引の生成
- 問合せ応答時間の短縮
- 実ワード・トークンの生成による問合せ精度の向上
- ストップワードのサポート

CHINESE_LEXER は新しいアルゴリズムを使用してトークンを生成するため、索引付けに要する時間は、CHINESE_VGRAM_LEXER を使用した場合より長くなります。

データベース・キャラクタ・セットが、Oracle でサポートしている中国語キャラクタ・セットまたは Unicode キャラクタ・セットである場合は、このレクサーを使用できます。

JAPANESE_VGRAM_LEXER

JAPANESE_VGRAM_LEXER 型は、テキスト索引作成用の日本語のトークンを識別します。属性はありません。

データベース・キャラクタ・セットが次のいずれかである場合は、このレクサーを使用できます。

- JA16SJIS
- JA16EUC
- UTF8

JAPANESE_LEXER

JAPANESE_LEXER 型は、テキスト索引作成用の日本語のトークンを識別します。属性はありません。

このレクサーは、JAPANESE_VGRAM_LEXER と比較して次の利点があります。

- 小型の索引の生成
- 問合せ応答時間の短縮
- 実ワード・トークンの生成による問合せ精度の向上

JAPANESE_LEXER は新しいアルゴリズムを使用してトークンを生成するため、索引付けに要する時間は、JAPANESE_VGRAM_LEXER を使用した場合より長くなります。

JAPANESE_LEXER は、次のキャラクタ・セットをサポートしています。

- JA16SJIS
- JA16EUC
- UTF8

日本語レクサーの例

テキスト索引の作成用に JAPANESE_LEXER を指定すると、JAPANESE_LEXER によって、文はワードに変換されます。

たとえば、次の複合語（自然言語処理）があるとします。

'自然言語処理'

このワードは、次の 3 つのトークンとして索引付けされます。

'自然','言語','処理'

文をワードに変換するために、内部ディクショナリが参照されます。内部ディクショナリでワードが検索できない場合、Oracle では、JAPANESE_VGRAM_LEXER を使用して変換します。

KOREAN_LEXER

KOREAN_LEXER 型は、テキスト索引作成用の韓国語テキストのトークンを識別します。

> **注意：** このレクサーは、この韓国語レクサーのみをサポートする旧バージョンの Oracle Text に対する下位互換用にサポートされています。新しいアプリケーションを構築する場合は、KOREAN_MORPH_LEXER の使用をお薦めします。

データベース・キャラクタ・セットが次のいずれかである場合は、このレクサーを使用できます。

- KO16KSC5601
- UTF8

KOREAN_LEXER を使用するときに、次のブール属性を指定できます。

属性	属性値
verb	動詞の索引付けに対して、TRUE または FALSE を指定します。デフォルトは TRUE です。
adjective	形容詞の索引付けに対して、TRUE または FALSE を指定します。デフォルトは TRUE です。
adverb	副詞の索引付けに対して、TRUE または FALSE を指定します。デフォルトは TRUE です。
onechar	1 つの文字の索引付けに対して、TRUE または FALSE を指定します。デフォルトは TRUE です。
number	数の索引付けに対して、TRUE または FALSE を指定します。デフォルトは TRUE です。
udic	ユーザー・ディクショナリの索引付けに対して、TRUE または FALSE を指定します。デフォルトは TRUE です。
xdic	X ユーザー・ディクショナリの索引付けに対して、TRUE または FALSE を指定します。デフォルトは TRUE です。
composite	コンポジット・ワードの索引付けに対して、TRUE または FALSE を指定します。

属性	属性値
morpheme	語形分析に対して、TRUE または FALSE を指定します。デフォルトは TRUE です。
toupper	英語の大文字への変換に対して、TRUE または FALSE を指定します。デフォルトは TRUE です。
tohangeul	ハンジャのハングルへの変換に対して、TRUE または FALSE を指定します。デフォルトは TRUE です。

制限事項

韓国語レクサーでは、文セクションおよび段落セクションはサポートされていません。

KOREAN_MORPH_LEXER

KOREAN_MORPH_LEXER 型は、Oracle Text の索引作成用の韓国語テキストのトークンを識別します。KOREAN_MORPH_LEXER レクサーは、KOREAN_LEXER と比較して次の利点があります。

- 韓国語テキストの語形分析機能の向上
- 索引付けの高速化
- 索引サイズの小型化
- 問合せ検索精度の向上

提供されるディクショナリ

KOREAN_MORPH_LEXER は次の 4 つのディクショナリを使用します。

ディクショナリ	ファイル
システム	$ORACLE_HOME/ctx/data/kolx/drk2sdic.dat
文法	$ORACLE_HOME/ctx/data/kolx/drk2gram.dat
ストップワード	$ORACLE_HOME/ctx/data/kolx/drk2xdic.dat
ユーザー定義	$ORACLE_HOME/ctx/data/kolx/drk2udic.dat

文法、ユーザー定義およびストップワードの各ディクショナリは、テキスト・フォーマット KSC 5601 です。これらのディクショナリは定義済みのルールを使用して変更できます。システム・ディクショナリは変更できません。

未登録のワードは、ユーザー定義ディクショナリ・ファイルに追加できます。新しいワードを指定するルールは、このファイルに含まれています。

サポートされているキャラクタ・セット

データベース・キャラクタ・セットが次のいずれかである場合は、KOREAN_MORPH_LEXER を使用できます。

- KO16KSC5601
- UTF8

属性

KOREAN_MORPH_LEXER を使用するときに、次の属性を指定できます。

属性	属性値
verb_adjective	動詞と形容詞の索引付けに対して、TRUE または FALSE を指定します。デフォルトは FALSE です。
one_char_word	1 つの音節の索引付けに対して、TRUE または FALSE を指定します。デフォルトは FALSE です。
number	数の索引付けに対して、TRUE または FALSE を指定します。デフォルトは FALSE です。
user_dic	ユーザー・ディクショナリの索引付けに対して、TRUE または FALSE を指定します。デフォルトは TRUE です。
stop_dic	ストップワード・ディクショナリの使用に対して、TRUE または FALSE を指定します。デフォルトは TRUE です。ストップワード・ディクショナリは、KOREAN_MORPH_LEXER に属します。
composite	コンポジット名詞の索引付けスタイルを指定します。 コンポジット名詞のみを索引付けするには、COMPOSITE_ONLY を指定します。 コンポジット名詞の名詞コンポーネントをすべて索引付けするには、NGRAM を指定します。 コンポジット名詞自体の他にコンポジット名詞の単一名詞コンポーネントを索引付けするには、COMPONENT_WORD を指定します。デフォルトは COMPONENT_WORD です。 NGRAM と COMPONENT_WORD の相違点については、例を参照してください。
morpheme	語形分析に対して、TRUE または FALSE を指定します。FALSE を設定すると、ドキュメント内の空白などのデリミタで分割されたワードからトークンが作成されます。デフォルトは TRUE です。
to_upper	英語の大文字への変換に対して、TRUE または FALSE を指定します。デフォルトは TRUE です。

属性	属性値
hanja	ハンジャ文字を索引付けするには、TRUE を指定します。FALSE に設定すると、ハンジャ文字はハングル文字に変換されます。デフォルトは FALSE です。
long_word	韓国語の 16 音節を超える長いワードを索引付けするには、TRUE を指定します。デフォルトは FALSE です。
japanese	KSC5601 コードの日本語の文字を索引付けするには、TRUE を指定します。デフォルトは FALSE です。
english	英数字文字列を索引付けするには、TRUE を指定します。デフォルトは TRUE です。

制限事項

韓国語レクサーでは、文セクションおよび段落セクションはサポートされていません。

KOREAN_MORPH_LEXER の例 : Composite 属性の設定

composite 属性を使用すると、コンポジット名詞の索引付け方法を制御できます。

NGRAM の例 composite 属性に対して NGRAM を指定すると、コンポジット名詞は、すべてのコンポーネント・トークンとともに索引付けされます。たとえば、次のようなコンポジット名詞（情報処理規則）があるとします。

'정보처리학회',

この名詞は、次の 6 つのトークンとして索引付けされます。

'정보', '처리', '학회', '정보처리',

'처리학회', '정보처리학회'

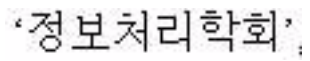
NGRAM を次のように指定して索引付けできます。

```
begin
ctx_ddl.create_preference('korean_lexer','KOREAN_MORPH_LEXER');
ctx_ddl.set_attribute('korean_lexer','COMPOSITE','NGRAM');
end
```

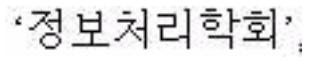
索引を作成するには、次のようにします。

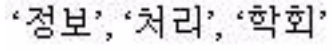
```
create index koreanx on korean(text) indextype is ctxsys.context
parameters ('lexer korean_lexer');
```

COMPONENT_WORD の例 composite 属性に対して COMPONENT_WORD を指定すると、コンポジット名詞およびそのコンポーネントが索引付けされます。たとえば、次のようなコンポジット名詞（情報処理規則）があるとします。



この名詞は、次の 4 つのトークンとして索引付けされます。





COMPONENT_WORD を次のように指定して索引付けできます。

```
begin
ctx_ddl.create_preference('korean_lexer','KOREAN_MORPH_LEXER');
ctx_ddl.set_attribute('korean_lexer','COMPOSITE','COMPONENT_WORD');
end
```

索引を作成するには、次のようにします。

```
create index koreanx on korean(text) indextype is ctxsys.context
parameters ('lexer korean_lexer');
```

USER_LEXER

USER_LEXER を使用して、ユーザーの言語に固有のレクサー処理ソリューションをプラグインします。USER_LEXER によって、Oracle Text でサポートされていない言語のレクサーを定義できます。また、サポートされている言語のレクサーがユーザーのアプリケーションに不適当な場合は、新しいレクサーを定義できます。

ユーザー定義レクサーを Oracle Text に登録するには、次の 2 つのルーチンの指定が必要です。

ユーザー定義ルーチン	説明
索引付けプロシージャ	ドキュメントとストップワードのトークン化を行うストアド・プロシージャ（PL/SQL）出力は、この項に記載されているとおりの XML ドキュメントであることが必要です。
問合せプロシージャ	問合せワードのトークン化を行うストアド・プロシージャ（PL/SQL）出力は、この項に記載されているとおりの XML ドキュメントであることが必要です。

制限事項

USER_LEXER では、次の機能はサポートされていません。

- CTX_DOC.GIST および CTX_DOC.THEMES
- CTX_QUERY.HFEEDBACK
- ABOUT 問合せ演算子
- CTXRULE 索引タイプ
- VGRAM 索引付けアルゴリズム

USER_LEXER 属性

USER_LEXER には、次の属性があります。

属性	サポートされる値
INDEX_PROCEDURE	ストアド・プロシージャの名前。デフォルトはありません。
INPUT_TYPE	VARCHAR2 または CLOB。デフォルトは CLOB です。
QUERY_PROCEDURE	ストアド・プロシージャの名前。デフォルトはありません。

INDEX_PROCEDURE

このコールバック・ストアド・プロシージャは、ドキュメントまたはストップリスト・オブジェクトにあるストップワードのトークン化が必要になると、Oracle Text によってコールされます。

要件 このプロシージャには、PL/SQL ストアド・プロシージャを指定できます。

CTXSYS ユーザーがこのストアド・プロシージャを所有し、索引所有者がこのストアド・プロシージャに対する実行権限を持っている必要があります。

このストアド・プロシージャは、索引の作成後に置換したり削除することはできません。索引の削除後に、このストアド・プロシージャを置換または削除することはできます。

パラメータ ユーザー定義レクサーの索引付けプロシージャでは、次の 2 つのインタフェースがサポートされています。

- VARCHAR2 インタフェース
- CLOB インタフェース

制限事項 このプロシージャは、次のいずれの操作も実行できません。

- ロールバック
- 現行トランザクションの明示的または暗黙的なコミット
- その他のトランザクション制御文の発行
- セッション言語または地域の変更

XML ドキュメントのルート要素トークンの子要素は、トークン化されるドキュメントまたはストップワードでトークンが出現した順序で戻される必要があります。

このストアド・プロシージャの動作は、すべてのパラメータに対して決定的であることが必要です。

INPUT_TYPE

ユーザー定義レクサーの索引付けプロシージャでは、2 つのインタフェースがサポートされています。1 つのインタフェースは、ドキュメントまたはストップワードと、XML としてコード化された対応するトークンを VARCHAR2 データ型で渡し、もう 1 つのインタフェースは CLOB データ型を使用します。この属性は、INDEX_PROCEDURE 属性で指定されたストアド・プロシージャが実装するインタフェースを示します。

VARCHAR2 インタフェース 表 2-1 では、トークン化するドキュメントまたはストップリスト・オブジェクトのストップワードを VARCHAR2 で Oracle Text からストアド・プロシージャに渡し、同様に、トークンを VARCHAR2 でストアド・プロシージャから Oracle Text に渡すインタフェースについて説明します。

ユーザー定義レクサーの索引付けプロシージャでは、索引付けする列の全ドキュメントが 32512 バイト以下で、そのトークンが 32512 バイト以下で表される場合は、このインタフェースを使用してください。この場合、表 2-2 に示した CLOB インタフェースも使用できますが、通常、VARCHAR2 インタフェースは CLOB インタフェースよりも高速に実行されます。

このプロシージャは、次のパラメータを使用して定義する必要があります。

表 2-1 INDEX_PROCEDURE の VARCHAR2 インタフェース

パラメータ順序	パラメータ・モード	パラメータのデータ型	説明
1	IN	VARCHAR2	トークン化するドキュメントまたはストップリスト・オブジェクトのストップワード。 ドキュメントが 32512 バイトを超える場合は、ドキュメント・レベルの索引付けエラーがレポートされます。
2	IN OUT	VARCHAR2	XML としてコード化されたトークン。 ドキュメントにトークンが含まれていない場合は、NULL を戻すか、または戻される XML ドキュメントのトークン要素に子要素を含めないでください。 データのバイト長は、32512 以下にしてください。 パフォーマンスを改善するには、このパラメータの宣言時に NOCOPY ヒントを使用してください。NOCOPY ヒントによって、データは値ではなく参照によって渡されます。 このプロシージャが戻す XML ドキュメントには不要な whitespace 文字（通常、可読性の向上のために使用する）を含めないでください。これによって、XML ドキュメントのサイズが縮小され、結果として送信時間が短縮されます。 パフォーマンス向上のために、index_procedure の実行時に、対応する XML Schema に対する XML ドキュメントの妥当性はチェックされません このパラメータは、パフォーマンスの目的で IN OUT になっています。ストアド・プロシージャで IN の値を指定する必要はありません。

表 2-1 INDEX_PROCEDURE の VARCHAR2 インタフェース（続き）

パラメータ順序	パラメータ・モード	パラメータのデータ型	説明
3	IN	BOOLEAN	位置情報が必要かどうか。 トークン化するドキュメントで検出されたトークンの文字オフセットと文字長が Oracle Text で必要な場合、このパラメータは TRUE に設定されます。 トークン化するドキュメントで検出されたトークンの文字オフセットと文字長が Oracle Text で不要な場合、このパラメータは FALSE に設定されます。これは、XML 属性の off と len を使用できないことを意味します。

CLOB インタフェース 表 2-2 では、トークン化するドキュメントまたはストップリスト・オブジェクトのストップワードを CLOB で Oracle Text からストアド・プロシージャに渡し、同様に、トークンを CLOB でストアド・プロシージャから Oracle Text に渡す CLOB インタフェースについて説明します。

ユーザー定義レクサーの索引付けプロシージャでは、索引付けする列の 1 つ以上のドキュメントが 32512 バイトを超える場合、または対応するトークンが 32512 バイトを超えるバイト数で表される場合は、このインタフェースを使用してください。

表 2-2 INDEX_PROCEDURE の CLOB インタフェース

パラメータ順序	パラメータ・モード	パラメータのデータ型	説明
1	IN	CLOB	表 2-1「INDEX_PROCEDURE の VARCHAR2 インタフェース」と同じです。
2	IN OUT	CLOB	表 2-1「INDEX_PROCEDURE の VARCHAR2 インタフェース」と同じです。 IN の値は、常に切り捨てられた CLOB となります。
3	IN	BOOLEAN	表 2-1「INDEX_PROCEDURE の VARCHAR2 インタフェース」と同じです。

1 番目と 2 番目のパラメータは一時 CLOB です。これらの CLOB ロケータを他のロケータ変数に割り当てないでください。仮パラメータの CLOB ロケータを他のロケータ変数に割り当てると、一時 CLOB の新しいコピーが作成され、パフォーマンスに影響を与えます。

QUERY_PROCEDURE

このコールバック・ストアド・プロシージャは、問合せ内のワードのトークン化が必要になると、Oracle Text によってコールされます。問合せ内の空白で区切られた文字のグループ（問合せ演算子を除く）は、ワードとして識別されます。

要件 このプロシージャには、PL/SQL ストアド・プロシージャを指定できます。

CTXSYS ユーザーがこのストアド・プロシージャを所有し、索引所有者がこのストアド・プロシージャに対する実行権限を持っている必要があります。

このストアド・プロシージャは、索引の作成後に置換したり削除することはできません。索引の削除後に、このストアド・プロシージャを置換または削除することはできます。

制限事項 このプロシージャは、次のいずれの操作も実行できません。

- ロールバック
- 現行トランザクションの明示的または暗黙的なコミット
- その他のトランザクション制御文の発行
- セッション言語または地域の変更

XML ドキュメントのルート要素トークンの子要素は、トークン化される問合せワードでトークンが出現した順序で戻される必要があります。

このストアド・プロシージャの動作は、すべてのパラメータに対して決定的であることが必要です。

パラメータ 表 2-3 では、ユーザー定義レクサーの問合せプロシージャのインタフェースについて説明します。

表 2-3

パラメータ順序	パラメータ・モード	パラメータのデータ型	説明
1	IN	VARCHAR2	トークン化する問合せワード。
2	IN	CTX_ULEXER.WILDCARD_TAB	問合せワード内のワイルド・カード文字（% および _）の文字オフセット。Oracle Text によって渡された問合せワードにワイルド・カード文字が含まれていない場合、この索引付き表は空になります。 問合せワード内のワイルド・カード文字は、ワイルド・カード問合せ機能が正しく実行されるように、戻されるトークンに保持する必要があります。 文字オフセットは 0（ゼロ）を基準にしています。
3	IN OUT	VARCHAR2	XML としてコード化されたトークン。 問合せワードにトークンが含まれていない場合は、NULL を戻すか、または戻される XML ドキュメントのトークン要素に子要素を含めないでください。 データの長さは、32512 バイト以下であることが必要です。

トークンの XML としてのコード化

ユーザーのストアド・プロシージャが戻すトークンの順序は、XML 1.0 ドキュメントで表す必要があります。この XML ドキュメントは、次の各項で示す XML Schema に対して妥当であることが必要です。

- Location XML Schema を使用しないユーザー定義索引付けプロシージャ
- Location XML Schema を使用したユーザー定義索引付けプロシージャ
- ユーザー定義レクサーの問合せプロシージャ

制限事項 この機能のパフォーマンスを向上させるために、Oracle Text の XML パーサーは妥当性チェックを実行しません。また、XML の機能に完全に適合する正規のパーサーではありません。つまり、最小限の XML 機能がサポートされます。次の XML 機能はサポートされません。

- ドキュメント・タイプ宣言（<!DOCTYPE [...]>）およびエンティティ宣言。lt、gt、amp、quot および apos の組込みエンティティのみ参照できます。
- CDATA セクション
- コメント
- 処理命令
- XML 宣言（<?xml version="1.0" ...?>）
- 名前空間
- 対応する XML Schema 以外で定義された要素および属性の使用
- 文字参照（例 : ট）
- xml:space 属性
- xml:lang 属性

Location XML Schema を使用しないユーザー定義索引付けプロシージャ

この項では、3 番目のパラメータが FALSE の場合に、ユーザー定義レクサーの索引付けプロシージャが戻す XML ドキュメントに適用される追加の制約について説明します。戻される XML ドキュメントは、次の XML Schema に対して妥当であることが必要です。

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="tokens">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="eos" type="EmptyTokenType"/>
          <xsd:element name="eop" type="EmptyTokenType"/>
          <xsd:element name="num" type="xsd:token"/>
          <xsd:group ref="IndexCompositeGroup"/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <!--
  Enforce constraint that compMem element must be preceeded by word element
  or compMem element for indexing
  -->
  <xsd:group name="IndexCompositeGroup">
    <xsd:sequence>
      <xsd:element name="word" type="xsd:token"/>
      <xsd:element name="compMem" type="xsd:token" minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:group>

  <!-- EmptyTokenType defines an empty element without attributes -->
  <xsd:complexType name="EmptyTokenType"/>

</xsd:schema>
```

次に、この XML Schema によって適用される制約をいくつか説明します。

- ルート要素はトークンで、指定は必須です。属性はありません。
- ルート要素には、子要素を指定しないことも、1 つ以上の子要素を指定することも可能です。子要素には、eos、eop、num、word および compMem のいずれかを指定できます。これらの子要素は、特定のトークン型を表します。
- compMem 要素は、word 要素または compMem 要素の後に指定する必要があります。
- eos 要素と eop 要素に属性はありません。これらは空要素であることが必要です。
- num、word および compMem 要素に属性はありません。これらの要素の内容は、Oracle Text によって正規化されます。whitespace 文字の空白文字への変換、隣接する複数の空白文字の単一空白文字への縮小、先頭および末尾の空白の削除、エンティティ参照の置換、および 64 バイトへの切捨てが行われます。

表 2-4 では、前述の XML Schema に定義されている要素名について説明します。

表 2-4 要素名

要素	説明
word	単純なワード・トークンを表します。この要素の内容はワード自体です。このトークンがストップワードまたは非ストップワードとして識別され、これに対応した処理が行われることはありません。
num	数値トークンを表します。この要素の内容は数値自体です。ストップリスト・プリファレンスによって NUMBERS がストップクラスとして追加されている場合、このトークンはストップワードとして処理されます。それ以外の場合、このトークンはワード・トークンと同様に処理されます。 このトークン型のサポートはオプションです。このトークン型をサポートしない場合、NUMERBS ストップクラスの追加による影響はありません。
eos	文の終わりを表すトークンです。この情報は、文内の問合せをサポートするために使用されます。 このトークン型のサポートはオプションです。このトークン型をサポートしない場合、SENTENCE セクションに対する問合せは期待どおりに実行されません。
eop	段落の終わりを表すトークンです。この情報は、段落内の問合せをサポートするために使用されます。 このトークン型のサポートはオプションです。このトークン型をサポートしない場合、PARAGRAPH セクションに対する問合せは期待どおりに実行されません。

表 2-4 要素名（続き）

要素	説明
compMem	前の順序にあるワード・トークンがコンポジット・トークンであり、この compMem トークンがそのワード・トークンの構成要素であることを示します。 このトークン型のサポートはオプションです。

例 ドキュメント : Vom Nordhauptbahnhof und aus der Innenstadt zum Messegelände

トークン :

```
<tokens>
  <word> VOM </word>
  <word> NORDHAUPTBAHNHOF </word>
  <compMem> HAUPT </compMem>
  <compMem> HAUPTBAHNHOF </compMem>
  <compMem> NORD </compMem>
  <word> UND </word>
  <word> AUS </word>
  <word> DER </word>
  <word> INNENSTADT </word>
  <word> ZUM </word>
  <word> MESSEGELÄNDE </word>
  <eos/>
</tokens>
```

例 ドキュメント : Oracle9*i* Release 2

トークン :

```
<tokens>
  <word> ORACLE9I</word>
  <word> RELEASE </word>
  <num> 2 </num>
</tokens>
```

例 ドキュメント : WHERE salary<25000.00 AND job = 'F&B Manager'

トークン :

```
<tokens>
  <word> WHERE </word>
  <word> salary&lt;2500.00 </word>
  <word> AND </word>
  <word> job </word>
  <word> F&amp;B </word>
  <word> Manager </word>
</tokens>
```

Location XML Schema を使用したユーザー定義索引付けプロシージャ

この項では、3 番目のパラメータが TRUE の場合に、ユーザー定義レクサーの索引付けプロシージャが戻す XML ドキュメントに適用される追加の制約について説明します。戻される XML ドキュメントは、次の XML Schema に対して妥当であることが必要です。

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="tokens">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="eos" type="EmptyTokenType"/>
          <xsd:element name="eop" type="EmptyTokenType"/>
          <xsd:element name="num" type="DocServiceTokenType"/>
          <xsd:group ref="DocServiceCompositeGroup"/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <!--
  Enforce constraint that compMem element must be preceeded by word element
  or compMem element for document service
  -->
  <xsd:group name="DocServiceCompositeGroup">
    <xsd:sequence>
      <xsd:element name="word" type="DocServiceTokenType"/>
      <xsd:element name="compMem" type="DocServiceTokenType"/>
    </xsd:sequence>
  </xsd:group>

  <!-- EmptyTokenType defines an empty element without attributes -->
  <xsd:complexType name="EmptyTokenType"/>
```

```
  <!--
  DocServiceTokenType defines an element with content and mandatory attributes
  -->
  <xsd:complexType name="DocServiceTokenType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:token">
        <xsd:attribute name="off" type="OffsetType" use="required"/>
        <xsd:attribute name="len" type="xsd:unsignedShort" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>

  <xsd:simpleType name="OffsetType">
    <xsd:restriction base="xsd:unsignedInt">
      <xsd:maxInclusive value="2147483647"/>
    </xsd:restriction>
  </xsd:simpleType>

</xsd:schema>
```

この XML Schema によって適用される制約は次のとおりです。

- ルート要素はトークンで、指定は必須です。属性はありません。
- ルート要素には、子要素を指定しないことも、1 つ以上の子要素を指定することも可能です。子要素には、eos、eop、num、word および compMem のいずれかを指定できます。これらの子要素は、特定のトークン型を表します。
- compMem 要素は、word 要素または compMem 要素の後に指定する必要があります。
- eos 要素と eop 要素に属性はありません。これらは空要素であることが必要です。
- num、word および compMem 要素には、必須属性の `off` と `len` があります。これらの要素の内容は、Oracle Text によって正規化されます。whitespace 文字の空白文字への変換、隣接する複数の空白文字の単一空白文字への縮小、先頭および末尾の空白の削除、エンティティ参照の置換、および 64 バイトへの切捨てが行われます。
- `off` 属性の値には 0（ゼロ）～ 2147483647 の整数を指定する必要があります。
- `len` 属性の値には 0（ゼロ）～ 65535 の整数を指定する必要があります。

表 2-4「要素名」では、前述の XML Schema に定義されている要素型について説明しています。

表 2-5「属性」では、前述の XML Schema に定義されている属性について説明します。

表 2-5 属性

属性	説明
off	トークン化するドキュメントに出現するトークンの文字オフセットを表します（SQL 機能の LENGTH のセマンティクスと同じです）。 このオフセットは、データストアでフェッチされたドキュメントではなく、ユーザー定義レクサーの索引付けプロシージャに渡された文字ドキュメントに対するものです。データストアでフェッチされたドキュメントは、フィルタ・オブジェクトまたはセクション・グループ（あるいはその両方）で事前処理されてから、ユーザー定義レクサーの索引付けプロシージャに渡される場合があります。 トークン化するドキュメントの最初の文字のオフセットは 0（ゼロ）です。
len	トークン化するドキュメントに出現するトークンの文字長を表します（SQL 機能の LENGTH のセマンティクスと同じです）。 この長さは、データストアでフェッチされたドキュメントではなく、ユーザー定義レクサーの索引付けプロシージャに渡された文字ドキュメントに対するものです。データストアでフェッチされたドキュメントは、フィルタ・オブジェクトまたはセクション・グループで事前処理されてから、ユーザー定義レクサーの索引付けプロシージャに渡される場合があります。

`off` 属性の値と `len` 属性の値の合計は、トークン化するドキュメントの合計文字数以下である必要があります。これによって、ドキュメントのオフセットと参照される文字がドキュメント境界内に保持されます。

例

ドキュメント : User-defined Lexer

トークン :

```
<tokens>
  <word off="0" len="4"> USE </word>
  <word off="5" len="7"> DEF </word>
  <word off="13" len="5"> LEX </word>
  <eos/>
</tokens>
```

ユーザー定義レクサーの問合せプロシージャ

この項では、ユーザー定義レクサーの問合せプロシージャが戻す XML ドキュメントに適用される追加の制約について説明します。戻される XML ドキュメントは、次の XML Schema に対して妥当であることが必要です。

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="tokens">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="num" type="QueryTokenType"/>
          <xsd:group ref="QueryCompositeGroup"/>
        </xsd:choice>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <!--
  Enforce constraint that compMem element must be preceeded by word element
  or compMem element for query
  -->
  <xsd:group name="QueryCompositeGroup">
    <xsd:sequence>
      <xsd:element name="word" type="QueryTokenType"/>
      <xsd:element name="compMem" type="QueryTokenType"/>
    </xsd:sequence>
  </xsd:group>

  <!--
  QueryTokenType defines an element with content and with an optional attribute
  -->
  <xsd:complexType name="QueryTokenType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:token">
        <xsd:attribute name="wildcard" type="WildcardType" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>

  <xsd:simpleType name="WildcardType">
    <xsd:restriction base="WildcardBaseType">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="64"/>
    </xsd:restriction>
  </xsd:simpleType>
```

```
  <xsd:simpleType name="WildcardBaseType">
    <xsd:list>
      <xsd:simpleType>
        <xsd:restriction base="xsd:unsignedShort">
          <xsd:maxInclusive value="378"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:list>
  </xsd:simpleType>

</xsd:schema>
```

次に、この XML Schema によって適用される制約をいくつか説明します。

- ルート要素はトークンで、指定は必須です。属性はありません。
- ルート要素には、子要素を指定しないことも、1 つ以上の子要素を指定することも可能です。子要素には、num、word および compMem のいずれかを指定できます。これらの子要素は、特定のトークン型を表します。
- compMem 要素は、word 要素または compMem 要素の後に指定する必要があります。
- num、word および compMem 要素には、オプション属性の wildcard があります。これらの要素の内容は、Oracle Text によって正規化されます。whitespace 文字の空白文字への変換、隣接する複数の空白文字の単一空白文字への縮小、先頭および末尾の空白の削除、エンティティ参照の置換、および 64 バイトへの切捨てが行われます。
- wildcard 属性の値は、空白で区切られた整数のリストです 整数の最小数は 1 で、最大数は 64 です。0（ゼロ）～ 378 の値を指定する必要があります。リスト内の構成は、任意の順序にできます。

表 2-4「要素名」では、前述の XML Schema に定義されている要素型について説明しています。

表 2-6「XML Schema の属性 : 問合せプロシージャ」では、前述の XML Schema に定義されている属性について説明します。

表 2-6 XML Schema の属性 : 問合せプロシージャ

属性	説明
wildcard	問合せ内の % 文字または _ 文字は、ユーザーがエスケープしないかぎり、他の文字で置換されるワイルド・カード文字とみなされます。問合せ内のワイルド・カード文字は、ワイルド・カード問合せ機能が正しく実行されるように、トークン化時に保持する必要があります。この属性は、要素の内容に含まれるワイルド・カード文字の文字オフセットを表します（SQL 機能の LENGTH のセマンティクスと同じです)。これらのオフセットは、要素の内容に対して実行される正規化に応じて調整されます。オフセットは、% 文字または _ 文字のいずれかを指し示す必要があります。 要素の内容に含まれる最初の文字のオフセットは 0（ゼロ）です。 トークンにワイルド・カード文字が含まれていない場合、この属性は指定できません。

例

問合せワード : pseudo-%morph%

トークン :

```
<tokens>
  <word> PSEUDO </word>
  <word wildcard="1 7"> %MORPH% </word>
</tokens>
```

例

```
Query word: <%>
Tokens:
<tokens>
  <word wildcard="5"> &lt;%&gt; </word>
</tokens>
```

ワードリスト型

ワードリスト・プリファレンスによって、使用している言語に対するステミングやファジー・マッチングなどの問合せオプションを使用可能にします。また、ワードリスト・プリファレンスを使用すると、サブストリングやプリフィックス索引付けを使用可能にでき、CONTAINS および CATSEARCH によるワイルド・カード問合せのパフォーマンスが改善されます。

ワードリスト・プリファレンスを作成するには、BASIC_WORDLIST を使用する必要があります。使用できるのはこの型のみです。

BASIC_WORDLIST

BASIC_WORDLIST 型を使用して、テキスト索引に対するステミングおよびファジー・マッチングを使用可能にしたり、プリフィックスの索引を作成することができます。

> **関連項目：** STEM および FUZZY 演算子の詳細は、第 3 章「CONTAINS 問合せ演算子」を参照してください。

BASIC_WORDLIST には、次の属性があります。

表 2-7

属性	属性値
stemmer	使用する言語ステマーを指定します。次のうちのいずれかを指定できます。 NULL（ステミングなし） ENGLISH（英語の語形変化） DERIVATIONAL（英語の派生語） DUTCH FRENCH GERMAN ITALIAN SPANISH AUTO（ステミングに対する自動言語識別）

表 2-7（続き）

属性	属性値
fuzzy_match	使用するファジー・マッチング・クラスタを指定します。次のうちのいずれかを指定できます。 GENERIC JAPANESE_VGRAM KOREAN CHINESE_VGRAM ENGLISH DUTCH FRENCH GERMAN ITALIAN SPANISH OCR AUTO（ステミングに対する自動言語識別）
fuzzy_score	デフォルトのファジー・スコアの下限を指定します。0 ～ 80 の数を指定します。この数値未満のスコアを含むテキストは戻されません。デフォルトは 60 です。
fuzzy_numresults	ファジー拡張の最大数を指定します。0 ～ 5,000 の数を指定します。デフォルトは 100 です。
substring_index	TRUE を指定すると、サブストリング索引が作成されます。サブストリング索引を使用すると、*%ing* や *%benz%* のような左側切捨ておよび左右切捨てのワイルド・カード問合せが改善されます。デフォルトは FALSE です。
prefix_index	TRUE を指定すると、プリフィックス索引付けが使用可能になります。プリフィックス索引付けを行うと、TO% のような右側切捨てのワイルド・カード検索のパフォーマンスが改善されます。デフォルトは FALSE です。
prefix_length_min	索引付けされたプリフィックスの最小長を指定します。デフォルトは 1 です。
prefix_length_max	索引付けされたプリフィックスの最大長を指定します。デフォルトは 64 です。
wlidcard_maxterms	ワイルド・カード拡張での語句の最大数を指定します。1 ～ 15,000 の数を指定します。デフォルトは 5,000 です。

stemmer
テキスト問合せでワード・ステミングに使用するステマーを指定します。ステマーに対して値を指定しない場合、デフォルトは ENGLISH です。

AUTO を指定すると、セッションの言語設定に従って、システムが自動的にステミング言語を設定します。言語に対してステマーが存在しない場合、デフォルトは NULL です。NULL ステマーを使用すると、STEM 演算子は問合せで無視されます。

fuzzy_match
列に使用するファジー・マッチング・ルーチンを指定します。ファジー・マッチングは、現在、英語、日本語および一部のヨーロッパ言語に対してサポートされています。

> **注意：** 中国語および韓国語の fuzzy_match 属性値は、英語および日本語のファジー・マッチング・ルーチンが、中国語および韓国語のテキストに使用されないようにするためのダミー属性値です。

fuzzy_match のデフォルトは GENERIC です。

AUTO を指定すると、セッションの言語設定に従って、システムが自動的にファジー・マッチング言語を設定します。

fuzzy_score
デフォルトのファジー・スコアの下限を指定します。0 ～ 80 の数を指定します。この数値未満のスコアを含むテキストは戻されません。デフォルトは 60 です。

ファジー・スコアは、拡張されたワードが問合せワードにどれだけ近いかという計測値です。スコアが高いほど適合する度合いが高くなります。このパラメータを使用して、ファジー拡張を最も適合する度合いに制限できます。

fuzzy_numresults
ファジー拡張の最大数を指定します。0 ～ 5000 の数を指定します。デフォルトは 100 です。

ファジー拡張を設定することによって、拡張を最も適合するワードの指定値に制限できます。

substring_index
TRUE を指定すると、サブストリング索引が作成されます。サブストリング索引を使用すると、*%ing* や *%benz%* のような左側切捨てまたは左右切捨てのワイルド・カード問合せが改善されます。デフォルトは FALSE です。

サブストリングの索引付けによって、索引付け処理およびディスク・リソースに対して、次のような影響があります。

- 索引作成および DML 処理が、最大 4 倍遅くなります。
- 作成するサブストリング索引のサイズは、ワード表上の $X 索引とほぼ同じサイズになります。
- substring_index を使用可能にして索引を作成すると、サブストリング索引をオフにして作成するよりも、索引のフラッシュ時により多くのロールバック・セグメントが必要になります。サブストリング索引の作成時に、次のいずれかを実行することをお薦めします。
  - 通常の 2 倍のロールバックを使用可能にします。
  - ディスクに対する索引フラッシュのサイズを縮小するために索引メモリーを減らします。

prefix_index
yes を指定すると、プリフィックス索引付けが使用可能になります。プリフィックス索引付けを行うと、*TO%* のような右側切捨てのワイルド・カード検索のパフォーマンスが改善されます。デフォルトは NO です。

> **注意：** プリフィックス索引付けを行うと、索引サイズが増加します。

プリフィックス索引付けでは、トークンが複数のプリフィックスに切断され、$I 表に格納されます。たとえば、TOKEN や TOY は、$I 表で通常次のように索引付けされます。

トークン	型	情報
TOKEN	0	DOCID 1 POS 1
TOY	0	DOCID 1 POS 3

プリフィックス索引付けを使用すると、これらのトークンのプリフィックス・サブストリングが、6 というトークン型で索引付けされます。

トークン	型	情報
TOKEN	0	DOCID 1 POS 1
TOY	0	DOCID 1 POS 3
T	6	DOCID 1 POS 1 POS 3
TO	6	DOCID 1 POS 1 POS 3
TOK	6	DOCID 1 POS 1
TOKE	6	DOCID 1 POS 1
TOKEN	6	DOCID 1 POS 1
TOY	6	DOCID 1 POS 3

TO% のようなワイルド・カード検索が高速化されます。これは、語句の拡張と結果セットのマージが実行されないためです。結果の取得に必要なのは、行（TO、6）の検索のみです。

prefix_length_min
索引付けされたプリフィックスの最小長を指定します。デフォルトは 1 です。

たとえば、prefix_length_min を 3 に設定し、prefix_length_max を 5 に設定すると、3 ～ 5 文字の長さのすべてのプリフィックスが索引付けされます。

> **注意：** ワイルド・カード検索のパターンが最小長未満または最大長を超える場合は、同等化拡張およびマージという速度の遅い方法で検索が実行されます。

prefix_length_max
索引付けされたプリフィックスの最大長を指定します。デフォルトは 64 です。

たとえば、prefix_length_min を 3 に設定し、prefix_length_max を 5 に設定すると、3 ～ 5 文字の長さのすべてのプリフィックスが索引付けされます。

> **注意：** ワイルド・カード検索のパターンが最小長未満または最大長を超える場合は、同等化拡張およびマージという速度の遅い方法で検索が実行されます。

wildcard_maxterms
ワイルド・カード（%）拡張での語句の最大数を指定します。このパラメータを使用して、ワイルド・カード問合せパフォーマンスを受入れ可能な制限内に保持します。ワイルド・カード問合せ拡張がこの制限数を超えると、エラーが戻されます。

BASIC_WORDLIST の例

ファジー・マッチングおよびステミングの使用可能化

次の例では、英語に対するステミングおよびファジー・マッチングを使用可能にします。プリファレンス STEM_FUZZY_PREF によって、拡張数が、許容される最大数に設定されます。また、このプリファレンスによって、左右切捨て検索のパフォーマンスが向上するように、サブストリング索引がシステムで作成されます。

```
begin
  ctx_ddl.create_preference('STEM_FUZZY_PREF', 'BASIC_WORDLIST');
  ctx_ddl.set_attribute('STEM_FUZZY_PREF','FUZZY_MATCH','ENGLISH');
  ctx_ddl.set_attribute('STEM_FUZZY_PREF','FUZZY_SCORE','0');
  ctx_ddl.set_attribute('STEM_FUZZY_PREF','FUZZY_NUMRESULTS','5000');
  ctx_ddl.set_attribute('STEM_FUZZY_PREF','SUBSTRING_INDEX','TRUE');
  ctx_ddl.set_attribute('STEM_FUZZY_PREF','STEMMER','ENGLISH');
end;
```

索引を SQL で作成するには、次の文を発行します。

```
create index fuzzy_stem_subst_idx on mytable ( docs )
  indextype is ctxsys.context parameters ('Wordlist STEM_FUZZY_PREF');
```

サブストリングとプリフィックス索引付けの使用可能化

次の例では、プリフィックスとサブストリングの索引付けに対してワードリスト・プリファレンスを設定します。プリフィックス索引付けに対して、3 ～ 4 文字の長さのトークン・プリフィックスの作成を指定します。

```
begin
    ctx_ddl.create_preference('mywordlist', 'BASIC_WORDLIST');
    ctx_ddl.set_attribute('mywordlist','PREFIX_INDEX','TRUE');
    ctx_ddl.set_attribute('mywordlist','PREFIX_MIN_LENGTH',3);
    ctx_ddl.set_attribute('mywordlist','PREFIX_MAX_LENGTH', 4);
    ctx_ddl.set_attribute('mywordlist','SUBSTRING_INDEX', 'YES');
end
```

ワイルド・カード拡張に対する制限の設定

wildcard_maxterms 属性を使用して、ワイルド・カード拡張で使用できる語句の最大数を設定します。

```
--- create a sample table
drop table quick ;
create table quick
  (
    quick_id number primary key,
    text      varchar(80)
  );

--- insert a row with 10 expansions for 'tire%'
insert into quick ( quick_id, text )
  values ( 1, 'tire tirea tireb tirec tired tiree tiref tireg tireh tirei tirej') ;
commit;

--- create an index using wildcard_maxterms=100
begin
    Ctx_Ddl.Create_Preference('wildcard_pref', 'BASIC_WORDLIST');
    ctx_ddl.set_attribute('wildcard_pref', 'wildcard_maxterms', 100) ;
end;
/
create index wildcard_idx on quick(text)
    indextype is ctxsys.context
    parameters ('Wordlist wildcard_pref') ;

--- query on 'tire%' - should work fine
select quick_id from quick
  where contains ( text, 'tire%' ) > 0;

--- now re-create the index with wildcard_maxterms=5

drop index wildcard_idx ;

begin
    Ctx_Ddl.Drop_Preference('wildcard_pref');
    Ctx_Ddl.Create_Preference('wildcard_pref', 'BASIC_WORDLIST');
    ctx_ddl.set_attribute('wildcard_pref', 'wildcard_maxterms', 5) ;
end;
/

create index wildcard_idx on quick(text)
    indextype is ctxsys.context
    parameters ('Wordlist wildcard_pref') ;
```

```
--- query on 'tire%' gives "wildcard query expansion resulted in too many terms"
select quick_id from quick
  where contains ( text, 'tire%' ) > 0;
```

記憶域型

記憶域プリファレンスを使用して、テキスト索引に対応付けられた表に対して表領域および作成パラメータを指定します。システムには、BASIC_STORAGE という単一の記憶域型があります。

型	説明
BASIC_STORAGE	テキスト索引を構成するデータベース表および索引に対して、表領域および作成パラメータを指定するために使用する索引付けの型

BASIC_STORAGE

BASIC_STORAGE 型は、テキスト索引を構成するデータベース表および索引に対する表領域および作成パラメータを指定します。

指定した句は、索引作成時に内部の CREATE TABLE（i_index _clause の場合は CREATE INDEX）文に追加されます。記憶域、LOB 記憶域、パーティション化などの句を指定できます。ただし、索引構成表句は指定しないでください。

> **関連項目：** CREATE TABLE 文と CREATE INDEX 文の指定方法の詳細は、『Oracle9*i* SQL リファレンス』を参照してください。

BASIC_STORAGE には、次の属性があります。

属性	属性値
i_table_clause	dr$< 索引名 >$I 表作成用のパラメータ句。内部 CREATE TABLE 文の終わりに追加する storage 句および tablespace 句を指定します。 I 表は索引データ表です。
k_table_clause	dr$< 索引名 >$K 表作成用のパラメータ句。内部 CREATE TABLE 文の終わりに追加する storage 句および tablespace 句を指定します。 K 表はキーマップ表です。
r_table_clause	dr$< 索引名 >$R 表作成用のパラメータ句。内部 CREATE TABLE 文の終わりに追加する storage 句および tablespace 句を指定します。 R 表は ROWID 表です。 デフォルト句は、'LOB(DATA) STORE AS (CACHE)' です。

属性	属性値
n_table_clause	dr$< 索引名 >$N 表作成用のパラメータ句。内部 CREATE TABLE 文の終わりに追加する storage 句および tablespace 句を指定します。 N 表はネガティブリスト表です。
i_index_clause	dr$< 索引名 >$X 索引作成用のパラメータ句。内部 CREATE INDEX 文の終わりに追加する storage 句および tablespace 句を指定します。 デフォルト句は、'COMPRESS 2' で、この索引表を圧縮するように Oracle に指示します。 この圧縮によってディスク領域が節約され、問合せパフォーマンスが向上するため、デフォルトをオーバーライドする場合は、パラメータ句に COMPRESS 2 を指定してこの表を圧縮することをお薦めします。
p_table_clause	BASIC_WORDLIST で SUBSTRING_INDEX を使用可能にしている場合の、サブストリング索引用のパラメータ句。 内部 CREATE INDEX 文の終わりに追加する storage 句および tablespace 句を指定します。P 表は索引構成表であるため、指定する storage 句は、このタイプの表に適切である必要があります。

記憶域デフォルト動作

デフォルトでは、BASIC_STORAGE 属性は設定されていません。このような場合は、索引所有者のデフォルト表領域にテキスト索引表が作成されます。ユーザー IUSER によって発行され、BASIC_STORAGE 属性が設定されていない次の文について考えます。

```
create index IOWNER.idx on TOWNER.tab(b) indextype is ctxsys.context;
```

この例では、テキスト索引は IOWNER のデフォルトの表領域に作成されます。

記憶域の例

次の例では、索引表が 1KB の初期エクステントで foo 表領域に作成されるように指定します。

```
begin
ctx_ddl.create_preference('mystore', 'BASIC_STORAGE');
ctx_ddl.set_attribute('mystore', 'I_TABLE_CLAUSE',
                      'tablespace foo storage (initial 1K)');
ctx_ddl.set_attribute('mystore', 'K_TABLE_CLAUSE',
                      'tablespace foo storage (initial 1K)');
ctx_ddl.set_attribute('mystore', 'R_TABLE_CLAUSE',
                      'tablespace foo storage (initial 1K)');
ctx_ddl.set_attribute('mystore', 'N_TABLE_CLAUSE',
                      'tablespace foo storage (initial 1K)');
ctx_ddl.set_attribute('mystore', 'I_INDEX_CLAUSE',
                      'tablespace foo storage (initial 1K)');
```

```
ctx_ddl.set_attribute('mystore', 'P_TABLE_CLAUSE',
                      'tablespace foo storage (initial 1K)');
end;
```

セクション・グループ型

ドキュメント・セクションに対して WITHIN 問合せを発行するには、セクションを定義する前に、セクション・グループを作成する必要があります。セクション・グループは、CREATE INDEX の PARAMETERS 句で指定します。

セクション・グループを作成する場合は、CTX_DDL.CREATE_SECTION_GROUP プロシージャを使用して次のグループ・タイプのいずれかを指定できます。

セクション・グループ・プリファレンス	説明
NULL_SECTION_GROUP	どのセクションも定義しないか、または SENTENCE か PARAGRAPH セクションのみを定義する場合は、このグループ・タイプを使用します。これはデフォルトです。
BASIC_SECTION_GROUP	このグループ・タイプを使用して、開始および終了タグが <A> および </A> という形式のセクションを定義します。 **注意：**このグループ・タイプでは、対になっていないカッコ、コメント・タグおよび属性などの入力はサポートされません。これらを入力するには、HTML_SECTION_GROUP を使用してください。
HTML_SECTION_GROUP	このグループ・タイプを使用して、HTML ドキュメントを索引付けし、HTML ドキュメントのセクションを定義します。
XML_SECTION_GROUP	このグループ・タイプを使用して、XML ドキュメントを索引付けし、XML ドキュメントのセクションを定義します。

セクション・グループ・プリファレンス	説明
AUTO_SECTION_GROUP	このグループ・タイプを使用して、XML ドキュメントの開始タグ / 終了タグに対して自動的にゾーン・セクションを作成します。XML タグから導出されるセクション名は、XML 内と同様に大 / 小文字が区別されます。 属性セクションは、属性を持つ XML タグに対して自動的に作成されます。属性セクションは、attribute@tag という形式でネーミングされます。 停止セクション、空のタグ、処理命令およびコメントは、索引付けされません。 自動セクション・グループには次の制限事項が適用されます。 ■ ゾーン、フィールドまたは特殊セクションは、自動セクション・グループに追加できません。 ■ 自動セクション化は、XML ドキュメント・タイプ（ルート要素）を索引付けしません。ただし、ドキュメント・タイプに停止セクションを定義することはできます。 ■ プリフィックスおよび名前空間を含む、索引付けされたタグの長さは、64 文字以下です。これより長いタグは索引付けされません。
PATH_SECTION_GROUP	このグループ・タイプを使用して、XML ドキュメントを索引付けします。このタイプは、AUTO_SECTION_GROUP と同じように動作します。 相違点は、このセクション・グループを使用すると、INPATH および HASPATH 演算子でパス検索を実行できることです。タグまたは属性名の問合せでは、大 / 小文字が同様に区別されます。停止セクションは使用できません。
NEWS_SECTION_GROUP	このグループ・タイプを使用して、RFC 1036 に従ったニュース・グループ形式のドキュメントのセクションを定義します。

セクション・グループの例

HTML ドキュメントのセクション・グループの作成

次の文は、HTML グループ・タイプを使用して htmgroup というセクション・グループを作成します。

```
begin
ctx_ddl_create_section_group('htmgroup', 'HTML_SECTION_GROUP');
end;
```

CTX_DDL.ADD_SECTION プロシージャを使用して、このグループにオプションでセクションを追加できます。ドキュメントを索引付けするには、次の文を発行します。

```
create index myindex on docs(htmlfile) indextype is ctxsys.context
parameters('filter ctxsys.null_filter section group htmgroup');
```

XML ドキュメントのセクション・グループの作成

次の文は、XML_SECTION_GROUP グループ・タイプを使用して xmlgroup というセクション・グループを作成します。

```
begin
ctx_ddl_create_section_group('xmlgroup', 'XML_SECTION_GROUP');
end;
```

CTX_DDL.ADD_SECTION プロシージャを使用して、このグループにオプションでセクションを追加できます。ドキュメントを索引付けするには、次の文を発行します。

```
create index myindex on docs(htmlfile) indextype is ctxsys.context
parameters('filter ctxsys.null_filter section group xmlgroup');
```

XML ドキュメントの自動セクション化

次の文は、AUTO_SECTION_GROUP グループ・タイプを使用して auto というセクション・グループを作成します。このセクション・グループによって、XML ドキュメントのタグから自動的にセクションが作成されます。

```
begin
    ctx_ddl_create_section_group('auto', 'AUTO_SECTION_GROUP');
end;

CREATE INDEX myindex on docs(htmlfile) INDEXTYPE IS ctxsys.context
PARAMETERS('filter ctxsys.null_filter section group auto');
```

分類型

この項では、CTX_CLS.TRAIN のプリファレンスの作成に使用する分類型について説明します。

RULE_CLASSIFIER

RULE_CLASSIFIER 型を使用して、ルール生成プロシージャ CTX_CLS.TRAIN のプリファレンスを作成します。

この型の属性は、次のとおりです。

属性名	データ型	デフォルト	最小値	最大値	説明
THRESHOLD	I	50	1	99	ルール生成のしきい値（パーセント）。ルールは、その信頼度レベルがしきい値を超えた場合のみ出力されます。
MAX_TERMS	I	100	20	2000	各クラスについて、関連語句のリストがルール生成のために選択されます。この属性は、各クラスについて選択可能な語句の最大数を指定します。
MEMORY_SIZE	I	500	10	4000	トレーニング用の通常のメモリー使用量（MB）。大きい値を指定すると、パフォーマンスが向上します。
NT_THRESHOLD	F	0.001	0	0.90	語句選択のしきい値。関連語句を選択する際の 2 つのステップを導く 2 つのしきい値があります。このしきい値は、最初のステップの動作を制御します。このステップでは候補語句が選択されます。この候補語句は、次のステップでさらに検討されます。トレーニング・セット内のドキュメント数に対する語句の出現頻度の割合がこのしきい値を超えた場合に、その語句が選択されます。

属性名	データ型	デフォルト	最小値	最大値	説明
TERM_THRESHOLD	I	10	0	100	語句選択のしきい値（パーセント）。このしきい値は、2 番目のステップの語句選択を制御します。各候補語句には、特定クラスとの相関関係を示す計算された数量があります。クラス内の全候補語句の最大値に対するこの数量値の割合がこのしきい値を超えた場合のみ、候補語句がそのクラスに対して選択されます。

ストップリスト

ストップリストによって、言語にある索引付けしないワードが識別されます。英語の場合、索引付けしないストップテーマも識別できます。デフォルトでは、データベースの言語に対応してシステムが提供するストップリストを使用して、テキストの索引付けが行われます。

Oracle Text では、英語、フランス語、ドイツ語、スペイン語、オランダ語およびデンマーク語を含むほとんどの言語に対して、デフォルトのストップリストが提供されています。これらのデフォルト・ストップリストには、ストップワードのみが含まれます。

関連項目： 提供されるデフォルトのストップリストの詳細は、付録 D「提供されるストップリスト」を参照してください。

マルチ言語のストップリスト

言語固有のストップワードを保持するマルチ言語のストップリストを作成できます。マルチ言語のストップリストは、MULTI_LEXER を使用した英語、ドイツ語および日本語などの異なる言語のドキュメントを含む表の索引付けに有効です。

マルチ言語のストップリストを作成するには、CTX_DLL.CREATE_STOPLIST プロシージャを使用し、MULTI_STOPLIST というストップリスト・タイプを指定します。CTX_DDL.ADD_STOPWORD を使用して言語固有のストップワードを追加します。

索引付け時に、各ドキュメントの言語列が調べられ、その言語のストップワードのみが排除されます。問合せ時に、セッション言語の設定によって、アクティブなストップワードが特定されます。これは、マルチレクサーの使用時にアクティブなレクサーが特定されるのと同様です。

ストップリストの作成

CTX_DLL.CREATE_STOPLIST プロシージャを使用して、独自のストップリストを作成できます。このプロシージャを使用すると、単一言語ストップリストの BASIC_STOPLIST またはマルチ言語ストップリストの MULTI_STOPLIST を作成できます。

独自のストップリストを作成して使用する場合は、このリストを CREATE INDEX の PARAMETERS 句に指定する必要があります。

デフォルトのストップリストの変更

デフォルトのストップリストは、常に CTXSYS.DEFAULT_STOPLIST という名前です。このストップリストを変更するには、次のプロシージャを使用します。

- CTX_DDL.ADD_STOPWORD
- CTX_DDL.REMOVE_STOPWORD
- CTX_DDL.ADD_STOPTHEME
- CTX_DDL.ADD_STOPCLASS

CTX_DDL パッケージを使用して CTXSYS.DEFAULT_STOPLIST を変更した場合は、変更を有効にするために索引を再作成する必要があります。

ストップワードの動的な追加

ALTER INDEX を使用して、ストップワードをデフォルトまたはカスタムのストップリストに動的に追加できます。ストップワードを動的に追加する場合、ワードはすぐにストップワードになり、索引から削除されるため、再索引付けする必要はありません。

注意： ストップワードは、索引に動的に追加できますが、動的に削除することはできません。ストップワードを削除するには、CTX_DDL.REMOVE_STOPWORD を使用して索引を削除してから、再索引付けする必要があります。

関連項目： 第 1 章「SQL 文と演算子」の「ALTER INDEX」

システム定義プリファレンス

Oracle Text をインストールすると、いくつかの索引付けプリファレンスが作成されます。CREATE INDEX の PARAMETERS 句にこれらの索引付けプリファレンスを使用することもできますし、また独自で定義することもできます。

デフォルトの索引パラメータは、この項で説明するシステム定義プリファレンスのいくつかにマップされます。

> **関連項目：** デフォルトの索引パラメータの詳細は、2-89 ページの「デフォルトの索引付けパラメータ」を参照してください。

システム定義のプリファレンスは、次のカテゴリに分類されています。

- データ記憶域
- フィルタ
- レクサー
- セクション・グループ
- ストップリスト
- 記憶域
- ワードリスト

データ記憶域

CTXSYS.DEFAULT_DATASTORE

このプリファレンスは、DIRECT_DATASTORE 型を使用します。このプリファレンスを使用すると、テキストが列に直接格納されているテキスト列に対して索引を作成できます。

CTXSYS.FILE_DATASTORE

このプリファレンスは、FILE_DATASTORE 型を使用します。

CTXSYS.URL_DATASTORE

このプリファレンスは、URL_DATASTORE 型を使用します。

フィルタ

CTXSYS.NULL_FILTER

このプリファレンスは、NULL_FILTER 型を使用します。

CTXSYS.INSO_FILTER

このプリファレンスは、INSO_FILTER 型を使用します。

レクサー

CTXSYS.DEFAULT_LEXER

デフォルトのレクサーは、インストール時に使用した言語によって異なります。次の項では、各言語の CTXSYS.DEFAULT_LEXER のデフォルト設定について説明します。

アメリカ英語およびイギリス英語での設定 使用言語が英語の場合、このプリファレンスは index_themes 属性を使用禁止にして BASIC_LEXER を使用します。

デンマーク語での設定 使用言語がデンマーク語の場合、このプリファレンスは次のオプションを使用可能にして BASIC_LEXER を使用します。

- 代替スペル（alternate_spelling 属性を DANISH に設定）

オランダ語での設定 使用言語がオランダ語の場合、このプリファレンスは次のオプションを使用可能にして BASIC_LEXER を使用します。

- コンポジット索引付け（composite 属性を DUTCH に設定）

ドイツ語およびドイツ工業規格（DIN）での設定 使用言語がドイツ語の場合、このプリファレンスは次のオプションを使用可能にして BASIC_LEXER を使用します。

- 大 / 小文字が区別される索引付け（mixed_case 属性を使用可能）
- コンポジット索引付け（composite 属性を GERMAN に設定）
- 代替スペル（alternate_spelling 属性を GERMAN に設定）

フィンランド語、ノルウェー語およびスウェーデン語での設定 使用言語がフィンランド語、ノルウェー語またはスウェーデン語の場合、このプリファレンスは次のオプションを使用可能にして BASIC_LEXER を使用します。

- 代替スペル（alternate_spelling 属性を SWEDISH に設定）

日本語での設定 使用言語が日本語の場合、このプリファレンスは JAPANESE_VGRAM_LEXER を使用します。

韓国語での設定 使用言語が韓国語の場合、このプリファレンス KOREAN_MORPH_LEXER を使用します。KOREAN_MORPH_LEXER 用のすべての属性が使用可能になります。

中国語での設定 使用言語が中国語（簡体字または繁体字）の場合、このプリファレンスは CHINESE_VGRAM_LEXER を使用します。

その他の言語 この項に記載していないその他の言語については、このプリファレンスは、属性を設定せずに BASIC_LEXER を使用します。

> **関連項目：** これらのオプションの詳細は、2-35 ページの「BASIC_LEXER」を参照してください。

CTXSYS.BASIC_LEXER

このプリファレンスは、BASIC_LEXER を使用します。

セクション・グループ

CTXSYS.NULL_SECTION_GROUP

このプリファレンスは、NULL_SECTION_GROUP 型を使用します。

CTXSYS.HTML_SECTION_GROUP

このプリファレンスは、HTML_SECTION_GROUP 型を使用します。

CTXSYS.AUTO_SECTION_GROUP

このプリファレンスは、AUTO_SECTION_GROUP 型を使用します。

CTXSYS.PATH_SECTION_GROUP

このプリファレンスは、PATH_SECTION_GROUP 型を使用します。

ストップリスト

CTXSYS.DEFAULT_STOPLIST

このストップリスト・プリファレンスは、使用するデータベース言語のストップリストをデフォルトに設定します。

関連項目： 提供されるストップリストのストップ・ワードの完全なリストは、付録 D「提供されるストップリスト」を参照してください。

CTXSYS.EMPTY_STOPLIST

このストップリストにはワードがありません。

記憶域

CTXSYS.DEFAULT_STORAGE

このプリファレンスは、BASIC_STORAGE 型を使用します。

ワードリスト

CTXSYS.DEFAULT_WORDLIST

このプリファレンスは、使用するデータベース言語に対して言語ステマーを使用します。使用している言語が 2-69 ページの表 2-7 に記載されていない場合、このプリファレンスは、NULL ステマーおよび GENERIC ファジー・マッチング属性にデフォルト設定されます。

システム・パラメータ

この項では、Oracle Text のシステム・パラメータを説明します。システム・パラメータは次のカテゴリに分類されます。

- 汎用システム・パラメータ
- デフォルトの索引付けパラメータ

汎用システム・パラメータ

Oracle Text をインストールすると、システム定義プリファレンスの他に、次のシステム・パラメータが設定されます。

システム・パラメータ	説明
MAX_INDEX_MEMORY	CREATE INDEX および ALTER INDEX の PARAMETERS 句に指定できる最大の索引付けメモリー。
DEFAULT_INDEX_MEMORY	CREATE INDEX および ALTER INDEX とともに使用されるデフォルトの索引付けメモリー。
LOG_DIRECTORY	CTX_OUTPUT ログ・ファイル用のディレクトリ。
CTX_DOC_KEY_TYPE	CTX_DOC プロシージャに対するデフォルト入力キー・タイプ（ROWID または PRIMARY_KEY のいずれか）。インストール時に ROWID に設定されます。 **関連項目：**8-20 ページの「CTX_DOC.SET_KEY_TYPE」

システムのデフォルト値は、CTX_PARAMETERS ビューを問い合せると表示できます。CTX_ADM.SET_PARAMETER プロシージャを使用して、デフォルトを変更できます。

デフォルトの索引付けパラメータ

この項では、CONTEXT 索引および CTXCAT 索引の作成時に使用できる索引パラメータを説明します。

CONTEXT 索引のパラメータ

次のデフォルトのパラメータは、CONTEXT 索引の作成時に CREATE INDEX の PARAMETERS 句にプリファレンスを指定しない場合に使用されます。デフォルトの各パラメータは、データ記憶域、フィルタ処理、レクサー処理などに使用するシステム定義プリファレンスを指定します。

システム・パラメータ	使用する場合	デフォルト値
DEFAULT_DATASTORE	CREATE INDEX の PARAMETERS 句にデータストア・プリファレンスが指定されていない場合	CTXSYS.DEFAULT_DATASTORE
DEFAULT_FILTER_FILE	CREATE INDEX の PARAMETERS 句にフィルタ・プリファレンスが指定されず、次の条件のいずれかに該当する場合 ■ ファイルが外部ファイル（BFILE）に格納されている。 ■ FILE_DATASTORE を使用するデータストア・プリファレンスを指定している。	CTXSYS.INSO_FILTER
DEFAULT_FILTER_BINARY	CREATE INDEX の PARAMETERS 句にフィルタ・プリファレンスが指定されず、テキスト列のデータ型が RAW、LONG RAW または BLOB であることが検出されている場合	CTXSYS.INSO_FILTER
DEFAULT_FILTER_TEXT	CREATE INDEX の PARAMETERS 句にフィルタ・プリファレンスが指定されず、テキスト列のデータ型が LONG、VARCHAR2、VARCHAR、CHAR または CLOB のいずれかであることが検出されている場合	CTXSYS.NULL_FILTER
DEFAULT_SECTION_HTML	CREATE INDEX の PARAMETERS 句にセクション・グループが指定されず、次の条件のいずれかに該当する場合 ■ データストア・プリファレンスが URL_DATASTORE を使用している。 ■ フィルタ・プリファレンスが INSO_FILTER を使用している。	CTXSYS.HTML_SECTION_GROUP

システム・パラメータ	使用する場合	デフォルト値
DEFAULT_SECTION_TEXT	CREATE INDEX の PARAMETERS 句にセクション・グループが指定されず、URL_DATASTORE および INSO_FILTER のいずれも使用していない場合	CTXSYS.NULL_SECTION_GROUP
DEFAULT_STORAGE	CREATE INDEX の PARAMETERS 句に記憶域プリファレンスが指定されていない場合	CTXSYS.DEFAULT_STORAGE
DEFAULT_LEXER	CREATE INDEX の PARAMETERS 句にレクサー・プリファレンスが指定されていない場合	CTXSYS.DEFAULT_LEXER
DEFAULT_STOPLIST	CREATE INDEX の PARAMETERS 句にストップリスト・プリファレンスが指定されていない場合	CTXSYS.DEFAULT_STOPLIST
DEFAULT_WORDLIST	CREATE INDEX の PARAMETERS 句にワードリスト・プリファレンスが指定されていない場合	CTXSYS.DEFAULT_WORDLIST

CTXCAT 索引のパラメータ

次のデフォルトの各パラメータは、CREATE INDEX を使用した CTXCAT 索引の作成時に、そのパラメータ文字列にパラメータを指定しない場合に使用されます。CTXCAT 索引は、索引セット、レクサー、記憶域、ストップリストおよびワードリストのパラメータのみをサポートしています。デフォルトの各パラメータには、システム定義プリファレンス名を指定します。

システム・パラメータ	使用する場合	デフォルト値
DEFAULT_CTXCAT_INDEX_SET	CREATE INDEX の PARAMETERS 句に索引セットが指定されていない場合	
DEFAULT_CTXCAT_STORAGE	CREATE INDEX の PARAMETERS 句に記憶域プリファレンスが指定されていない場合	CTXSYS.DEFAULT_STORAGE
DEFAULT_CTXCAT_LEXER	CREATE INDEX の PARAMETERS 句にレクサー・プリファレンスが指定されていない場合	CTXSYS.DEFAULT_LEXER
DEFAULT_CTXCAT_STOPLIST	CREATE INDEX の PARAMETERS 句にストップリスト・プリファレンスが指定されていない場合	CTXSYS.DEFAULT_STOPLIST
DEFAULT_CTXCAT_WORDLIST	CREATE INDEX の PARAMETERS 句にワードリスト・プリファレンスが指定されていない場合 CTXCAT 索引にワードリスト・プリファレンスを指定できますが、CATSEARCH 問合せ言語は、ワイルド・カード、ファジーおよびステミングをサポートしていないため、ほとんどの属性は適用されません。有効な唯一の属性は、日本語データに対する PREFIX_INDEX です。	CTXSYS.DEFAULT_WORDLIST

CTXRULE 索引のパラメータ

次のデフォルトの各パラメータは、CREATE INDEX を使用した CTXRULE 索引の作成時に、そのパラメータ文字列にパラメータを指定しない場合に使用されます。CTXRULE 索引は、レクサー、記憶域、ストップリストおよびワードリストのパラメータのみをサポートしています。デフォルトの各パラメータには、システム定義プリファレンス名を指定します。

システム・パラメータ	使用する場合	デフォルト値
DEFAULT_CTXRULE_LEXER	CREATE INDEX の PARAMETERS 句にレクサー・プリファレンスが指定されていない場合	CTXSYS.DEFAULT_LEXER
DEFAULT_CTXRULE_STORAGE	CREATE INDEX の PARAMETERS 句に記憶域プリファレンスが指定されていない場合	CTXSYS.DEFAULT_STORAGE
DEFAULT_CTXRULE_STOPLIST	CREATE INDEX の PARAMETERS 句にストップリスト・プリファレンスが指定されていない場合	CTXSYS.DEFAULT_STOPLIST
DEFAULT_CTXRULE_WORDLIST	CREATE INDEX の PARAMETERS 句にワードリスト・プリファレンスが指定されていない場合	CTXSYS.DEFAULT_WORDLIST

デフォルト値の表示

システムのデフォルト値は、CTX_PARAMETERS ビューを問い合せると表示できます。たとえば、すべてのパラメータと値を表示する場合は、次の文を発行します。

```
SQL> SELECT par_name, par_value from ctx_parameters;
```

デフォルト値の変更

CTX_ADM.SET_PARAMETER プロシージャに別のカスタム・プリファレンス名またはシステム定義プリファレンス名を指定して、デフォルト値を変更することができます。

3

CONTAINS 問合せ演算子

この章では、演算子の優先順位を説明し、CONTAINS 演算子の説明、構文および例を示します。この章の内容は次のとおりです。

- 演算子の優先順位
- ABOUT
- ACCUMulate（,）
- AND（&）
- BROADER TERM（BT、BTG、BTP、BTI）
- EQUIValence（=）
- FUZZY
- HASPATH
- INPATH
- MINUS（-）
- NARROWER TERM（NT、NTG、NTP、NTI）
- NEAR（;）
- NOT（~）
- OR（|）
- PREFERRED TERM（PT）
- RELATED TERM（RT）
- SOUNDEX（!）
- STEM（$）
- ストアド・クエリー式（SQE）

- SYNONYM（SYN）
- THRESHOLD（>）
- TRANSLATION TERM（TR）
- TRANSLATION TERM SYNONYM（TRSYN）
- TOP TERM（TT）
- WEIGHT（*）
- ワイルド・カード（% _）
- WITHIN

演算子の優先順位

演算子の優先順位は、問合せ式のコンポーネントが評価される順序を決定します。テキスト問合せ演算子は、それぞれ独自の評価順を持つ 2 組の演算子に分けることができます。次に、これら 2 つのグループをグループ 1 およびグループ 2 として説明します。

どのような場合でも、問合せ式は演算子の優先順位に従って、左から右の順に評価されます。優先順位の高い演算子が最初に評価されます。優先順位が等しい演算子は、式の中で現れる順序に従って左から右へ適用されます。

グループ 1 演算子

問合せ式内のグループ 1 の演算子を優先順位の高い順に並べると次のようになります。

1. EQUIValence（=）
2. NEAR（;）
3. WEIGHT（*）、THRESHOLD（>）
4. MINUS（-）
5. NOT（~）
6. WITHIN
7. AND（&）
8. OR（|）
9. ACCUMulate（,）

グループ 2 の演算子および文字

問合せ式内のグループ 2 の演算子を評価順位の高い順に並べると次のようになります。

1. ワイルド・カード文字
2. ABOUT
3. STEM（$）
4. FUZZY
5. SOUNDEX（!）

プロシージャ型演算子

グループ 1 にもグループ 2 にもリストされていないその他の演算子は、プロシージャ型演算子です。これらの演算子には優先順位がありません。これらには、SQE およびシソーラス演算子が含まれます。

優先順位の例

問合せ式	評価順
w1 | w2 & w3	(w1) | (w2 & w3)
w1 & w2 | w3	(w1 & w2) | w3
?w1 , w2 | w3 & w4	(?w1) , (w2 | (w3 & w4))
abc = def ghi & jkl = mno	((abc = def) ghi) & (jkl=mno)
dog and cat WITHIN body	dog and（cat WITHIN body）

1 番目の例では、AND は OR よりも優先順位が高いため、問合せは *w1* を含むすべてのドキュメント、および *w2* と *w3* の両方を含むすべてのドキュメントを戻します。

2 番目の例では、問合せは *w1* と *w2* の両方を含むすべてのドキュメント、および *w3* を含むすべてのドキュメントを戻します。

3 番目の例では、最初に FUZZY 演算子が *w1* に適用されます。次に AND 演算子が引数 *w3* および *w4* に適用され、その後 OR 演算子が *w2* および AND 演算子の結果に適用されます。最後に、*w1* に対する FUZZY 演算子のスコアが OR 演算子のスコアに加えられます。

4 番目の例は、EQUIVALENCE 演算子が AND 演算子より優先順位が高いことを示しています。

5 番目の例は、AND 演算子が WITHIN 演算子より優先順位が低いことを示しています。

優先順位の変更

優先順位は、グループ化文字によって次のように変わります。

- カッコ内の拡張操作または実行は、演算子の優先順位に関係なく、他の拡張操作に優先して解決されます。
- カッコ内の式の評価に関しては、演算子の優先順位が有効です。
- 拡張演算子は、大カッコ内に入っていないかぎり、大カッコ内の式には適用されません。

関連項目： 第 4 章「問合せの特殊文字」の「グループ化文字」

ABOUT

動作概要

すべての言語では、ABOUT 問合せによって、この演算子がない同じ問合せよりも、戻される関連ドキュメント数が増加します。Oracle は、ABOUT 問合せの結果に、関連性スコアが最も高く、最も関連性のあるドキュメントを割り当てます。

英語とフランス語での動作

英語とフランス語では、ABOUT 演算子を使用して概念を問い合せます。システムは、索引のテーマ・コンポーネントの概念情報を検索します。

注意： 英語で ABOUT 問合せを発行する場合、索引にテーマ・コンポーネントは必要ありません。ただし、索引にテーマ・コンポーネントがあると、ABOUT 問合せの結果が最も向上します。

Oracle では、問合せワードまたは句に関連する概念を含むドキュメントを取り出します。たとえば、*California* についての ABOUT 問合せを発行すると、システムによって、California の都市である *Los Angeles* と *San Francisco* という語句を含むドキュメントが戻されます。この ABOUT 問合せで戻されるドキュメントには、*California* という語句が含まれている必要はありません。

ABOUT 問合せに指定したワードまたは句が、索引に格納されたテーマと完全に一致する必要はありません。Oracle は、索引の検索を実行する前に、ワードまたは句を正規化します。

ABOUT 演算子は、CONTAINS SQL 演算子および CATSEARCH SQL 演算子と併用できます。

ABOUT 結果の改善

ABOUT 演算子は、英語とフランス語では、提供されているナレッジ・ベースを使用して入力された句を解析します。したがって、ABOUT 問合せは、ナレッジ・ベースにある概念の認識と解析に制限されています。

使用しているアプリケーション固有の用語をナレッジ・ベースに追加すると、ABOUT 問合せの結果を改善できます。

関連項目： 第 14 章「実行可能ファイル」の「ナレッジ・ベースの拡張」

構文

構文	説明
about(*phrase*)	すべての言語において、ABOUT 問合せがない同じ問合せよりも、戻される関連ドキュメント数が増加します。*phrase* パラメータは、1 つのワード、句またはフリー・テキスト・フォーマットのワード文字列です。 英語では、*phrase* に関連する概念を含むドキュメントを戻します。 戻されたスコアは関連性スコアです。 *phrase* に組み込まれた問合せ演算子は無視されます。 索引にテーマ情報のみが含まれている場合、ABOUT 演算子およびオペランドをテキスト列の問合せに組み込む必要があります。そうしない場合、エラーが戻ります。 指定する *phrase* は、4000 文字以内です。

大 / 小文字の区別

問合せを適切な文字で表すと、ABOUT 問合せの結果が最も向上します。これは、問合せの正規化が、大 / 小文字が区別されるナレッジ・カタログに基づくためです。

ただし、ABOUT 問合せの結果を取得するために、大 / 小文字を正確に区別して問合せを入力する必要はありません。システムが、最適な方法で問合せを解釈します。たとえば、*CISCO* という文字で問合せを入力し、これがナレッジ・カタログで検出されない場合、システムでは検索用の関連概念として *Cisco* を使用する場合があります。

制限事項

- ABOUT 問合せで指定する句は、4000 文字以内です。
- WITHIN 演算子を *'ABOUT (xyz) WITHIN abc'* のように ABOUT 演算子と結合することはできません。
- ABOUT 演算子は、NEAR または WITHIN などのオフセット情報を含む演算子と結合できません。

例

単一のワード

サッカー（soccer）についてのドキュメントを検索するには、次の構文を使用します。

```
'about(soccer)'
```

句

句を問合せ語句として入力すると、問合せをさらに詳細化して、国際大会のサッカー・ルール（soccer rules in international competition）についてのドキュメントを含めることができます。

```
'about(soccer rules in international competition)'
```

この英語の例では、*soccer*、*rules* または *international competition* のテーマを持つすべてのドキュメントが戻されます。

通常、スコア付けの点では、これら 3 つのテーマをすべて含むドキュメントの方が、これらのテーマのうちの 1 つまたは 2 つのみを含むドキュメントより高いスコアを示します。

構成されていない句

次の例のように、構成されていない句で問い合せることもできます。

```
'about(japanese banking investments in indonesia)'
```

結合された問合せ

AND や NOT など、他の演算子を使用して、ABOUT 問合せをワード問合せと結合できます。

たとえば、結合された ABOUT およびワード問合せを次のように発行できます。

```
'about(dogs) and cat'
```

次のように、ABOUT 問合せを別の ABOUT 問合せと結合できます。

```
'about(dogs) not about(labradors)'
```

注意： ABOUT 演算子は、*'ABOUT (xyz) WITHIN abc'* のように WITHIN 演算子と結合できません。

CATSEARCH による ABOUT 問合せ

次のように、grammar に CONTEXT を設定した問合せテンプレートを使用して、CATSEARCH による ABOUT 問合せを発行できます。

```
select pk||' ==> '||text from test
where catsearch(text,
'<query>
  <textquery grammar="context">
     about(California)
  </textquery>
  <score datatype="integer"/>
</query>','')>0
order by pk;
```

ACCUMulate（,）

問合せ語句のいずれかが 1 つ以上含まれるドキュメントを検索するには、ACCUM 演算子を使用します。ACCUMULATE 演算子は、ドキュメントを語句の合計の重みによってランク付けします。

構文

構文	説明
term1,*term2* *term1* accum *term2*	*term1* または *term2* を含むドキュメントを戻します。語句の重みによってドキュメントをランク付けし、語句の合計の重みが最も高いドキュメントに最も高いスコアを割り当てます。

例

次の例では、*soccer*、*Brazil* または *cup* のいずれかを含むドキュメントが戻され、3 つの語句をすべて含むドキュメントに最も高いスコアが割り当てられます。

```
'soccer, Brazil, cup'
```

次の例でも、*soccer*、*Brazil* または *cup* のいずれかを含むドキュメントが戻されます。ただし、WEIGHT 演算子によって、*Brazil* を含むドキュメントの方が、*soccer* および *cup* のみを含むドキュメントよりスコアが高いことが保証されます。

```
'soccer, Brazil*3, cup'
```

注意

ACCUMULATE スコア付け

ACCUM は、次の 2 つの基準に基づいてドキュメントにスコアを割り当てます。

- ドキュメントの語句の重み
- ドキュメントの語句のスコア

語句の重みとは、問合せ語句に置く重みをいいます。x,y,z などの問合せでは、各語句の重みは 1 です。x, 3*y, z の問合せでは、語句の重みはそれぞれ 1、3、1 です。

ACCUMULATE スコア付けでは、ドキュメント A で *p* 個の語句の重みの合計が *m* であり、ドキュメント B で *q* 個の語句の重みの合計が *m*+1 の場合、*p* および *q* の数値に関係なく、ドキュメント B の関連性スコアは、ドキュメント A より確実に高くなります。

この 2 つのドキュメントの重みが同じ M の場合、語句平均の重みスコアが高いドキュメントに、高い関連性スコアが割り当てられます。

次の表に、ACCUMULATE スコア付けを示します。

ドキュメント	問合せ	スコア（x）	スコア（y）	スコア（z）	語句の合計の重み	スコア（問合せ）
A	x,y,z	10	0	0	1	3
B	x,y,z	10	20	0	2	38
C	x,y,z	10	20	30	3	73
D	x,y,z	50	50	0	2	50
E	x, y*3, z	100	0	100	2	40
F	x, y*3, z	0	1	0	3	41

表の各行は、ACCUMULATE 問合せのスコアを示しています。最初の 4 行は、ドキュメント A、B、C、D の問合せ x,y,z のスコアを示しています。次の 2 行は、ドキュメント E および F の問合せ x, y*3,z のスコアを示しています。x、y および z は 3 つの異なるワードを表します。ドキュメント E および F の問合せは、2 番目の問合せ語句を重み 3 とし、任意に最も重要な問合せ語句に指定しています。

語句の合計の重みが、ドキュメントごとに示されています。たとえば、ドキュメント A には 1 つの問合せ語句のみが一致するため、このドキュメントの重みは 1 です。同様に、ドキュメント C では、重み 1 のすべての問合せ語句が一致するため、このドキュメントの重みは 3 です。

この表は、問合せ語句の重みが高いドキュメントは、重みが低いドキュメントよりも常に高いスコアを割り当てられることを示しています。たとえば、ドキュメント C は、問合せ語句の重みが最も高いため、常にドキュメント A、B および D よりも高いスコアを示します。同様に、ドキュメント F は、ドキュメント E よりも重みが高いため、E よりも高いスコアを示します。

ドキュメント B や D など、ドキュメントの語句の重みが等しい場合は、ドキュメント D のように、語句平均の重みスコアが高い方に高いスコアが割り当てられます。

AND（&）

各問合せ語句が 1 つ以上含まれるドキュメントを検索するには、AND 演算子を使用します。

構文

構文	説明
term1&*term2* *term1* and *term2*	*term1* および *term2* が両方含まれているドキュメントを戻します。そのオペランドの最小スコアを戻します。すべての問合せ語句が存在する必要があり、スコアの低い方が選択されます。

例

blue、*black* および *red* という語句が含まれているすべてのドキュメントを取得するには、次の問合せを発行します。

```
'blue & black & red'
```

AND 問合せでは、各問合せ語句のスコアのうち最も低いスコアが戻ります。この例では、ドキュメント内の *blue*、*black* および *red* の 3 つの語句のスコアがそれぞれ 10、20 および 30 の場合、ドキュメントのスコアは 10 になります。

BROADER TERM（BT、BTG、BTP、BTI）

指定された問合せ語句の上位語としてシソーラスで定義されている語句が含まれるように問合せを拡張するには、BROADER TERM 演算子（BT、BTG、BTP、BTI）を使用します。上位語の上位語、さらにその上位語、というようにシソーラス階層を上がるように問合せを拡張できます。

構文

構文	説明
BT(*term*[(*qualifier*)][,*level*][,*thes*])	シソーラスで term に対する上位語として定義されたすべての語句が含まれるように、問合せを拡張します。
BTG(*term*[(*qualifier*)][,*level*][,*thes*])	シソーラスで term に対する上位汎用語として定義されたすべての語句が含まれるように、問合せを拡張します。
BTP(*term*[(*qualifier*)][,*level*][,*thes*])	シソーラスで term に対する上位語分語として定義されたすべての語句が含まれるように、問合せを拡張します。
BTI(*term*[(*qualifier*)][,*level*][,*thes*])	シソーラスで term に対する上位インスタンス語として定義されたすべての語句が含まれるように、問合せを拡張します。

term
BROADER TERM 演算子にオペランドを指定します。term は、thes で指定されたシソーラスで語句に対して定義されている上位語のエントリが含まれるように拡張されます。たとえば、*BTG(dog)* と指定すると、拡張には、*dog* に対する上位汎用語として定義されている語句のみが含まれます。拡張演算子は term 引数に指定できません。

この拡張に含まれる上位語の数は、level の値によって決定されます。

qualifier
term が thes の同じ階層分岐で 2 つ以上のノードに現れる同形異義語（同じスペルで複数の意味を持つワードまたは句）の場合は、term に修飾子を指定します。

BROADER TERM 問合せで同形異義語に修飾子が指定されていない場合は、すべての同形異義語の上位語が含まれるように問合せが拡張されます。

level
指定された語句に対する上位語を戻すために、シソーラス階層内を横断するレベル数を指定します。たとえば、BT のレベル 1 問合せは、上位語が定義されている場合、指定した問合せ語句の上位語エントリを戻します。レベル 2 問合せは、指定した問合せ語句の上位語エントリ、さらにそれらの上位語の上位語エントリを戻します。

level 引数はオプションで、デフォルト値は（1）です。level 引数に 0（ゼロ）または負の値を設定すると、元の問合せ語句のみを戻します。

thes
指定された問合せ語句の拡張問合せを戻すために使用するシソーラス名を指定します。thes 引数はオプションで、デフォルト値は DEFAULT です。このデフォルト値を使用する場合は、シソーラス表に DEFAULT という名前のシソーラスが存在している必要があります。

> **注意：** thes を指定する場合は、level も指定する必要があります。

例

次の問合せは、*tutorial* という問合せ語句、または DEFAULT シソーラスで *tutorial* の BT 語として定義された語句が含まれているすべてのドキュメントを戻します。

```
'BT(tutorial)'
```

シソーラス名を指定する場合は、次のように level も指定する必要があります。

```
'BT(tutorial, 2, mythes)'
```

同形異義語の BROADER TERM 演算子

machine が *crane*（building equipment）の上位語で、*bird* が *crane*（waterfowl）の上位語であり、上位語問合せに修飾子が指定されていない場合、問合せ

```
BT(crane)
```

は、次のように拡張されます。

```
'{crane} or {machine} or {bird}'
```

上位語問合せで *waterfowl* が *crane* の修飾子として指定されている場合、問合せ

```
BT(crane{(waterfowl)})
```

は、次のように拡張されます。

```
'{crane} or {bird}'
```

> **注意：** BROADER TERM または NARROWER TERM 問合せで修飾子を指定する場合、修飾子およびその表記法（カッコ付け）はこの例のようにエスケープする必要があります。

関連項目

CTX_THES パッケージのプロシージャを使用して、シソーラスをブラウズできます。

関連項目： シソーラスの上位語のブラウズについては、第 12 章「CTX_THES パッケージ」の「BT」を参照してください。

EQUIValence（=）

問合せ時に条件を満たすワードの置換を指定するには、EQUIV 演算子を使用します。

構文

構文	説明
term1=*term2* *term1* equiv *term2*	term2 が term1 の条件を満たす置換語であることを指定します。スコアは、両方の語句の出現回数の合計として計算されます。

例

次の例は、句 *alsatians are big dogs* または *labradors are big dogs* のいずれかを含むすべてのドキュメントを戻します。

```
'labradors=alsatians are big dogs'
```

演算子の優先順位

EQUIV 演算子は、拡張演算子（FUZZY、SOUNDEX、STEM）を除く他のすべての演算子より高い優先順位を持ちます。

FUZZY

指定された語句に類似するスペルのワードが含まれるように問合せを拡張するには、FUZZY 演算子を使用します。この拡張タイプは、ドキュメント・セットにスペル・ミスが多い場合に、より正確な結果を検索するために有効です。

新しいファジー構文を使用すると、結果セットのランク付けができます。このランク付けによって、問合せワードに対する類似度が高いワードを含むドキュメントには、類似度の低いワードを含むドキュメントより高いスコアが割り当てられます。また、拡張する語句の数も制限できます。

STEM 拡張とは異なり、FUZZY 拡張によって生成されるワード数は、索引内のワードによって異なります。したがって、結果は、索引の内容によって大幅に変わってきます。

サポート対象言語

Oracle Text では、英語、ドイツ語、イタリア語、オランダ語、スペイン語および OCR のファジー定義をサポートしています。

ストップワード

FUZZY 拡張がストップワードを戻す場合、そのストップワードは問合せの中には含まれず、CTX_DOC.HIGHLIGHT または CTX_DOC.MARKUP でハイライト表示もされません。

基本文字変換

テキスト列に基本文字変換を使用でき、問合せ式に FUZZY 演算子が指定されている場合は、問合せの基本文字書式が処理されます。

構文

```
fuzzy(term, score, numresults, weight)
```

パラメータ	説明
term	FUZZY 拡張の実行対象ワードを指定します。term は、索引内のワードのみが含まれるように拡張されます。
score	類似度スコアを指定します。この数値未満のスコアを割り当てられた拡張内の語句は、排除されます。1 ～ 80 の数を指定します。デフォルトは 60 です。
numresults	term の拡張に使用する語句の最大数を指定します。1 ～ 5000 の数を指定します。デフォルトは 100 です。
weight	結果に対して、それぞれの類似度スコアに応じて重みを付ける場合は、WEIGHT または W を指定します。 結果に重みを付けない場合は、NOWEIGHT または N を指定します。

例

次の CONTAINS 問合せについて考えます。

```
...CONTAINS(TEXT, 'fuzzy(government, 70, 6, weight)', 1) > 0;
```

この問合せは、類似度スコアが 70 を超える、索引にある *government* の最初の 6 ファジー・バリエーションに拡張されます。

さらに、結果セットのドキュメントには、*government* に対するそれぞれの類似度に応じて重み付けが行われます。government に最も類似しているワードが含まれているドキュメントに、最も高いスコアが割り当てられます。

不要なパラメータは、適切な数のカンマを使用してスキップできます。たとえば、次のようにします。

```
'fuzzy(government,,,weight)'
```

下位互換性のための構文

前のリリースの古いファジー構文も引き続きサポートされています。この構文は、次のとおりです。

パラメータ	説明
?term	スペルが似ているすべての語句が、指定した問合せ語句に含まれるように問合せ語句を拡張します。

HASPATH

指定したセクション・パスを含むすべての XML ドキュメントを検索するには、この演算子を使用します。また、この演算子を使用すると、セクションの等価性もテストできます。

この演算子を機能させるには、PATH_SECTION_GROUP を使用して索引を作成する必要があります。

構文

構文	説明
HASPATH(path)	XML ドキュメント・セットを検索し、*path* が存在するすべてのドキュメントに対してスコア 100 を戻します。親と子のパスは、/ 文字で区切ります。たとえば、*A/B/C* と指定できます。 例を参照してください。
HASPATH(A="value")	XML ドキュメント・セットを検索し、*value* の内容のみ含まれている要素 A があるすべてのドキュメントに対してスコア 100 を戻します。 例を参照してください。

例

パスのテスト

次の問合せの例を示します。

```
HASPATH(A/B/C)
```

この問合せは次のドキュメントを検索して、スコア 100 を戻します。

```
<A><B><C>dog</C></B></A>
```

この問合せでは、*dog* は参照されません。

セクションの等価性のテスト

次の問合せの例を示します。

```
dog INPATH A
```

この問合せは、次のドキュメントを検索します。

```
<A>dog</A>
```

さらに、次のドキュメントも検索します。

```
<A>dog park</A>
```

問合せを、語句 *dog* のみに制限し、それ以外の語句を検索しないようにする場合は、HASPATH 演算子によるセクションの等価性のテストを使用できます。たとえば、問合せ

```
HASPATH(A="dog")
```

は、最初のドキュメントを検索し、スコア 100 を戻します。2 番目のドキュメントは検索しません。

制限事項

XML のセクション・データの記録方法により、完全に空の XML セクションの場合は、次のように不適切な一致が発生することがあります。

<A><B><C></C></B><D><E></E></D></A>

HASPATH（A/B/E）または HASPATH（A/D/C）の問合せでは、このドキュメントと不適切に一致します。この種の不適切な一致は、空のタグの間にテキストを挿入することで回避できます。

INPATH

XML ドキュメントでパス検索を実行するには、この演算子を使用します。この演算子は、WITHIN 演算子と似ています。ただし、右側には、単一のセクション名ではなく、カッコで囲んだパスが置かれます。

INPATH 演算子を機能させるには、PATH_SECTION_GROUP を使用して索引を作成する必要があります。

構文

INPATH 演算子の構文は、次のとおりです。

トップレベルのタグ検索

構文	説明
term INPATH (/A) term INPATH (A)	トップレベルのタグ <A> および </A> 内に *term* が含まれているドキュメントを戻します。タグ A は、トップレベルのタグで、かつドキュメント・タイプのタグである必要があります。

任意レベルのタグ検索

構文	説明
term INPATH (//A)	任意レベルの <A> タグに *term* が含まれているドキュメントを戻します。この問合せは、*'term WITHIN A'* と同じです。

直接の親子関係のパス検索

構文	説明
term INPATH (A/B)	トップレベルの要素 A の直接の子である要素 B に *term* が含まれているドキュメントを戻します。 たとえば、次のドキュメント `<A><B>term</B></A>` が戻されます。

単一レベルのワイルド・カード検索

構文	説明
term INPATH (A/*/B)	トップレベルの要素 A の孫（2 レベル下位）である要素 B に *term* が含まれているドキュメントを戻します。 たとえば、次のドキュメント `<A><D><B>term</B></D></A>` が戻されます。

マルチレベルのワイルド・カード検索

構文	説明
term INPATH (A/*/B/*/*/C)	トップレベルの要素 A の 2 レベル下位（孫）にある要素 B から 3 レベル下位の要素 C に *term* が含まれているドキュメントを戻します。

任意レベルの子検索

構文	説明
term INPATH(A//B)	トップレベルの要素 A の子（任意レベル）である要素 B に term が含まれているドキュメントを戻します。

属性の検索

構文	説明
term INPATH (//A/@B)	任意レベルの要素 A の属性 B に *term* が含まれているドキュメントを戻します。属性は、直接の親にバインドされている必要があります。

子 / 属性の存在のテスト

構文	説明
term INPATH (A[B])	要素 B を直接の子として持つトップレベルの要素 A に term が含まれているドキュメントを戻します。
term INPATH (A[.//B])	任意レベルの要素 B を子として持つトップレベルの要素 A に term が含まれているドキュメントを戻します。
term INPATH (//A[@B])	属性 B を持つ任意レベルの要素 A に term が含まれているドキュメントを検索します。属性は、直接の親に結合されている必要があります。

属性値のテスト

構文	説明
term INPATH (A[@B = "value"])	値が *value* である属性 B を持つトップレベルの要素 A に *term* が含まれているすべてのドキュメントを検索します。
term INPATH (A[@B != "value"])	値が *value* 以外の属性 B を持つトップレベルの要素 A に *term* が含まれているすべてのドキュメントを検索します。

タグ値のテスト

構文	説明
term INPATH (A[B = "value"]))	値が *value* であるタグ B を持つタグ A に *term* が含まれているドキュメントを戻します。

NOT

構文	説明
term INPATH (A[NOT(B)])	要素 B を直接の子として持たないトップレベルの要素 A に *term* が含まれているドキュメントを検索します。

AND および OR のテスト

構文	説明
term INPATH (A[B and C])	要素 B と要素 C を直接の子として持つトップレベルの要素 A に term が含まれているドキュメントを検索します。
term INPATH (A[B and @C="value"]])	要素 B と値が *value* である属性 C を持つトップレベルの要素 A に *term* が含まれているドキュメントを検索します。
term INPATH (A [B OR C])	要素 B または要素 C を持つトップレベルの要素 A に *term* が含まれているドキュメントを検索します。

パスとノードの組合せのテスト

構文	説明
term INPATH (A[@B = "value"]/C/D)	値が *value* である属性 B を持つトップレベルの要素 A の子が要素 C の場合、要素 C の子である要素 D に *term* が含まれているドキュメントを戻します。

ネストされた INPATH

次のように、INPATH 式全体を別の INPATH 式内にネストできます。

```
(dog INPATH (//A/B/C)) INPATH (D)
```

ネストを実行しても、2 つの INPATH のパスは完全に独立しています。外部 INPATH のパスは、内部 INPATH のパスのコンテキスト・ノードを変更しません。次に例を示します。

```
(dog INPATH (A)) INPATH (D)
```

この式は、ドキュメントを検索しません。内部 INPATH は、トップレベルのタグ A 内の *dog* を検索し、外部 INPATH は、トップレベルのタグ D を持つドキュメントに検索を制限されているためです。したがって、この式では、ドキュメントは検索されません。

大 / 小文字の区別

パス検索では、タグ名と属性名の大 / 小文字が区別されます。次に例を示します。

```
dog INPATH (A)
```

この式では、<A>dog</A> は検索されますが、<a>dog は検索されません。この場合は、次の式を使用します。

```
dog INPATH (a)
```

例

トップレベルのタグ検索

トップレベルのタグ <A> に、語句 *dog* が含まれているすべてのドキュメントを検索するには、次の問合せを行います。

```
dog INPATH (/A)
```

または

```
dog INPATH(A)
```

任意レベルのタグ検索

任意レベルの <A> タグに語句 *dog* が含まれているすべてのドキュメントを検索するには、次の問合せを行います。

```
dog INPATH(//A)
```

この問合せは、次のドキュメントを検索します。

```
<A>dog</A>
```

および

```
<C><B><A>dog</A></B></C>
```

直接の親子関係の検索

トップレベルの要素 A の直接の子である要素 B に語句 *dog* が含まれているすべてのドキュメントを検索するには、次の問合せを行います。

```
dog INPATH(A/B)
```

この問合せは、次の XML ドキュメントを検索します。

```
<A><B>My dog is friendly.</B><A>
```

ただし、次のドキュメントは検索しません。

```
<C><B>My dog is friendly.</B></C>
```

タグ値のテスト

タグの値をテストできます。たとえば、次の問合せ

```
dog INPATH(A[B="dog"])
```

は、次のドキュメントを検索します。

```
<A><B>dog</B></A>
```

ただし、次のドキュメントは検索しません。

```
<A><B>My dog is friendly.</B></A>
```

属性の検索

属性の内容を検索できます。たとえば、次の問合せ

```
dog INPATH(//A/@B)
```

は、次のドキュメントを検索します。

```
<C><A  B="snoop dog"> </A> </C>
```

属性値のテスト

属性の値をテストできます。たとえば、次の問合せ

```
California INPATH (//A[@B = "home address"])
```

は、次のドキュメントを検索します。

```
<A B="home address">San Francisco, California, USA</A>
```

ただし、次のドキュメントは検索しません。

```
<A B="work address">San Francisco, California, USA</A>
```

パスのテスト

HASPATH 演算子を使用して、パスの存在をテストできます。たとえば、次の問合せ

```
HASPATH(A/B/C)
```

は、次のドキュメントを検索して、スコア 100 を戻します。

```
<A><B><C>dog</C></B></A>
```

この問合せでは、*dog* は参照されません。

制限事項

等価性のテスト

INPATH による等価性のテストの例を次に示します。

```
dog INPATH (A[@B = "foo"])
```

これらの式には、次の制限事項が適用されます。

- 等価性と非等価性のみがサポートされています。範囲演算子および範囲関数は、サポートされていません。
- 等式の左側には属性を置く必要があります。タグとリテラルは、左側に置くことができません。
- 等式の右側にはリテラルを置く必要があります。タグと属性は、右側に置くことができません。
- 等価性のテストは、使用しているレクサーの設定によって異なります。デフォルト設定の場合の例を示します。

  ```
  dog INPATH (A[@B= "pot of gold"])
  ```

この問合せは、次のセクションと一致します。

```
<A B="POT OF GOLD">dog</A>
```
および

```
<A B="pot of gold">dog</A>
```
これは、デフォルト設定のレクサーでは、大 / 小文字が区別されないためです。

```
<A B="POT BLACK GOLD">dog</A>
```
これは、英語では、OF がデフォルトのストップワードであり、その位置にある任意のワードが問合せと一致するためです。

```
<A B="POT_OF_GOLD">dog</A>
```
これは、デフォルトでは、アンダースコアが結合文字ではないためです。

MINUS（-）

特定の問合せ語句が含まれているドキュメントを検索し、2 番目の問合せ語句が存在するドキュメントのランクを低くする場合は、MINUS 演算子を使用します。MINUS 演算子は、不要なノイズ語句が含まれているドキュメントのスコアを低くする場合に有効です。

構文

構文	説明
term1-*term2* *term1* minus *term2*	*term1* が含まれているドキュメントを戻します。*term1* のスコアから *term2* のスコアを引いて、スコアを計算します。正数のスコアを持つドキュメントのみが戻されます。

例

たとえば *cars* という語句についての問合せは、常に *Ford cars* についてのドキュメントに高いスコアを戻すとします。次の式を使用して、Ford に関するドキュメントのスコアを低くできます。

```
'cars - Ford'
```

この式は語句 *cars*（*Ford* を含む場合もある）を含むドキュメントを戻します。ただし、戻されたドキュメントのスコアは、*cars* のスコアから *Ford* のスコアを差し引いた値になります。

NARROWER TERM（NT、NTG、NTP、NTI）

指定された問合せ語句の下位語としてシソーラスで定義されているすべての語句が含まれるように問合せを拡張するには、NARROWER TERM 演算子（NT、NTG、NTP、NTI）を使用します。また、下位語の下位語、さらにその下位語、というようにシソーラス階層を下がるように問合せを拡張できます。

構文

構文	説明
NT(*term*[(*qualifier*)][,*level*][,*thes*])	シソーラスで term に対する下位語として定義されたすべての下位レベルの語句が含まれるように、問合せを拡張します。
NTG(*term*[(*qualifier*)][,*level*][,*thes*])	シソーラスで term に対する下位汎用語として定義されたすべての下位レベルの語句が含まれるように、問合せを拡張します。
NTP(*term*[(*qualifier*)][,*level*][,*thes*])	シソーラスで term に対する下位部分語として定義されたすべての下位レベルの語句が含まれるように、問合せを拡張します。
NTI(*term*[(*qualifier*)][,*level*][,*thes*])	シソーラスで term に対する下位インスタンス語として定義されたすべての下位レベルの語句が含まれるように、問合せを拡張します。

term
NARROWER TERM 演算子にオペランドを指定します。term は、thes によって指定されたシソーラスで、語句に対して定義された下位語のエントリが含まれるように拡張されます。この拡張に含まれる下位語の数は、level の値によって決定されます。拡張演算子は term 引数に指定できません。

qualifier
term が thes の同じ階層分岐で 2 つ以上のノードに現れる同形異義語（同じスペルで複数の意味を持つワードまたは句）の場合は、term に修飾子を指定します。

NARROWER TERM 問合せで同形異義語に修飾子が指定されていない場合は、すべての同形異義語の下位語が含まれるように問合せが拡張されます。

level
指定された語句に対する下位語を戻すために、シソーラス階層内を横断するレベル数を指定します。たとえば、NT のレベル 1 問合せは、下位語が定義されている場合、指定した問合せ語句のすべての下位語エントリを戻します。レベル 2 問合せは、指定した問合せ語句の下位語エントリ、さらにそれらの下位語の下位語エントリを戻します。

level 引数はオプションで、デフォルト値は（1）です。level 引数に 0（ゼロ）または負の値を設定すると、元の問合せ語句のみを戻します。

thes
指定された問合せ語句の拡張問合せを戻すために使用するシソーラス名を指定します。thes 引数はオプションで、デフォルト値は DEFAULT です。このデフォルト値を使用する場合は、シソーラス表に DEFAULT という名前のシソーラスが存在している必要があります。

注意： thes を指定する場合は、level も指定する必要があります。

例

次の問合せは、*cat* という問合せ語句、または DEFAULT シソーラスで *cat* の NT 語として定義された語句のいずれかが含まれているすべてのドキュメントを戻します。

```
'NT(cat)'
```

シソーラス名を指定する場合は、次のように level も指定する必要があります。

```
'NT(cat, 2, mythes)'
```

次の問合せは、*fairy tale* という問合せ語句、または DEFAULT シソーラスで *fairy tale* の下位インスタンス語として定義された語句のいずれかが含まれているすべてのドキュメントを戻します。

```
'NTI(fairy tale)'
```

cinderella という語句および *snow white* という語句が、*fairy tale* に対する下位語インスタンスとして定義されている場合、Oracle は、*fairy tale*、*cinderella* または *snow white* が含まれているドキュメントを戻します。

注意

シソーラスの各階層は個別の分離されたブランチを表し、4 つの NARROWER TERM 演算子に対応します。NARROWER TERM 問合せでは、指定された NARROWER TERM 演算子に対応するブランチを使用してその問合せを拡張するのみです。

関連項目

CTX_THES パッケージのプロシージャを使用して、シソーラスをブラウズできます。

関連項目： シソーラスの下位語のブラウズについては、第 12 章「CTX_THES パッケージ」の「NT」を参照してください。

NEAR（;）

2 つ以上の問合せ語句の出現位置の近さを基準にしてスコアを戻すには、NEAR 演算子を使用します。ドキュメント内で互いに近接した語句には高いスコアが、互いに離れた語句には低いスコアが戻ります。

注意： NEAR 演算子はワード問合せのみで機能します。ABOUT 問合せでは NEAR を使用できません。

構文

構文
NEAR((*word1*, *word2*,..., *wordn*) [, max_span [, order]])

word*1-n*

問合せ語句をカンマで区切って指定します。問合せ語句は、1 つのワードまたは句の場合もあります。

max_span

オプションで、最大クランプのサイズを指定します。デフォルト値は 100 です。100 より大きい数を指定するとエラーが戻ります。

クランプはすべての問合せ語句が出現する最小のワード・グループです。すべてのクランプは問合せ語句で始まり、問合せ語句で終わります。

2 つの問合せ語句を持つ NEAR 問合せでは、max_span がこの 2 つの語句間の最大距離になります。たとえば、*dog* が *cat* の 6 ワード以内にある場合、*dog* および *cat* について問い合せるには、次の問合せを発行します。

```
'near((dog, cat), 6)'
```

order

指定した順序で語句を検索するには、TRUE を指定します。デフォルトは FALSE です。

たとえば、最大クランプ・サイズを 20 にして、*monday*、*tuesday* および *wednesday* の順でワード検索をする場合は、次の問合せを発行します。

```
'near((monday, tuesday, wednesday), 20, TRUE)'
```

注意： order を指定するには、常に max_span パラメータに数を指定する必要があります。

order フラグ設定が異なる同じ問合せ式を使用すると、同じドキュメントに対して異なるスコアが戻る場合があります。たとえば、次の問合せを発行すると、同じドキュメントに対して異なるスコアが戻ります。

```
'near((dog, cat), 50, FALSE)'
'near((dog, cat), 50, TRUE)'
```

NEAR スコア付け

NEAR 演算子のスコア付けは、語句の出現頻度および語句の出現位置の近さを組み合せて行われます。問合せに適合する各ドキュメントに対して、ドキュメント内のクランプの数に比例し、そのクランプの平均サイズに反比例する 1 ～ 100 までのスコアが戻ります。小さなクランプは問合せ語句が近接していることを示すため、ドキュメントに小さなクランプが多く存在するほど高いスコアになります。

問合せ語句の数もスコアに影響を与えます。多くの語句（7 個程度）で問い合せると、少ない語句（2 個程度）で問い合せたときよりも、通常 100 のスコアを得るために必要なドキュメントのクランプ数が少なくなります。

クランプはすべての問合せ語句が出現する最小のワード・グループです。すべてのクランプは問合せ語句で始まり、問合せ語句で終わります。この項で説明するとおり、クランプ・サイズは max_span パラメータで定義できます。

NEAR およびその他の演算子

AND や OR など他の演算子とともに NEAR 演算子を使用できます。スコアは通常の方法で計算されます。

たとえば、語句 *lion* および *tiger* が互いに 10 ワード以内のところに位置する場合、語句 tiger、lion および cheetah を含むすべてのドキュメントを検索するには、次の問合せを発行します。

```
'near((lion, tiger), 10) AND cheetah'
```

戻される各ドキュメントのスコアは、NEAR 演算子および語句 *cheetah* のうち低い方のスコアになります。

EQUIVALENCE 演算子を使用して、NEAR 問合せの 1 つの問合せ語句を置換することもできます。

```
'near((stock crash, Japan=Korea), 20)'
```

この問合せは、*Japan* または *Korea* の 20 ワード以内に語句 *stock crash* を含むすべてのドキュメントを検索します。

下位互換性のある NEAR 構文

以前の ConText リリースの構文を使用して NEAR 問合せを記述できます。たとえば、*lion* が *tiger* の近くに出現するすべてのドキュメントを検索するには、次のように記述します。

```
'lion near tiger'
```

または、セミコロンを使用して次のようにも記述できます。

```
'lion;tiger'
```

この問合せは次の問合せと同等です。

```
'near((lion, tiger), 100, FALSE)'
```

注意： NEAR 演算子の構文のみが下位互換性を持ちます。この例では、戻されるスコアはこの項で説明したクランプ方法を使用して計算されます。

NEAR 演算子でのハイライト表示

ハイライト表示を使用し、問合せに NEAR 演算子が含まれている場合は、出現位置の近さの要件を満たすすべての語句がハイライト表示されます。ハイライト表示される語句は、1 つのワードまたは句の場合もあります。

たとえば、次のようなテキストを含むドキュメントがあるとします。

```
Chocolate and vanilla are my favorite ice cream flavors.  I like chocolate served in
a waffle cone, and vanilla served in a cup with carmel syrup.
```

問合せが *near((chocolate, vanilla)), 100, FALSE)* である場合、次の語句がハイライト表示されます。

```
<<Chocolate>> and <<vanilla>> are my favorite ice cream flavors.  I like
<<chocolate>> served in a waffle cone, and <<vanilla>> served in a cup with carmel
syrup.
```

ただし、問合せが *near((chocolate, vanilla)), 4, FALSE)* である場合、次の語句のみがハイライト表示されます。

```
<<Chocolate>> and <<vanilla>> are my favorite ice cream flavors.  I like chocolate
served in a waffle cone, and vanilla served in a cup with carmel syrup.
```

関連項目： ハイライト表示に使用できるプロシージャの詳細は、第 8 章「CTX_DOC パッケージ」を参照してください。

セクション検索および NEAR

次のように、WITHIN 演算子とともに NEAR 演算子を使用して、セクションを検索できます。

```
'near((dog, cat), 10) WITHIN Headings'
```

このような式を評価する場合は、指定されたセクション内全体にあるクランプが検索されます。

この例では、セクション *Headings* 内全体にある *dog* および *cat* を含むクランプのみがカウントされます。つまり、語句 *dog* が *Headings* 内にあり、語句 *cat* が *dog* から 5 ワードのところにあるが *Headings* の外にある場合、これら 2 つのワードは式を満たさず、カウントされません。

NOT（~）

1 つの問合せ語句が含まれているが、もう 1 つの問合せ語句が含まれていないドキュメントを検索するには、NOT 演算子（~）を使用します。

構文

構文	説明
term1~term2 *term1* not *term2*	*term1* が含まれるが、*term2* が含まれていないドキュメントを戻します。

例

たとえば、*animals* という語句が含まれていて *dogs* という語句は含まれていないドキュメントを取得するには、次の式を使用します。

```
'animals ~ dogs'
```

同様に、*transportation* という語句が含まれていて、*automobiles* または *trains* という語句は含まれていないドキュメントを取得するには、次の式を使用します。

```
'transportation not (automobiles or trains)'
```

注意： NOT 演算子は、他の論理演算子が作成したスコアには影響を与えません。

OR（|）

問合せ語句が 1 つ以上含まれるドキュメントを検索するには、OR 演算子を使用します。

構文

構文	説明
term1|*term2* *term1* or *term2*	*term1* または *term2* を含むドキュメントを戻します。そのオペランドの最大スコアを戻します。少なくともどちらか 1 つの問合せ語句が存在する必要があり、スコアの高い方が選択されます。

例

たとえば、*cats* または *dogs* という語句が含まれているドキュメントを取得するには、次の式のいずれかを使用します。

```
'cats | dogs'
'cats OR dogs'
```

スコア付け

OR 問合せでは、問合せ語句の最も高いスコアが戻ります。この例では、ドキュメント内での *cats* および *dogs* のスコアが、それぞれ 30 および 40 の場合、ドキュメントのスコアは 40 になります。

PREFERRED TERM（PT）

問合せ内の問合せ語句を、その問合せ語句の優先語としてシソーラスで定義された語句に置換するには、PREFERRED TERM 演算子（PT）を使用します。

構文

構文	説明
PT(*term*[,*thes*])	問合せで指定されたワードを *term* に対する優先語で置換します。

term
PREFERRED TERM 演算子にオペランドを指定します。term は、指定されたシソーラスで優先語として定義されている語句に置換されます。ただし、term に対して PT エントリが定義されていない場合は、問合せ式で term は置換されず、term が拡張の結果となります。

拡張演算子は term 引数に指定できません。

thes
指定された問合せ語句の拡張問合せを戻すために使用するシソーラス名を指定します。thes 引数はオプションで、デフォルト値は DEFAULT です。したがって、シソーラス演算子を使用するには、DEFAULT という名前のシソーラスがシソーラス表に存在している必要があります。

例

automobile はシソーラスに *car* という優先語があるとします。*automobile* という問合せ語句の PT 問合せを行うと、*car* というワードが含まれているすべてのドキュメントが戻ります。*automobile* というワードが含まれているドキュメントは戻りません。

関連項目

CTX_THES パッケージのプロシージャを使用して、シソーラスをブラウズできます。

> **関連項目：** シソーラスの優先語のブラウズについては、第 12 章「CTX_THES パッケージ」の「PT」を参照してください。

RELATED TERM（RT）

問合せ語句の関連語としてシソーラスに定義されたすべての語句が含まれるように問合せを拡張するには、RELATED TERM 演算子（RT）を使用します。

構文

構文	説明
RT(*term*[,*thes*])	シソーラスで term に対する関連語として定義されたすべての語句が含まれるように、問合せを拡張します。

term
RELATED TERM 演算子にオペランドを指定します。term、および thes で term に対して定義されたすべての関連エントリが含まれるように、term が拡張されます。

拡張演算子は term 引数に指定できません。

thes
指定された問合せ語句の拡張問合せを戻すために使用するシソーラス名を指定します。thes 引数はオプションで、デフォルト値は DEFAULT です。したがって、シソーラス演算子を使用するには、DEFAULT という名前のシソーラスがシソーラス表に存在している必要があります。

例

dog という語句には、*wolf* という関連語があるとします。*dog* に対して RT 問合せを行うと、*dog* および *wolf* というワードを含むすべてのドキュメントが戻ります。

関連項目

CTX_THES パッケージのプロシージャを使用して、シソーラスをブラウズできます。

> **関連項目：** シソーラスの関連語のブラウズについては、第 12 章「CTX_THES パッケージ」の「RT」を参照してください。

SOUNDEX（!）

発音が類似したワード（他のワードと似た発音のワード）が含まれるように問合せを拡張するには、SOUNDEX（!）演算子を使用します。この演算子を使用すると、スペルは異なっても英語では同じ発音であるワードを比較できます。

構文

構文	説明
!*term*	発音が同じすべての語句が、指定した問合せ語句に含まれるように問合せを拡張します（英語のテキストのみ）。

例

```
SELECT ID, COMMENT FROM EMP_RESUME
WHERE CONTAINS (COMMENT, '!SMYTHE') > 0 ;

ID COMMENT
-- ------------
23 Smith is a hard worker who..
```

言語

SOUNDEX は、7 ビットのキャラクタ・セットを使用する英語などの言語に最適です。8 ビットのキャラクタ・セットを使用する多くの西ヨーロッパの言語でも、多少効果は落ちますが使用できます。

テキスト列に基本文字変換が指定され、問合せ式に SOUNDEX 演算子が指定されている場合は、問合せの基本文字書式が処理されます。

STEM（$）

問合せ語句と同じ語根を持つ語を検索するには、STEM（$）演算子を使用します。

BASIC_LEXER プリファレンスの index_stems 属性を使用して、ステミングのパフォーマンスを改善できます。

Oracle Text のステマーは、Xerox 社の XSoft Division からライセンスを受けています。この機能は、英語、フランス語、スペイン語、イタリア語、ドイツ語およびオランダ語をサポートしています。

構文

構文	説明
$*term*	同じ語幹または語根を持つすべての語句が、指定した問合せ語句に含まれるように問合せを拡張します。

例

入力	拡張結果
$scream	scream screaming screamed
$distinguish	distinguish distinguished distinguishes
$guitars	guitars guitar
$commit	commit committed
$cat	cat cats
$sing	sang sung sing

ストップワードによる動作

STEM がストップワードとして指定したワードを戻す場合、そのストップワードは問合せの中には含まれず、CTX_QUERY.HIGHLIGHT または CTX_QUERY.MARKUP でハイライト表示もされません。

ストアド・クエリー式（SQE）

CTX_QUERY.STORE_SQE プロシージャで作成されたストアド・クエリー式をコールするには、SQE 演算子を使用します。

ストアド・クエリー式は、ドキュメントの編成および分類用の事前定義済みの BIN を作成する場合、または最初の問合せが 1 つ以上の追加問合せによって詳細化されている反復問合せを実行する場合に使用できます。

構文

構文	説明
SQE(*SQE_name*)	ストアド・クエリー式 *SQE_name* の結果を戻します。

例

disasters という名前の SQE を作成するには、CTX_QUERY.STORE_SQE を次のように使用します。

```
begin
ctx_query.store_sqe('disasters', 'hurricane or earthquake or blizzard');
end;
```

このストアド・クエリー式は、*hurricane*、*earthquake* または *blizzard* のいずれかを含むすべてのドキュメントを戻します。

これで、*SQE* は次のような問合せ式内でコールできます。

```
SELECT SCORE(1), docid FROM news
WHERE CONTAINS(resume, 'sqe(disasters)', 1)> 0
ORDER BY SCORE(1);
```

SYNONYM（SYN）

指定した語句のシノニムとしてシソーラスに定義されているすべての語句が含まれるように問合せを拡張するには、SYNONYM 演算子（SYN）を使用します。

構文

構文	説明
SYN(*term*[,*thes*])	シソーラスで *term* に対するシノニムとして定義されたすべての語句が含まれるように、問合せを拡張します。

term
SYNONYM 演算子にオペランドを指定します。term、および thes で term に対して定義されたすべてのシノニムが含まれるように、term が拡張されます。

拡張演算子は term 引数に指定できません。

thes
指定された問合せ語句の拡張問合せを戻すために使用するシソーラス名を指定します。thes 引数はオプションで、デフォルト値は DEFAULT です。このデフォルト値を使用する場合は、シソーラス表に DEFAULT という名前のシソーラスが存在している必要があります。

例

次の問合せ式は、*dog* という問合せ語句、または DEFAULT シソーラスで *dog* のシノニムとして定義された語句が含まれているすべてのドキュメントを戻します。

```
'SYN(dog)'
```

SYNONYM 演算子内のコンパウンド句

シノニム問合せで問合せ語句のコンパウンド句の拡張問合せは、AND 接続詞によって戻ります。

たとえば、*temperature* + *measurement* + *instruments* というコンパウンド句は、*thermometer* という問合せ語句のシノニムとしてシソーラスに定義されています。*thermometer* のシノニム問合せは次のように拡張されます。

```
{thermometer} OR ({temperature}&{measurement}&{instruments})
```

関連項目

CTX_THES パッケージのプロシージャを使用して、シソーラスをブラウズできます。

関連項目： シソーラスのシノニム語句のブラウズについては、第 12 章「CTX_THES パッケージ」の「SYN」を参照してください。

THRESHOLD（>）

THRESHOLD 演算子（>）は、次の 2 通りの方法で使用します。

- 式レベル
- 問合せ語句レベル

式レベルの THRESHOLD 演算子は、しきい値の数値を下回るスコアのドキュメントを結果セットから排除します。

問合せ語句レベルの THRESHOLD 値演算子は、ある語句がドキュメント内でどのようにスコアを出すかに基づいてドキュメントを選択します。

構文

構文	説明
expression>n	結果セットでスコアがしきい値 *n* を超えるドキュメントのみを戻します。
term>n	式内で、スコアが最低 *n* の問合せ語句が含まれているドキュメントを戻します。

例

式レベルで、*relational databases* が含まれているドキュメントを検索し、スコアが 75 よりも大きいドキュメントのみを戻すには、次の式を使用します。

```
'relational databases > 75'
```

問合せ語句レベルで、*lion* のスコアが 30 以上で、さらに *tiger* が含まれているドキュメントを検索するには、次の式を使用します。

```
'(lion > 30) and tiger'
```

TRANSLATION TERM（TR）

定義されたすべての外国語の等価語句が含まれるように問合せを拡張するには、TRANSLATION TERM（TR）演算子を使用します。

構文

構文	説明
TR(*term*[, *lang*, [*thes*]])	*term* に対して定義されたすべての外国語の等価語が含まれるように、*term* を拡張します。

term
TRANSLATION TERM 演算子にオペランドを指定します。term は、thes で term に対して定義されたすべての外国語エントリが含まれるように拡張されます。拡張演算子は term 引数に指定できません。

lang
オプションで、拡張によってどの外国語の等価語を戻すかを指定します。指定する言語は thes に定義されている言語と一致している必要があります。このパラメータを省略すると、定義されたすべての外国語の語句を使用するように拡張されます。

thes
オプションで、指定された問合せ語句の拡張問合せを戻すために使用するシソーラス名を指定します。thes 引数のデフォルト値は DEFAULT です。したがって、シソーラス演算子を使用するには、DEFAULT という名前のシソーラスがシソーラス表に存在している必要があります。

注意： thes を指定する場合は、lang も指定する必要があります。

例

cat に対して次のエントリを持つシソーラス MY_THES があるとします。

```
cat
  SPANISH: gato
  FRENCH:  chat
```

cat および *cat* のスペイン語訳を含むすべてのドキュメントを検索するには、次の問合せを発行します。

```
'tr(cat, spanish, my_thes)'
```

この問合せは次のように拡張されます。

```
'{cat}|{gato}|{chat}'
```

関連項目

CTX_THES パッケージのプロシージャを使用して、シソーラスをブラウズできます。

> **関連項目：** シソーラスの関連語のブラウズについては、第 12 章「CTX_THES パッケージ」の「TR」を参照してください。

TRANSLATION TERM SYNONYM（TRSYN）

問合せ語句に対して定義されたすべての外国語の等価語、問合せ語句のシノニム、およびそのシノニムの外国語の等価語が含まれるように問合せを拡張するには、TRSYN 演算子を使用します。

構文

構文	説明
TRSYN(*term*[, *lang*, [*thes*]])	term の外国語の等価語、term のシノニム、およびそのシノニムの外国語の等価語が含まれるように、term を拡張します。

term
この演算子にオペランドを指定します。term は、thes で term に対して定義されたすべての外国語エントリが含まれるように拡張されます。拡張演算子は term 引数に指定できません。

lang
オプションで、拡張によってどの外国語の等価語を戻すかを指定します。指定する言語は thes に定義されている言語と一致している必要があります。このパラメータを省略すると、定義されたすべての外国語の語句を使用するように拡張されます。

thes
オプションで、指定された問合せ語句の拡張問合せを戻すために使用するシソーラス名を指定します。thes 引数のデフォルト値は DEFAULT です。したがって、シソーラス演算子を使用するには、DEFAULT という名前のシソーラスがシソーラス表に存在している必要があります。

注意： thes を指定する場合は、lang も指定する必要があります。

例

cat に対して次のエントリを持つシソーラス MY_THES があるとします。

```
cat
  SPANISH: gato
  FRENCH:  chat
  SYN lion
    SPANISH: leon
```

cat、*cat* のスペイン語での等価語、*cat* のシノニム、および *lion* のスペイン語での等価語を含むすべてのドキュメントを検索するには、次の問合せを発行します。

```
'trsyn(cat, spanish, my_thes)'
```

この問合せは次のように拡張されます。

```
'{cat}|{gato}|{lion}|{leon}'
```

関連項目

CTX_THES パッケージのプロシージャを使用して、シソーラスをブラウズできます。

> **関連項目：** シソーラスの訳語およびシノニムのブラウズについては、第12章「CTX_THES パッケージ」の「TRSYN」を参照してください。

TOP TERM（TT）

問合せ語句を、シソーラス内の標準階層（BT、NT）でその問合せ語句に定義された最上位語と置換するには、TOP TERM 演算子（TT）を使用します。類種関係（BTG、NTG）、全体部分関係（BTP、NTP）およびインスタンス（BTI、NTI）階層の最上位語は戻りません。

構文

構文	説明
TT(*term*[,*thes*])	問合せで指定されたワードを、term に対する標準的な階層（BT、NT）での最上位語で置換します。

term
TOP TERM 演算子にオペランドを指定します。term は、指定されたシソーラスで最上位語として定義されている語句に置換されます。ただし、term に対して TT エントリが定義されていない場合は、問合せ式で term は置換されず、term が拡張の結果となります。

拡張演算子は term 引数に指定できません。

thes
指定された問合せ語句の拡張問合せを戻すために使用するシソーラス名を指定します。thes 引数はオプションで、デフォルト値は DEFAULT です。このデフォルト値を使用する場合は、シソーラス表に DEFAULT という名前のシソーラスが存在している必要があります。

例

dog という問合せ語句には、シソーラスで標準階層内の *animal* という最上位語があるとします。*dog* の TT 問合せは、*animal* という句が含まれているすべてのドキュメントを戻します。*dog* というワードが含まれているドキュメントは戻りません。

関連項目

CTX_THES パッケージのプロシージャを使用して、シソーラスをブラウズできます。

関連項目： シソーラスの最上位語のブラウズについては、第 12 章「CTX_THES パッケージ」の「TT」を参照してください。

WEIGHT（*）

WEIGHT 演算子は、指定した係数をスコアに掛けます。積の最高値は 100 です。たとえば、問合せ *cat, dog*2* は、*cat* のスコアと *dog* のスコアの 2 倍を合計します。スコアが 100 を超える場合、最高は 100 になります。

複数の問合せ語句が含まれている式では、WEIGHT 演算子を使用して問合せ語句の相対的なスコアを調整します。1 より小さい数と WEIGHT 演算子を使用すると、問合せ語句のスコアを減らすことができます。1 より大きく 10 より小さい数と WEIGHT 演算子を使用すると、問合せ語句のスコアを増やすことができます。

式に複数の問合せ語句がある場合、WEIGHT 演算子は、ACCUMULATE 問合せ、OR 問合せまたは AND 問合せで有効です。個々の語句に重みを付けないと、出現回数の最も多い問合せ語句をスコアから知ることができません。語句の重み付けにより、語句のスコアを個別に変更でき、関心度の高い語句がドキュメント全体のランク付けに高く反映されるようにできます。

構文

構文	説明
*term*n*	term が含まれているドキュメントを戻します。term のロー・スコアに n を掛けてスコアを計算します。n は、0.1 ～ 10 の数値です。

例

スポーツ記事の収集がある場合を考えます。サッカー（soccer）の記事のうち、特にブラジルのサッカー（Brazilian soccer）に関心があるとします。*soccer or Brazil* で通常の問合せをすると、US サッカーに関する多くの記事が高いランクで戻ります。ブラジルのサッカーについての記事のランクを上げるには、次の問合せを発行します。

```
'soccer or Brazil*3'
```

表 3-1 は、WEIGHT 演算子によって、サッカーの情報が含まれている 3 つの架空のドキュメント A、B および C のランクが変更できる様子を示しています。表の列は、3 つのドキュメントでの異なる 4 つの問合せ式の合計スコアを示します。

表 3-1

	soccer	Brazil	soccer or Brazil	soccer or Brazil*3
A	20	10	20	30
B	10	30	30	90
C	50	20	50	60

問合せ *soccer or Brazil* を含む 3 列目のスコアが、最も高いスコア付けを行う語句のスコアです。問合せ *soccer or Brazil**3 を含む 4 列目のスコアは、1 列目の *soccer* のスコア、および *Brazil* のスコアの 3 倍（*Brazil**3）のスコアのうち、大きい方のスコアです。

最初の問合せ *soccer or Brazil* では、ドキュメントは C、B、A の順にランク付けされます。問合せ *soccer or Brazil**3 では、ドキュメントは B、C、A という優先順にランク付けされます。

ワイルド・カード（% _）

問合せ式でワイルド・カード文字を使用すると、ワード検索をパターン検索に拡張できます。次のワイルド・カード文字があります。

ワイルド・カード文字	説明
%	ワイルド・カードとしてのパーセント記号は、その位置に任意の文字が複数現れることを示します。
_	ワイルド・カードとしてのアンダースコアは、その位置に任意の 1 文字が現れることを示します。

注意： ワイルド・カード式によってストップワードに変換された場合、ストップワードは問合せの中には含まれず、CTX_DOC.HIGHLIGHT または CTX_DOC.MARKUP でハイライト表示もされません。

右側切捨て問合せ

右側切捨てでは、検索文字列の右側にワイルド・カードが置かれます。

たとえば、次の問合せ式では、パターン *scal* で始まるすべての語句を検索します。

```
'scal%'
```

左側切捨て問合せおよび左右切捨て問合せ

左側切捨てでは、検索文字列の左側にワイルド・カードが置かれます。

king、*wing*、*sing* などのワードを検索するには、次の問合せを記述します。

```
'_ing'
```

通常、この問合せは次のように記述します。

```
'%ing'
```

また、左側切捨て検索および右側切捨て検索を組み合せて、左右切捨て検索を作成できます。次の問合せでは、サブストリング *%benz%* を含むワードのあるすべてのドキュメントを検索します。

```
'%benz%'
```

ワイルド・カード問合せのパフォーマンス改善

サブストリングまたはプリフィックス索引を追加すると、ワイルド・カード問合せのパフォーマンスを改善できます。

ワイルド・カード問合せが左側切捨ておよび左右切捨ての場合、サブストリング索引を作成すると、問合せパフォーマンスが向上します。サブストリング索引によって、*%ed*、*_ing*、*%benz%* などの、あらゆる種類の左側切捨てワイルド・カード検索の問合せパフォーマンスが向上します。

ワイルド・カード問合せが右側切捨ての場合は、プリフィックス索引を作成すると、パフォーマンスが向上します。プリフィックス索引によって、*to%* などのワイルド・カード検索の問合せパフォーマンスが向上します。

関連項目： サブストリングおよびプリフィックスの索引作成の詳細は、第 2 章の「BASIC_WORDLIST」を参照してください。

WITHIN

WITHIN 演算子を使用して、問合せをドキュメントのセクションに絞り込むことができます。ドキュメントのセクションは次のいずれかです。

- ゾーン・セクション
- フィールド・セクション
- 属性セクション
- 特殊セクション（文または段落）

構文

構文	説明
expression WITHIN *section*	事前定義済みのゾーン、フィールドまたは属性 section 内で expression を検索します。 セクションがゾーンの場合、expression は、セクションがゾーンまたは特殊セクションである WITHIN 演算子（ネストされた WITHIN）を 1 つ以上含むことができます。 セクションがフィールドまたは属性セクションの場合、式は他の WITHIN 演算子を含むことができません。
expression WITHIN SENTENCE	文中に expression を含むドキュメントを検索します。expression には AND または NOT 問合せを指定します。 expression は、セクションがゾーンまたは特殊セクションである WITHIN 演算子（ネストされた WITHIN）を 1 つ以上含むことができます。
expression WITHIN PARAGRAPH	段落中に expression を含むドキュメントを検索します。expression には AND または NOT 問合せを指定します。 expression は、セクションがゾーンまたは特殊セクションである WITHIN 演算子（ネストされた WITHIN）を 1 つ以上含むことができます。

例

ゾーン・セクション内の問合せ

セクション *Headings* 内で *San Francisco* という語句が含まれているすべてのドキュメントを検索するには、次のような問合せを記述します。

```
'San Francisco WITHIN Headings'
```

セクション *Headings* 内に *sailing* および *San Francisco* という語句が含まれているすべてのドキュメントを検索するには、次の 2 つの方法のいずれかで問合せを記述します。

```
'(San Francisco WITHIN Headings) and sailing'

'sailing and San Francisco WITHIN Headings'
```

WITHIN と複合式 同じセクション *Headings* 内に *dog* および *cat* という語句が含まれているすべてのドキュメントを検索するには、次の問合せを記述します。

```
'(dog and cat) WITHIN Headings'
```

この問合せと次の問合せは、論理的に異なります。

```
'dog WITHIN Headings and cat WITHIN Headings'
```

この問合せは、*dog* および *cat* という語句が *Headings* セクションにある場合に、*dog* および *cat* を含むすべてのドキュメントを、これらの語句が同じ *Headings* セクションに出現しているか異なるセクションに出現しているかに関係なく、検索します。

WITHIN と NEAR *Headings* セクション内で、*dog* が *cat* の近くにあるすべてのドキュメントを検索するには、次の問合せを記述します。

```
'dog near cat WITHIN Headings'
```

注意： NEAR 演算子は WITHIN 演算子よりも優先順位が高いため、この例では中カッコは必要ありません。この問合せは、*(dog near cat) WITHIN Headings* と同等です。

ネストされた WITHIN 問合せ

WITHIN 演算子をネストすると、ゾーン・セクション内のゾーン・セクションを検索できます。

たとえば、あるドキュメント・セットのゾーン・セクション BOOK 内に、ゾーン・セクション AUTHOR がネストされているとします。ネストされた WITHIN 問合せを次のように記述すると、BOOK セクションの AUTHOR セクションに出現する *scott* をすべて検索できます。

```
'(scott WITHIN AUTHOR) WITHIN BOOK'
```

フィールド・セクション内の問合せ

フィールド・セクション内の問合せの構文は、ゾーン・セクション内の問合せと同じです。前述の「ゾーン・セクション内の問合せ」で示したほとんどの例の構文は、フィールド・セクションに適用されます。

ただし、フィールド・セクションの動作は、次の点でゾーン・セクションとは異なります。

- 可視性 : テキストは、フィールド・セクション内で非表示にできます。
- 反復性 : WITHIN 問合せは、繰り返されたフィールド・セクションを識別できません。
- ネスト性 : ネストされた WITHIN 問合せは、フィールド・セクションで発行できません。

次に、これらの違いについて説明します。

フィールド・セクションの visible フラグ フィールド・セクションが、CTX_DDL.ADD_FIELD_SECTION で FALSE と設定された visible フラグで作成されている場合、フィールド・セクション内のテキストは、WITHIN 演算子を使用してのみ問合せできます。

たとえば、TITLE フィールド・セクションの visible フラグが FALSE に設定されているとします。この場合、WITHIN 演算子のない *dog* の問合せは、次の内容を含むドキュメントを検索しません。

```
<TITLE>The dog</TITLE> I like my pet.
```

このようなドキュメントを検索するには、次のように WITHIN 演算子を使用します。

```
'dog WITHIN TITLE'
```

また、TITLE を CTX_DDL.ADD_FIELD_SECTION でフィールド・セクションとして定義するときは、visible フラグを TRUE に設定できます。

> **関連項目 :** フィールド・セクション作成の詳細は、第 7 章「CTX_DDL パッケージ」の「ADD_FIELD_SECTION」を参照してください。

繰返しフィールド・セクション WITHIN 問合せは、ドキュメント内の繰り返されたフィールド・セクションを識別できません。たとえば、繰返しセクション <author> を持つドキュメントがあるとします。

```
<author> Charles Dickens </author>
<author> Martin Luther King </author>
```

<author> がフィールド・セクションとして定義されているとすると、(*charles and martin*) *within author* のような問合せは、これらのワードが別々のタグに出現しても、そのドキュメントを戻します。

WITHIN 問合せに繰返しセクションを識別させるには、セクションをゾーン・セクションとして定義します。

ネストされたフィールド・セクション ネストされた WITHIN 問合せは、フィールド・セクションで発行できません。これを行うと、エラーが戻ります。

文または段落内の問合せ

同じ文または段落で出現するワードの組合せを検索するには、文または段落境界内での問合せが有効です。文または段落を問い合せるには、索引付けする前に、特殊セクションをセクション・グループに追加する必要があります。これは、CTX_DDL.ADD_SPECIAL_SECTION で行います。

同じ文に *dog* および *cat* を含むドキュメントを検索するには、次のように問い合せます。

```
'(dog and cat) WITHIN SENTENCE'
```

同じ段落に *dog* および *cat* を含むドキュメントを検索するには、次のように問い合せます。

```
'(dog and cat) WITHIN PARAGRAPH'
```

ワード *dog* はあるが *cat* はない文を含むドキュメントを検索するには、次のように問い合せます。

```
'(dog not cat) WITHIN SENTENCE'
```

属性セクション内での問合せ

セクション・グループ・タイプとして XML_SECTION_GROUP または AUTO_SECTION_GROUP のいずれかで索引付けする場合、属性セクション内で問い合せることができます。

次のような XML ドキュメントがあるとします。

```
<book title="Tale of Two Cities">It was the best of times.</book>
```

セクション title@book は、属性セクション title になるように定義できます。これは、CTX_DLL.ADD_ATTR_SECTION プロシージャを使用して行うか、ALTER INDEX で索引付けしてから動的に行うことができます。

注意： AUTO_SECTION_GROUP を使用して XML ドキュメントを索引付けする場合、システムは自動的に属性セクションを作成し、これらを attribute@tag 形式でネーミングします。

XML_SECTION_GROUP を使用する場合、属性セクションは CTX_DDL.ADD_ATTR_SECTION で任意にネーミングできます。

属性セクション title 内で *Tale* を検索するには、次の問合せを発行します。

```
'Tale WITHIN title'
```

属性セクションの問合せに対する制約 属性セクション内の問合せには、次の制約が適用されます。

- 属性テキストに通常の問合せを行うと、WITHIN 句で修飾されていない場合は、ドキュメントにヒットしません。次のような XML ドキュメントがあるとします。

```
<book title="Tale of Two Cities">It was the best of times.</book>
```

Tale の問合せそのものは、WITHIN title@book で修飾されていない場合は、ドキュメントにヒットしません（この動作は、visible フラグを FALSE に設定した場合のフィールド・セクションと似ています）。

- 属性セクションは、ネストされた WITHIN 問合せでは使用できません。
- 句は属性テキストを無視します。たとえば、元のドキュメントが次のようなものであるとします。

```
Now is the time for all good <word type="noun"> men </word> to come to the aid.
```

この場合、このドキュメントは、介入的な属性テキストを無視して、通常の問合せ *good men* にヒットします。

- WITHIN 問合せは、繰返し属性セクションを識別できます。この動作は、ゾーン・セクションには似ていますが、フィールド・セクションには似ていません。たとえば、次のドキュメントがあるとします。

```
<book title="Tale of Two Cities">It was the best of times.</book>
<book title="Of Human Bondage">The sky broke dull and gray.</book>
```

book はゾーン・セクションで、book@author は属性セクションであるとします。次の問合せについて考えてみます。

```
'(Tale and Bondage) WITHIN book@author'
```

この問合せはドキュメントにヒットしません。これは、*tale* および *bondage* の出現する属性セクション book@author が異なるためです。

注意

セクション名

WITHIN 演算子では、検索するセクション名がわかっている必要があります。定義済みセクションのリストは、CTX_SECTIONS ビューまたは CTX_USER_SECTIONS ビューを使用して取得できます。

セクション境界

特殊セクションおよびゾーン・セクションについては、問合せの語句が、問合せを満たすためにドキュメントの特定のセクション内に出現する必要があります。フィールド・セクションにはこの要件は適用されません。

たとえば、*bold* がゾーン・セクションである場合、次のような問合せを考えます。

```
'(dog and cat) WITHIN bold'
```

この問合せは、次の語句を検索します。

```
<B>dog cat</B>
```

ただし、次の語句は検索しません。

```
<B>dog</B><B>cat</B>
```

これは、dog および cat が同じ *bold* セクション内に存在する必要があるためです。

この動作は、特に特殊セクションに有効です。たとえば、

```
'(dog and cat) WITHIN sentence'
```

は、同じ文中にある *dog* および *cat* を検索します。

一方、フィールド・セクションは、タイトル・セクションなど、繰返しのない埋込みメタデータ用です。問合せ文字列が存在する各フィールド・セクションは、異なるセクションとして区別できません。すべて同一セクション内に存在する文字列とみなされます。たとえば、次の問合せ

```
(dog and cat) WITHIN title
```

は、次のようなドキュメントを検索します。

<TITLE>dog</TITLE><TITLE>cat</TITLE>

このフィールド・セクション制限およびオーバーラップとネスト制限に対して、フィールド・セクション問合せは、特にセクションが各ドキュメントで出現する場合、または検索語句が共通の場合は、通常、ゾーン・セクション問合せよりも高速です。

制限事項

WITHIN 演算子には、次の制限事項があります。

- WITHIN 句を句に埋め込むことはできません。たとえば、*term1 WITHIN section term2* と記述することはできません。
- WITHIN 演算子は、$! や * などの拡張演算子と結合できません。
- WITHIN は予約語であるため、このワードを検索するには中カッコでワードを囲んでエスケープする必要があります。
- WITHIN 演算子は、*'ABOUT (xyz) WITHIN abc'* のように ABOUT 演算子と結合できません。

4

問合せの特殊文字

この章では、テキスト問合せに使用できる特殊文字について説明します。さらに、Oracle Text が予約語として処理するワードおよび文字を示します。

この章の内容は次のとおりです。

- グループ化文字
- エスケープ文字
- 予約語

グループ化文字

グループ化文字は、問合せ式の問合せ語句および演算子をグループ化することによって、演算子の優先順位を制御します。次のグループ化文字があります。

グループ化文字	説明
( )	カッコ文字は、文字間の語句および演算子をグループ化します。
[ ]	大カッコ文字は、文字間の語句および演算子をグループ化します。ただし、拡張演算子（FUZZY、SOUNDEX、STEM）に対しては機能しません。

問合せ語句および演算子のグループは、グループ化文字の左カッコで始まります。グループの終わりは、左カッコに対応する右カッコで示されます。右カッコと左カッコの間に、他のグループを挿入することもできます。

たとえば、左のカッコはグループの始まりを示します。最初に現れる右カッコが、グループの終わりです。右カッコよりも前に別の左カッコがあれば、ネストされたグループを示します。

エスケープ文字

and & *or* | *accum* など、問合せ式に対して特別な意味を持つワードまたは記号を問い合せるには、これらをエスケープする必要があります。問合せ式で文字をエスケープするには、次の 2 通りの方法があります。

エスケープ文字	説明
{}	中カッコを使用して、文字または記号の文字列をエスケープします。中カッコで囲まれたものが、エスケープ・シーケンスの部分とみなされます。 中カッコを使用して単一の文字をエスケープすると、エスケープされた文字が問合せ内の別のトークンになります。
\	バックスラッシュ文字を使用して、単一の文字または記号をエスケープします。バックスラッシュの直後の文字のみがエスケープされます。

次の例では、それぞれの式にテキスト演算子または予約記号が入っているため、エスケープ・シーケンスが必要です。

```
'AT\&T'
'{AT&T}'

'high\-voltage'
'{high-voltage}'
```

注意： 中カッコを使用してワード内の個々の文字をエスケープすると、文字はエスケープされますが、ワードは 3 つのトークンに分割されます。

たとえば、*high{-}voltage* と書かれた問合せでは、ハイフンの両側に空白がある *high - voltage* が検索されます。

エスケープ文字の問合せ

左中カッコ { は、エスケープ・シーケンスの始まりを示し、右中カッコ } はシーケンスの終わりを示します。左中カッコと右中カッコの間のすべて（すべての左中カッコを含む）が、エスケープされる問合せ式の部分です。エスケープされた問合せ式に右中カッコを組み込むには、}} を使用します。

バックスラッシュ・エスケープ文字をエスケープするには、\\ を使用します。

予約語

次の表に、CONTAINS 問合せでの検索時にエスケープする必要がある Oracle Text の予約語を示します。

予約語（ワード）	予約語（文字）	演算子
ABOUT	（なし）	ABOUT
ACCUM	,	ACCUMULATE
AND	&	AND
BT	（なし）	BROADER TERM
BTG	（なし）	BROADER TERM GENERIC
BTI	（なし）	BROADER TERM INSTANCE
BTP	（なし）	BROADER TERM PARTITIVE
FUZZY	?	FUZZY

予約語（ワード）	予約語（文字）	演算子
（なし）	{}	エスケープ文字（複数）
（なし）	\	エスケープ文字（単一）
（なし）	()	グループ化文字
（なし）	[]	グループ化文字
HASPATH	（なし）	HASPATH
INPATH	（なし）	INPATH
MINUS	-	MINUS
NEAR	;	NEAR
NOT	~	NOT
NT	（なし）	NARROWER TERM
NTG	（なし）	NARROWER TERM GENERIC
NTI	（なし）	NARROWER TERM INSTANCE
NTP	（なし）	NARROWER TERM PARTITIVE
OR	|	OR
PT	（なし）	PREFERRED TERM
RT	（なし）	RELATED TERM
（なし）	$	STEM
（なし）	!	SOUNDEX
SQE	（なし）	ストアド・クエリー式
SYN	（なし）	SYNONYM
（なし）	>	THRESHOLD
TR	（なし）	TRANSLATION TERM
TRSYN	（なし）	TRANSLATION TERM SYNONYM
TT	（なし）	TOP TERM
（なし）	*	WEIGHT
（なし）	%	ワイルド・カード文字（複数）
（なし）	_	ワイルド・カード文字（単一）
WITHIN	（なし）	WITHIN

5

CTX_ADM パッケージ

この章では、サーバーおよびデータ・ディクショナリを管理するための PL/SQL パッケージ CTX_ADM の使用方法について説明します。

CTX_ADM には、次のストアド・プロシージャが含まれています。

名前	説明
RECOVER	削除したテキスト表のデータベース・オブジェクトをクリーン・アップします。
SET_PARAMETER	索引作成のシステム・レベルのデフォルトを設定します。

注意： CTX_ADM のプロシージャを使用できるのは、CTXSYS ユーザーのみです。

RECOVER

RECOVER プロシージャは、テキスト・データ・ディクショナリをクリーン・アップし、残りのプリファレンスなどのオブジェクトを削除します。

構文

```
CTX_ADM.RECOVER;
```

例

```
begin
ctx_adm.recover;
end;
```

注意

ctxsrv サーバーが稼働している場合、CTX_ADM.RECOVER をコールしてシステムのリカバリを実行する必要はありません。稼働中のすべての *ctxsrv* サーバーは、約 15 分間隔で自動的にシステムのリカバリを実行します。RECOVER を使用すると、ユーザーはコマンドでリカバリを実行できます。

SET_PARAMETER

SET_PARAMETER プロシージャは、索引作成用のシステム・レベルのパラメータを設定します。

構文

```
CTX_ADM.SET_PARAMETER(param_name IN VARCHAR2,
                      param_value IN VARCHAR2);
```

param_name
設定するパラメータの名前を指定します。次のいずれかを指定できます。

- max_index_memory（索引付けに使用可能な最大メモリー）
- default_index_memory（索引付けに割り当てられたデフォルトのメモリー）
- log_directory（ctx_ouput ファイルのディレクトリ）
- ctx_doc_key_type（CTX_DOC プロシージャのデフォルトの入力キー型）
- file_access_role
- default_datastore（デフォルトのデータストア・プリファレンス）
- default_filter_file（ファイルにおけるデータストア用のデフォルトのフィルタ・プリファレンス）
- default_filter_text（デフォルトのテキスト・フィルタ・プリファレンス）
- default_filter_binary（デフォルトのバイナリ・フィルタ・プリファレンス）
- default_section_html（デフォルトの html セクション・グループ・プリファレンス）
- default_section_xml（デフォルトの xml セクション・グループ・プリファレンス)
- default_section_text（デフォルトのテキスト・セクション・グループ・プリファレンス）
- default_lexer（デフォルトのレクサー・プリファレンス）
- default_wordlist（デフォルトのワードリスト・プリファレンス）
- default_stoplist（デフォルトのストップリスト・プリファレンス）
- default_storage（デフォルトの記憶域プリファレンス）
- default_ctxcat_lexer
- default_ctxcat_stoplist

- default_ctxcat_storage
- default_ctxcat_wordlist
- default_ctxrule_lexer
- default_ctxrule_stoplist
- default_ctxrule_storage
- default_ctxrule_wordlist

> **関連項目：** これらのパラメータに対するデフォルト値の詳細は、第 2 章の「システム・パラメータ」を参照してください。

param_value
パラメータに割り当てる値を指定します。max_index_memory および default_index_memory に対しては、指定する値に次の構文が必要です。

number[M|G|K]

M は MB（メガバイト）、G は GB（ギガバイト）、K は KB（キロバイト）を表します。

その他の各パラメータに対しては、プリファレンス名を指定し、索引付けのデフォルトとして使用します。

例

```
begin
ctx_adm.set_parameter('default_lexer', 'my_lexer');
end;
```

6

CTX_CLS パッケージ

この章では、ドキュメントのセットに対する CTXRULE ルールを生成するための PL/SQL パッケージ CTX_CLS の使用方法について説明します。

名前	説明
TRAIN	ドキュメントのカテゴリを定義するルールを生成します。出力は、入力されたトレーニング・ドキュメント・セットに基づいて行われます。

TRAIN

このプロシージャを使用して、ドキュメントのカテゴリを選択する問合せルールを生成します。分類済みのドキュメントで構成されたトレーニング・セットを指定する必要があります。各ドキュメントが 1 つ以上のカテゴリに属している必要があります。このプロシージャは、カテゴリを定義する問合せを生成し、結果を表に書き込みます。

ドキュメント表には、関連付けられた CONTEXT 索引が移入されている必要があります。最適な結果を得るために、このプロシージャの実行前に索引を同期化してください。

ドキュメント表とカテゴリ表の準備も必要です。Oracle Text がサポートしているすべてのドキュメント形式を使用できます。

たとえば、ドキュメント表とカテゴリ表を次のように定義できます。

```
create table trainingdoc(
    docid number primary key,
    text varchar2(4000));

create table category (
    docid CONSTRAINT fk_id REFERENCES trainingdoc(docid),
    categoryid number);
```

構文

```
CTX_CLS.TRAIN(
    index_name in varchar2,
    doc_id in varchar2,
    cattab in varchar2,
    catdocid in varchar2,
    catid in varchar2,
    restab in varchar2,
    rescatid in varchar2,
    resquery in varchar2,
    resconfid in varchar2,
    preference_name in varchar2 DEFAULT NULL
);
```

index_name
ドキュメント・トレーニング・セットに関連付けられた CONTEXT 索引の名前を指定します。

doc_id
ドキュメント表内のドキュメント ID 列の名前を指定します。この列には一意のドキュメント ID が含まれている必要があります。この列は NUMBER データ型であることが必要です。

cattab
カテゴリ表の名前を指定します。この表に対する SELECT 権限が必要です。

catdocid
カテゴリ表内のドキュメント ID 列の名前を指定します。この表のドキュメント ID は、ドキュメント表にも存在している必要があります。この列は NUMBER データ型であることが必要です。

catid
カテゴリ表内のカテゴリ ID 列の名前を指定します。この列は NUMBER データ型であることが必要です。

restab
結果表の名前を指定します。この表に対する INSERT 権限が必要です。

rescatid
結果表内のカテゴリ ID 列の名前を指定します。この列は NUMBER データ型であることが必要です。

resquery
結果表内の問合せ列の名前を指定します。この列は VARACHAR2、CHAR CLOB、NVARCHAR2 または NCHAR データ型であることが必要です。

この列に生成された問合せは、次のように AND または NOT 演算子を使用して問合せ語句を接続します。

'T1 & T2 ~ T3'

問合せ語句はテーマ・トークンとしても使用でき、次のように ABOUT 演算子で接続されます。

'about(T1) & about(T2) ~ about(T3)'

resconfid
結果表内の信頼度列の名前を指定します。この列には、トレーニング・データから推測した確率が含まれます。この確率は、ドキュメントが問合せに十分答えているかどうかを示します。

preference_name
プリファレンスの名前を指定します。属性については、第 2 章「索引付け」の「分類型」を参照してください。

注意： 現在、次の列は 0（ゼロ）以上の整数値のみをサポートしています。

```
doc_id
catdocid
catid
rescatid
```

例

ドキュメント表には、関連付けられた CONTEXT 索引が存在している必要があります。たとえば、ドキュメント表を次のように定義して移入できます。

```
set serverout on
exec dbms_output.put_line(TO_CHAR(SYSDATE,'MM-DD-YYYY HH24:MI:SS')||':start');

create table doc (id number primary key, text varchar2(2000));
insert into doc values(1,'In 2002, Europe changed its currency to the EURO');
insert into doc values(2,'The NASDAQ rose today in heavy stock trading.');
insert into doc values(3,'The EURO lost 1 cent today against the US dollar');
insert into doc values(4,'Salt Lake City hosts the winter Olympic games');
insert into doc values(5,'ESPN broadcasts World Cup Soccer games.');
insert into doc values(6,'Soccer champion Diego Maradona retires.');
```

CONTEXT 索引を次のように作成します。

```
exec ctx_ddl.drop_preference('my_lexer');
exec ctx_ddl.create_preference('my_lexer','BASIC_LEXER');
exec ctx_ddl.set_attribute('my_lexer','INDEX_THEMES','NO');
exec ctx_ddl.set_attribute('my_lexer','INDEX_TEXT','YES');
CREATE INDEX docx on doc(text) INDEXTYPE IS ctxsys.context
PARAMETERS('LEXER my_lexer');
```

また、次のようにカテゴリ表を作成し、ドキュメントをカテゴリに関連付けます。

```
create table category (doc_id number, cat_id number, cat_name varchar2(100));
insert into category values (1,1,'Finance');
insert into category values (2,1,'Finance');
insert into category values (3,1,'Finance');
insert into category values (4,2,'Sports');
insert into category values (5,2,'Sports');
insert into category values (6,2,'Sports');
```

CTX_CLS.TRAIN が書き込む結果表は、次のように定義できます。

```
create table restab (cat_id number, query VARCHAR2(400), conf number);
```

後で CTXRULE 索引付けするように結果表を移入するには、RULE_CLASSIFIER プリファレンス属性を設定し、CTX_CLS.TRAIN を次のようにコールします。

```
exec ctx_ddl.drop_preference('my_classifier');
exec ctx_ddl.create_preference('my_classifier','RULE_CLASSIFIER');
exec ctx_ddl.set_attribute('my_classifier','MAX_TERMS','20');
exec ctx_ddl.set_attribute('my_classifier','THRESHOLD','40');
exec ctx_ddl.set_attribute('my_classifier','NT_THRESHOLD','0.02');
exec ctx_ddl.set_attribute('my_classifier','MEMORY_SIZE','200');
exec ctx_ddl.set_attribute('my_classifier','TERM_THRESHOLD','20');
exec ctx_output.start_log('mylog');

exec
ctx_cls.train('docx','id','category','doc_id','cat_id','restab','cat_ id','query',
'conf','my_classifier');
exec ctx_output.end_log();

create table catname as (select distinct cat_id, cat_name from category);

set termout on
select rpad(id,6) doc_id , rpad(cat_name,8) cat_name, rpad(text,50) text
 from doc, category where id=doc_id;
select rpad(a.cat_id,8) cat_id, rpad(cat_name,8) cat_name,  rpad(query,30) rule
 from restab a, catname b where b.cat_id=a.cat_id;
```

トレーニング・セットは次のとおりです。

```
DOC_ID CAT_NAME TEXT
------ -------- --------------------------------------------------
1      Finance  In 2002, Europe changed its currency to the EURO
2      Finance  The NASDAQ rose today in heavy stock trading.
3      Finance  The EURO lost 1 cent today against the US dollar
4      Sports   Salt Lake City hosts the winter Olympic games
5      Sports   ESPN broadcasts World Cup Soccer games.
6      Sports   Soccer champion Diego Maradona retires.

6 rows selected.
```

次のように、FINANCE と SPORTS のカテゴリに対するルールが生成されます。

```
CAT_ID   CAT_NAME RULE
-------- -------- ------------------------------
1        Finance  EURO
1        Finance  TODAY ~ EURO
2        Sports   GAMES
2        Sports   SOCCER ~ GAMES
```

7

CTX_DDL パッケージ

この章では、テキスト索引に必要なプリファレンス、セクション・グループおよびストップリストを作成および管理するための PL/SQL パッケージ、CTX_DDL の使用方法について説明します。

CTX_DDL には、次のストアド・プロシージャおよびファンクションが含まれています。

名前	説明
ADD_ATTR_SECTION	属性セクションをセクション・グループに追加します。
ADD_FIELD_SECTION	フィールド・セクションを作成し、それを指定されたセクション・グループに割り当てます。
ADD_INDEX	索引をカタログ索引プリファレンスに追加します。
ADD_SPECIAL_SECTION	特殊セクションをセクション・グループに追加します。
ADD_STOPCLASS	ストップクラスをストップリストに追加します。
ADD_STOP_SECTION	停止セクションを自動セクション・グループに追加します。
ADD_STOPTHEME	ストップテーマをストップリストに追加します。
ADD_STOPWORD	ストップワードをストップリストに追加します。
ADD_SUB_LEXER	サブレクサーをマルチレクサー・プリファレンスに追加します。
ADD_ZONE_SECTION	ゾーン・セクションを作成し、それを指定されたセクション・グループに追加します。
CREATE_INDEX_SET	CTXCAT 索引タイプの索引セットを作成します。
CREATE_POLICY	ora:contains() で使用するポリシーを作成します。
CREATE_PREFERENCE	テキスト・データ・ディクショナリにプリファレンスを作成します。

名前	説明
CREATE_SECTION_GROUP	テキスト・データ・ディクショナリにセクション・グループを作成します。
CREATE_STOPLIST	ストップリストを作成します。
DROP_INDEX_SET	索引セットを削除します。
DROP_POLICY	ポリシーを削除します。
DROP_PREFERENCE	テキスト・データ・ディクショナリからプリファレンスを削除します。
DROP_SECTION_GROUP	テキスト・データ・ディクショナリからセクション・グループを削除します。
DROP_STOPLIST	ストップリストを削除します。
OPTIMIZE_INDEX	索引を最適化します。
REMOVE_INDEX	CTXCAT 索引プリファレンスから索引を削除します。
REMOVE_SECTION	セクション・グループからセクションを削除します。
REMOVE_STOPCLASS	セクション・グループからストップクラスを削除します。
REMOVE_STOPTHEME	ストップリストからストップテーマを削除します。
REMOVE_STOPWORD	セクション・グループからストップワードを削除します。
SET_ATTRIBUTE	プリファレンスの属性を設定します。
SYNC_INDEX	索引を同期化します。
UNSET_ATTRIBUTE	プリファレンスから属性の設定を削除します。
UPDATE_POLICY	ポリシーを更新します。

ADD_ATTR_SECTION

属性セクションを XML セクション・グループに追加します。このプロシージャは、XML ドキュメントの属性をセクションとして定義する場合に有効です。これによって、WITHIN 演算子を使用して XML 属性テキストを検索できます。

> **注意：** AUTO_SECTION_GROUP を使用すると、属性セクションが自動的に作成されます。自動的に作成された属性セクションは、tag@attribute という形式でネーミングされます。

構文

```
CTX_DDL.ADD_ATTR_SECTION(
  group_name     in    varchar2,
  section_name   in    varchar2,
  tag            in    varchar2);
```

group_name
XML セクション・グループの名前を指定します。属性セクションは、XML セクション・グループにのみ追加できます。

section_name
属性セクションの名前を指定します。この名前は、属性テキストに対する WITHIN 問合せで使用されます。

指定するセクション名には、コロン（:）、カンマ（,）またはドット（.）は使用できません。またセクション名は、group_name の中で一意であることが必要です。セクション名の大 / 小文字は区別されません。

属性セクション名の長さは、64 バイト以内にしてください。

tag
tag@attr という形式で属性の名前を指定します。このパラメータでは、大 / 小文字が区別されます。

例

次のように、TITLE 属性を持つ BOOK タグを定義する XML ファイルがあるとします。

```
<BOOK TITLE="Tale of Two Cities">
  It was the best of times.
</BOOK>
```

タイトル属性を属性セクションとして定義するには、次のように XML_SECTION_GROUP を作成し、属性セクションを定義します。

```
begin
ctx_ddl_create_section_group('myxmlgroup', 'XML_SECTION_GROUP');
ctx_ddl.add_attr_section('myxmlgroup', 'booktitle', 'BOOK@TITLE');
end;
```

このように TITLE 属性セクションを定義し、ドキュメント・セットを索引付けすると、次のように XML 属性テキストを問い合せることができます。

```
'Cities within booktitle'
```

ADD_FIELD_SECTION

フィールド・セクションを作成し、そのセクションを既存のセクション・グループに追加します。これによって、WITHIN 演算子を使用してフィールド・セクション内を検索できます。

フィールド・セクションは開始および終了タグで区切られます。デフォルトでは、フィールド・セクション内のテキストは、ドキュメントの残りの部分とは別のサブドキュメントとして索引付けされます。

ゾーン・セクションとは異なり、フィールド・セクションはネストまたはオーバーラップできません。このため、フィールド・セクションは非繰返しセクション、非オーバーラップ・セクション（電子メール型またはニュース型ドキュメントの TITLE や AUTHOR マークアップなど）に最適です。

フィールド・セクションの索引付けによって、フィールド・セクションに対する WITHIN 問合せは、通常、ゾーン・セクションに対する WITHIN 問合せより高速になります。

構文

```
CTX_DDL.ADD_FIELD_SECTION(
  group_name     in    varchar2,
  section_name   in    varchar2,
  tag            in    varchar2,
  visible        in    boolean default FALSE
);
```

group_name

section_name を追加するセクション・グループの名前を指定します。1 つのセクション・グループに対して最大 64 のフィールド・セクションを追加できます。同じグループ内では、セクション・ゾーン名とセクション・フィールド名は同じ名前にできません。

section_name

group_name に追加するセクションの名前を指定します。この名前は、問合せでセクションを識別する場合に使用します。英数字以外の文字（_ など）は問合せでエスケープされるため、これらの文字を含む名前は使用しないでください。セクション名の大 / 小文字は区別されません。

同じグループ内では、ゾーン・セクション名とフィールド・セクション名は同じ名前にできません。語句 *PARAGRAPH* および *SENTENCE* は、特殊セクション用に予約されています。

セクション名はタグ内で一意である必要はありません。詳細を検索しやすいように、同じセクション名を複数のタグに割り当てることができます。

tag
セクションの開始をマークするタグを指定します。たとえば、タグが <H1> の場合は、H1 を指定します。指定する開始タグは、セクション・グループ内で一意であることが必要です。

group_name が HTML_SECTION_GROUP の場合は、META タグの NAME/CONTENT 属性の組に対してフィールド・セクションを作成できます。それには、tag を meta@namevalue として指定します。この場合、namevalue は NAME 属性の値で、対応する CONTENT 属性がセクションとして索引付けられます。次の例を参照してください。

Oracle は、セクション・グループ作成時に指定した group_type パラメータから終了タグを認識します。

visible
ドキュメントの残りの部分でテキストを参照できるようにするには、TRUE を指定します。

デフォルトでは、visible フラグは FALSE です。これは、フィールド・セクション内のテキストがドキュメントの残りの部分とは別のサブドキュメントとして索引付けされることを意味します。 ただし、フィールド・セクション内のテキストをドキュメント全体の一部として索引付けする場合は、visible フラグを TRUE に設定できます。

例

Visible なフィールド・セクションおよび Invisible なフィールド・セクション

次のコードは、BASIC_SECTION_GROUP 型のセクション・グループ basicgroup を定義します。次に、<A> タグに対して Author というフィールド・セクションを basicgroup に作成します。また、visible フラグは FALSE に設定します。

```
begin
    ctx_ddl.create_section_group('basicgroup', 'BASIC_SECTION_GROUP');
    ctx_ddl.add_field_section('basicgroup', 'Author', 'A', FALSE);
end;
```

Author フィールド・セクションは visible フラグを FALSE に設定したため、Author セクションのテキストを検索するには、次のように WITHIN 演算子を使用する必要があります。

```
'(Martin Luther King) WITHIN Author'
```

WITHIN 演算子を使用しないで *Martin Luther King* を問い合せると、フィールド・セクションにこの語句のインスタンスが戻りません。WITHIN を指定しないでフィールド・セクション内のテキストを問い合せる場合は、セクション作成時に次のように visible フラグを TRUE に設定する必要があります。

```
begin
ctx_ddl.add_field_section('basicgroup', 'Author', 'A', TRUE);
end;
```

<META> タグに対するセクションの作成

HTML_SECTION_GROUP を使用する場合は、META タグに対してセクションを作成できます。

次のように、META タグを持つ HTML ドキュメントがあるとします。

```
<META NAME="author" CONTENT="ken">
```

<META NAME="author"> タグに対して CONTENT 属性を索引付けたフィールド・セクションを作成します。

```
begin
ctx_ddl.create_section_group('myhtmlgroup', 'HTML_SECTION_GROUP');
ctx_ddl.add_field_section('myhtmlgroup', 'author', 'META@AUTHOR');
end
```

セクション・グループ mygroup で索引付けした後、次のようにドキュメントを問い合せることができます。

```
'ken WITHIN author'
```

制限事項

ネストされたセクション

フィールド・セクションはネストできません。たとえば、フィールド・セクションを <TITLE> で始まるように定義し、別のフィールド・セクションを <FOO> で始まるように定義した場合、この 2 つのセクションは次のようにネストすることはできません。

```
<TITLE> dog <FOO> cat </FOO> </TITLE>
```

ネストされたセクションを使用するには、これらをゾーン・セクションとして定義します。

繰返しセクション

繰返しフィールド・セクションは使用できますが、WITHIN 問合せはこれらを 1 つのセクションとして処理します。次に、ドキュメントの繰返しフィールド・セクションの例を示します。

```
<TITLE> cat </TITLE>
<TITLE> dog </TITLE>
```

問合せ *dog and cat within title* は、これらのワードが別々のセクション内に出現する場合にも、ドキュメントを戻します。

WITHIN 問合せで繰返しセクションを区別するには、これらをゾーン・セクションとして定義します。

関連項目

第 3 章「CONTAINS 問合せ演算子」の「WITHIN」演算子

第 2 章「索引付け」の「セクション・グループ型」

「CREATE_SECTION_GROUP」

「ADD_ZONE_SECTION」

「ADD_SPECIAL_SECTION」

「REMOVE_SECTION」

「DROP_SECTION_GROUP」

ADD_INDEX

索引をカタログ索引プリファレンスに追加します。CTXCAT 型のカタログ索引を作成するには、このプリファレンスを作成します。

構文

```
CTX_DDL.ADD_INDEX(set_name in varchar2,
    column_list varchar2,
    storage_clause varchar2);
```

set_name
索引セットの名前を指定します。

column_list
索引にカンマで区切られた列のリストを指定します。

storage_clause
記憶域句を指定します。

例

次のスキーマを使用して AUCTION という表について考えます。

```
create table auction(
    item_id number,
    title varchar2(100),
    category_id number,
    price number,
    bid_close date);
```

表の問合せに、category_id に対する必須のテキスト問合せ句とオプションの構造化条件が含まれているとします。結果は、bid_close に基づいてソートする必要があります。

ユーザーが入力する可能性がある様々なタイプの構造化問合せをサポートするために、カタログ索引を作成できます。

索引を作成する場合は、最初に索引セット・プリファレンスを作成した後、そのプリファレンスに必要な索引を追加します。

```
begin
ctx_ddl.create_index_set('auction_iset');
ctx_ddl.add_index('auction_iset','bid_close');
ctx_ddl.add_index('auction_iset','category_id, bid_close');
ctx_ddl.add_index('auction_iset','price, bid_close');
end;
```

結合されたカタログ索引は、CREATE INDEX を使用して次のように作成します。

```
create index auction_titlex on AUCTION(title) indextype is CTXCAT parameters ('index
set auction_iset');
```

問合せ

ワード *pokemon* の title 列を問い合せるには、次のように、通常の複合問合せを発行できます。

```
select * from AUCTION where CATSEARCH(title, 'pokemon',NULL)> 0;
select * from AUCTION where CATSEARCH(title, 'pokemon', 'category_id=99 order by
bid_close desc')> 0;
```

ADD_SPECIAL_SECTION

特殊セクション（SENTENCE または PARAGRAPH）をセクション・グループに追加します。これによって、WITHIN 演算子を使用して、ドキュメントの文または段落内を検索できるようになります。

ドキュメントの特殊セクションは、ゾーン・セクションおよびフィールド・セクションとして明示的にタグ付けされていないセクションです。索引が作成されると、特殊セクションの開始および終了が検出されます。Oracle では、このような 2 つのセクション *PARAGRAPH* および *SENTENCE* をサポートします。

文および段落の境界は、レクサーが判断します。たとえば、レクサーは SENTENCE セクションまたは PARAGRAPH セクションの境界を次のように認識します。

表 7-1

特殊セクション	境界
SENTENCE	WORD/PUNCT/WHITESPACE
	WORD/PUNCT/NEWLINE
PARAGRAPH	WORD/PUNCT/NEWLINE/WHITESPACE（インデント・スタイルの段落）
	WORD/PUNCT/NEWLINE/NEWLINE（ブロック・スタイルの段落）

punctuation、whitespace および newline の各文字は、レクサーの設定によって判断されます。これらの設定は変更できます。

レクサーが境界を認識できない場合、SENTENCE セクションまたは PARAGRAPH セクションはいずれも索引付けされません。

構文

```
CTX_DDL.ADD_SPECIAL_SECTION(
                group_name    IN VARCHAR2,
                section_name  IN VARCHAR2);
```

group_name
セクション・グループの名前を指定します。

section_name
SENTENCE または PARAGRAPH を指定します。

例

次のコードは、HTML ドキュメントの文内での検索を可能にします。

```
begin
ctx_ddl.create_section_group('htmgroup', 'HTML_SECTION_GROUP');
ctx_ddl.add_special_section('htmgroup', 'SENTENCE');
end;
```

文検索の他にゾーン検索を可能にするために、ゾーン・セクションをグループに追加することもできます。次の例では、ゾーン・セクション Headline をセクション・グループ htmgroup に追加します。

```
begin
ctx_ddl.create_section_group('htmgroup', 'HTML_SECTION_GROUP');
ctx_ddl.add_special_section('htmgroup', 'SENTENCE');
ctx_ddl.add_zone_section('htmgroup', 'Headline', 'H1');
end;
```

ドキュメント内での文検索または段落検索のみが必要で、ゾーン・セクションまたはフィールド・セクションを定義する必要がない場合は、次のように NULL_SECTION_GROUP を使用できます。

```
begin
ctx_ddl.create_section_group('nullgroup', 'NULL_SECTION_GROUP');
ctx_ddl.add_special_section('nullgroup', 'SENTENCE');
end;
```

関連項目

第 3 章「CONTAINS 問合せ演算子」の「WITHIN」演算子

第 2 章「索引付け」の「セクション・グループ型」

「CREATE_SECTION_GROUP」

「ADD_ZONE_SECTION」

「ADD_FIELD_SECTION」

「REMOVE_SECTION」

「DROP_SECTION_GROUP」

ADD_STOPCLASS

ストップクラスをストップリストに追加します。ストップクラスは、索引付けされていないトークンのクラスです。

構文

```
CTX_DDL.ADD_STOPCLASS(
  stoplist_name  in   varchar2,
  stopclass      in   varchar2
);
```

stoplist_name
ストップリストの名前を指定します。

stopclass
stoplist_name に追加するストップクラスを指定します。現在は、NUMBERS クラスのみがサポートされています。

ストップリストに追加できるストップワード、ストップテーマおよびストップクラスの数の最大値は 4095 です。

例

次のコードは、ストップクラス NUMBERS をストップリスト mystop に追加します。

```
begin
ctx_ddl.add_stopclass('mystop', 'NUMBERS');
end;
```

関連項目

「CREATE_STOPLIST」

「REMOVE_STOPCLASS」

「DROP_STOPLIST」

ADD_STOP_SECTION

停止セクションを自動セクション・グループに追加します。停止セクションを追加すると、自動セクションの索引付け操作で、XML ドキュメント内の指定されたセクションが無視されます。

> **注意：** 停止セクションを追加すると、索引内にセクション情報が作成されません。ただし、停止セクション内のテキストは常に検索可能です。

停止セクションを追加することは、ドキュメントに下位レベルの情報タグが多数含まれる場合に有効です。また、停止セクションを追加すると、自動セクション・グループを使用した索引付けのパフォーマンスが向上します。

追加できる停止セクションの数は無制限です。

停止セクションにはセクション名がありません。したがって、セクション・ビューには記録されません。

構文

```
CTX_DDL.ADD_STOP_SECTION(
                section_group IN VARCHAR2,
                tag  IN VARCHAR2);
```

section_group
自動セクション・グループの名前を指定します。自動セクション・グループを指定しない場合、このプロシージャはエラーを戻します。

tag
索引付け時に無視するタグを指定します。このパラメータでは、大 / 小文字が区別されます。また、このように停止タグを定義すると、タグの属性セクションがある場合は、これも停止します。

タグは、(doctype)tag という形式のドキュメント・タイプで修飾できます。たとえば、`<fluff>` タグを mydoc ドキュメント・タイプ内のみの停止セクションにする場合は、`(mydoc)fluff` をタグに指定します。

例

停止セクションの定義

次のコードは、タグ <fluff> で識別された停止セクションを自動セクション・グループ myauto に追加します。

```
begin
ctx_ddl.add_stop_section('myauto', 'fluff');
end;
```

また、このコードは、<fluff> 内に含まれるすべての属性セクションも停止します。たとえば、次のコードを持つドキュメントがあるとします。

```
<fluff type="computer">
```

前述のコードは、属性セクション fluff@type も停止します。

ドキュメント・タイプを区別する停止セクション

次のコードは、mydoc のルート要素を持つドキュメント内のみで、タグ <fluff> に対する停止セクションを作成します。

```
begin
ctx_ddl.add_stop_section('myauto', '(mydoc)fluff');
end;
```

関連項目

第 1 章「SQL 文と演算子」の「ALTER INDEX」

「CREATE_SECTION_GROUP」

ADD_STOPTHEME

単一のストップテーマをストップリストに追加します。ストップテーマは、索引付けされていないテーマです。

英語では、ABOUT 演算子を使用して索引テーマを問い合せます。

構文

```
CTX_DDL.ADD_STOPTHEME(
  stoplist_name  in   varchar2,
  stoptheme      in   varchar2
);
```

stoplist_name
ストップリストの名前を指定します。

stoptheme
stoplist_name に追加するストップテーマを指定します。システムではナレッジ・ベースを使用して、入力されたストップテーマを正規化します。正規化されたテーマが複数の場合、システムではそのストップテーマを処理しません。したがって、単一のストップテーマを送信することをお薦めします。

ストップリストに追加できるストップワード、ストップテーマおよびストップクラスの数の最大値は 4095 です。

例

次の例では、ストップテーマ banking をストップリスト mystop に追加します。

```
begin
ctx_ddl.add_stoptheme('mystop', 'banking');
end;
```

関連項目

「CREATE_STOPLIST」

「REMOVE_STOPTHEME」

「DROP_STOPLIST」

第 3 章「CONTAINS 問合せ演算子」の「ABOUT」演算子

ADD_STOPWORD

単一のストップワードをストップリストに追加します。

ストップワードのリストを作成するには、ワードごとにこのプロシージャを 1 回コールする必要があります。

構文

```
CTX_DDL.ADD_STOPWORD(
    stoplist_name  in   varchar2,
    stopword       in   varchar2,
    language       in varchar2 default NULL
);
```

stoplist_name
ストップリストの名前を指定します。

stopword
追加するストップワードを指定します。

言語固有のストップワードは、その言語固有の他のストップワードとの間で一意であることが必要です。たとえば、ドイツ語の *die* と英語の *die* を同じストップリスト内に含めることは有効です。

ストップリストに追加できるストップワード、ストップテーマおよびストップクラスの数の最大値は 4095 です。

language
stoplist_name で指定するストップリストが MULTI_STOPLIST 型の場合は、stopword の言語を指定します。Oracle がサポートしている言語のグローバリゼーション・サポート名または略称を指定する必要があります。

ストップワードを複数の言語でアクティブにするには、このパラメータに ALL を指定します。たとえば、任意の言語で停止が必要な英語のフラグメントを含む国際文書の場合は、ALL ストップワードを定義すると便利です。

ALL ストップワードは、すべての言語でアクティブになります。マルチレクサーを使用すると、そのストップワードの言語固有のレクサー処理が発生します。この状態は、そのストップワードを複数の特定言語に繰り返し追加した場合と同じです。

マルチレクサーを使用しない場合は、NULL を指定します。

例

単一言語のストップリスト

次の例では、ストップワード *because*、*notwithstanding*、*nonetheless* および *therefore* をストップリスト mystop に追加します。

```
begin
    ctx_ddl.add_stopword('mystop', 'because');
    ctx_ddl.add_stopword('mystop', 'notwithstanding');
    ctx_ddl.add_stopword('mystop', 'nonetheless');
    ctx_ddl.add_stopword('mystop', 'therefore');
end;
```

マルチ言語のストップリスト

次の例では、ドイツ語のワード *die* をマルチ言語のストップリストに追加します。

```
begin
    ctx_ddl.add_stopword('mystop', 'Die','german');
end;
```

注意： ストップワードは、ALTER INDEX を使用して索引を作成した後に追加できます。

ALL ストップワードの追加

次の例では、ワード *the* を ALL ストップワードとして、マルチ言語のストップリスト *globallist* に追加します。

```
begin
    ctx_ddl.add_stopword('globallist','the','ALL');
end;
```

関連項目

「CREATE_STOPLIST」

「REMOVE_STOPWORD」

「DROP_STOPLIST」

第 1 章「SQL 文と演算子」の「ALTER INDEX」

付録 D「提供されるストップリスト」

ADD_SUB_LEXER

サブレクサーをマルチレクサー・プリファレンスに追加します。サブレクサーは、マルチレクサー（マルチ言語）・プリファレンスで言語を識別します。複数の言語を索引付けする必要がある場合は、マルチレクサー・プリファレンスを使用します。

制限事項

CTX_DDL.ADD_SUB_LEXER の使用には、次の制限が適用されます。

- 起動ユーザーは、マルチレクサーまたは CTXSYS の所有者である必要があります。
- lexer_name パラメータには、マルチレクサー・レクサーであるプリファレンスが指定されている必要があります。
- マルチレクサーを索引で使用できるようにするには、デフォルトのレクサーが定義されている必要があります。
- サブレクサー・プリファレンスの所有者は、マルチレクサー・プリファレンスの所有者と同じである必要があります。
- サブレクサー・プリファレンスは、マルチレクサー・レクサーにはできません。
- サブレクサー・プリファレンスは、マルチレクサー・プリファレンスで使用されている間は削除できません。
- CTX_DDL.ADD_SUB_LEXER は、参照のみを記録します。サブレクサーの値は索引作成時に、索引値の記憶域にコピーされます。

構文

```
CTX_DDL.ADD_SUB_LEXER(
             lexer_name in varchar2,
             language  in varchar2,
             sub_lexer in varchar2,
             alt_value in varchar2 default null
);
```

lexer_name
マルチレクサー・プリファレンスの名前を指定します。

language
サブレクサーのグローバリゼーション・サポート言語の名前または略称を指定します。たとえば、英語に対して ENGLISH または EN を指定できます。

sub_lexer で指定したサブレクサーは、言語列が、language の略称グローバリゼーション・サポート名と同等の値（大 / 小文字を区別しない）を持つ場合に使用されます。

元表内の言語列の値が NULL、無効またはサブレクサーにマップされていない場合は、DEFAULT を指定し、デフォルトのサブレクサーを割り当てます。DEFAULT レクサーも、ストップワードの解析に使用されます。

指定した language に対するサブレクサー定義がすでに存在する場合は、このコールで置換されます。

sub_lexer
この言語に対して使用するサブレクサーの名前を指定します。

alt_value
オプションで、language の代替値を指定します。

language に対して DEFAULT を指定した場合、alt_value は指定できません。

alt_value は 30 バイトに制限されており、グローバリゼーション・サポート言語の名前、略称または DEFAULT を指定することはできません。

例

この例は、マルチ言語テキスト表の作成方法、および表を索引付けするためのマルチレクサーの設定方法を示しています。

次のように、主キー、テキスト列および言語列を持つマルチ言語表を作成します。

```
create table globaldoc (
   doc_id number primary key,
   lang varchar2(3),
   text clob
);
```

保持するドキュメントのほとんどが英語で、ドイツ語または日本語のドキュメントが少しある表を考えてみます。3 つの言語を処理するには、英語、ドイツ語および日本語に対して 1 つずつの、3 つのサブレクサーを作成する必要があります。

```
ctx_ddl.create_preference('english_lexer','basic_lexer');
ctx_ddl.set_attribute('english_lexer','index_themes','yes');
ctx_ddl.set_attribtue('english_lexer','theme_language','english');
```

```
ctx_ddl.create_preference('german_lexer','basic_lexer');
ctx_ddl.set_attribute('german_lexer','composite','german');
ctx_ddl.set_attribute('german_lexer','mixed_case','yes');
ctx_ddl.set_attribute('german_lexer','alternate_spelling','german');

ctx_ddl.create_preference('japanese_lexer','japanese_vgram_lexer');
```

マルチレクサー・プリファレンスを作成します。

```
ctx_ddl.create_preference('global_lexer', 'multi_lexer');
```

格納されているドキュメントのほとんどが英語であるため、英語のレクサーをデフォルトにします。

```
ctx_ddl.add_sub_lexer('global_lexer','default','english_lexer');
```

ドイツ語および日本語のレクサーをそれぞれの言語に追加します。また、言語列が ISO 639-2 で表現されている場合は、これらを代替値として追加します。

```
ctx_ddl.add_sub_lexer('global_lexer','german','german_lexer','ger');
ctx_ddl.add_sub_lexer('global_lexer','japanese','japanese_lexer','jpn');
```

パラメータ文字列に、次のようにマルチレクサー・プリファレンスおよび言語列を指定して索引 globalx を作成します。

```
create index globalx on globaldoc(text) indextype is ctxsys.context
parameters ('lexer global_lexer language column lang');
```

ADD_ZONE_SECTION

ゾーン・セクションを作成し、そのセクションを既存のセクション・グループに追加します。これによって、WITHIN 演算子を使用してゾーン・セクションを検索できます。

ゾーン・セクションは、開始および終了タグで区切られたセクションです。たとえば、HTML の <B> および </B> タグは、ボールド体で表示されるワードの範囲をマークします。

ゾーン・セクションは、ゾーン・セクション内で互いにネストしたりオーバーラップすることが可能です。また、1 つのドキュメント内で複数出現させることが可能です。

構文

```
CTX_DDL.ADD_ZONE_SECTION(
  group_name     in    varchar2,
  section_name   in    varchar2,
  tag            in    varchar2
);
```

group_name
section_name を追加するセクション・グループの名前を指定します。

section_name
group_name に追加するセクションの名前を指定します。この名前は、WITHIN 問合せでセクションを識別する場合に使用します。英数字以外の文字（_ など）の多くは特殊であり、問合せでエスケープされるため、これらの文字を含む名前は使用しないでください。セクション名の大 / 小文字は区別されません。

同じグループ内では、ゾーン・セクション名とフィールド・セクション名は同じ名前にできません。語句 *PARAGRAPH* および *SENTENCE* は、特殊セクション用に予約されています。

セクション名はタグ内で一意である必要はありません。詳細を検索しやすいように、同じセクション名を複数のタグに割り当てることができます。

tag
セクションの開始をマークするパターンを指定します。たとえば、<H1> が HTML タグの場合は、tag に H1 を指定します。指定する開始タグは、セクション・グループ内で一意であることが必要です。

Oracle は、セクション・グループ作成時に指定した group_type パラメータから終了タグを認識します。

group_name が HTML_SECTION_GROUP の場合は、META タグの NAME/CONTENT 属性の組に対してゾーン・セクションを作成できます。それには、tag を meta@namevalue として指定します。この場合、namevalue は NAME 属性の値で、対応する CONTENT 属性がセクションとして索引付けられます。次の例を参照してください。

group_name が XML_SECTION_GROUP の場合は、オプションで、(doctype)tag という形式でドキュメント・タイプ（ルート要素）を使用して tag を修飾できます。その場合は、XML ドキュメント・タイプの宣言での section_name が区別されます。次の例を参照してください。

例

HTML のセクションの作成

次のコードは、HTML_SECTION_GROUP 型の htmgroup というセクション・グループを定義します。その後、<H1> タグで識別された headline という htmgroup にゾーン・セクションを作成します。

```
begin
ctx_ddl.create_section_group('htmgroup', 'HTML_SECTION_GROUP');
ctx_ddl.add_zone_section('htmgroup', 'heading', 'H1');
end;
```

セクション・グループ htmgroup で索引付けした後、次のように問合せを発行して heading セクション内を問い合せることができます。

```
'Oracle WITHIN heading'
```

<META NAME> タグに対するセクションの作成

HTML_SECTION_GROUP を使用すると、HTML META タグにゾーン・セクションを作成できます。

次のように、META タグを持つ HTML ドキュメントがあるとします。

```
<META NAME="author" CONTENT="ken">
```

META タグに対してすべての CONTENT 属性を索引付けるゾーン・セクションを作成します。この場合、META タグの NAME 値は author です。

```
begin
ctx_ddl.create_section_group('htmgroup', 'HTML_SECTION_GROUP');
ctx_ddl.add_zone_section('htmgroup', 'author', 'meta@author');
end
```

セクション・グループ htmgroup で索引付けした後、次のようにドキュメントを問い合せることができます。

```
'ken WITHIN author'
```

ドキュメント・タイプを区別するセクションの作成（XML ドキュメントのみ）

異なるドキュメント・タイプに対して宣言された <book> タグを持つ XML ドキュメント・セットがあるとします。各ドキュメント・タイプに対して個別の book セクションを作成する必要がある場合を考えてみます。

mydocname が XML ドキュメント・タイプ（ルート要素）として、次のように宣言されているとします。

```
<!DOCTYPE mydocname ... [...
```

mydocname の中で、要素 <book> が宣言されています。このタグに対し、タグのドキュメント・タイプを区別する mybooksec という名前のセクションを次のように作成できます。

```
begin
ctx_ddl.create_section_group('myxmlgroup', 'XML_SECTION_GROUP');
ctx_ddl.add_zone_section('myxmlgroup', 'mybooksec', '(mydocname)book');
end;
```

注意

繰返しセクション

ゾーン・セクションは繰返しが可能です。検索条件と一致した各文字列は別々のセクションとして処理されます。たとえば、<H1> が heading セクションを示す場合は、次のように同じドキュメント内で繰り返すことができます。

<H1> The Brown Fox </H1>

<H1> The Gray Wolf </H1>

これらのゾーン・セクションが Heading という名前の場合、問合せ *Brown WITHIN Heading* はこのドキュメントを戻します。ただし、*(Brown and Gray) WITHIN Heading* という問合せはできません。

セクションのオーバーラップ

ゾーン・セクションは互いにオーバーラップできます。たとえば、<B> および <I> が 2 つの異なるゾーン・セクションを表す場合、これらはドキュメント内で次のようにオーバーラップできます。

```
plain <B> bold <I> bold and italic </B> only italic </I>  plain
```

ネストされたセクション

ゾーン・セクションは（それ自体も含む）、次のようにネストできます。

```
<TD> <TABLE><TD>nested cell</TD></TABLE></TD>
```

WITHIN 演算子を使用して、セクション内のセクションのテキストを検索するための問合せを記述できます。たとえば、BOOK1、BOOK2 および AUTHOR のゾーン・セクションが、ドキュメント doc1 および doc2 で次のように存在するとします。

doc1:

```
<book1> <author>Scott Tiger</author> This is a cool book to read.<book1>
```

doc2:

```
<book2> <author>Scott Tiger</author> This is a great book to read.<book2>
```

次のようにネストされた問合せを実行します。

```
'Scott within author within book1'
```

この問合せは doc1 のみを戻します。

関連項目

第 3 章「CONTAINS 問合せ演算子」の「WITHIN」演算子

第 2 章「索引付け」の「セクション・グループ型」

「CREATE_SECTION_GROUP」

「ADD_FIELD_SECTION」

「ADD_SPECIAL_SECTION」

「REMOVE_SECTION」

「DROP_SECTION_GROUP」

CREATE_INDEX_SET

CTXCAT 索引タイプの索引セットを作成します。CTXCAT 索引の作成時に、CREATE INDEX の PARAMETERS 句でこの索引セット名を指定します。

構文

```
CTX_DDL.CREATE_INDEX_SET(set_name in    varchar2);
```

set_name
索引セットの名前を指定します。CTXCAT 索引の作成時に、CREATE INDEX の PARAMETERS 句でこの索引セット名を指定します。

CREATE_POLICY

ora:contains ファンクションで使用するポリシーを作成します。ora:contains は、existsNode を使用した XPath 問合せ式内で使用するファンクションです。

関連項目： 『Oracle9*i* アプリケーション開発者ガイド - XML』

構文

```
CTX_DDL.CREATE_POLICY(
         policy_name     IN VARCHAR2 DEFAULT NULL,
         filter          IN VARCHAR2 DEFAULT NULL,
         section_group   IN VARCHAR2 DEFAULT NULL,
         lexer           IN VARCHAR2 DEFAULT NULL,
         stoplist        IN VARCHAR2 DEFAULT NULL,
         wordlist        IN VARCHAR2 DEFAULT NULL);
```

policy_name
新しいポリシーの名前を指定します。

filter
使用するフィルタ・プリファレンスを指定します。

注意： このリリースでは、このパラメータは使用されません。

section_group
使用するセクション・グループを指定します。指定できるのは NULL_SECTION_GROUP のみです。特殊（SENTENCE および PARAGRAPH）セクションのみサポートされています。

lexer
使用するレクサー・プリファレンスを指定します。INDEX_THEMES 属性を使用禁止にする必要があります。

stoplist
使用するストップリストを指定します。

wordlist
使用するワードリストを指定します。

例

mylex という名前の mylex レクサー・プリファレンスを作成します。

```
begin
   ctx_ddl.create_preference('mylex', 'BASIC_LEXER');
   ctx_ddl.set_attribute('mylex', 'printjoins', '_-');
   ctx_ddl.set_attribute ( 'mylex', 'index_themes', 'NO');
   ctx_ddl.set_attribute ( 'mylex', 'index_text', 'YES');
end;
```

mystop という名前のストップリスト・プリファレンスを作成します。

```
begin
  ctx_ddl.create_stoplist('mystop', 'BASIC_STOPLIST');
  ctx_ddl.add_stopword('mystop', 'because');
  ctx_ddl.add_stopword('mystop', 'nonetheless');
  ctx_ddl.add_stopword('mystop', 'therefore');
end;
```

'mywordlist' という名前のワードリスト・プリファレンスを作成します。

```
begin
 ctx_ddl.create_preference('mywordlist', 'BASIC_WORDLIST');
 ctx_ddl.set_attribute('mywordlist','FUZZY_MATCH','ENGLISH');
 ctx_ddl.set_attribute('mywordlist','FUZZY_SCORE','0');
 ctx_ddl.set_attribute('mywordlist','FUZZY_NUMRESULTS','5000');
 ctx_ddl.set_attribute('mywordlist','SUBSTRING_INDEX','TRUE');
 ctx_ddl.set_attribute('mywordlist','STEMMER','ENGLISH');
end;


exec ctx_ddl.create_policy('my_policy', NULL, NULL, 'mylex', 'mystop',
'mywordlist');
```

または

```
exec ctx_ddl.create_policy(policy_name => 'my_policy',
                           lexer => 'mylex',
                           stoplist => 'mystop',
                           wordlist => 'mywordlist');
```

次に、独自に定義したポリシーを使用して、existsNode() 問合せを次のように発行できます。

```
select id from xmltab
  where existsNode(doc, '/book/chapter[ ora:contains(summary,"dog or cat", "my_
policy") >0 ]', 'xmlns:ora="http://xmlns.oracle.com/xdb" ')=1;
```

ポリシーを更新するには、次のようにします。

```
exec ctx_ddl.update_policy(policy_name => 'my_policy', lexer => 'my_new_lex');
```

ポリシーを削除するには、次のようにします。

```
exec ctx_ddl.drop_policy(policy_name => 'my_policy');
```

CREATE_PREFERENCE

テキスト・データ・ディクショナリにプリファレンスを作成します。プリファレンスは、CREATE INDEX または ALTER INDEX のパラメータ文字列で指定します。

構文

```
CTX_DDL.CREATE_PREFERENCE(preference_name  in varchar2,
                          object_name      in varchar2);
```

preference_name
作成するプリファレンスの名前を指定します。

object_name
プリファレンス型の名前を指定します。

> **関連項目：** プリファレンス型および関連する属性の完全なリストは、第 2 章「索引付け」を参照してください。

例

テキストのみの索引の作成

次の例では、テキストのみの索引を指定するレクサー・プリファレンスを作成します。この例では、CTX_DDL.CREATE_PREFERENCE を使用して my_lexer という BASIC_LEXER プリファレンスを作成します。その後、CTX_DDL.SET_ATTRIBUTE を 2 回コールします。最初に INDEX_TEXT 属性に対して YES を指定し、次に INDEX_THEMES 属性に対して NO を指定します。

```
begin
ctx_ddl.create_preference('my_lexer', 'BASIC_LEXER');
ctx_ddl.set_attribute('my_lexer', 'INDEX_TEXT', 'YES');
ctx_ddl.set_attribute('my_lexer', 'INDEX_THEMES', 'NO');
end;
```

ファイル・データ記憶域の指定

次の例では、索引付けするファイルがオペレーティング・システムに格納されていることをシステムに知らせる mypref というデータ記憶域プリファレンスを作成します。その後、CTX_DDL.SET_ATTRIBUTE を使用し、PATH 属性をディレクトリ /docs に設定します。

```
begin
ctx_ddl.create_preference('mypref', 'FILE_DATASTORE');
ctx_ddl.set_attribute('mypref', 'PATH', '/docs');
end;
```

関連項目： データ記憶域の詳細は、第 2 章「索引付け」の「データストア型」を参照してください。

マスター表 / ディテール表の関係の作成

CTX_DDL.CREATE_PREFERENCE を使用して、DETAIL_DATASTORE を持つプリファレンスを作成します。CTX_DDL.SET_ATTRIBUTE を使用して、このプリファレンスに対して属性を設定します。次に、これを行う方法を示します。

```
begin
ctx_ddl.create_preference('my_detail_pref', 'DETAIL_DATASTORE');
ctx_ddl.set_attribute('my_detail_pref', 'binary', 'true');
ctx_ddl.set_attribute('my_detail_pref', 'detail_table', 'my_detail');
ctx_ddl.set_attribute('my_detail_pref', 'detail_key', 'article_id');
ctx_ddl.set_attribute('my_detail_pref', 'detail_lineno', 'seq');
ctx_ddl.set_attribute('my_detail_pref', 'detail_text', 'text');
end;
```

関連項目： マスター表 / ディテール表の詳細は、第 2 章「索引付け」の「DETAIL_DATASTORE」を参照してください。

記憶域属性の指定

次の例では、索引表が 1KB の初期エクステントで foo 表領域に作成されるように指定します。

```
begin
ctx_ddl.create_preference('mystore', 'BASIC_STORAGE');
ctx_ddl.set_attribute('mystore', 'I_TABLE_CLAUSE',
                      'tablespace foo storage (initial 1K)');
ctx_ddl.set_attribute('mystore', 'K_TABLE_CLAUSE',
                      'tablespace foo storage (initial 1K)');
ctx_ddl.set_attribute('mystore', 'R_TABLE_CLAUSE',
                      'tablespace foo storage (initial 1K)');
ctx_ddl.set_attribute('mystore', 'N_TABLE_CLAUSE',
                      'tablespace foo storage (initial 1K)');
ctx_ddl.set_attribute('mystore', 'I_INDEX_CLAUSE',
                      'tablespace foo storage (initial 1K)');
end;
```

関連項目： 第 2 章「索引付け」の「記憶域型」

属性のないプリファレンスの作成

属性のない型を持つプリファレンスを作成する場合は、次の例で示すように、フィルタを NULL_FILTER に設定したプリファレンスを作成する必要があります。

```
begin
ctx_ddl.create_preference('my_null_filter', 'NULL_FILTER');
end;
```

関連項目

「SET_ATTRIBUTE」

「DROP_PREFERENCE」

第 1 章「SQL 文と演算子」の「CREATE INDEX」

第 1 章「SQL 文と演算子」の「ALTER INDEX」

第 2 章「索引付け」

CREATE_SECTION_GROUP

テキスト列にセクションを定義するためのセクション・グループを作成します。

セクション・グループを作成する場合は、ADD_ZONE_SECTION、ADD_FIELD_SECTION または ADD_SPECIAL_SECTION を使用してそのセクション・グループにゾーン、フィールドまたは特殊セクションを追加できます。

索引付けする場合は、CREATE INDEX または ALTER INDEX のパラメータ文字列にセクション・グループを指定します。

索引付けした後、定義したセクション内を WITHIN 演算子を使用して問い合せることができます。

構文

```
CTX_DDL.CREATE_SECTION_GROUP(
  group_name     in    varchar2,
  group_type     in    varchar2
);
```

group_name
[user.]section_group_name として作成するセクション・グループ名を指定します。このパラメータは、1 人の所有者内で一意であることが必要です。

group_type
セクション・グループのタイプを指定します。group_type パラメータは、次のいずれかです。

セクション・グループ・プリファレンス	説明
NULL_SECTION_GROUP	どのセクションも定義しないか、または SENTENCE か PARAGRAPH セクションのみを定義する場合は、このグループ・タイプを使用します。これはデフォルトです。
BASIC_SECTION_GROUP	このグループ・タイプを使用して、開始および終了タグが <A> および </A> という形式のセクションを定義します。 **注意**: このグループ・タイプでは、対になっていないカッコ、コメント・タグおよび属性などの入力はサポートされません。これらを入力するには、HTML_SECTION_GROUP を使用してください。
HTML_SECTION_GROUP	このグループ・タイプを使用して、HTML ドキュメントを索引付けし、HTML ドキュメントのセクションを定義します。

セクション・グループ・プリファレンス	説明
XML_SECTION_GROUP	このグループ・タイプを使用して、XML ドキュメントを索引付けし、XML ドキュメントのセクションを定義します。
AUTO_SECTION_GROUP	このグループ・タイプを使用して、XML ドキュメントの開始タグ / 終了タグに対して自動的にゾーン・セクションを作成します。XML タグから導出されるセクション名は、XML 内と同様に大 / 小文字が区別されます。 属性セクションは、属性を持つ XML タグに対して自動的に作成されます。属性セクションは、attribute@tag という形式でネーミングされます。 停止セクション、空のタグ、処理の指示およびコメントは、索引付けされません。 自動セクション・グループには次の制限事項が適用されます。 ■ ゾーン、フィールドまたは特殊セクションは、自動セクション・グループに追加できません。 ■ 自動セクション化は、XML ドキュメント・タイプ（ルート要素）を索引付けしません。ただし、ドキュメント・タイプに停止セクションを定義することはできます。 ■ プリフィックスおよび名前空間を含む、索引付けされたタグの長さは、64 文字以下です。これより長いタグは索引付けされません。
PATH_SECTION_GROUP	このグループ・タイプを使用して、XML ドキュメントを索引付けします。このタイプは、AUTO_SECTION_GROUP と同じように動作します。 相違点は、このセクション・グループを使用すると、INPATH および HASPATH 演算子でパス検索を実行できることです。タグまたは属性名の問合せでは、大 / 小文字が同様に区別されます。
NEWS_SECTION_GROUP	このグループ・タイプを使用して、RFC 1036 に従ったニュース・グループ形式のドキュメントのセクションを定義します。

例

次のコマンドは、HTML グループ・タイプを使用して htmgroup というセクション・グループを作成します。

```
begin
    ctx_ddl.create_section_group('htmgroup', 'HTML_SECTION_GROUP');
end;
```

次のコマンドは、AUTO_SECTION_GROUP グループ・タイプを使用して auto というセクション・グループを作成し、XML ドキュメントのタグを自動的に索引付けします。

```
begin
    ctx_ddl.create_section_group('auto', 'AUTO_SECTION_GROUP');
end;
```

関連項目

第 3 章「CONTAINS 問合せ演算子」の「WITHIN」演算子

第 2 章「索引付け」の「セクション・グループ型」

「ADD_ZONE_SECTION」

「ADD_FIELD_SECTION」

「ADD_SPECIAL_SECTION」

「REMOVE_SECTION」

「DROP_SECTION_GROUP」

CREATE_STOPLIST

新しい空のストップリストを作成します。ストップリストには、索引付けしないワードまたはテーマを含めることができます。

言語固有のストップワードを保持するマルチ言語のストップリストを作成することもできます。マルチ言語のストップリストは英語、ドイツ語および日本語などの異なる言語のドキュメントを含む表の索引付けに有効です。索引付けを実行する場合は、テキスト表に言語列が含まれている必要があります。

ADD_STOPWORD、ADD_STOPCLASS または ADD_STOPTHEME を使用して、ストップワード、ストップクラスまたはストップテーマのいずれかをストップリストに追加できます。

ストップリストは、CREATE INDEX または ALTER INDEX のパラメータ文字列に指定し、デフォルトのストップリスト CTXSYS.DEFAULT_STOPLIST をオーバーライドできます。

構文

```
CTX_DDL.CREATE_STOPLIST(
    stoplist_name IN VARCHAR2,
    stoplist_type IN VARCHAR2 DEFAULT 'BASIC_STOPLIST');
```

stoplist_name
作成するストップリストの名前を指定します。

stoplist_type
単一言語のストップリストを作成するには、BASIC_STOPLIST を指定します。これはデフォルトです。

言語固有のストップワードを持つストップリストを作成するには、MULTI_STOPLIST を指定します。

索引付け時に、各ドキュメントの言語列が調べられ、その言語のストップワードのみが排除されます。問合せ時に、セッション言語の設定によって、アクティブなストップワードが特定されます。これは、マルチレクサーの使用時にアクティブなレクサーが特定されるのと同様です。

> **注意：** マルチ言語のストップリストを持つマルチ言語表を索引付けする場合は、表に言語列が含まれている必要があります。

例

単一言語のストップリスト

次のコードは、mystop というストップリストを作成します。

```
begin
ctx_ddl.create_stoplist('mystop', 'BASIC_STOPLIST');
end;
```

マルチ言語のストップリスト

次のコードは、multistop というマルチ言語のストップリストを作成し、次に 2 つの言語固有のストップワードを追加します。

```
begin
ctx_ddl.create_stoplist('multistop', 'MULTI_STOPLIST');
ctx_ddl.add_stopword('mystop', 'Die','german');
ctx_ddl.add_stopword('mystop', 'Or','english');
end;
```

関連項目

「ADD_STOPWORD」

「ADD_STOPCLASS」

「ADD_STOPTHEME」

「DROP_STOPLIST」

第 1 章「SQL 文と演算子」の「CREATE INDEX」および「ALTER INDEX」

付録 D「提供されるストップリスト」

DROP_INDEX_SET

索引セットを削除します。

構文

```
CTX_DDL.DROP_INDEX_SET(set_name in varchar2);
```

set_name
削除する索引セットの名前を指定します。

DROP_POLICY

CREATE_POLICY で作成したポリシーを削除します。

構文

```
CTX_DDL.DROP_POLICY(policy_name IN VARCHAR2);
```

policy_name
削除するポリシーの名前を指定します。

DROP_PREFERENCE

DROP_PREFERENCE プロシージャは、指定したプリファレンスをテキスト・データ・ディクショナリから削除します。プリファレンスを削除しても、そのプリファレンスを使用して作成された索引には影響を与えません。

構文

```
CTX_DDL.DROP_PREFERENCE(preference_name IN VARCHAR2);
```

preference_name
削除するプリファレンスの名前を指定します。

例

次のコードは、プリファレンス my_lexer を削除します。

```
begin
ctx_ddl.drop_preference('my_lexer');
end;
```

関連項目

「CREATE_PREFERENCE」

DROP_SECTION_GROUP

DROP_SECTION_GROUP プロシージャは、指定したセクション・グループおよびそのグループ内のすべてのセクションをテキスト・データ・ディクショナリから削除します。

構文

```
CTX_DDL.DROP_SECTION_GROUP(group_name IN VARCHAR2);
```

group_name
削除するセクション・グループの名前を指定します。

例

次のコードは、セクション・グループ htmgroup およびそのすべてのセクションを削除します。

```
begin
ctx_ddl.drop_section_group('htmgroup');
end;
```

関連項目

「CREATE_SECTION_GROUP」

DROP_STOPLIST

テキスト・データ・ディクショナリからストップリストを削除します。ストップリストを削除した場合は、変更を有効にするために索引を再作成または再構築する必要があります。

構文

```
CTX_DDL.DROP_STOPLIST(stoplist_name in varchar2);
```

stoplist_name
ストップリストの名前を指定します。

例

次のコードは、ストップリスト mystop を削除します。

```
begin
ctx_ddl.drop_stoplist('mystop');
end;
```

関連項目

「CREATE_STOPLIST」

OPTIMIZE_INDEX

索引を最適化します。索引の最適化は、同期化の後に行います。索引を最適化すると、古いデータが削除され、索引の断片化が最小限度に抑えられます。索引の最適化によって、問合せ応答時間を短縮できます。

高速モード、完全モードまたはトークン・モードで最適化できます。トークン・モードでは、最適化対象の特定のトークンを指定します。トークン・モードでは、参照頻度の低いトークンの最適化に要する時間を節約しつつ、検索頻度の高い索引トークンを最適化できます。トークンを最適化すると、そのトークンの問合せ応答時間が短縮できます。

> **注意：** 索引を最適化することによって、初期の索引操作後、元表に対するドキュメントの挿入、削除または更新操作のみが必要な場合は、応答時間がかなり短縮できます。

索引の最適化には、ALTER INDEX 文よりもこのプロシージャを使用することをお薦めします。

制限事項

CTX_DDL.OPTIMIZE_INDEX プロシージャでは、最大 16,000 のドキュメント ID が最適化されます。最大数を超えるドキュメント ID の最適化を続行するには、このプロシージャを再実行してください。

構文

```
CTX_DDL.OPTIMIZE_INDEX(
    idx_name  IN  VARCHAR2,
    optlevel  IN  VARCHAR2,
    maxtime   IN  NUMBER DEFAULT NULL,
    token     IN VARCHAR2 DEFAULT NULL,
    part_name IN VARCHAR2 DEFAULT NULL,
    parallel_degree IN VARCHAR2);
);
```

idx_name
索引の名前を指定します。索引名を指定しない場合は、Oracle によって最適化する索引が 1 つ選択されます。

optlevel
最適化レベルを文字列で指定します。最適化には、次のいずれかの方法を指定できます。

値	説明
FAST または CTX_DDL.OPTLEVEL_FAST	この方法では、断片化した行が圧縮されます。ただし、古いデータは削除されません。
FULL または CTX_DDL.OPTLEVEL_FULL	このモードでは、索引全体または索引の一部を最適化できます。この方法では行が圧縮され、古いデータ（削除済みの行）が削除されます。完全モードでの最適化は、削除済みの行が存在しない場合でも実行されます。
TOKEN	この方法では、最適化する特定のトークンを指定します。Oracle では、token で指定されたトークンに対して完全モードでの最適化を実行します。 検索頻度の高いトークンの最適化には、この方法を使用します。 トークン・モードでの最適化は、CTXRULE 索引に対してサポートされていません。

maxtime
FULL モードでの最適化の最大時間を分単位で指定します。

記号 CTX_DDL.MAXTIME_UNLIMITED を指定した場合（または NULL で渡した場合）、索引全体が最適化されます。これはデフォルトです。

token
最適化するトークンを指定します。

part_name
最適化する索引パーティション名を指定します。

parallel_degree
パラレル最適化に対する並列度を数値で指定します。実際の並列度は、リソースによって異なります。

例

次の 2 つの例では、索引を高速モードで最適化します。

```
begin
ctx_ddl.optimize_index('myidx','FAST');
end;

begin
ctx_ddl.optimize_index('myidx',CTX_DDL.OPTLEVEL_FAST);
end;
```

次の例では、索引トークン *Oracle* を最適化します。

```
begin
ctx_ddl.optimize_index('myidx','token', TOKEN=>'Oracle');
end;
```

関連項目

第 1 章「SQL 文と演算子」の「ALTER INDEX」

REMOVE_INDEX

CTXCAT 索引セット・プリファレンスから、指定した列リストを持つ索引を削除します。

注意： このプロシージャは、既存の索引から CTXCAT サブ索引を削除しません。削除するには、索引を削除し、変更した索引セット・プリファレンスを使用して再索引付けする必要があります。

構文

```
CTX_DDL.REMOVE_INDEX(
    set_name in varchar2,
    column_list in varchar2
    language in varchar2 default NULL
    );
```

set_name
索引セットの名前を指定します。

column_list
削除する列リストの名前を指定します。

REMOVE_SECTION

REMOVE_SECTION プロシージャは、指定されたセクションを指定されたセクション・グループから削除します。セクションは、名前または ID によって指定できます。セクション ID は、CTX_USER_SECTIONS ビューで参照できます。

構文 1

次の構文を使用して、セクション名によるセクションの削除を行います。

```
CTX_DDL.REMOVE_SECTION(
  group_name       in    varchar2,
  section_name     in    varchar2
);
```

group_name
section_name を削除するセクション・グループの名前を指定します。

section_name
group_name から削除するセクションの名前を指定します。

構文 2

次の構文を使用して、セクション ID よるセクションの削除を行います。

```
CTX_DDL.REMOVE_SECTION(
  group_name     in    varchar2,
  section_id     in    number
);
```

group_name
section_id を削除するセクション・グループの名前を指定します。

section_id
group_name から削除するセクションのセクション ID を指定します。

例

次のコードは、Title というセクションを htmgroup から削除します。

```
begin
ctx_ddl.remove_section('htmgroup', 'Title');
end;
```

関連項目

「ADD_FIELD_SECTION」

「ADD_SPECIAL_SECTION」

「ADD_ZONE_SECTION」

REMOVE_STOPCLASS

ストップクラスをストップリストから削除します。

構文

```
CTX_DDL.REMOVE_STOPCLASS(
  stoplist_name  in   varchar2,
  stopclass      in   varchar2
);
```

stoplist_name
ストップリストの名前を指定します。

stopclass
削除するストップクラスの名前を指定します。

例

次のコードは、ストップクラス NUMBERS をストップリスト mystop から削除します。

```
begin
ctx_ddl.remove_stopclass('mystop', 'NUMBERS');
end;
```

関連項目

「ADD_STOPCLASS」

REMOVE_STOPTHEME

ストップテーマをストップリストから削除します。

構文

```
CTX_DDL.REMOVE_STOPTHEME(
  stoplist_name   in   varchar2,
  stoptheme       in   varchar2
);
```

stoplist_name
ストップリストの名前を指定します。

stoptheme
stoplist_name から削除するストップテーマを指定します。

例

次のコードは、ストップテーマ *banking* をストップリスト mystop から削除します。

```
begin
ctx_ddl.remove_stoptheme('mystop', 'banking');
end;
```

関連項目

「ADD_STOPTHEME」

REMOVE_STOPWORD

ストップワードをストップリストから削除します。ストップワードの削除を索引に反映させる場合は、索引を再構築する必要があります。

構文

```
CTX_DDL.REMOVE_STOPWORD(
    stoplist_name  in   varchar2,
    stopword       in   varchar2,
    language       in   varchar2 default NULL
);
```

stoplist_name
ストップリストの名前を指定します。

stopword
stoplist_name から削除するストップワードを指定します。

language
stoplist_name に指定するストップリストが、MULTI_STOPLIST 型の場合は、削除する stopword の言語を指定します。Oracle がサポートしている言語のグローバリゼーション・サポート名または略称を指定する必要があります。また、ALL ストップワードも削除できます。

例

次のコードは、ストップワード *because* をストップリスト mystop から削除します

```
begin
    ctx_ddl.remove_stopword('mystop','because');
end;
```

関連項目

「ADD_STOPWORD」

SET_ATTRIBUTE

プリファレンスの属性を設定します。CTX_DDL.CREATE_PREFERENCE を使用してプリファレンスを作成した後、このプロシージャを使用します。

構文

```
ctx_ddl.set_attribute(preference_name in varchar2,
                      attribute_name  in varchar2,
                      attribute_value in varchar2);
```

preference_name
プリファレンスの名前を指定します。

attribute_name
属性の名前を指定します。

attribute_value
属性の値を指定します。ブール値は、TRUE または FALSE、T または F、YES または NO、Y または N、ON または OFF、あるいは 1 または 0（ゼロ）で指定できます。

例

ファイル・データ記憶域の指定

次の例では、索引付けするファイルがオペレーティング・システムに格納されていることをシステムに知らせる filepref というデータ記憶域プリファレンスを作成します。その後、CTX_DDL.SET_ATTRIBUTE を使用し、PATH 属性をディレクトリ /docs に設定します。

```
begin
ctx_ddl.create_preference('filepref', 'FILE_DATASTORE');
ctx_ddl.set_attribute('filepref', 'PATH', '/docs');
end;
```

関連項目： データ記憶域の詳細は、第 2 章「索引付け」の「データストア型」を参照してください。

SET_ATTRIBUTE を使用した例は、「CREATE_PREFERENCE」を参照してください。

SYNC_INDEX

索引を同期化し、元表に対する挿入、更新および削除を実行します。

構文

```
ctx_ddl.sync_index(
    idx_name  IN  VARCHAR2 DEFAULT NULL
    memory IN VARCHAR2 DEFAULT NULL,
    part_name IN VARCHAR2 DEFAULT NULL
    parallel_degree IN NUMBER DEFAULT 1);
```

idx_name
索引の名前を指定します。

memory
同期化に使用するランタイム・メモリー容量を指定します。この値は、DEFAULT_INDEX_MEMORY システム・パラメータをオーバーライドします。

メモリー・パラメータは、索引をディスクにフラッシュする前に、同期化操作に使用するメモリー量を指定します。大きいメモリー量を指定すると、次のような効果が得られます。

- I/O が削減されるため索引付けのパフォーマンスが向上します。
- 断片化が削減されるため問合せのパフォーマンスとメンテナンスが向上します。

小さいメモリー量を指定すると、ディスク I/O および索引の断片化が増加しますが、ランタイム・メモリー容量が不足している場合に有効なことがあります。

part_name
同期化する索引パーティション名を指定します。

parallel_degree
パラレル同期化に対する並列度を指定します。1 より大きい数値を指定すると、パラレル同期化が実行されます。実際の並列度は、リソースによっては多少低くなります。

例

次の例は、2MB のメモリーで索引 myindex を同期化します。

```
begin
    ctx_ddl.sync_index('myindex', '2M');
end;
```

次の例は、2MB のメモリーで part1 索引パーティションを同期化します。

```
begin
    ctx_ddl.sync_index('myindex', '2M', 'part1');
end;
```

関連項目

第 1 章「SQL 文と演算子」の「ALTER INDEX」

UNSET_ATTRIBUTE

プリファレンスから属性の設定を削除します。

構文

```
CTX_DDL.UNSET_ATTRIBUTE(preference_name varchar2,
                        attribute_name  varchar2);
```

preference_name
プリファレンスの名前を指定します。

attribute_name
属性の名前を指定します。

例

代替スペルの使用可能 / 使用禁止

次の例では、ドイツ語の代替スペルを使用可能にする方法および CTX_DDL.UNSET_ATTRIBUTE を使用して代替スペルを使用禁止にする方法を示します。

```
begin
ctx_ddl.create_preference('GERMAN_LEX', 'BASIC_LEXER');
ctx_ddl.set_attribute('GERMAN_LEX', 'ALTERNATE_SPELLING', 'GERMAN');
end;
```

代替スペルを使用禁止にするには、次のように CTX_DDL.UNSET_ATTRIBUTE プロシージャを使用します。

```
begin
ctx_ddl.unset_attribute('GERMAN_LEX', 'ALTERNATE_SPELLING');
end;
```

関連項目

「SET_ATTRIBUTE」

UPDATE_POLICY

CREATE_POLICY で作成したポリシーを更新します。ポリシーのプリファレンスを置換します。NULL の引数は置換されません。

構文

```
CTX_DDL.UPDATE_POLICY(
         policy_name     IN VARCHAR2 DEFAULT NULL,
         filter          IN VARCHAR2 DEFAULT NULL,
         section_group   IN VARCHAR2 DEFAULT NULL,
         lexer           IN VARCHAR2 DEFAULT NULL,
         stoplist        IN VARCHAR2 DEFAULT NULL,
         wordlist        IN VARCHAR2 DEFAULT NULL);
```

policy_name
更新するポリシーの名前を指定します。

filter
使用するフィルタ・プリファレンスを指定します。

> **注意：** このリリースでは、このパラメータは使用されません。

section_group
使用するセクション・グループを指定します。

lexer
使用するレクサー・プリファレンスを指定します。

stoplist
使用するストップリストを指定します。

wordlist
使用するワードリストを指定します。

8

CTX_DOC パッケージ

この章では、ドキュメント・サービスを要求するための PL/SQL パッケージ CTX_DOC について説明します。

> **注意：** このパッケージを使用できるのは、索引タイプが CONTEXT の場合のみです。このパッケージは、CTXCAT の索引タイプをサポートしていません。

CTX_DOC パッケージには、次のプロシージャおよびファンクションが含まれています。

名前	説明
FILTER	プレーン・テキストまたは HTML 形式のドキュメントを生成します。
GIST	ドキュメントの要点またはテーマ・サマリーを生成します。
HIGHLIGHT	プレーン・テキストまたは HTML 形式のドキュメントのハイライト表示するオフセット情報を生成します。
IFILTER	プレーン・テキスト形式のバイナリ・データを生成します。USER_DATASTORE プロシージャからコールできます。
MARKUP	プレーン・テキストまたは HTML 形式のドキュメントを、問合せ語句をハイライト表示して生成します。
PKENCODE	他の CTX_DOC プロシージャで使用するコンポジット・テキストキー文字列（値）をコード化します。
SET_KEY_TYPE	CTX_DOC プロシージャを設定し、ROWID または主キーのドキュメント識別子を受け取ります。
THEMES	ドキュメントのテーマのリストを生成します。
TOKENS	ドキュメントのすべての索引トークンを生成します。

FILTER

CTX_DOC.FILTER プロシージャを使用して、プレーン・テキストまたは HTML 形式のいずれかのドキュメントを生成します。表示されたドキュメントは、結果表またはメモリーのいずれかに格納できます。このプロシージャは、通常、問合せの後でフィルタ処理するドキュメントを決定してから使用します。

構文 1: メモリー内の結果記憶域

```
CTX_DOC.FILTER(
         index_name  IN VARCHAR2,
         textkey     IN VARCHAR2,
         restab      IN OUT NOCOPY CLOB,
         plaintext   IN BOOLEAN  DEFAULT FALSE);
```

構文 2: 結果表記憶域

```
CTX_DOC.FILTER(
         index_name  IN VARCHAR2,
         textkey     IN VARCHAR2,
         restab      IN VARCHAR2,
         query_id    IN NUMBER DEFAULT 0,
         plaintext   IN BOOLEAN  DEFAULT FALSE);
```

index_name
textkey で識別されるドキュメントを含むテキスト列に関連する索引の名前を指定します。

textkey
一意のドキュメントの ID（通常は主キー）を指定します。

textkey パラメータは、次のいずれかです。

- 単一列の主キーの値。
- コンポジット（複数列）主キーのコード化された仕様。CTX_DOC.PKENCODE を使用します。
- ドキュメントを含む行の ROWID。

主キーと ROWID の識別を切り替えるには、CTX_DOC.SET_KEY_TYPE を使用します。

restab
このプロシージャで、マークアップされたテキストを表またはメモリー内 CLOB のどちらに格納するかを指定できます。

結果を表に格納するには、表の名前を指定します。このコールを行う前に、結果表が存在している必要があります。

> **関連項目：** フィルタ結果表の構造の詳細は、付録 A「結果表」の「フィルタ表」を参照してください。

結果をメモリーに格納するには、CLOB ロケータの名前を指定します。restab が NULL の場合は、一時 CLOB が割り当てられて戻ります。使用後は、ロケータの割当てを解除する必要があります。

restab が NULL でない場合は、操作の前に CLOB が切り捨てられます。

query_id
restab に挿入した行の識別に使用する識別子を指定します。

query_id が指定されていないか、または NULL に設定されている場合、デフォルトは 0（ゼロ）です。restab に指定されている表を手動で切り捨てる必要があります。

plaintext
プレーン・テキスト形式のドキュメントを生成するには、TRUE を指定します。Inso フィルタを使用している場合または HTML ドキュメントを索引付けしている場合は、HTML 形式のドキュメントを生成するために、FALSE を指定します。

例

メモリー内フィルタ

次のコードは、ドキュメントをメモリー内の HTML にフィルタ処理する方法を示しています。

```
declare
mklob clob;
amt number := 40;
line varchar2(80);

begin
 ctx_doc.filter('myindex','1', mklob, FALSE);
 -- mklob is NULL when passed-in, so ctx-doc.filter will allocate a temporary
 -- CLOB for us and place the results there.
 dbms_lob.read(mklob, amt, 1, line);
 dbms_output.put_line('FIRST 40 CHARS ARE:'||line);
 -- have to de-allocate the temp lob
 dbms_lob.freetemporary(mklob);
 end;
```

フィルタ処理されたドキュメントを格納するために、次のようにフィルタ結果表を作成します。

```
create table filtertab (query_id  number,
                        document  clob);
```

テキストキー 20 でプレーン・テキスト形式のドキュメントを取得するには、次の文を発行します。

```
begin
ctx_doc.filter('newsindex', '20', 'filtertab', '0', TRUE);
end;
```

GIST

CTX_DOC.GIST プロシージャを使用して、ドキュメントの要点およびテーマ・サマリーを生成します。段落レベルまたは文レベルの要点またはテーマ・サマリーを生成できます。

構文 1: メモリー内記憶域

```
CTX_DOC.GIST(
    index_name    IN VARCHAR2,
    textkey       IN VARCHAR2,
    restab        IN OUT CLOB,
    glevel        IN VARCHAR2 DEFAULT 'P',
    pov           IN VARCHAR2 DEFAULT 'GENERIC',
    numParagraphs IN NUMBER DEFAULT NULL,
    maxPercent    IN NUMBER DEFAULT NULL,
    num_themes   IN NUMBER DEFAULT 50);
```

構文 2: 結果表記憶域

```
CTX_DOC.GIST(
    index_name    IN VARCHAR2,
    textkey       IN VARCHAR2,
    restab        IN VARCHAR2,
    query_id      IN NUMBER DEFAULT 0,
    glevel        IN VARCHAR2 DEFAULT 'P',
    pov           IN VARCHAR2 DEFAULT NULL,
    numParagraphs IN NUMBER DEFAULT NULL,
    maxPercent    IN NUMBER DEFAULT NULL,
    num_themes     IN NUMBER DEFAULT 50);
```

index_name

textkey で識別されるドキュメントを含むテキスト列に関連する索引の名前を指定します。

textkey

一意のドキュメントの ID（通常は主キー）を指定します。

textkey パラメータは、次のいずれかです。

- 単一列の主キーの値。
- コンポジット（複数列）主キーのコード化された仕様。コンポジット・テキストキーをコード化するには、CTX_DOC.PKENCODE プロシージャを使用します。
- ドキュメントを含む行の ROWID。

主キーと ROWID の識別を切り替えるには、CTX_DOC.SET_KEY_TYPE を使用します。

restab
このプロシージャで、要点およびテーマ・サマリーを表またはメモリー内 CLOB のどちらに格納するかを指定できます。

結果を表に格納するには、表の名前を指定します。

> **関連項目：** 要点結果表の構造の詳細は、付録 A「結果表」の「要点表」を参照してください。

結果をメモリーに格納するには、CLOB ロケータの名前を指定します。restab が NULL の場合は、一時 CLOB が割り当てられて戻ります。使用後は、ロケータの割当てを解除する必要があります。

restab が NULL でない場合は、操作の前に CLOB が切り捨てられます。

query_id
restab に挿入した行の識別に使用する識別子を指定します。

glevel
生成する要点またはテーマ・サマリーのタイプを指定します。有効な値は、次のとおりです。

- 段落の場合は *P*
- 文の場合は *S*

デフォルトは *P* です。

pov
要点または単一のテーマ・サマリーを生成するかどうかを指定します。生成される要点またはテーマ・サマリーのタイプ（文レベルまたは段落レベル）は、glevel に指定した値によって決まります。

ドキュメント全体の要点を生成するには、pov に値 GENERIC を指定します。ドキュメントの単一のテーマに対するテーマ・サマリーを生成するには、そのテーマを pov の値として指定します。

結果表記憶域を使用し、pov の値を指定しない場合、このプロシージャはドキュメントに対して、全体の要点に加えて最大 50 個までのテーマ・サマリーを戻します。

メモリー内の結果記憶域を CLOB に対して使用する場合は、pov を指定する必要があります。ただし、pov を指定しない場合、このプロシージャはドキュメントの全体の要点のみを生成します。

注意： pov パラメータでは、大 / 小文字が区別されます。ドキュメントの要点を戻すには、すべて大文字で「GENERIC」を指定します。テーマ・サマリーを戻すには、ドキュメントに対して生成されたとおりのテーマを正確に指定します。

ドキュメントに対して THEMES で生成されたテーマのみを、pov の入力として使用できます。

---

numParagraphs
ドキュメントの要点またはテーマ・サマリーに対して選択されたドキュメントの段落（または文）の最大数を指定します。デフォルトは 0（ゼロ）です。

注意： numParagraphs パラメータが使用されるのは、このパラメータによって生成される要点またはテーマ・サマリーのサイズが、maxPercent パラメータによって生成される要点またはテーマ・サマリーのサイズよりも小さいときのみです。

つまり、システムは、常に最小サイズの要点またはテーマ・サマリーを戻します。

---

maxPercent
ドキュメントの要点またはテーマ・サマリーに対して選択されたドキュメントの段落（または文）の最大数を、ドキュメントの合計段落数（または合計文数）に対する割合で指定します。デフォルトは 0（ゼロ）です。

注意： maxPercent パラメータが使用されるのは、このパラメータによって生成される要点またはテーマ・サマリーのサイズが、numParagraphs パラメータによって生成される要点またはテーマ・サマリーのサイズよりも小さいときのみです。

つまり、システムは、常に最小サイズの要点またはテーマ・サマリーを戻します。

---

num_themes
pov の値を指定しない場合は、生成するテーマ・サマリーの数を指定します。たとえば、10 を指定すると、トップ 10 のテーマ・サマリーが戻ります。デフォルトは 50 です。

0（ゼロ）または NULL を指定すると、ドキュメント内の全テーマが戻ります。51 以上のテーマがドキュメントに格納されている場合、概念階層が表示されるのは、トップ 50 のテーマのみです。

例

メモリー内要点

次の例では、10 個以内の段落のデフォルト・サイズでない全体の要点を生成します。結果は CLOB ロケータの中のメモリーに格納されます。使用後、コードは戻された CLOB ロケータの割当てを解除します。

```
set serveroutput on;
declare
  gklob clob;
  amt number := 40;
  line varchar2(80);

begin
 ctx_doc.gist('newsindex','34',gklob, pov => 'GENERIC',numParagraphs => 10);
  -- gklob is NULL when passed-in, so ctx-doc.gist will allocate a temporary
  -- CLOB for us and place the results there.

  dbms_lob.read(gklob, amt, 1, line);
  dbms_output.put_line('FIRST 40 CHARS ARE:'||line);
  -- have to de-allocate the temp lob
  dbms_lob.freetemporary(gklob);
 end;
```

結果表の要点

次の例では、要点表 CTX_GIST を作成します。

```
create table CTX_GIST (query_id  number,
                       pov       varchar2(80),
                       gist      CLOB);
```

要点およびテーマ・サマリー 次の例では、ドキュメント 34 に対するデフォルト・サイズの段落レベルの要点とドキュメント内の全テーマに対するトップ 10 のテーマ・サマリーを戻します。

```
begin
   ctx_doc.gist('newsindex','34','CTX_GIST', 1, num_themes=10);
end;
```

次の例では、10 個以内の段落のデフォルト・サイズでない要点表を生成します。

```
begin
  ctx_doc.gist('newsindex','34','CTX_GIST',1,pov ='GENERIC',numParagraphs=>10);
end;
```

次の例では、段落数がドキュメントの合計段落数の 10 パーセント以下である要点表を生成します。

```
begin
  ctx_doc.gist('newsindex','34','CTX_GIST',1,pov => 'GENERIC',  maxPercent => 10);
end;
```

テーマ・サマリー 次の例では、ドキュメント 34 の *insects* に対する段落レベルのテーマ・サマリーを戻します。デフォルト・サイズのテーマ・サマリーが戻されます。

```
begin
   ctx_doc.gist('newsindex','34','CTX_GIST',1, pov => 'insects');
end;
```

HIGHLIGHT

CTX_DOC.HIGHLIGHT プロシージャを使用して、ドキュメントのハイライト・オフセットを生成します。オフセット情報は、指定する問合せを満たすドキュメント内の語句に対して生成されます。これらのハイライト表示される語句は、ワード問合せを満たすワードまたは ABOUT 問合せを満たすテーマのいずれかです。

プレーン・テキスト形式または HTML 形式のいずれかのドキュメントに対して、ハイライト・オフセットを生成できます。オフセット情報は、CTX_DOC.FILTER でフィルタ処理された同じドキュメントに適用できます。

このプロシージャは、通常、問合せの後で処理するドキュメントを決めてから使用します。

ハイライト・オフセットは、メモリー内 PL/SQL 表または結果表のいずれかに格納できます。

構文 1: メモリー内の結果記憶域

```
CTX_DOC.HIGHLIGHT(
       index_name  IN VARCHAR2,
       textkey     IN VARCHAR2,
       text_query  IN VARCHAR2,
       restab      IN OUT NOCOPY HIGHLIGHT_TAB,
       plaintext   IN BOOLEAN  DEFAULT FALSE);
```

構文 2: 結果表記憶域

```
CTX_DOC.HIGHLIGHT(
         index_name  IN VARCHAR2,
         textkey     IN VARCHAR2,
         text_query  IN VARCHAR2,
         restab      IN VARCHAR2,
         query_id    IN NUMBER   DEFAULT 0,
         plaintext   IN BOOLEAN  DEFAULT FALSE);
```

index_name
textkey で識別されるドキュメントを含むテキスト列に関連する索引の名前を指定します。

textkey
一意のドキュメントの ID（通常は主キー）を指定します。

textkey パラメータは、次のいずれかです。

- 単一列の主キーの値。
- コンポジット（複数列）主キーのコード化された仕様。CTX_DOC.PKENCODE プロシージャを使用します。
- ドキュメントを含む行の ROWID。

主キーと ROWID の識別を切り替えるには、CTX_DOC.SET_KEY_TYPE を使用します。

text_query
ドキュメントを取り出すために使用された元の問合せ式を指定します。NULL の場合、ハイライトは生成されません。

text_query にワイルド・カード、ステミング、ファジー・マッチングが指定されていて、結果としてストップワードが戻る場合、HIGHLIGHT はストップワードをハイライト表示しません。

text_query に THRESHOLD 演算子がある場合、その演算子は無視されます。HIGHLIGHT プロシージャは、常に結果セット全体に対するハイライト情報を戻します。

restab
このプロシージャが、ハイライト・オフセットを表またはメモリー内 PL/SQL 表のいずれかに格納することを指定できます。

結果を表に格納するには、表の名前を指定します。このプロシージャのコール前に、表が存在している必要があります。

> **関連項目：** ハイライト結果表の構造の詳細は、付録 A「結果表」の「ハイライト表」を参照してください。

結果をメモリー内表に格納するには、CTX_DOC.HIGHLIGHT_TAB 型のメモリー内表の名前を指定します。HIGHLIGHT_TAB データ型は、次のように定義されます。

```
type highlight_rec is record (
  offset number,
  length number
);
type highlight_tab is table of highlight_rec index by binary_integer;
```

操作の前に、CTX_DOC.HIGHLIGHT は HIGHLIGHT_TAB を消去します。

query_id
restab に挿入した行の識別に使用する識別子を指定します。

query_id が指定されていないか、または NULL に設定されている場合、デフォルトは 0（ゼロ）です。restab に指定されている表を手動で切り捨てる必要があります。

plaintext
プレーン・テキスト・オフセットのドキュメントを生成するには、TRUE を指定します。

Inso フィルタを使用している場合または HTML ドキュメントを索引付けしている場合は、HTML オフセットのドキュメントを生成するために、FALSE を指定します。

例

ハイライト表の作成

ハイライト・オフセット情報を格納するために、ハイライト表を作成します。

```
create table hightab(query_id number,
                     offset number,
                     length number);
```

ワード・ハイライト・オフセット

ドキュメント 20 の *dog* という語句に対する HTML ハイライト・オフセット情報を取得するには、次の問合せを発行します。

```
begin
ctx_doc.highlight('newsindex', '20', 'dog', 'hightab', 0, FALSE);
end;
```

テーマ・ハイライト・オフセット

索引 *newsindex* にテーマ・コンポーネントがある場合、次の問合せを発行することによって、*politics* というテーマ問合せに対する HTML ハイライト・オフセット情報を取得します。

```
begin
ctx_doc.highlight('newsindex', '20', 'about(politics)', 'hightab', 0, FALSE);
end;
```

この文に対する出力は、ドキュメントの *politics* のテーマを表すハイライト表示されたワードおよび句に対するオフセットです。

IFILTER

バイナリ・データをテキストにフィルタ処理するには、このプロシージャを使用します。

このプロシージャは、バイナリ・データ（BLOB IN）を取り出し、そのデータを Inso フィルタでフィルタ処理し、そのテキスト形式データを CLOB に書き込みます。CTX_DOC.IFILTER では、セーフ・コールアウトを使用するため、ユーザー・データストアのプロシージャからコールできます。また、索引の使用は不要です。CTX_DOC.FILTER の場合は、索引が必要となります。

要件

CTX_DOC.IFILTER では、セーフ・コールアウト・メカニズムを採用しているため、extproc エージェントの起動には、SQL*Net リスナーの実行と構成が必要です。

構文

```
CTX_DOC.IFILTER(data IN BLOB, text IN OUT NOCOPY CLOB);
```

data
フィルタ処理するバイナリ・データを指定します。

text
宛先の CLOB を指定します。CLOB には、フィルタ処理済みのデータが置かれます。このパラメータは、書込み可能な有効な CLOB ロケータであることが必要です。NULL または書込み禁止の CLOB を渡すと、エラーになります。フィルタ処理済みのテキストは、既存のデータがある場合、その末尾に追加されます。

MARKUP

CTX_DOC.MARKUP プロシージャは、問合せ指定およびドキュメントのテキストキーを取り出し、問合せ語句がマークアップされた形式のドキュメントを戻します。これらのマークアップされた語句は、ワード問合せを満たすワードまたは ABOUT 問合せを満たすテーマのいずれかです。

マークアップされた出力は、プレーン・テキストまたは HTML のいずれかに設定できます。

ハイライト表示された語句をマークするために、事前定義済みのタグセットの 1 つ（HTML ナビゲーションを可能にするタグ順序を含む）を使用できます。

CTX_DOC.MARKUP は、通常、問合せの後で処理するドキュメントを決めてから使用します。

マークアップされたドキュメントは、メモリーまたは結果表のいずれかに格納できます。

構文 1: メモリー内結果記憶域

```
CTX_DOC.MARKUP(
    index_name      IN VARCHAR2,
    textkey         IN VARCHAR2,
    text_query      IN VARCHAR2,
    restab          IN OUT NOCOPY CLOB,
    plaintext       IN BOOLEAN   DEFAULT FALSE,
    tagset          IN VARCHAR2  DEFAULT 'TEXT_DEFAULT',
    starttag        IN VARCHAR2  DEFAULT NULL,
    endtag          IN VARCHAR2  DEFAULT NULL,
    prevtag         IN VARCHAR2  DEFAULT NULL,
    nexttag         IN VARCHAR2  DEFAULT NULL);
```

構文 2: 結果表記憶域

```
CTX_DOC.MARKUP(
    index_name      IN VARCHAR2,
    textkey         IN VARCHAR2,
    text_query      IN VARCHAR2,
    restab          IN VARCHAR2,
    query_id        IN NUMBER    DEFAULT 0,
    plaintext       IN BOOLEAN   DEFAULT FALSE,
    tagset          IN VARCHAR2  DEFAULT 'TEXT_DEFAULT',
    starttag        IN VARCHAR2  DEFAULT NULL,
    endtag          IN VARCHAR2  DEFAULT NULL,
    prevtag         IN VARCHAR2  DEFAULT NULL,
    nexttag         IN VARCHAR2  DEFAULT NULL);
```

index_name
textkey で識別されるドキュメントを含むテキスト列に関連する索引の名前を指定します。

textkey
一意のドキュメントの ID（通常は主キー）を指定します。

textkey パラメータは、次のいずれかです。

- 単一列の主キーの値。
- コンポジット（複数列）主キーのコード化された仕様。CTX_DOC.PKENCODE プロシージャを使用します。
- ドキュメントを含む行の ROWID。

主キーと ROWID の識別を切り替えるには、CTX_DOC.SET_KEY_TYPE を使用します。

text_query
ドキュメントを取り出すために使用された元の問合せ式を指定します。

text_query にワイルド・カード、ステミング、ファジー・マッチングが指定されていて、結果としてストップワードが戻る場合、MARKUP はストップワードをハイライト表示しません。

text_query に THRESHOLD 演算子がある場合、その演算子は無視されます。MARKUP プロシージャは、常に結果セット全体に対するハイライト情報を戻します。

restab
このプロシージャで、マークアップされたテキストを表またはメモリー内 CLOB のどちらに格納するかを指定できます。

結果を表に格納するには、表の名前を指定します。このプロシージャのコール前に、結果表が存在している必要があります。

> **関連項目：** マークアップ結果表の構造の詳細は、付録 A「結果表」の「マークアップ表」を参照してください。

結果をメモリーに格納するには、CLOB ロケータの名前を指定します。restab が NULL の場合は、一時 CLOB が割り当てられて戻ります。使用後は、ロケータの割当てを解除する必要があります。

restab が NULL でない場合は、操作の前に CLOB が切り捨てられます。

query_id
restab に挿入した行の識別に使用する識別子を指定します。

query_id が指定されていないか、または NULL に設定されている場合、デフォルトは 0（ゼロ）です。restab に指定されている表を手動で切り捨てる必要があります。

plaintext
プレーン・テキストのマークアップ・ドキュメントを生成するには、TRUE を指定します。Inso フィルタを使用している場合または HTML ドキュメントを索引付けしている場合は、マークアップされた HTML 形式のドキュメントを生成するために、FALSE を指定します。

tagset
次の事前定義済みのタグセットの 1 つを指定します。この表の 2 列目および 3 列目は、各タグセットに対して定義された 4 つの異なるタグを示します。

タグセット	タグ	タグ値
TEXT_DEFAULT	starttag	<<<
	endtag	>>>
	prevtag	
	nexttag	
HTML_DEFAULT	starttag	<B>
	endtag	</B>
	prevtag	
	nexttag	
HTML_NAVIGATE	starttag	<A NAME=ctx%CURNUM><B>
	endtag	</B></A>
	prevtag	<A HREF=#ctx%PREVNUM><</A>
	nexttag	<A HREF=#ctx%NEXTNUM>></A>

starttag
ハイライト表示された語句の開始を示すために、MARKUP によって挿入された文字を指定します。

ハイライト表示されたワードに対する starttag、endtag、prevtag および nexttag の順序は、次のとおりです。

```
... prevtag starttag word endtag nexttag...
```

endtag
ハイライト表示された語句の終了を示すために、MARKUP によって挿入された文字を指定します。

prevtag
前のハイライト表示された部分にユーザーをナビゲートするタグを定義するマークアップ順序を指定します。

マークアップ順序 prevtag および nexttag で、動的に設定される次のオフセット変数を指定できます。

オフセット変数	値
%CURNUM	現在のオフセット番号
%PREVNUM	前のオフセット番号
%NEXTNUM	次のオフセット番号

例については、HTML_NAVIGATE タグセットの記述を参照してください。

nexttag
次のハイライト表示された部分にユーザーをナビゲートするタグを定義するマークアップ順序を指定します。

マークアップ順序内で、prevtag に使用するものと同じオフセット変数を使用できます。例については、prevtag および HTML_NAVIGATE タグセットの記述を参照してください。

例

メモリー内マークアップ

次のコードは、マークアップ・ドキュメントを生成し、メモリー内に格納します。コードは、NULL CLOB ロケータを MARKUP に渡し、使用後戻された CLOB ロケータの割当てを解除します。

```
set serveroutput on

declare
mklob clob;
amt number := 40;
line varchar2(80);
```

```
begin
 ctx_doc.markup('myindex','1','dog & cat', mklob);
 -- mklob is NULL when passed-in, so ctx-doc.markup will allocate a temporary
 -- CLOB for us and place the results there.
 dbms_lob.read(mklob, amt, 1, line);
 dbms_output.put_line('FIRST 40 CHARS ARE:'||line);
 -- have to de-allocate the temp lob
 dbms_lob.freetemporary(mklob);
 end;
```

マークアップ表

マークアップされたドキュメントを格納するために、次のようにハイライト・マークアップ表を作成します。

```
create table markuptab (query_id  number,
                        document  clob);
```

HTML のワード・ハイライト

ドキュメント 23 の *dog* または *cat* というワードに対する HTML ハイライト・マークアップを作成するには、次の文を発行します。

```
begin
  ctx_doc.markup(index_name => 'my_index',
                      textkey => '23',
                      text_query => 'dog|cat',
                      restab => 'markuptab',
                      query_id => '1',
                      tagset => 'HTML_DEFAULT');
end;
```

HTML のテーマ・ハイライト

ドキュメント 23 の *politics* というテーマに対する HTML ハイライト・マークアップを作成するには、次の文を発行します。

```
begin
  ctx_doc.markup(index_name => 'my_index',
                      textkey => '23',
                      text_query => 'about(politics)',
                      restab => 'markuptab',
                      query_id => '1',
                      tagset => 'HTML_DEFAULT');
end;
```

PKENCODE

CTX_DOC.PKENCODE ファンクションは、コンポジット・テキストキーのリストを 1 つの文字列に変換し、文字列を戻します。

PKENCODE によって作成された文字列は、CTX_DOC.THEMES および CTX_DOC.GIST などの他の CTX_DOC プロシージャで、主キー・パラメータ textkey として使用できます。

構文

```
CTX_DOC.PKENCODE(
         pk1     IN VARCHAR2,
         pk2     IN VARCHAR2 DEFAULT NULL,
         pk4     IN VARCHAR2 DEFAULT NULL,
         pk5     IN VARCHAR2 DEFAULT NULL,
         pk6     IN VARCHAR2 DEFAULT NULL,
         pk7     IN VARCHAR2 DEFAULT NULL,
         pk8     IN VARCHAR2 DEFAULT NULL,
         pk9     IN VARCHAR2 DEFAULT NULL,
         pk10    IN VARCHAR2 DEFAULT NULL,
         pk11    IN VARCHAR2 DEFAULT NULL,
         pk12    IN VARCHAR2 DEFAULT NULL,
         pk13    IN VARCHAR2 DEFAULT NULL,
         pk14    IN VARCHAR2 DEFAULT NULL,
         pk15    IN VARCHAR2 DEFAULT NULL,
         pk16    IN VARCHAR2 DEFAULT NULL)
RETURN VARCHAR2;
```

pk1-pk16
各 PK 引数はコンポジット・テキストキー・リストの列要素を指定します。最大 16 列の要素をコード化できます。

戻り値

コンポジット・テキストキーのコード化された値を表す文字列です。

例

```
begin
ctx_doc.gist('newsindex',CTX_DOC.PKENCODE('smith', 14), 'CTX_GIST');
end;
```

この例の *smith* および *14* は、ドキュメントのコンポジット・テキストキーの値を構成します。

SET_KEY_TYPE

このプロシージャを使用して CTX_DOC プロシージャを設定し、ROWID または PRIMARY_KEY のドキュメントの識別子を受け取ります。この設定は、セッションの起動にのみ影響を与えます。

構文

```
ctx_doc.set_key_type(key_type in varchar2);
```

key_type
ROWID または PRIMARY_KEY のいずれかを、CTX_DOC プロシージャに対する入力キー・タイプ（ドキュメント識別子）として指定します。

このパラメータのデフォルトは、CTX_DOC_KEY_TYPE システム・パラメータの値です。

> **注意：** 元表に主キーがない場合は、PRIMARY_KEY に key_type を指定しても無視されます。CTX_DOC プロシージャに指定する textkey パラメータは、ROWID として解釈されます。

例

CTX_DOC プロシージャを設定し、主キーのドキュメント識別子を受け取るには、次の文を発行します。

```
begin
ctx_doc.set_key_type('PRIMARY_KEY');
end
```

THEMES

CTX_DOC.THEMES プロシージャを使用して、1 つのドキュメントに対してテーマ・リストを 1 つ生成します。各テーマは、指定した結果表またはメモリー内 PL/SQL 表のいずれかに行として格納できます。

構文 1: メモリー内表記憶域

```
CTX_DOC.THEMES(
    index_name      IN VARCHAR2,
    textkey         IN VARCHAR2,
    restab          IN OUT NOCOPY THEME_TAB,
    full_themes     IN BOOLEAN DEFAULT FALSE,
    num_themes       IN NUMBER DEFAULT 50);
```

構文 2: 結果表記憶域

```
CTX_DOC.THEMES(
    index_name      IN VARCHAR2,
    textkey         IN VARCHAR2,
    restab          IN VARCHAR2,
    query_id        IN NUMBER DEFAULT 0,
    full_themes     IN BOOLEAN DEFAULT FALSE,
    num_themes       IN NUMBER DEFAULT 50);
```

index_name
テキスト列の索引名を指定します。

textkey
一意のドキュメントの ID（通常は主キー）を指定します。

textkey パラメータは、次のいずれかです。

- 単一列の主キーの値。
- コンポジット（複数列）主キーのコード化された仕様。textkey がコンポジット・テキストキーの場合、CTX_DOC.PKENCODE プロシージャを使用して、コンポジット・テキストキー文字列をコード化する必要があります。
- ドキュメントを含む行の ROWID。

主キーと ROWID の識別を切り替えるには、CTX_DOC.SET_KEY_TYPE を使用します。

restab
このプロシージャが、結果を表またはメモリー内 PL/SQL 表のいずれかに格納することを指定できます。

結果を表に格納するには、表の名前を指定します。

> **関連項目：** テーマ結果表の構造の詳細は、付録 A「結果表」の「テーマ表」を参照してください。

結果をメモリー内表に格納するには、THEME_TAB 型のメモリー内表の名前を指定します。THEME_TAB データ型は、次のように定義されます。

```
type theme_rec is record (
   theme varchar2(2000),
   weight number
);

type theme_tab is table of theme_rec index by binary_integer;
```

操作の前に、CTX_DOC.THEMES は、ユーザーが指定した THEME_TAB を消去します。

query_id
restab に挿入した行の識別に使用する識別子を指定します。

full_themes
このプロシージャが、各ドキュメント・テーマに対して単一のテーマを作成するか、親テーマ（全テーマ）の階層型リストを作成するかを指定します。

TRUE を指定すると、このプロシージャは全テーマを結果表の THEME 列に書き込みます。

FALSE を指定すると、このプロシージャは単一テーマ情報を結果表の THEME 列に書き込みます。これはデフォルトです。

num_themes
取り出すテーマの数を指定します。たとえば、10 を指定すると、ドキュメントのトップ 10 のテーマが戻ります。デフォルトは 50 です。

0（ゼロ）または NULL を指定すると、ドキュメント内の全テーマが戻ります。51 以上のテーマがドキュメントに格納されている場合、概念階層が表示されるのは、トップ 50 のテーマのみです。

例

メモリー内テーマ

次の例では、ドキュメント 1 のトップ 10 のテーマを生成し、それを the_themes というメモリー内表に格納します。その後、表全体をループし、ドキュメントのテーマを表示します。

```
declare
 the_themes ctx_doc.theme_tab;

begin
 ctx_doc.themes('myindex','1',the_themes, numthemes=>10);
 for i in 1..the_themes.count loop
  dbms_output.put_line(the_themes(i).theme||':'||the_themes(i).weight);
  end loop;
end;
```

テーマ表

次の例では、CTX_THEMES というテーマ表を作成します。

```
create table CTX_THEMES (query_id number,
                         theme varchar2(2000),
                         weight number);
```

単一テーマ

リストの各要素が単一テーマである場合、トップ 20 のテーマのリストを取得するには、次の文を発行します。

```
begin
     ctx_doc.themes('newsindex','34','CTX_THEMES',1,full_themes => FALSE,
     num_themes=> 20);
end;
```

全テーマ

リストの各要素が親テーマの階層リストの場合、トップ 20 のテーマのリストを取得するには、次の文を発行します。

```
begin
    ctx_doc.themes('newsindex','34','CTX_THEMES',1,full_themes => TRUE,
    num_themes=>20);
end;
```

TOKENS

このプロシージャを使用して、ドキュメント内のすべてのテキスト・トークンを識別します。戻されるトークンは、索引に挿入されるトークンです。この機能は、ドキュメントの分類、ルーティングまたはクラスタ化を行う場合に有効です。

ストップワードは戻されません。セクション・タグは、テキスト・トークンではないため、戻されません。

構文 1: メモリー内表記憶域

```
CTX_DOC.TOKENS(index_name      IN VARCHAR2,
               textkey         IN VARCHAR2,
               restab          IN OUT NOCOPY TOKEN_TAB);
```

構文 2: 結果表記憶域

```
CTX_DOC.TOKENS(index_name      IN VARCHAR2,
               textkey         IN VARCHAR2,
               restab          IN VARCHAR2,
               query_id        IN NUMBER DEFAULT 0);
```

index_name
テキスト列の索引名を指定します。

textkey
一意のドキュメントの ID（通常は主キー）を指定します。

textkey パラメータは、次のいずれかです。

- 単一列の主キーの値。
- コンポジット（複数列）主キーのコード化された仕様。コンポジット・テキストキーをコード化するには、CTX_DOC.PKENCODE プロシージャを使用します。
- ドキュメントを含む行の ROWID。

主キーと ROWID の識別を切り替えるには、CTX_DOC.SET_KEY_TYPE を使用します。

restab
このプロシージャが、結果を表またはメモリー内 PL/SQL 表のいずれかに格納することを指定できます。

戻されるトークンは、textkey で指定したドキュメント（または行）の索引に挿入されるトークンです。ストップワードは戻されません。セクション・タグは、テキスト・トークンではないため、戻されません。

トークン表の指定 結果を表に格納するには、表の名前を指定します。トークン表には自由に名前を付けることができますが、次の名前およびデータ型を持つ列が組み込まれている必要があります。

表 8-1

列名	型	説明
QUERY_ID	NUMBER	CTX_DOC.TOKENS への特定のコールで生成された結果の識別子（複数の TOKEN コールの結果を表に格納するときのみ移入されます）。
TOKEN	VARCHAR2(64)	テキスト内のトークン文字列。
OFFSET	NUMBER	ドキュメント内のトークンの位置（ドキュメントの開始位置 1 との相対的な位置）。
LENGTH	NUMBER	トークンの文字の長さ。

メモリー内表の指定 結果をメモリー内表に格納するには、TOKEN_TAB 型のメモリー内表の名前を指定します。TOKEN_TAB データ型は、次のように定義されます。

```
type token_rec is record (
    token varchar2(64),
    offset number,
    length number
);

type token_tab is table of token_rec index by binary_integer;
```

操作の前に、CTX_DOC.TOKENS は、ユーザーが指定した TOKEN_TAB を消去します。

query_id

restab に挿入した行の識別に使用する識別子を指定します。

例

メモリー内トークン

次の例では、ドキュメント 1 のトークンを生成し、それをメモリー内表に格納して、the_tokens として宣言します。その後、表全体をループし、ドキュメントのトークンを表示します。

```
declare
 the_tokens ctx_doc.token_tab;

begin
 ctx_doc.tokens('myindex','1',the_tokens);
 for i in 1..the_tokens.count loop
  dbms_output.put_line(the_tokens(i).token);
  end loop;
end;
```

9

CTX_OUTPUT パッケージ

この章では、PL/SQL パッケージ CTX_OUTPUT の使用方法について説明します。

CTX_OUTPUT には、次のストアド・プロシージャが含まれています。

名前	説明
ADD_EVENT	索引ログにイベントを追加します。
END_LOG	索引およびドキュメント・サービスの要求のロギングを停止します。
LOGFILENAME	現在のログ・ファイルの名前を戻します。
REMOVE_EVENT	索引ログからイベントを削除します。
START_LOG	索引およびドキュメント・サービスの要求のロギングを開始します。

ADD_EVENT

詳細なログ出力を表示するために、イベントを索引ログに追加します。現在、追加できるイベントは、CTX_OUTPUT.EVENT_INDEX_PRINT_ROWID のみです。このイベントは索引付けが終了した各行の ROWID をロギングします。これは、失敗した索引操作のデバッグに有効です。

構文

```
CTX_OUTPUT.ADD_EVENT(event in varchar2);
```

event
ロギングする索引イベントの型を指定します。現在、追加できるイベントは、CTX_OUTPUT.EVENT_INDEX_PRINT_ROWID のみです。このイベントは索引付けが終了した各行の ROWID をロギングします。

例

```
begin
    CTX_OUTPUT.ADD_EVENT(CTX_OUTPUT.EVENT_INDEX_PRINT_ROWID);
end;
```

END_LOG

索引およびドキュメント・サービスの要求のロギングを停止します。

構文

```
CTX_OUTPUT.END_LOG;
```

例

```
begin
CTX_OUTPUT.END_LOG;
end;
```

LOGFILENAME

現在のログのファイル名を戻します。このプロシージャは、LOG_DIRECTORY システム・パラメータで指定されたディレクトリの中でログ・ファイルを検索します。

構文

```
CTX_OUTPUT.LOGFILENAME RETURN VARCHAR2;
```

戻り値

ログ・ファイル名です。

例

```
declare
   logname varchar2(100);
begin
   logname := CTX_OUTPUT.LOGFILENAME;
   dbms_output.put_line('The current log file is: '||logname);
end;
```

REMOVE_EVENT

索引ログからイベントを削除します。

構文

```
CTX_OUTPUT.REMOVE_EVENT(event in varchar2);
```

event
ログから削除する索引イベントの型を指定します。現在、追加および削除できるイベントは、CTX_OUTPUT.EVENT_INDEX_PRINT_ROWID のみです。

例

```
begin
    CTX_OUTPUT.REMOVE_EVENT(CTX_OUTPUT.EVENT_INDEX_PRINT_ROWID);
end;
```

START_LOG

索引およびドキュメント・サービスの要求のロギングを開始します。

構文

```
CTX_OUTPUT.START_LOG(logfile in varchar2);
```

logfile
ログ・ファイルの名前を指定します。ログは、システム・パラメータ LOG_DIRECTORY で指定されたディレクトリに格納されています。

例

```
begin
CTX_OUTPUT.START_LOG('mylog1');
end;
```

10

CTX_QUERY パッケージ

この章では、問合せフィードバックの生成、ヒット数のカウントおよびストアド・クエリー式の生成に使用できる PL/SQL パッケージ CTX_QUERY について説明します。

> **注意：** このパッケージを使用できるのは、索引タイプが CONTEXT の場合のみです。このパッケージは、CTXCAT の索引タイプをサポートしていません。

CTX_QUERY パッケージには、次のプロシージャおよびファンクションが含まれています。

名前	説明
BROWSE_WORDS	索引内のシード・ワードに関するワードを戻します。
COUNT_HITS	問合せへのヒット数を戻します。
EXPLAIN	問合せ式の解析および拡張情報を生成します。
HFEEDBACK	階層問合せフィードバック情報（上位語、下位語および関連語）を生成します。
REMOVE_SQE	指定したストアド・クエリー式を SQL 表から削除します。
STORE_SQE	問合せを実行して、その結果をストアド・クエリー式表に格納します。

BROWSE_WORDS

このプロシージャを使用すると、Oracle Text の索引内のワードをブラウズできます。シード・ワードを指定すると、BROWSE_WORDS は索引内のそのシード・ワードに関するワード、および各ワードが含まれているドキュメントのおおよその数を戻します。

この機能は、問合せの詳細化に有効です。次の内容を識別できます。

- 非選択的ワード（そのワードを含むドキュメント件数が少ないもの）
- ドキュメント・セットの中でスペルが間違っているワード

構文 1: 表への結果の格納

```
ctx_query.browse_words(
    index_name  IN   VARCHAR2,
    seed        IN   VARCHAR2,
    restab      IN   VARCHAR2,
    browse_id   IN   NUMBER   DEFAULT 0,
    numwords    IN   NUMBER   DEFAULT 10,
    direction   IN   VARCHAR2 DEFAULT BROWSE_AROUND,
    part_name   IN   VARCHAR2 DEFAULT NULL
);
```

構文 2: メモリーへの結果の格納

```
ctx_query.browse_words(
    index_name  IN      VARCHAR2,
    seed        IN      VARCHAR2,
    resarr      IN OUT  BROWSE_TAB,
    numwords    IN      NUMBER   DEFAULT 10,
    direction   IN      VARCHAR2 DEFAULT BROWSE_AROUND,
    part_name   IN      VARCHAR2 DEFAULT NULL
);
```

index
索引の名前を指定します。schema.name で指定できます。ローカル索引である必要があります。

seed
シード・ワードを指定します。このワードは、ブラウズ拡張の前にレクサー処理されます。ワードがトークン表に存在する必要はありません。seed は単一のワードである必要があります。複数のワードをシードとして使用すると、エラーになります。

restab
結果表の名前を指定します。restab は、schema.name として入力できます。このプロシージャのコール前に、表が存在している必要があります。また、表に対する INSERT 権限が必要です。この表は、次のスキーマを持つ必要があります。

列	データ型
browse_id	number
word	varchar2(64)
doc_count	number

restab の中にある行が、BROWSE_WORDS がコールされる前に削除されることはありません。

resarr
結果配列の名前を指定します。resarr は、ctx_query.browse_tab 型です。

```
type browse_rec is record (
   word varchar2(64),
   doc_count number
);
type browse_tab is table of browse_rec index by binary_integer;
```

browse_id
0 ～ 2^{32} の間で数値識別子を指定します。このブラウズのために生成された行は、restab の中の browse_id 列に値を持ちます。browse_id を指定しない場合、デフォルトは 0（ゼロ）です。

numwords
戻すワード数を指定します。

direction
ブラウズの方向を指定します。次のいずれかを指定できます。

値	動作
BEFORE	シード・ワードおよびシードの前のワードをアルファベット順にブラウズします。
AROUND	シード・ワードおよびシードの前後のワードをアルファベット順にブラウズします。
AFTER	シード・ワードおよびシードの後のワードをアルファベット順にブラウズします。

記号 CTX_QUERY.BROWSE_BEFORE、CTX_QUERY.BROWSE_AROUND および CTX_QUERY.BROWSE_AFTER も、これらのリテラル値に対して定義されています。

part_name
ブラウズする索引パーティション名を指定します。

例

結果表でのワードのブラウズ

```
begin
ctx_query.browse_words('myindex','dog','myres',numwords=>5,direction=>'AROUND');
end;

select word, doc_count from myres order by word;

WORD       DOC_COUNT
--------   ----------
CZAR       15
DARLING    5
DOC        73
DUNK       100
EAR        3
```

結果配列でのワードのブラウズ

```
set serveroutput on;
declare
  resarr ctx_query.browse_tab;
begin
ctx_query.browse_words('myindex','dog',resarr,5,CTX_QUERY.BROWSE_AROUND);
for i in 1..resarr.count loop
  dbms_output.put_line(resarr(i).word || ':' || resarr(i).doc_count);
end loop;
end;
```

COUNT_HITS

指定された問合せへのヒット数を戻します。実測または予測モードで COUNT_HITS をコールできます。実測モードは問合せへの正確なヒット数を戻します。予測モードは上限の見積りを戻しますが、実測モードよりも高速に実行します。

構文

```
CTX_QUERY.COUNT_HITS (
    index_name  IN VARCHAR2,
    text_query  IN VARCHAR2,
    exact       IN BOOLEAN  DEFAULT TRUE,
    part_name   IN VARCHAR2 DEFAULT NULL
) RETURN NUMBER;
```

index_name
索引の名前を指定します。

text_query
問合せを指定します。

exact
実測カウントには TRUE を指定します。上限を見積るには FALSE を指定します。FALSE を指定することによって戻された数値はあまり正確ではありませんが、より高速に実行されます。

part_name
問い合せる索引パーティション名を指定します。

注意

問合せが構造化基準を含む場合は、SELECT COUNT(*) を使用してください。

EXPLAIN

CTX_QUERY.EXPLAIN を使用して、問合せ式に実行計画情報を生成します。実行計画では、テキスト問合せ式に対して解析ツリーがグラフィカル表示されます。この情報は結果表に格納されます。

このプロシージャは問合せを実行しません。かわりに、問合せの発行前に、問合せの拡張および解析の方法をユーザーに知らせます。これは、特に STEM、ワイルド・カード、シソーラス、FUZZY、SOUNDEX または ABOUT 問合せに有効です。解析ツリーには、次の情報も表示されます。

- 実行の順序（演算子の優先順位）
- ABOUT 問合せの正規化
- 問合せ式の最適化
- ストップワード変換
- コンポジット・ワード・トークンの分類

Oracle による問合せの評価方法を理解しておくと、問合せの詳細化やデバッグを行うときに役立ちます。また、実行計画情報を使用してユーザーがより高度な問合せを作成できるように、アプリケーションを設計できます。

制限事項

リモート問合せで EXPLAIN は使用できません。

構文

```
CTX_QUERY.EXPLAIN(
        index_name     IN VARCHAR2,
        text_query     IN VARCHAR2,
        explain_table  IN VARCHAR2,
        sharelevel     IN NUMBER DEFAULT 0,
        explain_id     IN VARCHAR2 DEFAULT NULL,
        part_name      IN VARCHAR2 DEFAULT NULL
);
```

index_name
問い合せる索引名を指定します。

text_query
行を選択する基準として使用する問合せ式を指定します。

text_query にワイルド・カード、FUZZY または SOUNDEX 演算子が含まれている場合、このプロシージャは索引表を調べて拡張を判断します。

ワイルド・カード、FUZZY（?）および SOUNDEX（!）式のフィードバックでは、通常の問合せで削除が遅延した場合、その理由を判断できません。

explain_table
text_query に対する解析ツリー表現の格納に使用する表の名前を指定します。EXPLAIN からの結果を格納するには、表に対して少なくとも INSERT 権限および DELETE 権限が必要です。

> **関連項目：** 実行計画表の構造の詳細は、付録 A「結果表」の「実行計画表」を参照してください。

sharelevel
explain_table を複数の EXPLAIN コールで共有するかどうかを指定します。排他使用には 0（ゼロ）を、共有使用には 1 を指定します。このパラメータのデフォルトは 0（排他使用）です。

0（ゼロ）を指定すると、システムは次の EXPLAIN のコールの前に自動的に結果表を切り捨てます。

共有使用の 1 を指定すると、このプロシージャは結果表を切り捨てません。同じ explain_id を持つ結果のみが更新されます。同じ explain_id の結果が存在しない場合は、新しい結果が実行計画表に追加されます。

explain_id
複数の EXPLAIN コールが同じ共有実行計画表を使用する場合は、EXPLAIN プロシージャが戻した実行計画結果を識別する名前を指定します。このパラメータのデフォルトは NULL です。

part_name
問い合せる索引パーティション名を指定します。

例

実行計画表の作成

たとえば、test_explain という名前の実行計画表を作成するには、次の SQL 文を使用します。

```
create table test_explain(
         explain_id varchar2(30),
         id number,
         parent_id number,
         operation varchar2(30),
         options varchar2(30),
         object_name varchar2(64),
         position number,
         cardinality number);
```

CTX_QUERY.EXPLAIN の実行

comp% OR ?smith などの問合せ式の拡張を取得するには、次のように CTX_QUERY.EXPLAIN を使用します。

```
ctx_query.explain(
         index_name => 'newindex',
         text_query => 'comp% OR ?smith',
         explain_table => 'test_explain',
         sharelevel => 0,
         explain_id => 'Test');
```

実行計画表からのデータの取出し

実行計画表を読み込むには、次のように列を選択できます。

```
select explain_id, id, parent_id, operation, options, object_name, position
from test_explain order by id;
```

次のように、出力は ID 順に並べられ、階層問合せをシミュレートします。

```
EXPLAIN_ID    ID PARENT_ID OPERATION    OPTIONS OBJECT_NAME POSITION
---------- ---- --------- ------------ ------- ----------- --------
Test           1         0 OR           NULL    NULL          1
Test           2         1 EQUIVALENCE  NULL    COMP%         1
Test           3         2 WORD         NULL    COMPTROLLER   1
Test           4         2 WORD         NULL    COMPUTER      2
Test           5         1 EQUIVALENCE  (?)     SMITH         2
Test           6         5 WORD         NULL    SMITH         1
Test           7         5 WORD         NULL    SMYTHE        2
```

関連項目

第 3 章「CONTAINS 問合せ演算子」

付録 H「ストップワード変換」

HFEEDBACK

英語またはフランス語の場合、このプロシージャは、指定された問合せへの階層問合せフィードバック情報（上位語、下位語および関連語）を生成します。

上位語、下位語および関連語情報はナレッジ・ベースから取得されます。ただし、索引中にも存在するナレッジ・ベース語句のみが、問合せフィードバック情報として戻ります。これによって、HFEEDBACK プロシージャから戻された語句が、現在索引付けされているドキュメント・セットよりヒットする可能性が高くなります。

階層問合せフィードバック情報は、他の問合せ語句をユーザーに提示する場合に有効です。

注意： CTX_QUERY.HFEEDBACK は、英語とフランス語のみでサポートされています。

構文

```
CTX_QUERY.HFEEDBACK(
           index_name     IN VARCHAR2,
           text_query     IN VARCHAR2,
           feedback_table IN VARCHAR2,
           sharelevel     IN NUMBER DEFAULT 0,
           feedback_id    IN VARCHAR2 DEFAULT NULL,
           part_name      IN VARCHAR2 DEFAULT NULL
           );
```

index_name
問い合せるテキスト列の索引の名前を指定します。

text_query
行を選択する基準として使用する問合せ式を指定します。

feedback_table
フィードバック語の格納に使用する表の名前を指定します。

関連項目： 実行計画表の構造の詳細は、付録 A「結果表」の「HFEEDBACK 表」を参照してください。

sharelevel
feedback_table を複数の HFEEDBACK コールで共有するかどうかを指定します。排他使用には 0（ゼロ）を、共有使用には 1 を指定します。このパラメータのデフォルトは 0（排他使用）です。

0（ゼロ）を指定すると、システムは次の HFEEDBACK のコールの前に自動的にフィードバック表を切り捨てます。

共有使用の 1 を指定すると、このプロシージャはフィードバック表を切り捨てません。同じ feedback_id を持つ結果のみが更新されます。同じ feedback_id の結果が存在しない場合は、新しい結果がフィードバック表に追加されます。

feedback_id
複数の HFEEDBACK コールが同じ共有フィードバック表を使用する場合は、HFEEDBACK のコールによって戻されたフィードバック結果を識別する値を指定します。このパラメータのデフォルトは NULL です。

part_name
問い合せる索引パーティション名を指定します。

例

HFEEDBACK 結果表の作成

CTX_QUERY.HFEEDBACK で使用する結果表を次のように作成します。

```
CREATE TABLE restab (
  feedback_id VARCHAR2(30),
  id          NUMBER,
  parent_id   NUMBER,
  operation   VARCHAR2(30),
  options     VARCHAR2(30),
  object_name VARCHAR2(80),
  position    NUMBER,
  bt_feedback ctx_feedback_type,
  rt_feedback ctx_feedback_type,
  nt_feedback ctx_feedback_type
) NESTED TABLE bt_feedback STORE AS res_bt
  NESTED TABLE rt_feedback STORE AS res_rt
  NESTED TABLE nt_feedback STORE AS res_nt;
```

CTX_FEEDBACK_TYPE は CTXSYS スキーマのシステム定義型です。

関連項目： HFEEDBACK 表の構造の詳細は、付録 A「結果表」の「HFEEDBACK 表」を参照してください。

CTX_QUERY.HFEEDBACK のコール

次のコードは、*computer industry* の問合せを持つ hfeedback プロシージャをコールします。

```
BEGIN
ctx_query.hfeedback (index_name     => 'my_index',
                     text_query     => 'computer industry',
                     feedback_table => 'restab',
                     sharelevel     => 0,
                     feedback_id    => 'query10'
                    );
END;
```

結果表からの選択

次のコードは、結果表からフィードバック・データを取り出します。このコードによって、上位語、下位語および関連語フィードバックを NESTED TABLE から個別に取り出します。

```
DECLARE
  i NUMBER;
BEGIN
  FOR frec IN (
    SELECT object_name, bt_feedback, rt_feedback, nt_feedback
    FROM restab
    WHERE feedback_id = 'query10' AND object_name IS NOT NULL
  ) LOOP

    dbms_output.put_line('Broader term feedback for ' || frec.object_name ||
':');
    i := frec.bt_feedback.FIRST;
    WHILE i IS NOT NULL LOOP
      dbms_output.put_line(frec.bt_feedback(i).text);
      i := frec.bt_feedback.NEXT(i);
    END LOOP;

    dbms_output.put_line('Related term feedback for ' || frec.object_name ||
':');
    i := frec.rt_feedback.FIRST;
    WHILE i IS NOT NULL LOOP
      dbms_output.put_line(frec.rt_feedback(i).text);
      i := frec.rt_feedback.NEXT(i);
    END LOOP;

    dbms_output.put_line('Narrower term feedback for ' || frec.object_name ||
':');
    i := frec.nt_feedback.FIRST;
    WHILE i IS NOT NULL LOOP
      dbms_output.put_line(frec.nt_feedback(i).text);
```

```
      i := frec.nt_feedback.NEXT(i);
    END LOOP;

  END LOOP;
END;
```

サンプル出力

次の出力は、*computer industry* について問い合せる前述の例に対するものです。

```
Broader term feedback for computer industry:
hard sciences
Related term feedback for computer industry:
computer networking
electronics
knowledge
library science
mathematics
optical technology
robotics
satellite technology
semiconductors and superconductors
symbolic logic
telecommunications industry
Narrower term feedback for computer industry:
ABEND - abnormal end of task
AT&T Starlans
ATI Technologies, Incorporated
ActivCard
Actrade International Ltd.
Alta Technology
Amiga Format
Amiga Library Services
Amiga Shopper
Amstrat Action
Apple Computer, Incorporated
..
```

注意： 取得した HFEEDBACK 情報は、索引およびナレッジ・ベースの内容に依存するため、前述の内容とは異なる可能性があります。

REMOVE_SQE

CTX_QUERY.REMOVE_SQE プロシージャは、指定されたストアド・クエリー式を削除します。

構文

```
CTX_QUERY.REMOVE_SQE(query_name IN VARCHAR2);
```

query_name
削除するストアド・クエリー式の名前を指定します。

例

```
begin
ctx_query.remove_sqe('disasters');
end;
```

STORE_SQE

このプロシージャは、ストアド・クエリー式を作成します。問合せ定義のみが格納されます。

サポートされている演算子

ストアド・クエリー式は、すべての CONTAINS 問合せ演算子をサポートしています。また、すべての特殊文字、および他のストアド・クエリー式も含めて、問合せ式で使用できるすべてのコンポーネントをサポートしています。

権限

ユーザーは、所有しているストアド・クエリー式を作成および削除できます。ユーザーは、誰が所有するストアド・クエリー式でも使用できます。CTXSYS ユーザーは、どのユーザーのストアド・クエリー式も作成および削除できます。

構文

```
CTX_QUERY.STORE_SQE(query_name      IN VARCHAR2,
                    text_query      IN VARCHAR2);
```

query_name
作成するストアド・クエリー式の名前を指定します。CTXSYS ユーザーの場合は、この名前を user.name で指定できます。

text_query
query_name に関連付ける問合せ式を指定します。

例

```
begin
ctx_query.store_sqe('disasters', 'hurricanes | earthquakes');
end;
```

11

CTX_REPORT パッケージ

この章では、CTX_REPORT パッケージを使用して索引レポートを作成する方法を説明します。この章の内容は次のとおりです。

- CTX_REPORT のプロシージャ
- 複数バージョンのファンクションの使用方法

CTX_REPORT のプロシージャ

CTX_REPORT パッケージには、次のプロシージャが含まれています。

名前	説明
DESCRIBE_INDEX	索引を説明するレポートを作成します。
DESCRIBE_POLICY	ポリシーを説明するレポートを作成します。
CREATE_INDEX_SCRIPT	指定した索引を複製する SQL*Plus スクリプトを作成します。
CREATE_POLICY_SCRIPT	指定したポリシーを複製する SQL*Plus スクリプトを作成します。
INDEX_SIZE	索引の内部オブジェクト、その表領域および使用サイズを表示するレポートを作成します。
INDEX_STATS	索引の様々な統計を表示するレポートを作成します。
TOKEN_INFO	デコードされたトークンに関する情報を表示するレポートを作成します。
TOKEN_TYPE	名前を変換し、数値のトークン型を戻します。

複数バージョンのファンクションの使用方法

CTX_REPORT パッケージの一部のプロシージャには、ファンクション・バリエーションがあります。これらのファンクションは、次のようにコールできます。

```
select ctx_report.describe_index('MYINDEX') from dual;
```

SQL*Plus では、サポートの目的で、次のように出力ファイルを生成して送信できます。

```
set long 64000
set pages 0
set heading off
set feedback off
spool outputfile
select ctx_report.describe_index('MYINDEX') from dual;
spool off
```

DESCRIBE_INDEX

索引を説明するレポートを作成します。このレポートには、索引メタデータの設定、使用されている索引付けオブジェクト、オブジェクトの属性の設定および索引パーティションの説明（ある場合）が含まれます。

この操作は、IN OUT CLOB パラメータを指定したプロシージャとして、またはレポートを CLOB で戻すファンクションとしてコールできます。

構文

```
procedure CTX_REPORT.DESCRIBE_INDEX(
  index_name IN VARCHAR2,
  report     IN OUT NOCOPY CLOB
);

function CTX_REPORT.DESCRIBE_INDEX(
  index_name IN VARCHAR2
) return CLOB;
```

index_name
説明する索引の名前を指定します。

report
レポートを書き込む CLOB ロケータを指定します。

report が NULL の場合は、セッション中に一時 CLOB が作成されて戻ります。この一時 CLOB は、必要に応じてコール元で解放してください。

report CLOB はレポートの生成前に切り捨てられます。したがって、既存の内容はこのコールによって上書きされます。

DESCRIBE_POLICY

ポリシーを説明するレポートを作成します。このレポートには、ポリシー・メタデータの設定、使用されている索引付けオブジェクト、オブジェクトの属性の設定が含まれます。

この操作は、IN OUT CLOB パラメータを指定したプロシージャとして、またはレポートを CLOB で戻すファンクションとしてコールできます。

構文

```
procedure CTX_REPORT.DESCRIBE_POLICY(
  policy_name IN VARCHAR2,
  report      IN OUT NOCOPY CLOB
);

function CTX_REPORT.DESCRIBE_POLICY(
  policy_name IN VARCHAR2
) return CLOB;
```

policy_name
説明するポリシーの名前を指定します。

report
レポートを書き込む CLOB ロケータを指定します。

report が NULL の場合は、セッション中に一時 CLOB が作成されて戻ります。この一時 CLOB は、必要に応じてコール元で解放してください。

report CLOB はレポートの生成前に切り捨てられます。したがって、既存の内容はこのコールによって上書きされます。

CREATE_INDEX_SCRIPT

指定したテキスト索引を複製する SQL*Plus スクリプトを作成します。

作成されたスクリプトには、指定したテキスト索引に使用されているプリファレンスと同一のプリファレンスが含まれます。ただし、プリファレンス名は別の名前になります。

この操作は、IN OUT CLOB パラメータを指定したプロシージャとして、またはレポートを CLOB で戻すファンクションとしてコールできます。

構文

```
procedure CTX_REPORT.CREATE_INDEX_SCRIPT(
  index_name      in varchar2,
  report          in out nocopy clob,
  prefname_prefix in varchar2 default null
);

function CTX_REPORT.CREATE_INDEX_SCRIPT(
  index_name      in varchar2,
  prefname_prefix in varchar2 default null
) return clob;
```

index_name
索引の名前を指定します。

report
スクリプトを書き込む CLOB ロケータを指定します。

report が NULL の場合は、セッション中に一時 CLOB が作成されて戻ります。この一時 CLOB は、必要に応じてコール元で解放してください。

report CLOB はレポートの生成前に切り捨てられます。したがって、既存の内容はこのコールによって上書きされます。

prefname_prefix
オプションで、プリファレンス名に使用するプリフィックスを指定します。

prefname_prefix を省略または NULL にした場合は、索引名が使用されます。prefname_prefix は、索引の長さに関する制限に従います。

CREATE_POLICY_SCRIPT

指定したテキスト・ポリシーを複製する SQL*Plus スクリプトを作成します。

作成されたスクリプトには、指定したテキスト・ポリシーに使用されているプリファレンスと同一のプリファレンスが含まれます。

この操作は、IN OUT CLOB パラメータを指定したプロシージャとして、またはレポートを CLOB で戻すファンクションとしてコールできます。

構文

```
procedure CTX_REPORT.CREATE_POLICY_SCRIPT(
  policy_name      in varchar2,
  report          in out nocopy clob,
  prefname_prefix in varchar2 default null
);

function CTX_REPORT.CREATE_POLICY_SCRIPT(
  policy_name      in varchar2,
  prefname_prefix in varchar2 default null
) return clob;
```

policy_name
ポリシーの名前を指定します。

report
スクリプトを書き込むロケータを指定します。

report が NULL の場合は、セッション中に一時 CLOB が作成されて戻ります。この一時 CLOB は、必要に応じてコール元で解放してください。

report CLOB はレポートの生成前に切り捨てられます。したがって、既存の内容はこのコールによって上書きされます。

prefname_prefix
オプションで、プリファレンス名に使用するプリフィックスを指定します。prefname_prefix を省略または NULL にした場合は、ポリシー名が使用されます。prefname_prefix は、ポリシーの長さに関する制限に従います。

INDEX_SIZE

テキスト索引またはテキスト索引パーティションの内部オブジェクト、その割り当て済み表領域および使用サイズを表示するレポートを作成します。

この操作は、IN OUT CLOB パラメータを指定したプロシージャとして、またはレポートを CLOB で戻すファンクションとしてコールできます。

構文

```
procedure CTX_REPORT.INDEX_SIZE(
  index_name IN VARCHAR2,
  report     IN OUT NOCOPY CLOB,
  part_name  IN VARCHAR2 DEFAULT NULL
);

function CTX_REPORT.INDEX_SIZE(
  index_name  IN VARCHAR2,
  part_name   IN VARCHAR2 DEFAULT NULL
) return clob;
```

index_name
説明する索引の名前を指定します。

report
レポートを書き込む CLOB ロケータを指定します。

report が NULL の場合は、セッション中に一時 CLOB が作成されて戻ります。この一時 CLOB は、必要に応じてコール元で解放してください。

report CLOB はレポートの生成前に切り捨てられます。したがって、既存の内容はこのコールによって上書きされます。

part_name
索引パーティションの名前を指定します（オプション）。part_name が NULL で、索引がローカル・パーティション・テキスト索引の場合は、すべてのパーティションに関する全オブジェクトが表示されます。part_name を指定した場合は、特定のパーティションのオブジェクトのみが表示されます。

INDEX_STATS

テキスト索引について計算された様々な統計を表示するレポートを作成します。

このプロシージャでは、テキスト索引表全体がスキャンされるため、大規模な索引の場合は実行時間が長くなる可能性があります。

セッション中に CTXSYS 一時表領域に一時表が作成され、使用されます。

```
procedure index_stats(
  index_name in varchar2,
  report     in out nocopy clob,
  part_name  in varchar2 default null,
  frag_stats in boolean default TRUE,
  list_size  in number  default 100
);
```

index_name
説明する索引の名前を指定します。CONTEXT、CTXCAT、CTXRULE または CTXXPATH の各索引を指定できます。

report
レポートを書き込む CLOB ロケータを指定します。`report` が NULL の場合は、セッション中に一時 CLOB が作成されて戻ります。この一時 CLOB は、必要に応じてコール元で解放してください。

`report` CLOB はレポートの生成前に切り捨てられます。したがって、既存の内容はこのコールによって上書きされます。

part_name
索引パーティションの名前を指定します。索引がローカル・パーティション索引の場合は、`part_name` を指定する必要があります。指定した索引パーティションに関する統計が計算されます。

frag_stats
断片化の統計を計算する場合は、TRUE を指定します。FALSE を指定すると、索引データのサイズに関する統計は表示されませんが、トークンの統計を計算する時間が短縮され、必要なリソースも少なくなります。

list_size
各コンパイル済みリストに表示する要素数を指定します。`list_size` の最大値は 1000 です。

例

CONTEXT 索引に関する INDEX_STATS のサンプル出力を次に示します。このレポートは、明確にするために一部が切り捨てられています。トークン統計の一部と断片化統計のすべてが記載されています。

断片化統計は、レポートの最後にあります。このレポートによって、最適な行の断片化、予想される索引内の不要データ量、および最も断片化されているトークンのリストを確認できます。CTX_DDL.OPTIMIZE_INDEX の実行によって、索引がクリーン・アップされます。

```
=============================================================
              STATISTICS FOR "DR_TEST"."TDRBPRX21"
=============================================================

indexed documents:                                        53
allocated docids:                                         68
$I rows:                                              16,259

-------------------------------------------------------------
                        TOKEN STATISTICS
-------------------------------------------------------------

unique tokens:                                        13,445
average $I rows per token:                              1.21
tokens with most $I rows:
  telecommunications industry (THEME)                      6
  science and technology (THEME)                           6
  EMAIL (FIELD SECTION "SOURCE")                           6
  DEC (FIELD SECTION "TIMESTAMP")                          6
  electronic mail (THEME)                                  6
  computer networking (THEME)                              6
  communications (THEME)                                   6
  95 (FIELD SECTION "TIMESTAMP")                           6
  15 (FIELD SECTION "TIMESTAMP")                           6
  HEADLINE (ZONE SECTION)                                  6

average size per token:                                    8
tokens with largest size:
  T (NORMAL)                                             405
  SAID (NORMAL)                                          313
  HEADLINE (ZONE SECTION)                                272
  NEW (NORMAL)                                           267
  I (NORMAL)                                             230
  MILLION (PREFIX)                                       222
  D (NORMAL)                                             219
  MILLION (NORMAL)                                       215
```

```
  U (NORMAL)                                        192
  DEC (FIELD SECTION "TIMESTAMP")                   186

average frequency per token:                       2.00
most frequent tokens:
  HEADLINE (ZONE SECTION)                           68
  DEC (FIELD SECTION "TIMESTAMP")                   62
  95 (FIELD SECTION "TIMESTAMP")                    62
  15 (FIELD SECTION "TIMESTAMP")                    62
  T (NORMAL)                                        61
  D (NORMAL)                                        59
  881115 (THEME)                                    58
  881115 (NORMAL)                                   58
  I (NORMAL)                                        55
  geography (THEME)                                 52

token statistics by type:
  token type:                                   NORMAL
    unique tokens:                               6,344
    total rows:                                  7,631
    average rows:                                 1.20
    total size:                        67,445 (65.86 KB)
    average size:                                   11
    average frequency:                            2.33
    most frequent tokens:
      T                                             61
      D                                             59
      881115                                        58
      I                                             55
      SAID                                          45
      C                                             43
      NEW                                           36
      MILLION                                       32
      FIRST                                         28
      COMPANY                                       27

  token type:                                   THEME
    unique tokens:                               4,563
    total rows:                                  5,523
    average rows:                                 1.21
    total size:                        21,930 (21.42 KB)
    average size:                                    5
    average frequency:                            2.40
    most frequent tokens:
      881115                                        58
      political geography                           52
      geography                                     52
```

```
      United States                                          51
      business and economics                                 50
      abstract ideas and concepts                            48
      North America                                          48
      science and technology                                 46
      NKS                                                    34
      nulls                                                  34
```

このレポートの断片化統計の部分は、次のとおりです。

```
------------------------------------------------------------------
                    FRAGMENTATION STATISTICS
------------------------------------------------------------------

total size of $I data:                     116,772 (114.04 KB)

$I rows:                                            16,259
estimated $I rows if optimal:                       13,445
estimated row fragmentation:                           17 %

garbage docids:                                        15
estimated garbage size:                     21,379 (20.88 KB)

most fragmented tokens:
  telecommunications industry (THEME)                  83 %
  science and technology (THEME)                       83 %
  EMAIL (FIELD SECTION "SOURCE")                       83 %
  DEC (FIELD SECTION "TIMESTAMP")                      83 %
  electronic mail (THEME)                              83 %
  computer networking (THEME)                          83 %
  communications (THEME)                               83 %
  95 (FIELD SECTION "TIMESTAMP")                       83 %
  HEADLINE (ZONE SECTION)                              83 %
  15 (FIELD SECTION "TIMESTAMP")                       83 %
```

TOKEN_INFO

デコードされたトークンに関する情報を表示するレポートを作成します。このプロシージャでは、トークンの情報全体がスキャンされるため、大規模なトークンの場合は実行時間が長くなる可能性があります。

この操作は、IN OUT CLOB パラメータを指定したプロシージャとして、またはレポートを CLOB で戻すファンクションとしてコールできます。

構文

```
procedure CTX_REPORT.TOKEN_INFO(
  index_name      in varchar2,
  report          in out nocopy clob,
  token           in varchar2,
  token_type      in number,
  part_name       in varchar2 default null,
  raw_info        in boolean  default FALSE,
  decoded_info    in boolean  default TRUE
);

function CTX_REPORT.TOKEN_INFO(
  index_name      in varchar2,
  token           in varchar2,
  token_type      in number,
  part_name       in varchar2 default null,
  raw_info        in varchar2 default 'N',
  decoded_info    in varchar2 default 'Y'
) return clob;
```

index_name
索引の名前を指定します。

report
レポートを書き込む CLOB ロケータを指定します。

report が NULL の場合は、セッション中に一時 CLOB が作成されて戻ります。この一時 CLOB は、必要に応じてコール元で解放してください。

report CLOB はレポートの生成前に切り捨てられます。したがって、既存の内容はこのコールによって上書きされます。トークンは、渡したトークン型によっては大 / 小文字が区別されます。

token
トークン・テキストを指定します。

token_type
トークン型を指定します。THEME、ZONE、ATTR、PATH および PATH ATTR のトークン型は、大 / 小文字が区別されます。

これ以外のトークン型でも、大 / 小文字を区別する索引のレクサーを通じて渡された場合は、トークン入力で大 / 小文字が区別されます。

part_name
索引パーティションの名前を指定します。

索引がローカル・パーティション索引の場合は、`part_name` を指定する必要があります。TOKEN_INFO は、指定した索引パーティションのみに適用されます。

raw_info
索引データの 16 進ダンプを含める場合は、TRUE を指定します。TRUE を指定すると、token_info 列に RAW データの 16 進ダンプが含まれます。

decoded_info
デコードを指定し、DOCID およびオフセット・データを追加します。FALSE を指定すると、トークン情報はデコードされません。これは、データのダンプのみを行う場合に便利です。

TOKEN_TYPE

英語名を数値のトークン型に変換する補助的なファンクションです。token_info または token_type を取得するその他の CTX API で使用すると便利です。

```
function token_type(
  index_name in varchar2,
  type_name  in varchar2
) return number;

TOKEN_TYPE_TEXT      constant number := 0;
TOKEN_TYPE_THEME     constant number := 1;
TOKEN_TYPE_ZONE_SEC  constant number := 2;
TOKEN_TYPE_ATTR_TEXT constant number := 4;
TOKEN_TYPE_ATTR_SEC  constant number := 5;
TOKEN_TYPE_PREFIX    constant number := 6;
TOKEN_TYPE_PATH_SEC  constant number := 7;
TOKEN_TYPE_PATH_ATTR constant number := 8;
TOKEN_TYPE_STEM      constant number := 9;
```

index_name
索引の名前を指定します。

type_name
token_type の英語名を指定します。次に、適切な入力文字列を示します。すべての入力で、大 / 小文字は区別されません。

入力	意味	戻される型
TEXT	通常のテキスト・トークン	0
THEME	テーマ・トークン	1
ZONE SEC	ゾーン・トークン	2
ATTR TEXT	属性内のテキスト	4
ATTR SEC	属性セクション	5
PREFIX	プリフィックス・トークン	6
PATH SEC	パス・セクション	7
PATH ATTR	パス属性セクション	8
STEM	語幹トークン	9

入力	意味	戻される型
FIELD <name> TEXT	フィールド・セクション <name> のテキスト・トークン	16-79
FILED <name> PREFIX	フィールド・セクション <name> のプリフィックス・トークン	616-916
FIELD <name> STEM	フィールド・セクション <name> の語幹トークン	916-979

FIELD 型では、索引メタデータを読み込む必要があります。したがって、この目的のためにコールが大量に発生する場合は、token_type を繰返しコールするのではなく、ローカル変数に値をキャッシュしたほうが効率的です。

定数型（0 ～ 9）にも、このパッケージで定義済みの定数があります。

例

```
typenum := ctx_report.token_type('myindex', 'field author text');
```

12

CTX_THES パッケージ

この章では、シソーラスを管理および参照するための CTX_THES パッケージの使用方法について説明します。これらのシソーラスのファンクションは、特に指定されていない場合は、ISO-2788 および ANSI Z39.19 規格に基づいています。

シソーラスへの情報の格納方法を理解しておくと、シソーラス演算子を使用した問合せを記述する際に役立ちます。また、シソーラスを使用して、英語とフランス語での ABOUT 問合せおよびドキュメント・テーマの生成に使用するナレッジ・ベースを拡張できます。

CTX_THES には、次のストアド・プロシージャおよびファンクションが含まれています。

名前	説明
ALTER_PHRASE	シソーラス句を変更します。
ALTER_THESAURUS	シソーラスを改名または切り捨てます。
BT	句のすべての上位語を戻します。
BTG	句のすべての上位汎用語を戻します。
BTI	句のすべての上位インスタンス語を戻します。
BTP	句のすべての上位部分語を戻します。
CREATE_PHRASE	指定されたシソーラスに句を追加します。
CREATE_RELATION	2 つの句の間にリレーションを作成します。
CREATE_THESAURUS	指定されたシソーラスを作成します。
CREATE_TRANSLATION	句に対する新しい翻訳を作成します。
DROP_PHRASE	シソーラスから句を削除します。
DROP_RELATION	2 つの句の間のリレーションを削除します。
DROP_THESAURUS	指定されたシソーラスをシソーラス表から削除します。

名前	説明
DROP_TRANSLATION	句に対する翻訳を削除します。
HAS_RELATION	シソーラス・リレーションの存在の有無をテストします。
NT	句のすべての下位語を戻します。
NTG	句のすべての下位汎用語を戻します。
NTI	句のすべての下位インスタンス語を戻します。
NTP	句のすべての下位部分語を戻します。
OUTPUT_STYLE	拡張ファンクションの出力スタイルを設定します。
PT	句の優先語を戻します。
RT	句の関連語を戻します。
SN	句のスコープ・ノートを戻します。
SYN	句のシノニム語句を戻します。
THES_TT	句のすべての最上位語を戻します。
TR	句の外国語の等価語を戻します。
TRSYN	句の外国語の等価語、句のシノニムおよびシノニムの外国語の等価語を戻します。
TT	句の最上位語を戻します。
UPDATE_TRANSLATION	既存の翻訳を更新します。

関連項目： シソーラス演算子の詳細は、第 3 章「CONTAINS 問合せ演算子」を参照してください。

ALTER_PHRASE

シソーラスの既存の句を変更します。句を変更できるのは、CTXSYS またはシソーラスの所有者のみです。

構文

```
CTX_THES.ALTER_PHRASE(tname      in varchar2,
                      phrase     in varchar2,
                      op         in varchar2,
                      operand    in varchar2 default null);
```

tname
シソーラス名を指定します。

phrase
変更する句を指定します。

op
変更操作を文字列または記号として指定します。次の操作の 1 つを、op と operand の組で指定できます。

op	意味	operand
RENAME または CTX_THES.OP_RENAME	句を改名します。シソーラスに新しい句がすでに存在している場合は、このプロシージャによって例外が発生します。	新しい句を指定します。修飾子を挿入し、句に対する修飾子を変更、追加または削除できます。
PT または CTX_THES.OP_PT	句を優先語にします。シノニム・リングにある既存の優先語は、非優先シノニムになります。	ありません。
SN または CTX_THES.OP_SN	句のスコープ・ノートを変更します。	新しいスコープ・ノートを指定します。

operand
引数を指定し、操作を変更します。前述の表を参照してください。

例

シソーラスの中のスペルの間違っているワードを修正します。

```
ctx_thes.alter_phrase('thes1', 'tee', 'rename', 'tea');
```

mercury (metal) から修飾子を削除します。

```
ctx_thes.alter_phrase('thes1', 'mercury (metal)', 'rename', 'mercury');
```

mercury に修飾子を追加します。

```
ctx_thes.alter_phrase('thes1', 'mercury', 'rename', 'mercury (planet)');
```

シノニム・リング内で Kowalski を優先語にします。

```
ctx_thes.alter_phrase('thes1', 'Kowalski', 'pt');
```

view cameras のスコープ・ノートを変更します。

```
ctx_thes.alter_phrase('thes1', 'view cameras', 'sn', 'Cameras with lens focusing');
```

ALTER_THESAURUS

このプロシージャを使用して、既存のシソーラスを改名または切り捨てます。指定されたシソーラスでこのファンクションをコールできるのは、シソーラスの所有者または CTXSYS ユーザーのみです。

構文

```
CTX_THES.ALTER_THESAURUS(tname     in   varchar2,
                         op        in   varchar2,
                         operand   in   varchar2 default null);
```

tname
シソーラス名を指定します。

op
変更操作を文字列または記号として指定します。次の 2 つの操作のいずれかを指定できます。

op	意味	operand
RENAME または CTX_THES.OP_RENAME	シソーラスを改名します。新しい名前がすでに存在している場合は、エラーが戻ります。	新しいシソーラス名を指定します。
TRUNCATE または CTX_THES.OP_TRUNCATE	シソーラスを切り捨てます。	ありません。

operand
引数を指定し、操作を変更します。前述の表を参照してください。

例

シソーラス THES1 を MEDICAL に改名します。

```
ctx_thes.alter_thesaurus('thes1', 'rename', 'medical');
```

または

```
ctx_thes.alter_thesaurus('thes1', ctx_thes.op_rename, 'medical');
```

すべての op 引数に対して記号を使用できますが、これ以降の例ではすべて文字列を使用しています。

シソーラス THES1 からすべての句およびリレーションを削除します。

```
ctx_thes.alter_thesaurus('thes1', 'truncate');
```

BT

このファンクションは、指定されたシソーラスに記録されているとおりに、句のすべての上位語を戻します。

構文 1: 表結果

```
CTX_THES.BT(restab IN OUT NOCOPY EXP_TAB,
            phrase IN VARCHAR2,
            lvl    IN NUMBER DEFAULT 1,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.BT(phrase IN VARCHAR2,
            lvl    IN NUMBER DEFAULT 1,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

lvl

戻す上位語のレベルを指定します。たとえば、2 を指定することによって、句の上位語の上位語までをすべて取得します。

tname

シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で上位語の文字列を戻します。

```
{bt1}|{bt2}|{bt3} ...
```

例

文字列結果

cat に対して次のようなエントリを持つ MY_THES というシソーラスがあるとします。

```
cat
   BT1 feline
     BT2 mammal
       BT3 vertebrate
         BT4 animal
```

cat に対する上位語を 2 レベル上まで検索するには、次の文を発行します。

```
set serveroutput on

declare
  terms varchar2(2000);
begin
  terms := ctx_thes.bt('CAT', 2, 'MY_THES');
  dbms_output.put_line('The broader expansion for CAT is: '||terms);
end;
```

このコードは、次の出力結果を戻します。

```
The broader expansion for CAT is: {cat}|{feline}|{mammal}
```

表結果

次のコードは、表結果を使用して *white wolf* に対する上位語の検索を実行します。

```
set serveroutput on

declare
  xtab ctx_thes.exp_tab;
begin
  ctx_thes.bt(xtab, 'white wolf', 2, 'my_thesaurus');
  for i in 1..xtab.count loop
    dbms_output.put_line(xtab(i).rel||' '||xtab(i).phrase);
  end loop;
end;
```

このコードは、次の出力結果を戻します。

```
PHRASE WHITE WOLF
BT WOLF
BT CANINE
BT ANIMAL
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「BROADER TERM（BT、BTG、BTP、BTI）」

BTG

このファンクションは、指定されたシソーラスに記録されているとおりに、句のすべての上位汎用語を戻します。

構文 1: 表結果

```
CTX_THES.BTG(restab IN OUT NOCOPY EXP_TAB,
             phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.BTG(phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

lvl

戻す上位語のレベルを指定します。たとえば、2 を指定することによって、句の上位語の上位語までをすべて取得します。

tname

シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で上位汎用語の文字列を戻します。

```
{bt1}|{bt2}|{bt3} ...
```

例

cat に対する上位汎用語を 2 レベル上まで検索するには、次の文を発行します。

```
set serveroutput on
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.btg('CAT', 2, 'MY_THES');
  dbms_output.put_line('the broader expansion for CAT is: '||terms);
end;
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「BROADER TERM（BT、BTG、BTP、BTI）」

BTI

このファンクションは、指定されたシソーラスに記録されているとおりに、句のすべての上位インスタンス語を戻します。

構文 1: 表結果

```
CTX_THES.BTI(restab IN OUT NOCOPY EXP_TAB,
             phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.BTI(phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

lvl

戻す上位語のレベルを指定します。たとえば、2 を指定することによって、句の上位語の上位語までをすべて取得します。

tname

シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で上位インスタンス語の文字列を戻します。

```
{bt1}|{bt2}|{bt3} ...
```

例

cat に対する上位インスタンス語を 2 レベル上まで検索するには、次の文を発行します。

```
set serveroutput on
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.bti('CAT', 2, 'MY_THES');
  dbms_output.put_line('the broader expansion for CAT is: '||terms);
end;
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「BROADER TERM（BT、BTG、BTP、BTI）」

BTP

このファンクションは、指定されたシソーラスに記録されているとおりに、句のすべての上位部分語を戻します。

構文 1: 表結果

```
CTX_THES.BTP(restab IN OUT NOCOPY EXP_TAB,
             phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.BTP(phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

lvl

戻す上位語のレベルを指定します。たとえば、2 を指定することによって、句の上位語の上位語までをすべて取得します。

tname

シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で上位語の文字列を戻します。

```
{bt1}|{bt2}|{bt3} ...
```

例

cat に対する 2 つの上位部分語を検索するには、次の文を発行します。

```
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.btp('CAT', 2, 'MY_THES');
  dbms_output.put_line('the broader expansion for CAT is: '||terms);
end;
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「BROADER TERM（BT、BTG、BTP、BTI）」

CREATE_PHRASE

CREATE_PHRASE プロシージャは、新しい句を指定されたシソーラスに追加します。

> **注意：** このプロシージャでシソーラスのリレーションを作成できますが、シソーラスにリレーションを作成するには、CTX_THES.CREATE_PHRASE よりも CTX_THES.CREATE_RELATION を使用することをお薦めします。

構文

```
CTX_THES.CREATE_PHRASE(tname    IN VARCHAR2,
                       phrase   IN VARCHAR2,
                       rel      IN VARCHAR2 DEFAULT NULL,
                       relname  IN VARCHAR2 DEFAULT NULL);
```

tname
新しい句を追加するシソーラスまたは既存の句があるシソーラスの名前を指定します。

phrase
シソーラスに追加する句または新しい関係を作成する句を指定します。

rel
phrase および *relname* の間の新しい関係を指定します。このパラメータは、下位互換用にのみサポートされます。CTX_THES.CREATE_RELATION を使用して、新しいリレーションをシソーラスに作成します。

relname
phrase に関連する既存の句を指定します。このパラメータは、下位互換用にのみサポートされます。CTX_THES.CREATE_RELATION を使用して、新しいリレーションをシソーラスに作成します。

戻り値

エントリの ID です。

例

句に対するエントリの作成

この例では、2 つの新しい句（*os* および *operating system*）が tech_thes という名前のシソーラスに作成されます。

```
begin
   ctx_thes.create_phrase('tech_thes','os');
   ctx_thes.create_phrase('tech_thes','operating system');
end;
```

CREATE_RELATION

シソーラスの中の 2 つの句の間にリレーションを作成します。

> **注意：** シソーラスにリレーションを作成するには、CTX_THES.CREATE_PHRASE よりも CTX_THES.CREATE_RELATION を使用することをお薦めします。

指定されたシソーラスでこのプロシージャをコールできるのは、シソーラスの所有者および CTXSYS ユーザーのみです。

構文

```
CTX_THES.CREATE_RELATION(tname     in    varchar2,
                         phrase    in    varchar2,
                         rel       in    varchar2,
                         relphrase in    varchar2);
```

tname
シソーラス名を指定します。

phrase
変更または作成する句を指定します。phrase が明確な同形異義語の場合は、修飾子を指定する必要があります。phrase がシソーラスに存在しない場合は作成されます。

rel
作成するリレーションを指定します。phrase から relphrase へのリレーションになります。次のリレーションのいずれかを指定できます。

リレーション	意味	relphrase
BT*/NT*	階層関係を追加します。	関連付けられた句を指定します。関連は、phrase から relphrase へ解析されます。
RT	対応関係を追加します。	対応付ける句を指定します。
SYN	句をシノニム・リングに追加します。	シノニム・リングの中の既存の句を指定します。
言語の指定	句に対する翻訳を追加します。	新しい翻訳句を指定します。

relphrase
関連付けられた句を指定します。relphrase が tname に存在しない場合に、relphrase が作成されます。前述の表を参照してください。

注意

rel に対して指定したリレーションは、phrase から relphrase へ解析されます。たとえば、上位語 animal を持つ dog があるとします。

```
dog
  BT animal
```

このリレーションを追加するには、引数を次のように指定します。

```
begin
CTX_THES.CREATE_RELATION('thes','dog','BT','animal');
end;
```

> **注意：** CTX_THES.CREATE_RELATION に対して指定する引数の順序は、CTX_THES.CREATE_PHRASE で指定する順序とは異なります。

例

リレーション VEHICLE NT CAR を作成します。

```
ctx_thes.create_relation('thes1', 'vehicle', 'NT', 'car');
```

you に対する日本語の翻訳を作成します。

```
ctx_thes.create_relation('thes1', 'you', 'JAPANESE:', 'kimi');
```

CREATE_THESAURUS

CREATE_THESAURUS プロシージャは、シソーラス表に、指定された名前を持つ空のシソーラスを作成します。

構文

```
CTX_THES.CREATE_THESAURUS(name           IN VARCHAR2,
                          casesens       IN BOOLEAN DEFAULT FALSE);
```

name
作成するシソーラスの名前を指定します。シソーラスの名前は一意である必要があります。指定した名前を持つシソーラスがすでに存在している場合、CREATE_THESAURUS はエラーを戻し、シソーラスは作成されません。

casesens
作成するシソーラスで大 / 小文字を区別するかどうかを指定します。casesens が *TRUE* の場合、Oracle は、指定されたシソーラスに入力されるすべての語句の大 / 小文字の区別を維持します。その結果、指定のシソーラスを使用する問合せの大 / 小文字が区別されます。

例

```
begin
   ctx_thes.create_thesaurus('tech_thes', FALSE);
end;
```

CREATE_TRANSLATION

このプロシージャを使用して、指定した言語で句に対する新しい翻訳を作成します。

構文

```
CTX_THES.CREATE_TRANSLATION(tname       in    varchar2,
                            phrase      in    varchar2,
                            language    in    varchar2,
                            translation in    varchar2);
```

tname
シソーラスの名前を 30 文字以内で指定します。

phrase
翻訳を追加するシソーラス内の句を指定します。その句がシソーラスにすでに存在している必要があります。存在しない場合はエラーが発生します。

language
翻訳の言語を 10 文字以内で指定します。

translation
翻訳済みの語句を 256 文字以内で指定します。

この句の翻訳がすでに存在している場合、この新しい翻訳は、元の翻訳を削除せずに追加されます。ただし、元の翻訳と同じでない場合に限ります。同じ翻訳を 2 回追加すると、エラーになります。

例

次のコードは、*dog* に対するスペイン語の翻訳を *my_thes* に追加します。

```
begin
   ctx_thes.create_translation('my_thes', 'dog', 'SPANISH', 'PERRO');
end;
```

DROP_PHRASE

シソーラスから句を削除します。指定されたシソーラスでこのプロシージャをコールできるのは、シソーラスの所有者および CTXSYS ユーザーのみです。

構文

```
CTX_THES.DROP_PHRASE(tname      in varchar2,
                     phrase     in varchar2);
```

tname
シソーラス名を指定します。

phrase
削除する句を指定します。phrase が明確な同形異義語の場合は、修飾子を挿入する必要があります。句が tname に存在しない場合は、このプロシージャによって例外が発生します。

BT* / NT* リレーションは、削除された句の前後にパッチされます。たとえば、A が BT B を持ち、B が BT C を持つ場合、B が削除された後、A は BT C を持ちます。

ワードが複数の上位語を持つ場合、各上位語に対する各下位語の関係が設定されます。

BT、BTG、BTP および BTI は別々の階層です。したがって、A が BTG B を持ち、B が BTI C を持つ場合、B が削除されると、A と C の間に暗黙的なリレーションは作成されないことに注意してください。

RT リレーションはパッチされません。たとえば、A が RT B を持ち、B が RT C を持つ場合、B が削除されると、A と C の間に対応関係は作成されません。

例

mythes に定義された次のリレーションがあるとします。

```
wolf
   BT canine
canine
   BT animal
```

句 *canine* を削除します。

```
begin
ctx_thes.drop_phrase('mythes', 'canine');
end;
```

シソーラスの結果はパッチされ、次のようになります。

```
wolf
   BT animal
```

DROP_RELATION

2 つの句の間のリレーションをシソーラスから削除します。

> **注意：** CTX_THES.DROP_RELATION は、2 つの句の間のリレーションのみを削除します。このコールで、句が削除されることはありません。

指定されたシソーラスでこのプロシージャをコールできるのは、シソーラスの所有者および CTXSYS ユーザーのみです。

構文

```
CTX_THES.DROP_RELATION(tname     in    varchar2,
                       phrase    in    varchar2,
                       rel       in    varchar2,
                       relphrase in    varchar2 default null);
```

tname
シソーラス名を指定します。

phrase
ファイリング句を指定します。

rel
削除するリレーションを指定します。phrase から relphrase へのリレーションになります。次のリレーションのいずれかを指定できます。

リレーション	意味	relphrase
BT*/NT*	階層関係を削除します。	オプションで relphrase を指定します。指定されない場合、句に対するこの種類のすべてのリレーションが削除されます。
RT	対応関係を削除します。	オプションで relphrase を指定します。指定されない場合、句に対するのすべての RT リレーションが削除されます。
SYN	句をそのシノニム・リングから削除します。	ありません。
PT	優先語の指定を句から削除します。句はシノニム・リングに残ります。	ありません。

リレーション	意味	relphrase
language	翻訳を句から削除します。	オプションで relphrase を指定します。ある言語の 1 つの句に対して複数の翻訳がある場合は、relphrase を指定できます。また、1 つの翻訳のみでも削除できます。 relphrase が NULL の場合は、その言語の句に対するすべての翻訳が削除されます。

relphrase
関連付けられた句を指定します。

注意

rel に対して指定したリレーションは、phrase から relphrase へ解析されます。たとえば、上位語 animal を持つ dog があるとします。

```
dog
  BT animal
```

このリレーションを削除するには、引数を次のように指定します。

```
begin
CTX_THES.DROP_RELATION('thes','dog','BT','animal');
end;
```

また、次のように NT を使用してこのリレーションを削除することもできます。

```
begin
CTX_THES.DROP_RELATION('thes','animal','NT','dog');
end;
```

例

リレーション VEHICLE NT CAR を削除します。

```
ctx_thes.drop_relation('thes1', 'vehicle', 'NT', 'car');
```

vehicle に対するすべての下位語リレーションを削除します。

```
ctx_thes.drop_relation('thes1', 'vehicle', 'NT');
```

me に対する日本語の翻訳を削除します。

```
ctx_thes.drop_relation('thes1', 'me', 'JAPANESE:');
```

me に対する指定した日本語の翻訳を削除します。

```
ctx_thes.drop_relation('thes1', 'me', 'JAPANESE:', 'boku')
```

DROP_THESAURUS

DROP_THESAURUS プロシージャは、指定されたシソーラスおよびそのシソーラス表にあるすべてのエントリを削除します。

構文

```
CTX_THES.DROP_THESAURUS(name IN VARCHAR2);
```

name
削除するシソーラスの名前を指定します。

例

```
begin
ctx_thes.drop_thesaurus('tech_thes');
end;
```

DROP_TRANSLATION

このプロシージャを使用して、句に対する 1 つ以上の翻訳を削除します。

構文

```
CTX_THES.DROP_TRANSLATION (tname       in    varchar2,
                           phrase      in    varchar2,
                           language    in    varchar2 default null,
                           translation in    varchar2 default null);
```

tname
シソーラスの名前を 30 文字以内で指定します。

phrase
翻訳を削除するシソーラス内の句を指定します。その句がシソーラスにすでに存在している必要があります。存在しない場合はエラーが発生します。

language
オプションで、翻訳の言語を 10 文字以内で指定します。指定しない場合は、その翻訳も指定できません。その結果、その句に対するすべての言語での翻訳が削除されます。その句に翻訳がない場合は、エラーが発生します。

translation
オプションで、削除する翻訳済みの語句を 256 文字以内で指定します。指定した翻訳が存在しない場合は、エラーが発生します。

例

次のコードは、*dog* に対するスペイン語の翻訳を削除します。

```
begin
   ctx_thes.drop_translation('my_thes', 'dog', 'SPANISH', 'PERRO');
end;
```

次のコードは、dog に対するすべての言語での全翻訳を削除します。

```
begin
   ctx_thes.drop_translation('my_thes', 'dog');
end;
```

HAS_RELATION

このプロシージャを使用して、実際に拡張操作を実行せずに、シソーラス・リレーションの存在の有無をテストします。その句に対するリレーションが指定したリスト内にある場合は、TRUE を戻します。

構文

```
CTX_THES.HAS_RELATION(phrase in varchar2,
                      rel in varchar2,
                      tname in varchar2 default 'DEFAULT')
  returns boolean;
```

phrase
句を指定します。

rel
単一のシソーラス・リレーションまたはカンマで区切られたリレーションのリストを指定します。ただし、PT は除外します。すべてのリレーションを示すには、ANY を指定します。

tname
シソーラス名を指定します。

例

次の例では、DEFAULT シソーラス内の句 *cat* に上位語または上位汎用語がある場合は、TRUE を戻します。

```
set serveroutput on
result boolean;

begin
 result := ctx_thes.has_relation('cat','BT,BTG');
 if (result) then dbms_output.put_line('TRUE');
  else  dbms_output.put_line('FALSE');
 end if;
end;
```

NT

このファンクションは、指定されたシソーラスに記録されているとおりに、句のすべての下位語を戻します。

構文 1: 表結果

```
CTX_THES.NT(restab IN OUT NOCOPY EXP_TAB,
            phrase IN VARCHAR2,
            lvl    IN NUMBER DEFAULT 1,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.NT(phrase IN VARCHAR2,
            lvl    IN NUMBER DEFAULT 1,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

関連項目： EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

lvl

戻す下位語のレベルを指定します。たとえば、2 を指定することによって、句の下位語の下位語までをすべて取得します。

tname

シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で下位語の文字列を戻します。

```
{nt1}|{nt2}|{nt3} ...
```

例

文字列結果

cat に対して次のようなエントリを持つ MY_THES というシソーラスがあるとします。

```
cat
 NT domestic cat
 NT wild cat
 BT mammal
mammal
 BT animal
domestic cat
 NT Persian cat
 NT Siamese cat
```

cat に対する下位語を 2 レベル下まで検索するには、次の文を発行します。

```
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.nt('CAT', 2, 'MY_THES');
  dbms_output.put_line('the narrower expansion for CAT is: '||terms);
end;
```

このコードは、次の出力結果を戻します。

```
the narrower expansion for CAT is: {cat}|{domestic cat}|{Persian cat}|{Siamese cat}|
{wild cat}
```

表結果

次のコードは、表結果を使用して *canine* に対する下位語の検索を実行します。

```
declare
  xtab ctx_thes.exp_tab;
begin
  ctx_thes.nt(xtab, 'canine', 2, 'my_thesaurus');
  for i in 1..xtab.count loop
    dbms_output.put_line(lpad(' ', 2*xtab(i).xlevel) ||
    xtab(i).xrel || ' ' || xtab(i).xphrase);
  end loop;
end;
```

このコードは、次の出力結果を戻します。

```
PHRASE CANINE
 NT WOLF (Canis lupus)
   NT WHITE WOLF
   NT GREY WOLF
 NT DOG (Canis familiaris)
   NT PIT BULL
   NT DASCHUND
   NT CHIHUAHUA
NT HYENA (Canis mesomelas)
NT COYOTE (Canis latrans)
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「NARROWER TERM（NT、NTG、NTP、NTI）」

NTG

このファンクションは、指定されたシソーラスに記録されているとおりに、句のすべての下位汎用語を戻します。

構文 1: 表結果

```
CTX_THES.NTG(restab IN OUT NOCOPY EXP_TAB,
             phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.NTG(phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

lvl

戻す下位語のレベルを指定します。たとえば、2 を指定することによって、句の下位語の下位語までをすべて取得します。

tname

シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で下位汎用語の文字列を戻します。

```
{nt1}|{nt2}|{nt3} ...
```

例

cat に対する下位汎用語を 2 レベル下まで検索するには、次の文を発行します。

```
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.ntg('CAT', 2, 'MY_THES');
  dbms_output.put_line('the narrower expansion for CAT is: '||terms);
end;
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「NARROWER TERM（NT、NTG、NTP、NTI）」

NTI

このファンクションは、指定されたシソーラスに記録されているとおりに、句のすべての下位インスタンス語を戻します。

構文 1: 表結果

```
CTX_THES.NTI(restab IN OUT NOCOPY EXP_TAB,
             phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.NTI(phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

lvl

戻す下位語のレベルを指定します。たとえば、2 を指定することによって、句の下位語の下位語までをすべて取得します。

tname

シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で下位インスタンス語の文字列を戻します。

```
{nt1}|{nt2}|{nt3} ...
```

例

cat に対する下位インスタンス語を 2 レベル下まで検索するには、次の文を発行します。

```
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.nti('CAT', 2, 'MY_THES');
  dbms_output.put_line('the narrower expansion for CAT is: '||terms);
end;
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「NARROWER TERM（NT、NTG、NTP、NTI）」

NTP

このファンクションは、指定されたシソーラスに記録されているとおりに、句のすべての下位部分語を戻します。

構文 1: 表結果

```
CTX_THES.NTP(restab IN OUT NOCOPY EXP_TAB,
             phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.NTP(phrase IN VARCHAR2,
             lvl    IN NUMBER DEFAULT 1,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

lvl

戻す下位語のレベルを指定します。たとえば、2 を指定することによって、句の下位語の下位語までをすべて取得します。

tname

シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で下位部分語の文字列を戻します。

```
{nt1}|{nt2}|{nt3} ...
```

例

cat に対する下位部分語を 2 レベル下まで検索するには、次の文を発行します。

```
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.ntp('CAT', 2, 'MY_THES');
  dbms_output.put_line('the narrower expansion for CAT is: '||terms);
end;
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「NARROWER TERM（NT、NTG、NTP、NTI）」

OUTPUT_STYLE

CTX_THES 拡張ファンクションの戻り文字列に対する出力形式を設定します。このプロシージャは、CTX_THES 拡張ファンクションに対する表結果には影響を与えません。

構文

```
CTX_THES.OUTPUT_STYLE (
      showlevel     IN BOOLEAN DEFAULT FALSE,
      showqualify   IN BOOLEAN DEFAULT FALSE,
      showpt        IN BOOLEAN DEFAULT FALSE,
      showid        IN BOOLEAN DEFAULT FALSE
);
```

showlevel
BT/NT 拡張のレベルを示すには、TRUE を指定します。

showqualify
句修飾子を示すには、TRUE を指定します。

showpt
アスタリスク * を持つ優先語を示すには、TRUE を指定します。

showid
句 ID を示すには、TRUE を指定します。

注意

CTX_THES 拡張ファンクションに対する戻り文字列の一般的な構文は次のとおりです。

```
{pt indicator:phrase (qualifier):level:phraseid}
```

優先語のインジケータは、句の始めにアスタリスク、次にコロンです。修飾子は、句の終わりの空白の次のカッコの中です。レベルは数字です。

次は、bird の turkey に対する戻り文字列の例です。

```
*:TURKEY (BIRD):1:1234
```

PT

このファンクションは、指定されたシソーラスに記録されているとおりに、句の優先語を戻します。

構文 1: 表結果

```
CTX_THES.PT(restab IN OUT NOCOPY EXP_TAB,
            phrase IN VARCHAR2,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN varchar2;
```

構文 2: 文字列結果

```
CTX_THES.PT(phrase IN VARCHAR2,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN varchar2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

関連項目： EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

tname

シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、優先語を次の形式の文字列として戻します。

```
{pt}
```

例

automobile に対して次の優先語定義を持つシソーラス MY_THES があるとします。

```
AUTOMOBILE
  PT CAR
```

automobile に対する優先語を検索するには、次のコードを実行します。

```
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.pt('AUTOMOBILE','MY_THES');
  dbms_output.put_line('The prefered term for automobile is: '||terms);
end;
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「PREFERRED TERM（PT）」

RT

このファンクションは、指定されたシソーラスの語句の関連語を戻します。

構文 1: 表結果

```
CTX_THES.RT(restab IN OUT NOCOPY EXP_TAB,
            phrase IN VARCHAR2,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.RT(phrase IN VARCHAR2,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN varchar2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

関連項目： EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

tname

シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で関連語の文字列を戻します。

```
{rt1}|{rt2}|{rt3}| ...
```

例

dog に対して次の関連語定義を持つシソーラス MY_THES があるとします。

```
DOG
  RT WOLF
  RT HYENA
```

dog に対する関連語を検索するには、次のコードを実行します。

```
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.rt('DOG','MY_THES');
  dbms_output.put_line('The related terms for dog are: '||terms);
end;
```

このコードは、次の出力結果を戻します。

```
The related terms for dog are: {dog}|{wolf}|{hyena}
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「RELATED TERM（RT）」

SN

このファンクションは、指定された句のスコープ・ノートを戻します。

構文

```
CTX_THES.SN(phrase IN VARCHAR2,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

phrase
シソーラスで検索する句を指定します。

tname
シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、スコープ・ノートを文字列として戻します。

例

```
declare
  note varchar2(80);
begin
  note := ctx_thes.sn('camera','mythes');
  dbms_output.put_line('CAMERA');
  dbms_output.put_line(' SN ' || note);
end;

sample output:

CAMERA
 SN Optical cameras
```

SYN

このファンクションは、指定されたシソーラスに記録されているとおりに、すべての句のシノニムを戻します。

構文 1: 表結果

```
CTX_THES.SYN(restab IN OUT NOCOPY EXP_TAB,
             phrase IN VARCHAR2,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.SYN(phrase IN VARCHAR2,
             tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab
オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase
シソーラスで検索する句を指定します。

tname
シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式の文字列を戻します。

```
{syn1}|{syn2}|{syn3} ...
```

例

文字列結果

cat に対して次のエントリを持つ、ANIMALS というシソーラスがあるとします。

```
CAT
  SYN KITTY
  SYN FELINE
```

cat に対するシノニムを検索し、結果を文字列として取得するには、次の文を発行します。

```
declare
  synonyms varchar2(2000);
begin
  synonyms := ctx_thes.syn('CAT','ANIMALS');
  dbms_output.put_line('the synonym expansion for CAT is: '||synonyms);
end;
```

このコードは、次の出力結果を戻します。

```
the synonym expansion for CAT is: {CAT}|{KITTY}|{FELINE}
```

表結果

次のコードは、*canine* に対するシノニムを検索し、結果を表に取得します。表の内容は、標準出力で出力されます。

```
declare
  xtab ctx_thes.exp_tab;
begin
  ctx_thes.syn(xtab, 'canine', 'my_thesaurus');
  for i in 1..xtab.count loop
    dbms_output.put_line(lpad(' ', 2*xtab(i).xlevel) ||
    xtab(i).xrel || ' ' || xtab(i).xphrase);
  end loop;
end;
```

このコードは、次の出力結果を戻します。

```
PHRASE CANINE
 PT DOG
SYN PUPPY
SYN MUTT
SYN MONGREL
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「SYNONYM（SYN）」

THES_TT

このプロシージャは、シソーラスのすべての最上位語を検索して戻します。最上位語は、下位語を持つが上位語を持たないすべての語句として定義されます。

このプロシージャは、シソーラス全体を検索し、すべての最上位語を検索します。TT は句の中で、その句の最上位語を検索します。この点で、THES_TT と TT は異なります。

大規模なシソーラス

このプロシージャはシソーラス全体を検索するため、大規模なシソーラスを処理することもできます。このようなシソーラスに対して、このファンクションを頻繁にコールしないでください。かわりに、アプリケーションでこれを 1 回コールして結果を別の表に格納し、この格納された結果を使用してください。

構文

```
CTX_THES.THES_TT(restab IN OUT NOCOPY EXP_TAB,
                 tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

restab
結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

tname
シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このプロシージャはすべての最上位語を戻し、それを restab に格納します。

TR

このファンクションは、指定された単一言語のシソーラスに対し、シソーラスに記録されているとおりに句の外国語の等価語を戻します。

> **注意：** 外国語の翻訳は、ISO-2788 または ANSI Z39.19 シソーラス規格に準拠するものではありません。TR の動作は、Oracle Text 固有です。

構文 1: 表結果

```
CTX_THES.TR(restab IN OUT NOCOPY EXP_TAB,
            phrase IN VARCHAR2,
            lang   IN VARCHAR2 DEFAULT NULL,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT')
```

構文 2: 文字列結果

```
CTX_THES.TR(phrase IN VARCHAR2,
            lang   IN VARCHAR2 DEFAULT NULL,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase

シソーラスで検索する句を指定します。

lang

外国語を指定します。phrase のすべての翻訳を示すには、ALL を指定します。

tname
シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で外国語の語句の文字列を戻します。

```
{ft1}|{ft2}|{ft3} ...
```

例

cat に対して次のエントリを持つシソーラス MY_THES があるとします。

```
cat
  SPANISH: gato
  FRENCH:  chat
  SYN lion
    SPANISH: leon
```

cat に対する翻訳を検索するには、次の文を発行します。

```
declare
  trans      varchar2(2000);
  span_trans varchar2(2000);
begin
  trans := ctx_thes.tr('CAT','ALL','MY_THES');
  span_trans := ctx_thes.tr('CAT','SPANISH','MY_THES')
  dbms_output.put_line('the translations for CAT are: '||trans);
  dbms_output.put_line('the Spanish translations for CAT are: '||span_trans);
end;
```

このコードは、次の出力結果を戻します。

```
the translations for CAT are: {CAT}|{CHAT}|{GATO}
the Spanish translations for CAT are: {CAT}|{GATO}
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「TRANSLATION TERM（TR）」

TRSYN

このファンクションは、指定された単一言語のシソーラスに対し、シソーラスに記録されているとおりに句の外国語の等価語、句のシノニムおよびシノニムの外国語の等価語を戻します。

注意： 外国語の翻訳は、ISO-2788 または ANSI Z39.19 シソーラス規格に準拠するものではありません。TRSYN の動作は、Oracle Text 固有です。

構文 1: 表結果

```
CTX_THES.TRSYN(restab IN OUT NOCOPY EXP_TAB,
               phrase IN VARCHAR2,
               lang   IN VARCHAR2 DEFAULT NULL,
               tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.TRSYN(phrase IN VARCHAR2,
           lang   IN VARCHAR2 DEFAULT NULL,
           tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN VARCHAR2;
```

restab

オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

関連項目： EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase
シソーラスで検索する句を指定します。

lang
外国語を指定します。*phrase* のすべての翻訳を示すには、ALL を指定します。

tname
シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で外国語の語句の文字列を戻します。

```
{ft1}|{ft2}|{ft3} ...
```

例

cat に対して次のエントリを持つシソーラス MY_THES があるとします。

```
cat
  SPANISH: gato
  FRENCH:  chat
  SYN lion
    SPANISH: leon
```

cat に対する翻訳およびシノニムを検索するには、次の文を発行します。

```
declare
  synonyms   varchar2(2000);
  span_syn   varchar2(2000);
begin
  synonyms := ctx_thes.trsyn('CAT','ALL','MY_THES');
  span_syn := ctx_thes.trsyn('CAT','SPANISH','MY_THES')
  dbms_output.put_line('all synonyms for CAT are: '||synonyms);
  dbms_output.put_line('the Spanish synonyms for CAT are: '||span_syn);
end;
```

このコードは、次の出力結果を戻します。

```
all synonyms for CAT are: {CAT}|{CHAT}|{GATO}|{LION}|{LEON}
the Spanish synonyms for CAT are: {CAT}|{GATO}|{LION}|{LEON}
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「TRANSLATION TERM SYNONYM（TRSYN）」

TT

このファンクションは、指定されたシソーラスに記録されているとおりに、句の最上位語を戻します。

構文 1: 表結果

```
CTX_THES.TT(restab IN OUT NOCOPY EXP_TAB,
            phrase IN VARCHAR2,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT');
```

構文 2: 文字列結果

```
CTX_THES.TT(phrase IN VARCHAR2,
            tname  IN VARCHAR2 DEFAULT 'DEFAULT')
RETURN varchar2;
```

restab
オプションで、結果を格納する拡張表の名前を指定します。この表は、次のようにシステムが定義する EXP_TAB 型である必要があります。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);
type exp_tab is table of exp_rec index by binary_integer;
```

> **関連項目：** EXP_TAB の詳細は、付録 A「結果表」の「CTX_THES 結果表およびデータ型」を参照してください。

phrase
シソーラスで検索する句を指定します。

tname
シソーラス名を指定します。指定しない場合は、システムのデフォルト・シソーラスが使用されます。

戻り値

このファンクションは、次の形式で最上位語の文字列を戻します。

```
{tt}
```

例

dog に対して次の上位語エントリを持つシソーラス MY_THES があるとします。

```
DOG
   BT1 CANINE
     BT2 MAMMAL
       BT3 VERTEBRATE
         BT4 ANIMAL
```

DOG に対する最上位語を検索するには、次のコードを実行します。

```
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.tt('DOG','MY_THES');
  dbms_output.put_line('The top term for DOG is: '||terms);
end;
```

このコードは、次の出力結果を戻します。

```
The top term for dog is: {ANIMAL}
```

関連項目

「OUTPUT_STYLE」

第 3 章「CONTAINS 問合せ演算子」の「TOP TERM（TT）」

UPDATE_TRANSLATION

このプロシージャを使用して、既存の翻訳を更新します。

構文

```
CTX_THES.UPDATE_TRANSLATION(tname       in     varchar2,
                            phrase      in     varchar2,
                            language    in     varchar2,
                            translation in     varchar2,
                            new_translation in varchar2);
```

tname
シソーラスの名前を 30 文字以内で指定します。

phrase
翻訳を更新するシソーラス内の句を指定します。その句がシソーラスにすでに存在している必要があります。存在しない場合はエラーが発生します。

language
翻訳の言語を 10 文字以内で指定します。

translation
更新する翻訳済みの語句を指定します。指定した翻訳が存在しない場合は、エラーが発生します。

phase に対する翻訳が 1 つのみの場合は、NULL を指定できます。指定した言語の語句に対して複数の翻訳がある場合は、エラーが発生します。

new_translation
オプションで、翻訳済み語句の新しい形式を指定します。

例

次のコードは、*dog* に対するスペイン語の翻訳を更新します。

```
begin
   ctx_thes.update_translation('my_thes', 'dog', 'SPANISH:', 'PERRO', 'CAN');
end;
```

13

CTX_ULEXER パッケージ

この章では、ユーザー・レクサーで使用するための PL/SQL パッケージ CTX_ULEXER の使用方法について説明します。

CTX_ULEXER は、次の型を宣言します。

名前	説明
WILDCARD_TAB	索引付き表型を使用して、ユーザー定義レクサーの問合せプロシージャでワイルド・カード文字として処理する文字のオフセットを指定します。

WILDCARD_TAB

```
TYPE WILDCARD_TAB IS TABLE OF NUMBER INDEX BY BINARY_INTEGER;
```

この索引付き表型を使用して、ユーザー定義レクサーの問合せプロシージャでワイルド・カード文字として処理する問合せワードの文字オフセットを指定します。

14

実行可能ファイル

この章では、Oracle Text で提供される実行可能ファイルについて説明します。この章の内容は次のとおりです。

- シソーラス・ローダー（ctxload）
- ナレッジ・ベース拡張コンパイラ（ctxkbtc）

シソーラス・ローダー（ctxload）

ctxload を使用して、シソーラスに関する次の操作を実行します。

- シソーラスのファイルを Oracle Text のシソーラス表にインポートします。
- ロード済みのシソーラスを、ユーザー指定のオペレーティング・システム・ファイルにエクスポートします。

インポート・ファイルは、ASCII フラット・ファイルで、問合せを拡張するときに使用するシノニム、上位語、下位語または関連語のエントリを含みます。

関連項目： シソーラスのインポート用のインポート・ファイルの例は、付録 C「ロード例」の「ctxload シソーラス・インポート・ファイルの構造」を参照してください。

テキストのロード

現在、ctxload プログラムはテキスト列のロードをサポートしていません。ファイルをバッチ内のテキスト列にロードする場合は、SQL*Loader の使用をお薦めします。

関連項目： 付録 C「ロード例」の「SQL*Loader 例」

ctxload 構文

```
ctxload -user username[/password][@sqlnet_address]
        -name object_name
        -file file_name

       [-thes]
       [-thescase y|n]
       [-thesdump]
       [-log file_name]
       [-trace]
       [-pk]
       [-export]
       [-update]
```

必須の引数

-user
ctxload を実行するユーザー名およびパスワードを指定します。

ユーザー名およびパスワードの直後に *@sqlnet_address* を指定すると、リモート・データベースにログインできるようになります。*sqlnet_address* の値はデータベースの接続文字列です。環境変数 TWO_TASK がリモート・データベースに設定されている場合は、データベースへの接続で *sqlnet_address* の値を指定する必要はありません。

-name object_name
ctxload を使用してシソーラスをエクスポート / インポートする場合、エクスポート / インポートするシソーラスの名前を指定するには object_name を使用します。

シソーラス演算子を使用する問合せでシソーラスを識別するには、object_name を使用します。

注意： シソーラス名は、一意である必要があります。シソーラスに指定した名前が既存のシソーラスと同一である場合、ctxload はエラーを戻し、既存のシソーラスが上書きされることはありません。

---

ctxload を使用してテキスト・フィールドを更新 / エクスポートする場合、テキスト列に対応付けられている索引を指定するには、object_name を使用します。

-file file_name
ctxload を使用してシソーラスをインポートする場合、シソーラスのエントリがあるインポート・ファイルの名前を指定するには、file_name を使用します。

ctxload を使用してシソーラスをエクスポートする場合、ctxload で作成されたエクスポート・ファイルの名前を指定するには、file_name を使用します。

注意： シソーラス・ダンプ・ファイルに指定した名前が既存のファイル名と同一である場合、ctxload により既存のファイルが上書きされます。

オプションの引数

-thes
シソーラスをインポートします。-file 引数でソース・ファイルを指定します。-name でインポートするシソーラスの名前を指定します。

-thescase y | n
y を指定すると、-name で指定した名前で大 / 小文字が区別されるシソーラスを作成し、そのシソーラスに -file で指定したシソーラス・インポート・ファイルのエントリを移入します。-thescase を *y*（大 / 小文字が区別されるシソーラス）にすると、ctxload はインポート・ファイルに登録されているとおりに語句をシソーラスに入力します。

-thescase のデフォルト値は *n*（大 / 小文字が区別されないシソーラス）です。

注意： -thescase は、-thes 引数と併用した場合にのみ有効です。

-thesdump
シソーラスをエクスポートします。-name 引数で、エクスポートするシソーラスの名前を指定します。-file 引数で宛先ファイルを指定します。

-log
ファイルのロード時に ctxload によって生成される、各国語対応（グローバリゼーション・サポート）メッセージの書込み先ログ・ファイル名を指定します。ログ・ファイルの名前を指定しない場合、メッセージは標準出力に書き出されます。

-trace
ALTER SESSION SET SQL_TRACE TRUE を使用した SQL 文のトレースを使用可能にします。このコマンドによって、トレース・ファイルにあるすべての処理済み SQL 文を獲得し、デバッグに使用できます。トレース・ファイルの位置は、オペレーティング・システム依存であり、USER_DUMP_DEST 初期化パラメータを使用して変更できます。

関連項目： SQL トレースおよび USER_DUMP_DEST 初期化パラメータの詳細は、『Oracle9*i* データベース管理者ガイド』を参照してください。

-pk
更新またはエクスポートされる行の主キー値を指定します。

主キーがコンパウンドの場合、二重引用符で値を囲み、キーをカンマで区切る必要があります。

-export
データベース表の CLOB 列または BLOB 列の内容を、-file で指定したオペレーティング・システム・ファイルにエクスポートします。ctxload は、-pk で指定した行の CLOB 列または BLOB 列をエクスポートします。

-export を使用する場合、-pk で主キーを指定する必要があります。

-update
データベース表の CLOB 列または BLOB 列の内容を、-file で指定したオペレーティング・システム・ファイルの内容で更新します。ctxload は、-pk で指定した行の CLOB 列または BLOB 列を更新します。

-update を使用する場合、-pk で主キーを指定する必要があります。

ctxload の例

この項では、ctxload が実行可能な操作の例を示します。

> **関連項目：** ドキュメントのロード例の詳細は、付録 C「ロード例」を参照してください。

シソーラスのインポート例

次の例では、tech_doc というシソーラスを tech_thesaurus.txt というインポート・ファイルからインポートします。

```
ctxload -user jsmith/123abc -thes -name tech_doc -file tech_thesaurus.txt
```

シソーラスのエクスポート例

次の例では、tech_doc というシソーラスの内容のダンプを tech_thesaurus.out というファイルに出力します。

```
ctxload -user jsmith/123abc -thesdump -name tech_doc -file tech_thesaurus.out
```

ナレッジ・ベース拡張コンパイラ（ctxkbtc）

ナレッジ・ベースとは、テーマの索引付け、ABOUT 問合せの処理、CTX_DOC パッケージでのドキュメント・テーマの抽出などのテーマ分析を実行するために、Oracle Text で使用する情報ソースのことです。英語とフランス語のナレッジ・ベースが提供されています。

ctxkbtc コンパイラを使用して、次の操作を実行できます。

- Oracle Text のナレッジ・ベースで 1 つ以上のシソーラスをコンパイルし、使用しているナレッジ・ベースを拡張します。アプリケーション固有の語句および関連情報も拡張されます。テーマ分析時に、ナレッジ・ベースの拡張部分は、ナレッジ・ベースでオーバーラップする語句および関連情報を上書きします。
- 1 つ以上のシソーラスをコンパイルして新しいユーザー定義のナレッジ・ベースを作成します。英語とフランス語以外の言語では、この機能は、言語固有のナレッジ・ベースの作成に使用できます。

関連項目： Oracle Text のナレッジ・ベースの詳細は、付録 I「英語のナレッジ・ベースのカテゴリ階層」を参照してください。

ABOUT 演算子の詳細は、第 3 章「CONTAINS 問合せ演算子」の「ABOUT」演算子を参照してください。

ドキュメント・サービスの詳細は、第 8 章「CTX_DOC パッケージ」を参照してください。

ナレッジ・ベースのキャラクタ・セット

ナレッジ・ベースは、任意のシングルバイト・キャラクタ・セットで格納できます。提供されるナレッジ・ベースは、WE8ISO8859P1 です。拡張ナレッジ・ベースは、US7ASCII などの別のキャラクタ・セットで格納できます。

ctxkbtc 構文

```
ctxkbtc -user uname/passwd
    [-name thesname1 [thesname2 ... thesname16]]
    [-revert]
    [-stoplist stoplistname]
    [-verbose]
    [-log filename]
```

-user
拡張ナレッジ・ベースを作成する管理者用のユーザー名およびパスワードを指定します。このユーザーには、ORACLE_HOME ディレクトリへの書込み権限が必要です。

-name *thesname1 [thesname2 ... thesname16]*
拡張ナレッジ・ベースを作成するためにナレッジ・ベースでコンパイルするシソーラス（最大 16）の名前を指定します。指定するシソーラスは、ctxload で -thescase Y オプションを指定してロードされている必要があります。

-revert
拡張ナレッジ・ベースを、Oracle Text のデフォルトのナレッジ・ベースに回復します。

-stoplist *stoplistname*
ストップリストの名前を指定します。ストップリスト内のストップワードは、テーマまたはテーマの一部になることができない不使用ワードとしてナレッジ・ベースに追加されます。このコマンドの実行後、CTX_DLL.ADD_STOPTHEME を使用してストップテーマを追加することもできます。

-verbose
グローバリゼーション・サポート以外のメッセージを含む、すべての警告およびメッセージを標準出力に表示します。

-log
すべてのメッセージを格納するログ・ファイルを指定します。ログ・ファイルを指定すると、メッセージは標準出力にレポートされません。

ctxkbtc の使用上の注意

- ctxkbtc の実行前に、環境変数 NLS_LANG をデータベース・キャラクタ・セットに一致するように設定しておく必要があります。
- ctxkbtc を発行するユーザーには、ORACLE_HOME ディレクトリへの書込み権限が必要です。プログラムにより、このディレクトリにファイルが書き込まれるためです。
- コンパイル前に、各シソーラスが、ctxload の -thescase Y オプションによって大 / 小文字が区別されて Oracle Text にロードされている必要があります。
- ctxkbtc を 2 回実行すると、前の拡張は削除されます。

ctxkbtc の制限事項

ctxkbtc プログラムには、次の制限事項があります。

- データベースを別のリリースにアップグレードまたはダウングレードする場合は、テーマの索引付けや関連機能が正常に機能するように、拡張ナレッジ・ベースを新しい環境で再コンパイルすることをお薦めします。
- テーマ索引付けの実行中は、ナレッジ・ベースの拡張は実行できません。また、拡張ナレッジ・ベースを使用するには、Oracle Text の機能を使用しているすべての SQL セッションを終了して再オープンする必要があります。
- インストールごとに 1 ユーザーのみが拡張できます。ユーザー拡張は、そのインストールのすべてのユーザーに影響を与えるため、管理者または技術マネージャのみがナレッジ・ベースを拡張してください。

シソーラス語句に対する ctxkbtc の制約

語句は、大 / 小文字が区別されます。たとえば、シソーラスに大文字の語句がある場合、ドキュメント内で小文字で表記されている同一語句は認識されません。

1 つの語句の最大長は、80 文字です。

明確な同形異義語は、サポートされません。

シソーラス・リレーションに対する ctxkbtc の制約

次の制約が、シソーラス・リレーションに適用されます。

- BTG および BTP は BT と同じです。NTG および NTP は NT と同じです。
- 優先語のみが、BT、NT または RT を持つことができます。
- 語句が USE リレーションを持たない場合は、独自の優先語として扱われます。
- 語句のセットが SYN リレーションによって関連付けられている場合は、そのうちの 1 つのみが優先語になることがあります。
- 既存のカテゴリを、最上位語にすることはできません。
- BT および NT リレーションには、サイクルはありません。
- 1 つの語句は、最大 1 つの優先語および最大 1 つの BT を持つことができます。語句は、NT をいくつでも持つことができます。
- 語句の RT は、その語句の上位クラスまたは下位クラスになることができません。優先語は、32 以下であればいくつでも RT を持つことができます。
- ツリーの高さは、最上位語レベルを含めて最高 16 です。
- 複数のシソーラスがコンパイルされる場合、1 つのシソーラスの最上位語は、別のシソーラスの上位語を持つことはできません。

注意： シソーラスのコンパイラは、前述のルールの特定の違反を許容します。たとえば、語句が複数の BT を持つ場合、最後の BT のみ検出し、他はすべて無視します。

同様に、既存のナレッジ・ベースのカテゴリ間に BT がある場合も、警告メッセージが表示されるのみです。

このような違反は、好ましくない結果を引き起こす場合があるため、お薦めできません。

ナレッジ・ベースの拡張

Oracle Text のナレッジ・ベースで 1 つ以上のシソーラスをコンパイルし、提供されたナレッジ・ベースを拡張できます。アプリケーション固有の語句および関連情報も拡張されます。テーマ分析時に、ナレッジ・ベースの拡張部分は、ナレッジ・ベースでオーバーラップする語句および関連情報を上書きします。

ナレッジ・ベースの拡張時には、テーマを検証する場合に最適な結果が得られるように、新規語句をナレッジ・ベースのカテゴリの 1 つにリンクすることをお薦めします。

> **関連項目：** ナレッジ・ベースの詳細は、付録 I「英語のナレッジ・ベースのカテゴリ階層」を参照してください。

新規語句を既存のカテゴリから完全に切り離している場合は、新規語句から検証されるテーマが少なくなります。この結果、精度が低下するため ABOUT 問合せでコールしなおす必要があります。また、要点およびテーマ・ハイライトの品質も低下します。

既存語句を新規語句の上位語にすることによって、新規語句を既存語句にリンクします。

ナレッジ・ベースの拡張例

医学用語の階層を含む医学シソーラス medthes を購入します。このシソーラスの 4 つの最上位語は、次のとおりです。

- Anesthesia and Analgesia（麻酔および無痛）
- Anti-Allergic and Respiratory System Agents（抗アレルギー薬および呼吸器系薬）
- Anti-Inflammatory Agents, Antirheumatic Agents, and Inflammation Mediators（抗炎症薬、抗リウマチ薬および炎症伝達物質）
- Antineoplastic and Immunosuppressive Agents（抗腫瘍薬および免疫抑制薬）

これらの語句を既存のナレッジ・ベースにリンクするには、医学シソーラスに次のエントリを追加して、新規語句を既存の *health and medicine* ブランチにマップします。

```
health and medicine
 NT Anesthesia and Analgesia
 NT Anti-Allergic and Respiratory System Agents
 NT Anti-Inflamammatory Agents, Antirheumatic Agents, and Inflamation Mediators
 NT Antineoplastic and Immunosuppressive Agents
```

グローバリゼーション・サポート言語の環境変数をデータベース・キャラクタ・セットに一致するように設定します。たとえば、データベース・キャラクタ・セットが WE8ISO8859P1 で、アメリカ英語を使用している場合は、次のように NLS_LANG を設定します。

```
setenv NLS_LANG AMERICAN_AMERICA.WE8ISO8859P1
```

医学シソーラスが med.thes というファイルにあるとすると、次のように、ctxload でシソーラスを medthes としてロードします。

```
ctxload -thes -thescase y -name medthes -file med.thes -user ctxsys/ctxsys
```

ロードされたシソーラス medthes をナレッジ・ベースにリンクするには、次のように ctxkbtc を使用します。

```
ctxkbtc -user ctxsys/ctxsys -name medthes
```

言語固有のナレッジ・ベースの追加

シングルバイトの空白で区切られた言語の独自のナレッジ・ベースをロードすると、スペイン語など、英語やフランス語以外の言語に対してテーマ機能を拡張できます。

テーマ機能には、テーマの索引付け、ABOUT 問合せ、テーマ・ハイライト、および PL/SQL パッケージ CTX_DOC を使用したテーマ、要点およびテーマ・サマリーの生成などの機能が含まれます。

テーマ機能は、ユーザー定義ナレッジ・ベースを追加して拡張します。たとえば、スペイン語シソーラスからスペイン語のナレッジ・ベースを作成できます。

言語固有のナレッジ・ベースをロードする手順は、次のとおりです。

1. ユーザー定義シソーラスを ctxload を使用してロードします。
2. language 部分がターゲット言語になるように、NLS_LANG を設定します。charset 部分は、シングルバイト・キャラクタ・セットに設定する必要があります。
3. ロード済みのシソーラスを ctxkbtc を使用して次のようにコンパイルします。

```
ctxkbtc -user ctxsys/ctxsys -name my_lang_thes
```

このコマンドは、言語固有のナレッジ・ベースをロード済みのシソーラスからコンパイルします。索引付けおよび ABOUT 問合せ時のテーマ分析にこのナレッジ・ベースを使用するには、NLS_LANG 言語を BASIC_LEXER プリファレンスの THEME_LANGUAGE 属性値として指定します。

ナレッジ・ベースの追加に関する制限事項

ナレッジ・ベースの追加には、次の制限事項が適用されます。

- Oracle が提供するナレッジ・ベースは、英語とフランス語のみです。それ以外の言語に対しては、独自のシソーラスを用意する必要があります。
- ナレッジ・ベースは、シングルバイト・キャラクタ・セットを持つ言語に対してのみ追加できます。マルチバイト・キャラクタ・セットでのみ表現される言語に対しては、ナレッジ・ベースを作成できません。データベースが、UTF-8 などのマルチバイト・ユニバーサル・キャラクタ・セットの場合は、そのシソーラスのコンパイル時に、NLS_LANG パラメータを互換性のあるシングルバイト・キャラクタ・セットに設定する必要があります。
- ナレッジ・ベースの追加によって、空白で区切られた言語が最適に機能します。
- グローバリゼーション・サポート言語ごとに 1 つのナレッジ・ベースを持つことができます。
- 上位語、下位語および関連語などの階層問合せフィードバック情報の取得操作は、英語とフランス語以外の言語では機能しません。その他の言語では、ナレッジ・ベースがユーザーのシソーラスからのみ導出されるためです。したがって、使用しているシソーラスから直接階層情報を取得することをお薦めします。

複数のシソーラスの優先順位

複数のシソーラスがコンパイルされる場合、コンパイラへの引数に指定されているシソーラスの順序（最も優先度が高いものが最初）によって、優先順位が決定されます。ユーザーのシソーラスは、常に最初から組み込まれているナレッジ・ベースより優先されます。

拡張ナレッジ・ベースのサイズ制限

次の表は、拡張ナレッジ・ベースの作成およびコンパイルに対応付けられたサイズ制限を示しています。

パラメータの説明	制限
語句ごとの RT 数（from + to）	32
1 階層ごとの語句数（指定した最上位語のすべての下位語）	64000
拡張ナレッジ・ベースの新規語句数	100 万
ナレッジ・ベースへのユーザー拡張にコンパイル可能な個別のシソーラス数	16

A

結果表

この付録では、CTX_QUERY、CTX_DOC および CTX_THES パッケージのプロシージャによって生成された出力を格納するために使用する結果表の構造について説明します。

この付録の内容は次のとおりです。

- CTX_QUERY 結果表
- CTX_DOC 結果表
- CTX_THES 結果表およびデータ型

CTX_QUERY 結果表

結果を戻す CTX_QUERY プロシージャに対して、プロシージャがコールされる前に結果を格納するための表を作成しておく必要があります。表には自由に名前を付けることができますが、特定の名前およびデータ型を持つ列が組み込まれている必要があります。

この項では、次のタイプの結果表とその結果表に必要な列について説明します。

- 実行計画表
- HFEEDBACK 表

実行計画表

表 A-1 は、CTX_QUERY.EXPLAIN によって結果が書き込まれる表の構造を示しています。

表 A-1

列名	データ型	説明
EXPLAIN_ID	VARCHAR2(30)	FEEDBACK コールで指定した explain_id 引数の値。
ID	NUMBER	問合せ実行ツリーの各ノードに割り当てられた番号。ルート・オペレーション・ノードでは、ID=1 です。ノードは、解析ツリーでの表示のとおり、上から下、左から右の順に順序番号が付けられます。
PARENT_ID	NUMBER	ID ステップの出力で操作する実行ステップの ID。グラフィック的には、これは、問合せ実行ツリー内の親ノードです。ルート・オペレーション・ノード（ID=1）では、PARENT_ID=0 です。
OPERATION	VARCHAR2(30)	実行する内部操作の名前。有効な値は、表 A-2 を参照してください。
OPTIONS	VARCHAR2(30)	OPERATION 列で示した操作のバリエーションを表す文字。OPERATION に対応付けられた複数の OPTIONS がある場合、OPTIONS の値は処理の順に連結されます。有効な値は、表 A-3 を参照してください。
OBJECT_NAME	VARCHAR2(80)	セクション名、ワイルド・カード語句、重み、しきい値、または索引で検索する語句。
POSITION	NUMBER	すべて同じ PARENT_ID を持つノードの処理の順序。1 から開始し、昇順に順序番号が付けられます。
CARDINALITY	NUMBER	予備。将来の互換性のためにこの列を作成してください。

演算列値

表 A-2 は、実行計画表の OPERATION 列の有効な値を示しています。

表 A-2

演算値	問合せ演算子	等価記号
ABOUT	ABOUT	(なし)
ACCUMULATE	ACCUM	,
AND	AND	&
COMPOSITE	(なし)	(なし)
EQUIVALENCE	EQUIV	=
MINUS	MINUS	-
NEAR	NEAR	;
NOT	NOT	~
NO_HITS	(この問合せからのヒットはありません)	
OR	OR	|
PHRASE	(検索する句)	
SECTION	(セクション)	
THRESHOLD	>	>
WEIGHT	*	*
WITHIN	WITHIN	(なし)
WORD	(単一語)	

OPTIONS 列値

次の表は、実行計画表の OPTIONS 列の有効な値を示しています。

表 A-3

OPTIONS の値	説明
($)	STEM
(?)	FUZZY
(!)	SOUNDEX
(T)	順序付けされた NEAR の順序
(F)	順序付けされていない NEAR の順序
(n)	NEAR 演算子の max_span パラメータに対応付けられた数値

HFEEDBACK 表

表 A-4 は、CTX_QUERY.HFEEDBACK によって結果が書き込まれる表の構造を示しています。

表 A-4

列名	データ型	説明
FEEDBACK_ID	VARCHAR2(30)	HFEEDBACK コールで指定した *feedback_id* 引数の値。
ID	NUMBER	問合せ実行ツリーの各ノードに割り当てられた番号。ルート・オペレーション・ノードでは、ID=1 です。ノードは、解析ツリーでの表示のとおり、上から下、左から右の順に順序番号が付けられます。
PARENT_ID	NUMBER	ID ステップの出力で操作する実行ステップの ID。グラフィック的には、これは、問合せ実行ツリー内の親ノードです。ルート・オペレーション・ノード（ID=1）では、PARENT_ID=0 です。
OPERATION	VARCHAR2(30)	実行する内部操作の名前。有効な値は、表 A-5 を参照してください。

表 A-4（続き）

列名	データ型	説明
OPTIONS	VARCHAR2(30)	OPERATION 列で示した操作のバリエーションを表す文字。OPERATION に対応付けられた複数の OPTIONS がある場合、OPTIONS の値は処理の順に連結されます。有効な値は、表 A-6 を参照してください。
OBJECT_NAME	VARCHAR2(80)	セクション名、ワイルド・カード語句、重み、しきい値、または索引で検索する語句。
POSITION	NUMBER	すべて同じ PARENT_ID を持つノードの処理の順序。1 から開始し、昇順に順序番号が付けられます。
BT_FEEDBACK	CTX_FEEDBACK_TYPE	上位フィードバック語を格納します。表 A-7 を参照してください。
PT_FEEDBACK	CTX_FEEDBACK_TYPE	関連フィードバック語を格納します。表 A-7 を参照してください。
NT_FEEDBACK	CTX_FEEDBACK_TYPE	下位フィードバック語を格納します。表 A-7 を参照してください。

演算列値

表 A-5 は、HFEEDBACK 表の OPERATION 列の有効な値を示しています。

表 A-5

演算値	問合せ演算子	等価記号
ABOUT	ABOUT	（なし）
ACCUMULATE	ACCUM	,
AND	AND	&
EQUIVALENCE	EQUIV	=
MINUS	MINUS	-
NEAR	NEAR	;
NOT	NOT	~
OR	OR	|
SECTION	（セクション）	

表 A-5（続き）

演算値	問合せ演算子	等価記号
TEXT	テキスト問合せのワードまたは句	
THEME	ABOUT 問合せのワードまたは句	
THRESHOLD	>	>
WEIGHT	*	*
WITHIN	WITHIN	（なし）

OPTIONS 列値

次の表は、FEEDBACK 表の OPTIONS 列の値を示しています。

表 A-6

OPTIONS の値	説明
(T)	順序付けされた NEAR の順序
(F)	順序付けされていない NEAR の順序
(n)	NEAR 演算子の max_span パラメータに対応付けられた数値

CTX_FEEDBACK_TYPE

CTX_FEEDBACK_TYPE はオブジェクトの NESTED TABLE です。このデータ型は ctxsys スキーマで事前定義済みです。この型を使用して、BT_FEEDBACK、RT_FEEDBACK および NT_FEEDBACK 列を定義します。

この NESTED TABLE、CTX_FEEDBACK_TYPE は、ctxsys スキーマで事前定義済みの CTX_FEEDBACK_ITEM_TYPE 型のオブジェクトを格納します。このオブジェクトは次のように、3 つのメンバーと 1 つのメソッドで定義されています。

表 A-7

CTX_FEEDBACK_ITEM_TYPE メンバーおよびメソッド	データ型	説明
text	VARCHAR2(80)	フィードバック語
cardinality	NUMBER	（将来の拡張用）
score	NUMBER	（将来の拡張用）

これらのオブジェクトを定義する SQL コードは、次のとおりです。

```
CREATE OR REPLACE TYPE ctx_feedback_type AS TABLE OF ctx_feedback_item_type;

CREATE OR REPLACE TYPE ctx_feedback_item_type AS OBJECT
(text        VARCHAR2(80),
 cardinality NUMBER,
 score       NUMBER,
 MAP MEMBER FUNCTION rank RETURN REAL,
 PRAGMA RESTRICT_REFERENCES (rank, RNDS, WNDS, RNPS, WNPS)
);

CREATE OR REPLACE TYPE BODY ctx_feedback_item_type AS
   MAP MEMBER FUNCTION rank RETURN REAL IS
   BEGIN
      RETURN score;
   END rank;
END;
```

関連項目： HFEEDBACK 表およびその NESTED TABLE からの選択方法の例は、第 10 章「CTX_QUERY パッケージ」の「CTX_QUERY.HFEEDBACK」を参照してください。

CTX_DOC 結果表

CTX_DOC プロシージャは、表に格納された結果を戻します。プロシージャをコールする前に、表を作成しておく必要があります。表には自由に名前を付けることができますが、特定の名前およびデータ型を持つ列が組み込まれている必要があります。

この項では、次のタイプの結果表とその結果表に必要な列について説明します。

- フィルタ表
- 要点表
- ハイライト表
- マークアップ表
- テーマ表

フィルタ表

フィルタ表には、CTX_DOC.FILTER が戻したフィルタ処理されたドキュメントごとに 1 行が格納されます。フィルタ処理されたドキュメントは、プレーン・テキストまたは HTML です。

ドキュメントに CTX_DOC.FILTER をコールした場合、そのドキュメントはテキスト列に対して定義されたフィルタで処理され、その結果がユーザーが指定したフィルタ表に格納されます。

フィルタ表には自由に名前を付けることができますが、次の名前およびデータ型を持つ列が組み込まれている必要があります。

表 A-8

列名	データ型	説明
QUERY_ID	NUMBER	CTX_DOC.FILTER への特定のコールで生成された結果の識別子（複数の FILTER コールの結果を表に格納するときのみ移入されます）。
DOCUMENT	CLOB	プレーン・テキストまたは HTML で格納された、ドキュメントのテキスト。

要点表

要点表には、CTX_DOC.GIST が生成する要点 / テーマ・サマリーごとに 1 行が格納されます。

要点表には自由に名前を付けることができますが、次の名前およびデータ型を持つ列が組み込まれている必要があります。

表 A-9

列名	データ型	説明
QUERY_ID	NUMBER	問合せ ID。
POV	VARCHAR2(80)	ドキュメント・テーマ。ドキュメントでのテーマの使用状況またはナレッジ・ベース内での表現方法によって異なります。 POV は、ドキュメントの要点に対しては、GENERIC の値を持ちます。
GIST	CLOB	プレーン・テキストとして格納された、要点またはテーマ・サマリーのテキスト。

ハイライト表

ハイライト表には、ドキュメント内のハイライト表示された語句のオフセットおよび長さの情報が格納されます。この情報は、CTX_DOC.HIGHLIGHT によって生成されます。ハイライト表示された語句は、ワード問合せまたは ABOUT 問合せを満たすワードまたは句です。

ドキュメントが形式設定されている場合、テキストはプレーン・テキストまたは HTML のどちらかにフィルタ処理され、フィルタ処理されたテキストのオフセット情報が生成されます。オフセット情報を使用すると、CTX_DOC.FILTER でフィルタ処理された同じドキュメントの問合せ語句をハイライト表示することができます。

ハイライト表には自由に名前を付けることができますが、次の名前およびデータ型を持つ列が組み込まれている必要があります。

表 A-10

列名	データ型	説明
QUERY_ID	NUMBER	CTX_DOC.HIGHLIGHT への特定のコールで生成された結果の識別子（複数の HIGHLIGHT コールの結果を表に格納するときのみ移入されます）。
OFFSET	NUMBER	ドキュメント内のハイライト表示の位置（ドキュメントの開始位置 1 との相対的な位置）。
LENGTH	NUMBER	ハイライト表示の長さ。

マークアップ表

マークアップ表には、ドキュメント内にマークアップ・タグによってハイライト表示された問合せ語句があるプレーン・テキストまたは HTML 形式のドキュメントが格納されます。この情報は、ユーザーが CTX_DOC.MARKUP をコールしたときに生成されます。

マークアップ表には自由に名前を付けることができますが、次の名前およびデータ型を持つ列が組み込まれている必要があります。

表 A-11

列名	データ型	説明
QUERY_ID	NUMBER	CTX_DOC.MARKUP への特定のコールで生成された結果の識別子（複数の MARKUP コールの結果を表に格納するときのみ移入されます）。
DOCUMENT	CLOB	プレーン・テキストまたは HTML 形式で格納された、ドキュメントのマークアップされたテキスト。

テーマ表

テーマ表には、CTX_DOC.THEMES が生成したテーマごとに 1 行が格納されます。THEME 列に格納される値は、単一のテーマ句か、コロンで区切られた親テーマの文字列のいずれかです。

テーマ表には自由に名前を付けることができますが、次の名前およびデータ型を持つ列が組み込まれている必要があります。

表 A-12

列名	データ型	説明
QUERY_ID	NUMBER	問合せ ID。
THEME	VARCHAR2(2000)	テーマ句か、コロン（:）で区切られた親テーマの文字列。
WEIGHT	NUMBER	ドキュメントのその他のテーマ句と比較したテーマ句の重み。

トークン表

トークン表には、ドキュメントのテキスト・トークンが、CTX_DOC.TOKENS プロシージャの出力として格納されます。トークン表には自由に名前を付けることができますが、次の名前およびデータ型を持つ列が組み込まれている必要があります。

表 A-13

列名	データ型	説明
QUERY_ID	NUMBER	CTX_DOC.HIGHLIGHT への特定のコールで生成された結果の識別子（複数の HIGHLIGHT コールの結果を表に格納するときのみ移入されます）。
TOKEN	VARCHAR2(64)	テキスト内のトークン文字列。
OFFSET	NUMBER	ドキュメント内のトークンの位置（ドキュメントの開始位置 1 との相対的な位置）。
LENGTH	NUMBER	トークンの文字の長さ。

CTX_THES 結果表およびデータ型

BT、NT、SYN などの CTX_THES 拡張ファンクションは、EXP_TAB 型の表に拡張結果を戻すことができます。restab 引数で表の名前を指定できます。

EXP_TAB 表型

EXP_TAB 表型は EXP_REC 型の行の表です。

EXP_REC および EXP_TAB 型は、CTXSYS スキーマで次のように定義されます。

```
type exp_rec is record (
    xrel varchar2(12),
    xlevel number,
    xphrase varchar2(256)
);

type exp_tab is table of exp_rec index by binary_integer;
```

シソーラス拡張ファンクションをコールして restab を指定する場合、システムは拡張結果を EXP_TAB 表として戻します。この表の各行は、EXP_REC 型で、拡張のワードまたは句を表します。次の表は、EXP_REC のフィールドを示しています。

EXP_REC フィールド	説明
xrel	xrel フィールドには、語句と入力語句のリレーションが含まれます（SYN、PT、RT など）。入力語句が拡張に表示されている場合、xrel 値は PHRASE です。翻訳に対しては、xrel 値は言語です。
xlevel	xlevel フィールドは、リレーションのレベルです。これは、主に xrel が階層リレーション（BT*/NT*）の場合に使用されます。 xlevel フィールドは、xrel が PHRASE のときは 0（ゼロ）です。 xlevel フィールドは、TRSYN でのシノニムの翻訳の場合は 2 です。 xlevel フィールドは、PT や RT などの階層でない演算子の場合は 1 です。
xphrase	xphrase は関連語です。関連語に修飾子が存在する場合は、カッコ内で囲みます。コンパウンド語句は、非コンパウンド化されません。

B

サポートされているドキュメント形式

この付録では、Inso フィルタ処理テクノロジによってサポートされているドキュメント形式をリストに示します。この付録の内容は次のとおりです。

- ドキュメント・フィルタ処理テクノロジ
- サポートされているドキュメント形式
- サポートされていない形式

ドキュメント・フィルタ処理テクノロジ

Oracle Text では、Stellent Chicago, Inc. からライセンス許可を受けているドキュメント・フィルタ処理テクノロジを使用しています。このフィルタ処理テクノロジによって、ほとんどのドキュメント形式を索引付けできます。さらに、このテクノロジによって、ドキュメントを表示するために、CTX_DOC パッケージを使用してドキュメントを HTML に変換できます。

> **関連項目：** サポートされている形式のリストは、この付録の「サポートされているドキュメント形式」を参照してください。

索引付けおよび DML 処理に Inso フィルタ処理を使用するには、フィルタ・プリファレンスに INSO_FILTER オブジェクトを指定する必要があります。

CTX_DOC パッケージでドキュメントを HTML に変換するために Inso フィルタ処理テクノロジを使用する場合は、INSO_FILTER 索引付けプリファレンスを使用する必要はありません。ただし、使用している環境は、この付録で説明するとおり、このフィルタ処理テクノロジを使用するように設定しておく必要があります。

Inso フィルタ処理テクノロジは、Stellent Chicago, Inc. からライセンス許可を受けている共有ライブラリおよびデータ・ファイルに依存して、ドキュメントを HTML 形式に変換します。

次の各項では、サポートされているプラットフォームおよび異なるプラットフォームで Inso フィルタ処理を使用可能にする方法を説明します。

サポートされているプラットフォーム

サポートされているプラットフォーム

Inso フィルタ・テクノロジは、Oracle では以下のプラットフォームをサポートしています。

- Sun Solaris SPARC 32 ビットおよび 64 ビット（2.6、2.7、2.8）
- IBM AIX 64 ビット（4.2 ～ 4.3）
- HP-UX 64 ビット（11.0）
- Alpha OpenVMS/Compaq Tru64 UNIX（4.0）
- Microsoft Windows
  - Intel x86 搭載の WinNT（4.0 以上）
  - Intel x86 搭載の Win95、Win98 SE、Win2000 および Windows ME
- Intel x86 搭載の Linux

環境変数

Inso フィルタ処理に関連するすべての環境変数は、Oracle Text で参照できる必要があります。

UNIX プラットフォームに対する要件

次の要件は、Solaris、IBM AIX、HP-UX、Compaq Tru64 UNIX および Linux の各プラットフォームに適用されます。

- *.flt ファイルに、Oracle データベースおよび ctxsrv サーバーを実行しているオペレーティング・システム・ユーザーに対する実行権限が付与されていることを確認してください。
- 環境変数 $PATH は、*.flt ファイルの場所が含まれるように設定してください。具体的には、isunx2.flt ファイルの場所と Inso フィルタ処理の共有ライブラリの場所である $ORACLE_HOME/ctx/lib が含まれるように設定します。
- Inso テクノロジがファイルを $HOME ディレクトリのサブディレクトリ（.oit）に書き込めるように、環境変数 $HOME を設定してください。
- ベクトル・グラフィック・イメージ変換を行うには、実行中の X-Windows サーバーにアクセスする必要があります。

ベクトル・グラフィック形式のフィルタ処理

次のステップに従って、UNIX プラットフォームのベクトル・グラフィック形式をフィルタ処理してください。

- ベクトル・グラフィック形式をフィルタ処理するには、X サーバーを開始します。X サーバーが存在しない場合（システムが Xm、Xt、X11 などの X ライブラリを検出しない場合）、ベクトル・グラフィック・フィルタ処理は実行されません。ベクトル・グラフィック形式には、CAD 図面およびプレゼンテーション形式（Power Point 97 など）が含まれます。ビットマップ形式には、ビットマップの他に、GIF、JPEG および TIF 形式が含まれます。
- システムのベクトル・グラフィック変換の実行は X ライブラリに依存するため、X ライブラリに対するシステム固有のライブラリ・パスの環境変数が正しく設定されていることを確認してください。
- 環境変数 $DISPLAY を設定します。たとえば DISPLAY=:0.0 と設定すると、コンソールの X サーバーを使用することをシステムに知らせます。

OLE2 オブジェクトのサポート

Inso フィルタ・テクノロジが OLE2 オブジェクトで実行できる機能には制限があり、これらの制限はプラットフォームに依存します。プラットフォームでメタファイルのスナップショットが使用可能な場合、Inso テクノロジはこれを使用して OLE2 オブジェクトを変換します。

メタファイルのスナップショットが UNIX プラットフォームで使用できない場合、Inso テクノロジは OLE2 オブジェクトを変換できません。

ただし、メタファイルのスナップショットが Windows NT プラットフォームで使用できない場合は、元のアプリケーションを使用して（使用可能な場合）OLE2 オブジェクトを変換します。

サポートされているドキュメント形式

次の表は、フィルタ処理に対して Oracle Text がサポートするすべてのドキュメント形式を示しています。ドキュメント・フィルタ処理は、索引付け、DML およびドキュメントの HTML への変換（CTX_DOC パッケージを使用）に使用されます。このフィルタ処理テクノロジは、Stellent Chicago, Inc. からライセンス許可を受けた Outside In HTML Export および Outside In Content Access テクノロジに基づいています。

注意： このリストには、Oracle が処理できるすべての形式が記載されているわけではありません。すべての形式をプレーン・テキストにフィルタ処理できる外部フィルタが存在している場合、外部フィルタ・フレームワークを使用すると、Oracle ですべてのドキュメント形式を処理できるようになります。

ワード処理 - 共通

形式	バージョン
ASCII Text（7 および 8 ビット・バージョン）	すべてのバージョン
ANSI Text（7 および 8 ビット）	すべてのバージョン
Unicode Text	すべてのバージョン
HTML	バージョン 3.0 以下（一部制限あり）
IBM Revisable Form Text	すべてのバージョン
IBM FFT	すべてのバージョン
Microsoft Rich Text Format（RTF）	すべてのバージョン

ワード処理 - DOS

形式	バージョン
DEC WPS Plus（WPL）	バージョン 4.1 以下
DEC WPS Plus（DX）	バージョン 4.0 以下
DisplayWrite 2 & 3（TXT）	すべてのバージョン
DisplayWrite 4 & 5	リリース 2.0 以下のバージョン
Enable	バージョン 3.0、4.0 および 4.5
First Choice	バージョン 3.0 以下
Framework	バージョン 3.0
IBM Writing Assistant	バージョン 1.01
Lotus Manuscript	バージョン 2.0 以下
MASS11	バージョン 8.0 以下
Microsoft Word	バージョン 6.0 以下
Microsoft Works	バージョン 2.0 以下
MultiMate	バージョン 4.0 以下
Navy DIF	すべてのバージョン

形式	バージョン
Nota Bene	バージョン 3.0
Office Writer	バージョン 4.0 〜 6.0
PC-File Letter	バージョン 5.0 以下
PC-File+ Letter	バージョン 3.0 以下
PFS:Write	バージョン A、B および C
Professional Write	バージョン 2.1 以下
Q&A	バージョン 2.0
Samna Word	Samna Word IV+ 以下のバージョン
SmartWare II	バージョン 1.02
Sprint	バージョン 1.0 以下
Total Word	バージョン 1.2
Volkswriter 3 & 4	バージョン 1.0 以下
Wang PC（IWP）	バージョン 2.6 以下
WordMARC	Composer Plus 以下のバージョン
WordPerfect	バージョン 6.1 以下
WordStar	バージョン 7.0 以下
WordStar 2000	バージョン 3.0 以下
XyWrite	III Plus 以下のバージョン

ワード処理 - インターナショナル

形式	バージョン
JustSystems 一太郎	バージョン 5.0、6.0、8.0、9.0 および 10.0

ワード処理 - Windows

形式	バージョン
AMI/AMI Professional	バージョン 3.1 以下
Corel WordPerfect for Windows	バージョン 2002 以下
JustWrite	バージョン 3.0 以下
Legacy	バージョン 1.1 以下
Lotus WordPro（Intel 搭載の NT のみ）	SmartSuite 96、97、Millennium および Millennium 9.6
Lotus WordPro（Intel 搭載の NT を除くすべてのサポートされているプラットフォーム、テキストのみ）	SmartSuite 97、Millennium および Millennium 9.6
Microsoft Windows Works	バージョン 4.0 以下
Microsoft Windows Write	バージョン 3.0 以下
Microsoft Word 97	Word 97
Microsoft Word 2000	Word 2000
Microsoft Word 2002（Office XP）	Word 2002
Microsoft Word for Windows	バージョン 7.0 以下
Microsoft WordPad	すべてのバージョン
Novell Perfect Works	バージョン 2.0
Novell WordPerfect for Windows	バージョン 7.0 以下
Professional Write Plus	バージョン 1.0
Q&A Write for Windows	バージョン 3.0
Star Office Writer for Windows（テキストのみ）	バージョン 5.2
WordStar for Windows	バージョン 1.0

ワード処理 - Macintosh

形式	バージョン
Microsoft Word	バージョン 4.0 ～ 6.0
Microsoft Word 98	Word 98
WordPerfect	バージョン 1.02 ～ 3.0
Microsoft Works	バージョン 2.0 以下
MacWrite II	バージョン 1.1

ワード処理 - Unix

形式	バージョン
Star Office Writer for Windows	バージョン 5.2

デスクトップ・パブリッシング

形式	バージョン
Adobe FrameMaker	バージョン 6.0

スプレッドシート形式

形式	バージョン
Enable	バージョン 3.0、4.0 および 4.5
First Choice	バージョン 3.0 以下
Framework	バージョン 3.0
Lotus 1-2-3（DOS & Windows）	バージョン 5.0 以下
Lotus 1-2-3 for SmartSuite	SmartSuite 97、Millennium および Millennium 9.6
Lotus 1-2-3 Charts（DOS & Windows）	バージョン Millennium 9.6 以下
Lotus 1-2-3（OS/2）	バージョン 2.0 以下

形式	バージョン
Lotus 1-2-3 Charts（OS/2）	バージョン 2.0 以下
Lotus Symphony	バージョン 1.0、1.1 および 2.0
Microsoft Excel 97	Excel 97
Microsoft Excel 2000	Excel 2000
Microsoft Excel 2002（Office XP）	Excel 2002
Microsoft Excel Windows	バージョン 2.2 ～ 7.0
Microsoft Excel Macintosh	バージョン 3.0 ～ 4.0 および 98
Microsoft Excel Charts	バージョン 2.x ～ 7.0
Microsoft Multiplan	バージョン 4.0
Microsoft Windows Works	バージョン 4.0 以下
Microsoft Works（DOS）	バージョン 2.0 以下
Microsoft Works（Mac）	バージョン 2.0 以下
Mosaic Twin	バージョン 2.5
Novell Perfect Works	バージョン 2.0
QuattroPro for DOS	バージョン 5.0 以下
QuattroPro for Windows	バージョン 2002 以下
PFS:Professional Plan	バージョン 1.0
SuperCalc 5	バージョン 4.0
SmartWare II	バージョン 1.02
VP Planner 3D	バージョン 1.0

データベース形式

形式	バージョン
Access	バージョン 2.0 以下
dBASE	バージョン 5.0 以下
DataEase	バージョン 4.x
dBXL	バージョン 1.3
Enable	バージョン 3.0、4.0 および 4.5
First Choice	バージョン 3.0 以下
FoxBase	バージョン 2.1
Framework	バージョン 3.0
Microsoft Windows Works	バージョン 4.0 以下
Microsoft Works（DOS）	バージョン 2.0 以下
Microsoft Works（Mac）	バージョン 2.0 以下
Paradox（DOS）	バージョン 4.0 以下
Paradox（Windows）	バージョン 1.0 以下
Personal R:BASE	バージョン 1.0
R:BASE 5000	バージョン 3.1 以下
R:BASE System V	バージョン 1.0
Reflex	バージョン 2.0
Q & A	バージョン 2.0 以下
SmartWare II	バージョン 1.02

表示形式

形式	バージョン
PDF - Portable Document Format	Acrobat バージョン 2.1、3.0、4.0 および 5.0（日本語 PDF を含む）

プレゼンテーション形式

形式	バージョン
Corel Presentations	バージョン 8.0、9.0 および 2002
Novell Presentations	バージョン 3.0 および 7.0
Harvard Graphics for DOS	バージョン 2.x および 3.x
Harvard Graphics	Windows バージョン
Freelance 96	Freelance 96
Freelance for Windows	SmartSuite 97、Millennium および Millennium 9.6
Freelance for Windows	バージョン 1.0 および 2.0
Freelance for OS/2	バージョン 2.0 以下
Microsoft PowerPoint for Windows	バージョン 7.0 以下
Microsoft PowerPoint 97	PowerPoint 97
Microsoft PowerPoint 2000	PowerPoint 2000
Microsoft PowerPoint 2002（Office XP）	PowerPoint 2002
Microsoft PowerPoint for Macintosh	バージョン 4.0 および 98

標準グラフィック形式

次の表は、Inso フィルタが認識するグラフィック形式を示しています。これらのグラフィック形式を含むテキスト列は、索引付けしてもエラーは発生しません。つまり、列にこれらの形式のいずれが含まれていても問題はありません。

注意： Inso フィルタは、図形からテキスト情報を取り出すことはできません。

形式	バージョン
Binary Group 3 Fax	すべてのバージョン
BMP（RLE、ICO、CUR および OS/2 DIB を含む）	Windows
CALS Raster	Type 1 および II
CDR（TIFF イメージが埋め込まれている場合）	Corel Draw バージョン 2.0 ～ 9.0
CGM - Computer Graphics Metafile	ANSI、CALS、NIST、バージョン 3.0
DCX（複数ページの PCX）	Microsoft Fax
DRW - Micrografx Designer	バージョン 3.1
DRW - Micrografx Draw	バージョン 4.0
DXF（Binary および ASCII）AutoCAD Drawing Interchange Format	バージョン 14 以下
EMF	Windows Enhanced Metafile
EPS - Encapsulated PostScript	TIFF イメージが埋め込まれている場合
FPX - Kodak Flash Pix	固有のバージョンなし
GIF - Graphics Interchange Format	Compuserve
GP4 - Group 4 CALS format	Types I および II
HPGL - Hewlett Packard Graphics Language	バージョン 2.0
IMG - GEM Paint	固有のバージョンなし
JFIF（JPEG not in TIFF）	すべてのバージョン
JPEG	すべてのバージョン
Novell Perfect Works（Draw）	Novell バージョン 2.0
PBM - Portable Bitmap	固有のバージョンなし

形式	バージョン
PCD - Kodak Photo CD	バージョン 1.0
PCX Bitmap	PC Paintbrush
PGM - Portable Graymap	固有のバージョンなし
PIC	Lotus 1-2-3 Picture File Format - 固有のバージョンなし
PICT1 & PICT2（Raster）	Macintosh Standard
PNG - Portable Network Graphics Internet Format	バージョン 1.0
PNTG	MacPaint
PPM - Portable Pixmap	固有のバージョンなし
Progressive JPEG	固有のバージョンなし
PSP - Paintshop Pro（Intel 搭載の NT のみ）	バージョン 5.0 および 5.0.1
SDW	Ami Draw
Snapshot（Lotus）	すべてのバージョン
SRS - Sun Raster File Format	固有のバージョンなし
Targa	Truevision
TIFF	バージョン 6 以下
TIFF CCITT Group 3 & 4	Fax Systems
VISO	Visio 4（ページ・プレビューのみ）、5、2000、2002
WBMP	固有のバージョンなし
WMF	Windows Metafile
WordPerfect Graphics [WPG および WPG2]	バージョン 2.0 以下
XBM - X-Windows Bitmap	x10 互換
XPM - X-Windows Pixmap	x10 互換
XWD - X-Windows Dump	x10 互換

その他

形式	バージョン
Executable（EXE、DLL）	固有のバージョンなし
Executable for Windows NT	固有のバージョンなし
Microsoft Project（テキストのみ）	Project 98
MSG（テキストのみ）	Microsoft Outlook メール形式
vCard Electronic Business Card	Versit バージョン 2.1
WML	バージョン 5.2 互換

サポートされていない形式

パスワードで保護されているドキュメントおよびパスワードで保護されている内容を含むドキュメントは、Inso フィルタではサポートされていません。

C

ロード例

この付録では、テキスト列にテキストをロードする方法の例を示します。また、ctxload インポート・ファイルの構造についても説明します。

- SQL の INSERT 例
- SQL*Loader 例
- ctxload シソーラス・インポート・ファイルの構造

SQL の INSERT 例

テキスト表を移入するには、CREATE TABLE を使用して id および text の 2 列の表を作成し、INSERT 文を使用してデータをロードすると簡単です。この例では、id 列を主キーにしています。これはオプションです。テキスト列は、VARCHAR2 です。

```
create table docs (id number primary key, text varchar2(80));
```

テキスト列を移入するには、次のように INSERT 文を使用します。

```
insert into docs values(1, 'this is the text of the first document');
insert into docs values(12, 'this is the text of the second document');
```

SQL*Loader 例

次の例では、SQL*Loader を使用してオペレーティング・システムから BLOB 列に複合形式のドキュメントをロードする方法を説明します。例には、次の 2 つのステップがあります。

- 表の作成
- 制御ファイルを読み込み、データを表にロードする SQL*Loader コマンドの発行

関連項目： SQL*Loader の使用方法の詳細は、『Oracle9*i* データベース・ユーティリティ』を参照してください。

表の作成

この例では、articles_formatted 表を次のように作成します。

```
CREATE TABLE articles_formatted (
  ARTICLE_ID   NUMBER PRIMARY KEY ,
  AUTHOR       VARCHAR2(30),
  FORMAT       VARCHAR2(30),
  PUB_DATE     DATE,
  TITLE        VARCHAR2(256),
  TEXT         BLOB
);
```

article_id 列は、主キーです。ドキュメントは、BLOB 型であるテキスト列にロードされます。

SQL*Loader コマンドの発行

次のコマンドによって、制御ファイル LOADER1.DAT を読み込むローダーを起動します。

```
sqlldr userid=demo/demo control=loader1.dat log=loader.log
```

制御ファイル例：loader1.dat

この SQL*Loader 制御ファイルは、ロードされる列を定義し、loader2.dat から articles_formatted 表にデータを 1 行ずつロードするようにローダーに指示します。loader2.dat の各行は、ロードするフィールドをカンマで区切られたリストとして保持します。

```
-- load file example
load data
INFILE 'loader2.dat'
INTO TABLE articles_formatted
APPEND
FIELDS TERMINATED BY ','
(article_id SEQUENCE (MAX,1),
 author CHAR(30),
 format,
 pub_date SYSDATE,
 title,
 ext_fname FILLER CHAR(80),
 text LOBFILE(ext_fname) TERMINATED BY EOF)
```

この制御ファイルは、次の方法で、loader2.dat から articles_formatted 表にデータをロードするようにローダーに指示します。

1. loader2.dat のドキュメント・フィールドを示す行の順番を、article_id 列に書き込みます。
2. 行の 1 番目のフィールドを author 列に書き込みます。
3. 行の 2 番目のフィールドをフォーマット列に書き込みます。
4. SYSDATE によって指定された現在の日付を pub_date 列に書き込みます。
5. 行の 3 番目のフィールドであるドキュメントのタイトルを、title 列に書き込みます。
6. ロードするドキュメントのすべての名前を ext_fname 一時変数にロードし、実際のドキュメントを BLOB 型のテキスト列にロードします。

データ・ファイル例：loader2.dat

このファイルは、articles_formatted 表の各行にロードするデータを含みます。

各行には、articles_formatted にロードされるフィールドの、カンマで区切られたリストがあります。各行の最後のフィールドは、テキスト列にロードされるファイルの名前を示します。

```
Ben Kanobi, plaintext,Kawasaki news article,../sample_docs/kawasaki.txt,
Joe Bloggs, plaintext,Java plug-in,../sample_docs/javaplugin.txt,
John Hancock, plaintext,Declaration of Independence,../sample_docs/indep.txt,
M. S. Developer, Word7,Newsletter example,../sample_docs/newsletter.doc,
M. S. Developer, Word7,Resume example,../sample_docs/resume.doc,
X. L. Developer, Excel7,Common example,../sample_docs/common.xls,
X. L. Developer, Excel7,Complex example,../sample_docs/solvsamp.xls,
Pow R. Point, Powerpoint7,Generic presentation,../sample_docs/generic.ppt,
Pow R. Point, Powerpoint7,Meeting presentation,../sample_docs/meeting.ppt,
Java Man, PDF,Java Beans paper,../sample_docs/j_bean.pdf,
Java Man, PDF,Java on the server paper,../sample_docs/j_svr.pdf,
Ora Webmaster, HTML,Oracle home page,../sample_docs/oramnu97.html,
Ora Webmaster, HTML,Oracle Company Overview,../sample_docs/oraoverview.html,
John Constable, GIF,Laurence J. Ellison : portrait,../sample_docs/larry.gif,
Alan Greenspan, GIF,Oracle revenues : Graph,../sample_docs/oragraph97.gif,
Giorgio Armani, GIF,Oracle Revenues : Trend,../sample_docs/oratrend.gif,
```

ctxload シソーラス・インポート・ファイルの構造

シソーラスのインポート・ファイルのエントリには、次の形式を使用する必要があります。

```
phrase
 BT broader_term
 NT narrower_term1
 NT narrower_term2
. . .
 NT narrower_termN

 BTG broader_term
 NTG narrower_term1
 NTG narrower_term2
. . .
 NTG narrower_termN

 BTP broader_term
 NTP narrower_term1
 NTP narrower_term2
. . .
 NTP narrower_termN
```

```
 BTI broader_term
 NTI narrower_term1
 NTI narrower_term2
. . .
 NTI narrower_termN

 SYN synonym1
 SYN synonym2
. . .
 SYN synonymN

 USE synonym1  or  SEE synonym1 or  PT  synonym1

 RT related_term1
 RT related_term2
. . .
 RT related_termN

 SN text

 language_key: term
```

phrase

シノニム、上位語、下位語または関連語（あるいはそのすべて）を持つワードまたは句です。

ISO-2788 標準に従って、TT マーカーを句の前に置くことで、句が階層の最上位語であることを明示できますが、TT マーカーが必ず必要であるということはありません。実際、ctxload はインポート時に TT マーカーを無視します。

最上位語は上位語（BT、BTG、BTP または BTI）を持たない句として識別されます。

注意： シソーラス問合せ演算子（SYN、PT、BT、BTG、BTP、BTI、NT、NTG、NTP、NTI および RT）は予約語であるため、シソーラスのエントリで句として使用できません。

BT、BTG、BTP、BTI broader_termN
broader_termN が phrase の上位（汎用、部分またはインスタンス）語であることを示すマーカーです。

broader_termN は、phrase の一般的な説明またはカテゴリ（上位語）を概念的に提供するワードまたは句です。たとえば、*elephant* というワードには、*land mammal* という上位語があります。

NT、NTG、NTP、NTI narrower_termN
narrower_termN が phrase の下位（汎用、部分またはインスタンス）語であることを示すマーカーです。

phrase に上位（汎用、部分またはインスタンス）語がなく、1 つ以上の下位（汎用、部分またはインスタンス）語がある場合、phrase は各階層内で最上位語として作成されます（Oracle Text のシソーラスでは、BT/NT、BTG/NTG、BTP/NTP および BTI/NTI 階層は個別の構造体です）。

narrower_termN は、phrase の特定の説明（下位語）を概念的に提供するワードまたは句です。たとえば、*elephant* というワードには、*indian elephant* および *african elephant* という下位語があります。

SYN synonymN
phrase および synonymN がシノニム・リング内でシノニムであることを示すマーカーです。

synonymN は、phrase と同じ意味を持つワードまたは句です。たとえば、*dog* というワードには、*canine* というシノニムがあります。

注意： シノニム・リングは、Oracle Text のシソーラスでは明示的に定義されていません。これらは、シノニムの推移的な性質によって作成されます。

USE SEE PT synonym1
phrase および synonym1 がシノニム・リング内でシノニムであることを示すマーカーです（SYN に似ています）。

また、USE マーカー、SEE マーカーまたは PT マーカーは、synonym1 がシノニム・リングに対する優先語であることを示します。すべてのマーカーがシノニム・リングに対する優先語を定義するために使用できます。

RT related_termN
related_termN が phrase の関連語であることを示すマーカーです。

related_termN は、phrase に関連する意味を持つが、必ずしもシノニムではないワードまたは句です。たとえば、*dog* というワードには、*wolf* という関連語があります。

> **注意：** 関連語には推移性はありません。句に 2 つ以上の関連語がある場合、その語はそれぞれ親句と関連しているだけであり、関連語同士が互いに関連しているわけではありません。

SN text
後続の text が、前のエントリのスコープ・ノート（コメント）であることを示すマーカーです。

language_key term
term は、language_key で指定された言語に翻訳された句です。

代替階層構造

シソーラスの規格に従い、ロード・ファイルは、最上位語の下に語句をインデントし、語句のレベルを含む NT（または NTG、NTP、NTI）マーカーを使用することによって、書式設定階層（BT/NT、BTG/NTG、BTP、NTP、BTI/NTI）をサポートします。

```
phrase
   NT1 narrower_term1
      NT2 narrower_term1.1
      NT2 narrower_term1.2
          NT3 narrower_term1.2.1
          NT3 narrower_term1.2.2
   NT1 narrower_term2
   . . .
   NT1 narrower_termN
```

この方法によって、最上位語のブランチ全体を、ロード・ファイル内で階層的に表示できます。

インポート・ファイル内の語句の使用上の注意

インポート・ファイルのエントリの構造には、次の条件が適用されます。

- 各エントリ（phrase、BT、NT または SYN）は、改行に続く 1 行以内でまとめる必要があります。
- エントリは、1 つのワードまたは句で構成されます。
- エントリ（phrase、BT、NT または SYN）の最大長は、255 文字です。この中には BT、NT および SYN のマーカーまたは改行は含まれません。
- エントリにカッコまたはプラス符号は指定できません。
- 関連語句（BT、NT など）で始まるファイルのすべての行は、行頭に 1 つ以上の空白が必要です。
- ファイルには、同じ phrase を 2 回以上使用できます。
- 各 phrase には 1 つ以上の下位語エントリ（NT、NTG、NTP）、上位語エントリ（BT、BTG、BTP）、シノニム・エントリおよび関連語エントリを指定できます。
- 上位語、下位語、シノニムおよび優先語のエントリは、それぞれ適切なマーカーで始める必要があります。また、マーカーは大文字で指定します。
- phrase には、上位語、下位語およびシノニムをどのような順序でも指定できます。
- 同形異義語を使用する場合は、その後に意味を限定する語をカッコで囲んで指定する必要があります。

  たとえば、`cranes (birds)`、`cranes (lifting equipment)` のように指定します。
- コンパウンド語句は、各要素の間にプラス記号を入れます（例 : buildings + construction）。
- コンパウンド語句は、他の語句に対するシノニムまたは優先語としてのみ使用できます。コンパウンド語句自体は語句として使用できません。また、階層関係にも使用できません。
- 語句に続く行に、最大 2000 文字までのスコープ・ノート（SN）を指定できます。

- 複数行に渡るスコープ・ノートを作成できますが、ノートの各行に SN マーカーが必要です。

  不適切な SN の使用例

  ```
  VIEW CAMERAS
   SN Cameras with through-the lens focusing and a
  range of movements of the lens plane relative to
  the film plane
  ```

  適切な SN の使用例

  ```
  VIEW CAMERAS
   SN Cameras with through-the lens focusing and a
   SN range of movements of the lens plane relative
   SN to the film plane
  ```

- 複数ワードの語句を、予約語で始めることはできません（たとえば、*use* という語句は予約語であるため、*use other door* は語句として使用できません。ただし、*use* という語句は使用可能です）。

インポート・ファイル内の関連語句の使用上の注意

インポート・ファイルのエントリに定義される関連語句には、次の条件が適用されます。

- 関連語エントリは、句または別の関連語エントリの後に続けて指定する必要があります。
- 関連語エントリは 1 つ以上の空白から始め、RT マーカーの後に空白を入れ、その同じ行に関連語を指定します。
- 複数の関連語には複数の RT マーカーが必要です。

  不適切な RT の使用例

  ```
  MOVING PICTURE CAMERAS
   RT CINE CAMERAS
  TELEVISION CAMERAS
  ```

  適切な RT の使用例

  ```
  MOVING PICTURE CAMERAS
   RT CINE CAMERAS
   RT TELEVISION CAMERAS
  ```

- 語句には、複数の上位語、下位語および関連語を指定できます。

インポート・ファイル例

この項では、正しく形式設定されたシソーラスのインポート・ファイルの例を 3 つ示します。

例 1（フラット構造）

```
cat
 SYN feline
 NT domestic cat
 NT wild cat
 BT mammal
mammal
 BT animal
domestic cat
 NT Persian cat
 NT Siamese cat
wild cat
 NT tiger
tiger
 NT Bengal tiger
dog
 BT mammal
 NT domestic dog
 NT wild dog
 SYN canine
domestic dog
 NT German Shepard
wild dog
 NT Dingo
```

例 2（階層構造）

```
animal
   NT1 mammal
        NT2 cat
             NT3 domestic cat
                  NT4 Persian cat
                  NT4 Siamese cat
             NT3 wild cat
                  NT4 tiger
                       NT5 Bengal tiger
        NT2 dog
             NT3 domestic dog
                  NT4 German Shepard
             NT3 wild dog
                  NT4 Dingo
cat
 SYN feline
dog
 SYN canine
```

例 3

```
35MM CAMERAS
 BT MINIATURE CAMERAS
CAMERAS
 BT OPTICAL EQUIPMENT
 NT MOVING PICTURE CAMERAS
 NT STEREO CAMERAS
LAND CAMERAS
 USE VIEW CAMERAS
VIEW CAMERAS
 SN Cameras with through-the lens focusing and a range of
 SN movements of the lens plane relative to the film plane
 UF LAND CAMERAS
 BT STILL CAMERAS
```

D

提供されるストップリスト

この付録では、サポートされるすべての言語に対するデフォルトのストップリストについて説明し、言語ごとにストップワードを示します。次のストップリストについて説明します。

- 英語のデフォルト・ストップリスト
- 中国語（繁体字）のストップリスト
- 中国語（簡体字）のストップリスト
- デンマーク語（dk）のデフォルト・ストップリスト
- オランダ語（nl）のデフォルト・ストップリスト
- フィンランド語（sf）のデフォルト・ストップリスト
- フランス語（f）のデフォルト・ストップリスト
- ドイツ語（d）のデフォルト・ストップリスト
- イタリア語（i）のデフォルト・ストップリスト
- ポルトガル語（pt）のデフォルト・ストップリスト
- スペイン語（e）のデフォルト・ストップリスト
- スウェーデン語（s）のデフォルト・ストップリスト

英語のデフォルト・ストップリスト

次の英語のワードが、ストップワードとして定義されています。

ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード
a	be	had	it	only	she	was
about	because	has	its	of	some	we
after	been	have	last	on	such	were
all	but	he	more	one	than	when
also	by	her	most	or	that	which
an	can	his	mr	other	the	who
any	co	if	mrs	out	their	will
and	corp	in	ms	over	there	with
are	could	inc	mz	s	they	would
as	for	into	no	so	this	up
at	from	is	not	says	to	

中国語（繁体字）のストップリスト

次の中国語（繁体字）のワードが、この言語に対するデフォルトのストップリストに定義されています。

前
目
有關
但是
必須
至於
除了
繼續
過去
立即
進入
只要

於
由
不過
相當
以上
一般
不少
另外
所有
左右
並不
否則

此
因
可以
其中
之後
不是
最後
共同
不會
經過
了解
據
未
並

們
他
果
如
他
其
以
所
能
不
是
就
有
只
來
以
其
尤
在
現
麼
什

能
可
於
對
然
雖
及
以
且
而
別
分
解
了
何
任
得
使
是
只
此
如

有
沒
為
因
們
我
多
許
起
引
強
加
據
根
直
一
關
相
要
需
要
不

望
希
否
是
括
包
近
最
何
如
至
甚
經
已
同
不
時
當
因
原

中国語（簡体字）のストップリスト

次の中国語（簡体字）のワードが、この言語に対するデフォルトのストップリストに定義されています。

必将
超过
达到
都是
各种
回到
尽管
之一
如何
为了
也是

必须
成为
大量
对于
共同
获得了
就是
没有
什么
我们
以及

并非
除了
带来
这个
还将
或者
具有
目前
实在
下去
已经

由于
处在
带着
而且
还有
基本上
可能
哪里
所需
现在
以上

一同
此项
但是
而言
很少
基于
可以
那里
所有
相当
因此

一再
从而
当时
方面
很有
即可
来自
却是
它的
新的
因为

一得
存在着
得到
各方面
还是
较大
两个
如果
他们
许多

デンマーク語（dk）のデフォルト・ストップリスト

次のデンマーク語のワードが、この言語に対するデフォルトのストップリストに定義されています。

ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード
af	en	god	hvordan	med	og	udenfor
aldrig	et	han	I	meget	oppe	under
alle	endnu	her	De	mellem	på	ved
altid	få	hos	i	mere	rask	vi
bagved	lidt	hovfor	imod	mindre	hurtig	
de	fjernt	hun	ja	når	sammen	
der	for	hvad	jeg	hvonår	temmelig	
du	foran	hvem	langsom	nede	nok	
efter	fra	hvor	mange	nej	til	
eller	gennem	hvorhen	måske	nu	uden	

オランダ語（nl）のデフォルト・ストップリスト

次のオランダ語のワードが、この言語に対するデフォルトのストップリストに定義されています。

ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード
aan	betreffende	eer	had	juist	na	overeind	van	weer
aangaande	bij	eerdat	hadden	jullie	naar	overigens	vandaan	weg
aangezien	binnen	eerder	hare	kan	nadat	pas	vanuit	wegens
achter	binnenin	eerlang	heb	klaar	net	precies	vanwege	wel
achterna	boven	eerst	hebben	kon	niet	reeds	veeleer	weldra
afgelopen	bovenal	elk	hebt	konden	noch	rond	verder	welk
al	bovendien	elke	heeft	krachtens	nog	rondom	vervolgens	welke
aldaar	bovengenoemd	en	hem	kunnen	nogal	sedert	vol	wie
aldus	bovenstaand	enig	hen	kunt	nu	sinds	volgens	wiens
alhoewel	bovenvermeld	enigszins	het	later	of	sindsdien	voor	wier
alias	buiten	enkel	hierbeneden	liever	ofschoon	slechts	vooraf	wij

ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード
alle	daar	er	hierboven	maar	om	sommige	vooral	wijzelf
allebei	daarheen	erdoor	hij	mag	omdat	spoedig	vooralsnog	zal
alleen	daarin	even	hoe	meer	omhoog	steeds	voorbij	ze
alsnog	daarna	eveneens	hoewel	met	omlaag	tamelijk	voordat	zelfs
altijd	daarnet	evenwel	hun	mezelf	omstreeks	tenzij	voordezen	zichzelf
altoos	daarom	gauw	hunne	mij	omtrent	terwijl	voordien	zij
ander	daarop	gedurende	ik	mijn	omver	thans	voorheen	zijn
andere	daarvanlangs	geen	ikzelf	mijnent	onder	tijdens	voorop	zijne
anders	dan	gehad	in	mijner	ondertussen	toch	vooruit	zo
anderszins	dat	gekund	inmiddels	mijzelf	ongeveer	toen	vrij	zodra
behalve	de	geleden	inzake	misschien	ons	toenmaals	vroeg	zonder
behoudens	die	gelijk	is	mocht	onszelf	toenmalig	waar	zou
beide	dikwijls	gemoeten	jezelf	mochten	onze	tot	waarom	zouden
beiden	dit	gemogen	jij	moest	ook	totdat	wanneer	zowat
ben	door	geweest	jijzelf	moesten	op	tussen	want	zulke
beneden	doorgaand	gewoon	jou	moet	opnieuw	uit	waren	zullen
bent	dus	gewoonweg	jouw	moeten	opzij	uitgezonderd	was	zult
bepaald	echter	haar	jouwe	mogen	over	vaak	wat	

フィンランド語（sf）のデフォルト・ストップリスト

次のフィンランド語のワードが、この言語に対するデフォルトのストップリストに定義されています。

ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード
aina	hyvin	kesken	me	nyt	takia	yhdessä
alla	hoikein	kukka	mikä	oikea	tässä	ylös
ansiosta	ilman	kyllä	miksi	oikealla	te	
ei	ja	kylliksi	milloin	paljon	ulkopuolella	
enemmän	jälkeen	tarpeeksi	milloinkan	siellä	vähän	
ennen	jos	lähellä	koskaan	sinä	vahemmän	
etessa	kanssa	läpi	minä	ssa	vasen	
haikki	kaukana	liian	missä	sta	vasenmalla	
hän	kenties	lla	miten	suoraan	vastan	
he	ehkä	luona	kuinkan	tai	vielä	
hitaasti	keskellä	lla	nopeasti	takana	vieressä	

フランス語（f）のデフォルト・ストップリスト

次のフランス語のワードが、この言語に対するデフォルトのストップリストに定義されています。

ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード
a	beaucoup	comment	encore	lequel	moyennant	près	ses	toujours
afin	ça	concernant	entre	les	ne	puis	sien	tous
ailleurs	ce	dans	et	lesquelles	ni	puisque	sienne	toute
ainsi	ceci	de	étaient	lesquels	non	quand	siennes	toutes
alors	cela	dedans	était	leur	nos	quant	siens	très
après	celle	dehors	étant	leurs	notamment	que	soi	trop
attendant	celles	déjà	etc	lors	notre	quel	soi-même	tu
au	celui	delà	eux	lorsque	notres	quelle	soit	un
aucun	cependant	depuis	furent	lui	nôtre	quelqu''un	sont	une
aucune	certain	des	grâce	ma	nôtres	quelqu''une	suis	vos
au-dessous	certaine	desquelles	hormis	mais	nous	quelque	sur	votre
au-dessus	certaines	desquels	hors	malgré	nulle	quelques-unes	ta	vôtre
auprès	certains	dessus	ici	me	nulles	quelques-uns	tandis	vôtres
auquel	ces	dès	il	même	on	quels	tant	vous
aussi	cet	donc	ils	mêmes	ou	qui	te	vu
aussitôt	cette	donné	jadis	mes	où	quiconque	telle	y
autant	ceux	dont	je	mien	par	quoi	telles	
autour	chacun	du	jusqu	mienne	parce	quoique	tes	
aux	chacune	duquel	jusque	miennes	parmi	sa	tienne	
auxquelles	chaque	durant	la	miens	plus	sans	tiennes	
auxquels	chez	elle	laquelle	moins	plusieurs	sauf	tiens	
avec	combien	elles	là	moment	pour	se	toi	
à	comme	en	le	mon	pourquoi	selon	ton	

ドイツ語（d）のデフォルト・ストップリスト

次のドイツ語のワードが、この言語に対するデフォルトのストップリストに定義されています。

ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード
ab	dann	des	es	ihnen	keinem	obgleich	sondern	welchem
aber	daran	desselben	etwa	ihr	keinen	oder	sonst	welchen
allein	darauf	dessen	etwas	ihre	keiner	ohne	soviel	welcher
als	daraus	dich	euch	Ihre	keines	paar	soweit	welches
also	darin	die	euer	ihrem	man	sehr	über	wem
am	darüber	dies	eure	Ihrem	mehr	sei	um	wen
an	darum	diese	eurem	ihren	mein	sein	und	wenn
auch	darunter	dieselbe	euren	Ihren	meine	seine	uns	wer
auf	das	dieselben	eurer	Ihrer	meinem	seinem	unser	weshalb
aus	dasselbe	diesem	eures	ihrer	meinen	seinen	unsre	wessen
außer	daß	diesen	für	ihres	meiner	seiner	unsrem	wie
bald	davon	dieser	fürs	Ihres	meines	seines	unsren	wir
bei	davor	dieses	ganz	im	mich	seit	unsrer	wo
beim	dazu	dir	gar	in	mir	seitdem	unsres	womit
bin	dazwischen	doch	gegen	ist	mit	selbst	vom	zu
bis	dein	dort	genau	ja	nach	sich	von	zum
bißchen	deine	du	gewesen	je	nachdem	Sie	vor	zur
bist	deinem	ebenso	her	jedesmal	nämlich	sie	während	zwar
da	deinen	ehe	herein	jedoch	neben	sind	war	zwischen
dabei	deiner	ein	herum	jene	nein	so	wäre	zwischens
dadurch	deines	eine	hin	jenem	nicht	sogar	wären	
dafür	dem	einem	hinter	jenen	nichts	solch	warum	
dagegen	demselben	einen	hintern	jener	noch	solche	was	
dahinter	den	einer	ich	jenes	nun	solchem	wegen	
damit	denn	eines	ihm	kaum	nur	solchen	weil	
danach	der	entlang	ihn	kein	ob	solcher	weit	
daneben	derselben	er	Ihnen	keine	ober	solches	welche	

イタリア語（i）のデフォルト・ストップリスト

次のイタリア語のワードが、この言語に対するデフォルトのストップリストに定義されています。

ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード
a	da	durante	lo	o	seppure	un
affinchè	dachè	e	loro	onde	si	una
agl''	dagl''	egli	ma	oppure	siccome	uno
agli	dagli	eppure	mentre	ossia	sopra	voi
ai	dai	essere	mio	ovvero	sotto	vostro
al	dal	essi	ne	per	su	
all''	dall''	finché	neanche	perchè	subito	
alla	dalla	fino	negl''	perciò	sugl''	
alle	dalle	fra	negli	però	sugli	
allo	dallo	giacchè	nei	poichè	sui	
anzichè	degl''	gl''	nel	prima	sul	
avere	degli	gli	nell''	purchè	sull''	
bensì	dei	grazie	nella	quand''anche	sulla	
che	del	I	nelle	quando	sulle	
chi	dell''	il	nello	quantunque	sullo	
cioè	delle	in	nemmeno	quasi	suo	
come	dello	inoltre	neppure	quindi	talchè	
comunque	di	io	noi	se	tu	
con	dopo	l''	nonchè	sebbene	tuo	
contro	dove	la	nondimeno	sennonchè	tuttavia	
cosa	dunque	le	nostro	senza	tutti	

ポルトガル語（pt）のデフォルト・ストップリスト

次のポルトガル語のワードが、この言語に対するデフォルトのストップリストに定義されています。

ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード
a	bem	e	longe	para	se	você
abaixo	com	ela	mais	por	sem	vocês
adiante	como	elas	menos	porque	sempre	
agora	contra	êle	muito	pouco	sim	
ali	debaixo	eles	não	próximo	sob	
antes	demais	em	ninguem	qual	sobre	
aqui	depois	entre	nós	quando	talvez	
até	depressa	eu	nunca	quanto	todas	
atras	devagar	fora	onde	que	todos	
bastante	direito	junto	ou	quem	vagarosamente	

スペイン語（e）のデフォルト・ストップリスト

次のスペイン語のワードが、この言語に対するデフォルトのストップリストに定義されています。

ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード	ストップワード
a	aquí	cuantos	esta	misma	nosotras	querer	tales	usted
acá	cada	cuán	estar	mismas	nosotros	qué	tan	ustedes
ahí	cierta	cuánto	estas	mismo	nuestra	quien	tanta	varias
ajena	ciertas	cuántos	este	mismos	nuestras	quienes	tantas	varios
ajenas	cierto	de	estos	mucha	nuestro	quienesquiera	tanto	vosotras
ajeno	ciertos	dejar	hacer	muchas	nuestros	quienquiera	tantos	vosotros
ajenos	como	del	hasta	muchísima	nunca	quién	te	vuestra
al	cómo	demasiada	jamás	muchísimas	os	ser	tener	vuestras
algo	con	demasiadas	junto	muchísimo	otra	si	ti	vuestro
alguna	conmigo	demasiado	juntos	muchísimos	otras	siempre	toda	vuestros
algunas	consigo	demasiados	la	mucho	otro	sí	todas	y
alguno	contigo	demás	las	muchos	otros	sín	todo	yo
algunos	cualquier	el	lo	muy	para	Sr	todos	
algún	cualquiera	ella	los	nada	parecer	Sra	tomar	
allá	cualquieras	ellas	mas	ni	poca	Sres	tuya	
allí	cuan	ellos	más	ninguna	pocas	Sta	tuyo	
aquel	cuanta	él	me	ningunas	poco	suya	tú	
aquella	cuantas	esa	menos	ninguno	pocos	suyas	un	
aquellas	cuánta	esas	mía	ningunos	por	suyo	una	
aquello	cuántas	ese	mientras	no	porque	suyos	unas	
aquellos	cuanto	esos	mío	nos	que	tal	unos	

スウェーデン語（s）のデフォルト・ストップリスト

次のスウェーデン語のワードが、この言語に対するデフォルトのストップリストに定義されています。

ストップワード	ストップワード	ストップワード	ストップワード
ab	efter	ja	sin
aldrig	efterät	jag	skall
all	eftersom	långsamt	som
alla	ej	långt	till
alltid	eller	lite	tillräckligt
än	emot	man	tillsammans
ännu	en	med	trots att
ånyo	ett	medan	under
är	fastän	mellan	uppe
att	för	mer	ut
av	fort	mera	utan
avser	framför	mindre	utom
avses	från	mot	vad
bakom	genom	myckett	väl
bra	gott	när	var
bredvid	hamske	nära	varför
dä	han	nej	vart
där	här	nere	varthän
de	hellre	ni	vem
dem	hon	nu	vems
den	hos	och	vi
denna	hur	oksa	vid
deras	i	om	vilken
dess	in	över	
det	ingen	på	
detta	innan	så	
du	inte	sådan	

E

代替スペルの規則

この付録では、Oracle Text がドイツ語、デンマーク語およびスウェーデン語で使用する代替スペルの規則について説明します。代替スペルを使用可能にする方法についても説明します。

この付録の内容は次のとおりです。

- 概要
- ドイツ語の代替スペル
- デンマーク語の代替スペル
- スウェーデン語の代替スペル

概要

この付録では、Oracle Text がドイツ語、デンマーク語およびスウェーデン語に対して使用する代替スペルの規則を示します。これらの言語には、指定された複数のスペルを持つワードが含まれています。

複数のスペル方法があるワードが言語に含まれている場合、Oracle はそのワードを基本形式で索引付けします。たとえばドイツ語では、*ä* 文字の基本形式は *ae* であるため、*ä* 文字を含むワードは代替として *ae* で索引付けされます。

また、Oracle は、検索前に問合せ語句をその基本形式に変換します。したがって、ユーザーはいずれかのスペルでワードを問い合せることができます。

代替スペルの使用可能

GERMAN、DANISH または SWEDISH のいずれかを代替スペルの BASIC_LEXER 属性に指定することによって、代替スペルを使用可能にできます。たとえば、次の文を発行して、ドイツ語の代替スペルを使用可能にできます。

```
begin
ctx_ddl.create_preference('GERMAN_LEX', 'BASIC_LEXER');
ctx_ddl.set_attribute('GERMAN_LEX', 'ALTERNATE_SPELLING', 'GERMAN');
end;
```

代替スペルの使用禁止

代替スペルを使用禁止にするには、次のように CTX_DDL.UNSET_ATTRIBUTE プロシージャを使用します。

```
begin
ctx_ddl.unset_attribute('GERMAN_LEX', 'ALTERNATE_SPELLING');
end;
```

ドイツ語の代替スペル

ドイツ語のアルファベットは、英語のアルファベットに、ä ö ü ß などの他の文字を追加したものです。次の表は、Oracle Text がこれらの文字に対して使用する代替スペルの規則を示しています。

文字	代替スペル
ä	ae
ü	ue
ö	oe
Ä	AE
Ü	UE
Ö	OE
ß	ss

デンマーク語の代替スペル

デンマーク語のアルファベットは、*w* を除くラテン・アルファベットに、ø æ å などの特殊文字を追加したものです。次の表は、Oracle Text がこれらの文字に対して使用する代替スペルの規則を示しています。

文字	代替スペル
æ	ae
ø	oe
å	aa
Æ	AE
Ø	OE
Å	AA

スウェーデン語の代替スペル

スウェーデン語のアルファベットは、*w* を除く英語のアルファベットに、å ä ö などの文字を追加したものです。次の表は、Oracle Text がこれらの文字に対して使用する代替スペルの規則を示しています。

文字	代替スペル
ä	ae
å	aa
ö	oe
Ä	AE
Å	AA
Ö	OE

F

スコア付けのアルゴリズム

この付録では、ワード問合せのスコア付けのアルゴリズムについて説明します。SCORE 演算子を使用してスコアを取得します。

注意： この付録では、Oracle がワード問合せのスコアを計算する方法について説明します。これは、英語の ABOUT 問合せのスコアの計算方法とは異なります。

ワード問合せのスコア付けのアルゴリズム

ワード問合せで戻されたドキュメントの関連性スコアを計算するために、Oracle では Salton の式に基づいた逆頻度アルゴリズムを使用します。

逆頻度スコア付けでは、ドキュメント・セットで頻出する語句はノイズ語句とみなされるため、これらの語句のスコアは低くなります。ドキュメントのスコアを高くするには、問合せ語句がそのドキュメント内では頻出し、ドキュメント・セット全体としてはほとんど出現しないことが必要です。

次の表は、Oracle の逆頻度スコア付けを示しています。最初の列はドキュメント・セット内のドキュメントの数を示し、2 番目の列はスコアが 100 になるために必要な、ドキュメント内の問合せ語句の出現回数を示しています。

この表では、ドキュメント・セット内の 1 つのドキュメントのみが問合せ語句を含んでいると仮定します。

ドキュメント・セット内のドキュメント数	スコアが 100 になるために必要なドキュメント内の問合せ語句の出現回数
1	34
5	20
10	17
50	13
100	12
500	10
1,000	9
10,000	7
100,000	5
1,000,000	4

表では、ドキュメント・セット内に 5 つのドキュメントがあり、そのうち 1 つのドキュメントのみが問合せ語句を含んでいる場合、スコアが 100 になるには問合せ語句がドキュメントに 20 回出現する必要があることを示しています。ただし、ドキュメント・セットにドキュメントが 1,000,000 ある場合、スコアが 100 になるには、問合せ語句は 4 回のみ出現すれば十分です。

例

化学に関するドキュメントが 5000 あり、語句 *chemical* がすべてのドキュメントに 1 回以上は出現するとします。この場合、ドキュメント・セットに対して、語句 *chemical* は頻出用語となります。

chemical が 5 回出現し、語句 *hydrogen* が 5 回出現するドキュメントが 1 つあるとします。その他のドキュメントには、語句 *hydrogen* は含まれていません。この場合、ドキュメント・セットに対して、語句 hydrogen は頻出用語になりません。

ドキュメント・セットで *chemical* は頻繁に出現するため、この問合せ語句のドキュメントに対するスコアは、ドキュメント・セット全体ではほとんど出現しない *hydrogen* と比べて低くなります。このため、*hydrogen* に対するスコアは、*chemical* に対するスコアより高くなります。両方の語句がそのドキュメントに 5 回ずつ出現する場合でも、結果は同じです。

注意： ドキュメントで語句 *hydrogen* が 4 回出現し、語句 *chemical* が 5 回出現する場合でも、*hydrogen* のスコアが高くなります。ドキュメント・セットでは、*chemical* が頻出（5000 回以上）するためです。

また、逆頻度スコア付けでも、*hydrogen* を含むドキュメントの追加によってドキュメントでのその語句のスコアが下がり、*hydrogen* を含まないドキュメントの追加によってスコアが上がります。

DML およびスコア付け

スコア付けのアルゴリズムはドキュメント・セット内のドキュメント数に基づいているため、ドキュメント・セット内でのドキュメントの挿入、更新または削除によって、DML の前後に指定した任意の語句のスコアが変更されます。

DML が重い場合、開発者または Oracle 管理者は索引を最適化する必要があります。索引を最適化した直後に問合せを実行すると、完全に関連ランク付けされます。

DML が軽い場合でも、非常に正確な関連ランク付けが行われます。

どちらの場合も、開発者または Oracle 管理者は、CTX_DDL.SYNC_INDEX を使用して索引を同期化する必要があります。

G

ビュー

この付録では、Oracle Text が提供するすべてのビューについて説明します。システムが提供するビューは次のとおりです。

- CTX_CLASSES
- CTX_INDEXES
- CTX_INDEX_ERRORS
- CTX_INDEX_OBJECTS
- CTX_INDEX_PARTITIONS
- CTX_INDEX_SETS
- CTX_INDEX_SET_INDEXES
- CTX_INDEX_SUB_LEXERS
- CTX_INDEX_SUB_LEXER_VALUES
- CTX_INDEX_VALUES
- CTX_OBJECTS
- CTX_OBJECT_ATTRIBUTES
- CTX_OBJECT_ATTRIBUTE_LOV
- CTX_PARAMETERS
- CTX_PENDING
- CTX_PREFERENCES
- CTX_PREFERENCE_VALUES
- CTX_SECTIONS
- CTX_SECTION_GROUPS

- CTX_SQES
- CTX_STOPLISTS
- CTX_STOPWORDS
- CTX_SUB_LEXERS
- CTX_THESAURI
- CTX_THES_PHRASES
- CTX_USER_INDEXES
- CTX_USER_INDEX_ERRORS
- CTX_USER_INDEX_OBJECTS
- CTX_USER_INDEX_PARTITIONS
- CTX_USER_INDEX_SETS
- CTX_USER_INDEX_SET_INDEXES
- CTX_USER_INDEX_SUB_LEXERS
- CTX_USER_INDEX_SUB_LEXER_VALS
- CTX_USER_INDEX_VALUES
- CTX_USER_PENDING
- CTX_USER_PREFERENCES
- CTX_USER_PREFERENCE_VALUES
- CTX_USER_SECTIONS
- CTX_USER_SECTION_GROUPS
- CTX_USER_SQES
- CTX_USER_STOPLISTS
- CTX_USER_STOPWORDS
- CTX_USER_SUB_LEXERS
- CTX_USER_THESAURI
- CTX_USER_THES_PHRASES
- CTX_VERSION

CTX_CLASSES

このビューは、テキスト・データ・ディクショナリに登録されているすべてのプリファレンス・カテゴリを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
CLA_NAME	VARCHAR2(30)	クラス名
CLA_DESCRIPTION	VARCHAR2(80)	クラスの説明

CTX_INDEXES

このビューは、カレント・ユーザーのテキスト・データ・ディクショナリに登録されているすべての索引を表示します。CTXSYS で、このビューを問い合せることができます。

列名	型	説明
IDX_ID	NUMBER	内部索引 ID
IDX_OWNER	VARCHAR2(30)	索引の所有者
IDX_NAME	VARCHAR2(30)	索引名
IDX_TABLE_OWNER	VARCHAR2(30)	表の所有者
IDX_TABLE	VARCHAR2(30)	表名
IDX_KEY_NAME	VARCHAR2(256)	主キー列
IDX_TEXT_NAME	VARCHAR2(30)	テキスト列名
IDX_DOCID_COUNT	NUMBER	索引付けされたドキュメント数
IDX_STATUS	VARCHAR2(12)	ステータス
IDX_LANGUAGE_COLUMN	VARCHAR2(256)	元表の言語列名
IDX_FORMAT_COLUMN	VARCHAR2(256)	元表のフォーマット列名
IDX_CHARSET_COLUMN	VARCHAR2(256)	元表のキャラクタ・セット列名

CTX_INDEX_ERRORS

このビューは、DML エラーを表示し、CTXSYS で問い合せることができます。

列名	型	説明
ERR_INDEX_OWNER	VARCHAR2(30)	索引の所有者
ERR_INDEX_NAME	VARCHAR2(30)	索引名
ERR_TIMESTAMP	DATE	エラーの発生時間
ERR_TEXTKEY	VARCHAR2(18)	エラーが発生したドキュメントの ROWID、またはエラーが発生した操作の名前（ALTER INDEX など）
ERR_TEXT	VARCHAR2(4000)	エラー・テキスト

CTX_INDEX_OBJECTS

このビューは、索引中の各クラス用に使用されるオブジェクトを表示します。CTXSYS で、このビューを問い合せることができます。

列名	型	説明
IXO_INDEX_OWNER	VARCHAR2(30)	索引の所有者
IXO_INDEX_NAME	VARCHAR2(30)	索引名
IXO_CLASS	VARCHAR2(30)	クラス名
IXO_OBJECT	VARCHAR2(30)	オブジェクト名

CTX_INDEX_PARTITIONS

このビューは、すべての索引パーティションを表示します。CTXSYS で、このビューを問い合せることができます。

列名	型	説明
IXP_ID	NUMBER(38)	索引パーティション ID
IXP_INDEX_OWNER	VARCHAR2(30)	索引の所有者
IXP_INDEX_NAME	VARCHAR2(30)	索引名
IXP_INDEX_PARTITION_NAME	VARCHAR2(30)	索引パーティション名
IXP_TABLE_OWNER	VARCHAR2(30)	表の所有者
IXP_TABLE_NAME	VARCHAR2(30)	表名
IXP_TABLE_PARTITION_NAME	VARCHAR2(30)	表パーティション名
IXP_DOCID_COUNT	NUMBER(38)	パーティションに関連付けられたドキュメント数
IXP_STATUS	VARCHAR2(12)	パーティションのステータス

CTX_INDEX_SETS

このビューは、すべての索引セット名を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
IXS_OWNER	VARCHAR2(30)	索引セットの所有者
IXS_NAME	VARCHAR2(30)	索引セット名

CTX_INDEX_SET_INDEXES

このビューは、索引セット内のすべてのサブ索引を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
IXX_INDEX_SET_OWNER	VARCHAR2(30)	索引セットの所有者
IXX_INDEX_SET_NAME	VARCHAR2(30)	索引セット名
IXX_COLLIST	VARCHAR2(500)	サブ索引の列リスト
IXX_STORAGE	VARCHAR2(500)	サブ索引の記憶域句

CTX_INDEX_SUB_LEXERS

このビューは、索引ごとに各言語のサブレクサーを表示します。CTXSYS で、このビューを問い合せることができます。

列名	型	説明
ISL_INDEX_OWNER	VARCHAR2(30)	索引の所有者
ISL_INDEX_NAME	VARCHAR2(30)	索引名
ISL_LANGUAGE	VARCHAR2(30)	サブレクサーの言語
ISL_ALT_VALUE	VARCHAR2(30)	言語の代替値
ISL_OBJECT	VARCHAR2(30)	この言語で使用するレクサー・オブジェクト名

CTX_INDEX_SUB_LEXER_VALUES

このビューは、サブレクサー属性とその値を表示します。CTXSYS で、このビューにアクセスできます。

列名	型	説明
ISV_INDEX_OWNER	VARCHAR2(30)	索引の所有者
ISV_INDEX_NAME	VARCHAR2(30)	索引名
ISV_LANGUAGE	VARCHAR2(30)	サブレクサーの言語
ISV_OBJECT	VARCHAR2(30)	この言語で使用するレクサー・オブジェクト名
ISV_ATTRIBUTE	VARCHAR2(30)	サブレクサー属性の名前
ISV_VALUE	VARCHAR2(500)	サブレクサーの属性値

CTX_INDEX_VALUES

このビューは、索引で使用される各オブジェクトの属性値を表示します。CTXSYS で、このビューを問い合せることができます。

列名	型	説明
IXV_INDEX_OWNER	VARCHAR2(30)	索引の所有者
IXV_INDEX_NAME	VARCHAR2(30)	索引名
IXV_CLASS	VARCHAR2(30)	クラス名
IXV_OBJECT	VARCHAR2(30)	オブジェクト名
IXV_ATTRIBUTE	VARCHAR2(30)	属性名
IXV_VALUE	VARCHAR2(500)	属性値

CTX_OBJECTS

このビューは、テキスト・データ・ディクショナリに登録されているすべてのテキスト・オブジェクトを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
OBJ_CLASS	VARCHAR2(30)	オブジェクト・クラス（データストア、フィルタ、レクサーなど）
OBJ_NAME	VARCHAR2(30)	オブジェクト名
OBJ_DESCRIPTION	VARCHAR2(80)	オブジェクトの説明

CTX_OBJECT_ATTRIBUTES

このビューは、各オブジェクトのプリファレンスに割り当てられる属性を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
OAT_CLASS	VARCHAR2(30)	オブジェクト・クラス（データストア、フィルタ、レクサーなど）
OAT_OBJECT	VARCHAR2(30)	オブジェクト名
OAT_ATTRIBUTE	VARCHAR2(64)	属性名
OAT_DESCRIPTION	VARCHAR2(80)	属性の説明
OAT_REQUIRED	VARCHAR2(1)	要求される属性（Y または N）
OAT_STATIC	VARCHAR2(1)	現在は使用されていません。
OAT_DATATYPE	VARCHAR2(64)	属性データ型
OAT_DEFAULT	VARCHAR2(500)	属性のデフォルト値
OAT_MIN	NUMBER	最小値
OAT_MAX	NUMBER	最大値
OAT_MAX_LENGTH	NUMBER	最大長

CTX_OBJECT_ATTRIBUTE_LOV

このビューは、Oracle Text によって提供される特定のオブジェクト属性に対して使用可能な値を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
OAL_CLASS	NUMBER(38)	オブジェクトのクラス
OAL_OBJECT	VARCHAR2(30)	オブジェクト名
OAL_ATTRIBUTE	VARCHAR2(32)	属性名
OAl_LABEL	VARCHAR2(30)	属性値ラベル
OAL_VALUE	VARCHAR2(64)	属性値
OAL_DESCRIPTION	VARCHAR2(80)	属性値の説明

CTX_PARAMETERS

このビューは、CTXSYS によって定義されたすべてのシステム定義パラメータを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
PAR_NAME	VARCHAR2(30)	パラメータ名 ■ max_index_memory ■ ctx_doc_key_type ■ default_index_memory ■ default_datastore ■ default_filter_binary ■ default_filter_text ■ default_filter_file ■ default_section_html ■ default_section_xml ■ default_section_text ■ default_lexer ■ default_stoplist ■ default_storage ■ default_wordlist ■ default_ctxcat_lexer ■ default_ctxcat_index_set ■ default_ctxcat_stoplist ■ default_ctxcat_storage ■ default_ctxcat_wordlist ■ default_ctxrule_lexer ■ default_ctxrule_stoplist ■ default_ctxrule_storage ■ default_ctxrule_wordlist ■ log_directory ■ file_access_role
PAR_VALUE	VARCHAR2(500)	パラメータ値 max_index_memory および default_index_memory の場合、PAR_VALUE はメモリー量で構成される文字列を格納します。他のパラメータ名の場合、PAR_VALUE は索引作成用デフォルトとして使用されるプリファレンス名を格納します。

CTX_PENDING

このビューは、DML キュー内のユーザーの各エントリの行を表示します。CTXSYS で、このビューを問い合せることができます。

列名	型	説明
PND_INDEX_OWNER	VARCHAR2(30)	索引の所有者
PND_INDEX_NAME	VARCHAR2(30)	索引名
PND_PARTITION_NAME	VARCHAR2(30)	ローカル・パーティション索引のパーティション名。通常の索引の場合は NULL です。
PND_ROWID	ROWID	索引付けされる ROWID
PND_TIMESTAMP	DATE	変更の時間

CTX_PREFERENCES

このビューは、Oracle Text ユーザーによって作成されるプリファレンスおよび Oracle Text にあるすべてのシステム定義のプリファレンスを表示します。このビューは、各プリファレンスにつき 1 行を含みます。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
PRE_OWNER	VARCHAR2(30)	プリファレンス所有者のユーザー名
PRE_NAME	VARCHAR2(30)	プリファレンスの名前
PRE_CLASS	VARCHAR2(30)	プリファレンス・クラス
PRE_OBJECT	VARCHAR2(30)	使用されるオブジェクト

CTX_PREFERENCE_VALUES

このビューは、テキスト・データ・ディクショナリのすべてのプリファレンスに割り当てられた値を表示します。このビューは、各値につき 1 行を含みます。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
PRV_OWNER	VARCHAR2(30)	プリファレンス所有者のユーザー名
PRV_PREFERENCE	VARCHAR2(30)	プリファレンスの名前
PRV_ATTRIBUTE	VARCHAR2(64)	属性名
PRV_VALUE	VARCHAR2(500)	属性値

CTX_SECTIONS

このビューは、テキスト・データ・ディクショナリで作成されたすべてのセクションの情報を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SEC_OWNER	VARCHAR2(30)	セクション・グループの所有者
SEC_SECTION_GROUP	VARCHAR2(30)	セクション・グループ名
SEC_TYPE	VARCHAR2(30)	セクションのタイプ（ZONE、FIELD、SPECIAL、ATTR または STOP のいずれか）
SEC_ID	NUMBER	セクション ID
SEC_NAME	VARCHAR2(30)	セクション名
SEC_TAG	VARCHAR2(64)	セクション・タグ
SEC_VISIBLE	VARCHAR2(1)	Y または N（フィールド・セクションのみが参照できるインジケータ）

CTX_SECTION_GROUPS

このビューは、テキスト・データ・ディクショナリで作成されたすべてのセクション・グループの情報を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SGP_OWNER	VARCHAR2(30)	セクション・グループの所有者
SGP_NAME	VARCHAR2(30)	セクション・グループ名
SGP_TYPE	VARCHAR2(30)	セクション・グループのタイプ

CTX_SQES

このビューは、ユーザーが作成したすべての SQE の定義を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SQE_OWNER	VARCHAR2(30)	SQE の所有者
SQE_NAME	VARCHAR2(30)	SQE の名前
SQE_QUERY	VARCHAR2(2000)	問合せテキスト

CTX_STOPLISTS

このビューは、ストップリストを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SPL_OWNER	VARCHAR2(30)	ストップリストの所有者
SPL_NAME	VARCHAR2(30)	ストップリスト名
SPL_COUNT	NUMBER	ストップワード数
SPL_TYPE	VARCHAR2(30)	ストップリストのタイプ（MULTI または BASIC）

CTX_STOPWORDS

このビューは、各ストップリストのストップワードを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SPW_OWNER	VARCHAR2(30)	ストップリストの所有者
SPW_STOPLIST	VARCHAR2(30)	ストップリスト名
SPW_TYPE	VARCHAR2(10)	ストップ・タイプ（STOP_WORD、STOP_CLASS または STOP_THEME のいずれか）
SPW_WORD	VARCHAR2(80)	ストップワード
SPW_LANGUAGE	VARCHAR2(30)	ストップワードの言語

CTX_SUB_LEXERS

このビューは、マルチレクサーおよびそれに含まれるサブレクサー・プリファレンスの情報を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SLX_OWNER	VARCHAR2(30)	マルチレクサー・プリファレンスの所有者
SLX_NAME	VARCHAR2(30)	マルチレクサー・プリファレンスの名前
SLX_LANGUAGE	VARCHAR2(30)	参照されるレクサーの言語（略称ではなくフル・ネーム）
SLX_ALT_VALUE	VARCHAR2(30)	言語の代替値
SLX_SUB_OWNER	VARCHAR2(30)	サブレクサーの所有者
SLX_SUB_NAME	VARCHAR2(30)	サブレクサーの名前

CTX_THESAURI

このビューは、テキスト・データ・ディクショナリで作成されたすべてのシソーラスについての情報を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
THS_OWNER	VARCHAR2(30)	シソーラスの所有者
THS_NAME	VARCHAR2(30)	シソーラス名

CTX_THES_PHRASES

このビューは、テキスト・データ・ディクショナリにあるすべてのシソーラスの句に関する情報を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
THP_THESAURUS	VARCHAR2(30)	シソーラス名
THP_PHRASE	VARCHAR2(256)	シソーラスの句
THP_QUALIFIER	VARCHAR2(256)	シソーラスの修飾子
THP_SCOPE_NOTE	VARCHAR2(2000)	シソーラスのスコープ・ノート

CTX_USER_INDEXES

このビューは、カレント・ユーザーのテキスト・データ・ディクショナリに登録されているすべての索引を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
IDX_ID	NUMBER	内部索引 ID
IDX_NAME	VARCHAR2(30)	索引名
IDX_TYPE	VARCHAR2(30)	索引のタイプ（CONTEXT、CTXCAT または CTXRULE）
IDX_TABLE_OWNER	VARCHAR2(30)	表の所有者
IDX_TABLE	VARCHAR2(30)	表名
IDX_KEY_NAME	VARCHAR2(256)	主キー列
IDX_TEXT_NAME	VARCHAR2(30)	テキスト列名
IDX_DOCID_COUNT	NUMBER	索引付けされたドキュメント数
IDX_STATUS	VARCHAR2(12)	ステータス（INDEXED または INDEXING）
IDX_LANGUAGE_COLUMN	VARCHAR2(256)	元表の言語列名
IDX_FORMAT_COLUMN	VARCHAR2(256)	元表のフォーマット列名
IDX_CHARSET_COLUMN	VARCHAR2(256)	元表のキャラクタ・セット列名

CTX_USER_INDEX_ERRORS

このビューは、カレント・ユーザーの索引エラーを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
ERR_INDEX_NAME	VARCHAR2(30)	索引名
ERR_TIMESTAMP	DATE	エラーの発生時間
ERR_TEXTKEY	VARCHAR2(18)	エラーが発生したドキュメントのROWID、またはエラーが発生した操作の名前（ALTER INDEX など）
ERR_TEXT	VARCHAR2(4000)	エラー・テキスト

CTX_USER_INDEX_OBJECTS

このビューは、カレント・ユーザー用に定義された索引に連結されているプリファレンスを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
IXO_INDEX_NAME	VARCHAR2(30)	索引名
IXO_CLASS	VARCHAR2(30)	オブジェクト名
IXO_OBJECT	VARCHAR2(80)	オブジェクトの説明

CTX_USER_INDEX_PARTITIONS

このビューは、カレント・ユーザーのすべての索引パーティションを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
IXP_ID	NUMBER(38)	索引パーティション ID
IXP_INDEX_NAME	VARCHAR2(30)	索引名
IXP_INDEX_PARTITION_NAME	VARCHAR2(30)	索引パーティション名
IXP_TABLE_OWNER	VARCHAR2(30)	表の所有者

列名	型	説明
IXP_TABLE_NAME	VARCHAR2(30)	表名
IXP_TABLE_PARTITION_NAME	VARCHAR2(30)	表パーティション名
IXP_DOCID_COUNT	NUMBER(38)	索引パーティションに関連付けられたドキュメント数
IXP_STATUS	VARCHAR2(12)	パーティションのステータス

CTX_USER_INDEX_SETS

このビューは、カレント・ユーザーに属するすべての索引セット名を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
IXS_NAME	VARCHAR2(30)	索引セット名

CTX_USER_INDEX_SET_INDEXES

このビューは、カレント・ユーザーに属する索引セット内のすべての索引を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
IXX_INDEX_SET_NAME	VARCHAR2(30)	索引セット名
IXX_COLLIST	VARCHAR2(500)	索引の列リスト
IXX_STORAGE	VARCHAR2(500)	索引の記憶域句

CTX_USER_INDEX_SUB_LEXERS

このビューは、問合せを実行するユーザーに対して、索引ごとに各言語のサブレクサーを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
ISL_INDEX_NAME	VARCHAR2(30)	索引名
ISL_LANGUAGE	VARCHAR2(30)	サブレクサーの言語
ISL_ALT_VALUE	VARCHAR2(30)	言語の代替値
ISL_OBJECT	VARCHAR2(30)	この言語で使用するレクサー・オブジェクト名

CTX_USER_INDEX_SUB_LEXER_VALS

このビューは、問合せを実行するユーザーに対して、サブレクサー属性とその値を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
ISV_INDEX_NAME	VARCHAR2(30)	索引名
ISV_LANGUAGE	VARCHAR2(30)	サブレクサーの言語
ISV_OBJECT	VARCHAR2(30)	この言語で使用するレクサー・オブジェクト名
ISV_ATTRIBUTE	VARCHAR2(30)	サブレクサー属性の名前
ISV_VALUE	VARCHAR2(500)	サブレクサー属性の値

CTX_USER_INDEX_VALUES

このビューは、カレント・ユーザーの索引で使用される各オブジェクトの属性値を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
IXV_INDEX_NAME	VARCHAR2(30)	索引名
IXV_CLASS	VARCHAR2(30)	クラス名
IXV_OBJECT	VARCHAR2(30)	オブジェクト名
IXV_ATTRIBUTE	VARCHAR2(30)	属性名
IXV_VALUE	VARCHAR2(500)	属性値

CTX_USER_PENDING

このビューは、DML キュー内のユーザーの各エントリの行を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
PND_INDEX_NAME	VARCHAR2(30)	索引名
PND_PARTITION_NAME	VARCHAR2(30)	ローカル・パーティション索引のパーティション名。通常の索引の場合は NULL です。
PND_ROWID	ROWID	索引付けされる ROWID
PND_TIMESTAMP	DATE	変更時間

CTX_USER_PREFERENCES

このビューは、カレント・ユーザーによって定義されたすべてのプリファレンスを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
PRE_NAME	VARCHAR2(30)	プリファレンスの名前
PRE_CLASS	VARCHAR2(30)	プリファレンス・クラス
PRE_OBJECT	VARCHAR2(30)	使用されるオブジェクト

CTX_USER_PREFERENCE_VALUES

このビューは、カレント・ユーザーによって定義されたプリファレンスのすべての値を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
PRV_PREFERENCE	VARCHAR2(30)	プリファレンスの名前
PRV_ATTRIBUTE	VARCHAR2(64)	属性名
PRV_VALUE	VARCHAR2(500)	属性値

CTX_USER_SECTIONS

このビューは、カレント・ユーザーのテキスト・データ・ディクショナリに作成されたセクションに関する情報を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SEC__SECTION_GROUP	VARCHAR2(30)	セクション・グループ名
SEC_TYPE	VARCHAR2(30)	セクションのタイプ（ZONE、FIELD、SPECIAL、STOP または ATTR のいずれか）
SEC_ID	NUMBER	セクション ID
SEC_NAME	VARCHAR2(30)	セクション名
SEC_TAG	VARCHAR2(64)	セクション・タグ
SEC_VISIBLE	VARCHAR2(1)	Y または N（フィールド・セクションのみが参照できるインジケータ）

CTX_USER_SECTION_GROUPS

このビューは、カレント・ユーザーのテキスト・データ・ディクショナリに作成されたセクション・グループに関する情報を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SGP_NAME	VARCHAR2(30)	セクション・グループ名
SGP_TYPE	VARCHAR2(30)	セクション・グループのタイプ

CTX_USER_SQES

このビューは、カレント・ユーザーによって作成されたすべてのシステム SQE およびセッション SQE の定義を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SQE_OWNER	VARCHAR2(30)	SQE の所有者
SQE_NAME	VARCHAR2(30)	SQE の名前
SQE_QUERY	VARCHAR2(2000)	問合せテキスト

CTX_USER_STOPLISTS

このビューは、カレント・ユーザーのストップリストを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SPL_NAME	VARCHAR2(30)	ストップリスト名
SPL_COUNT	NUMBER	ストップワード数
SPL_TYPE	VARCHAR2(30)	ストップリストのタイプ（MULTI または BASIC）

CTX_USER_STOPWORDS

このビューは、カレント・ユーザーの各ストップリストのストップワードを表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SPW_STOPLIST	VARCHAR2(30)	ストップリスト名
SPW_TYPE	VARCHAR2(10)	ストップ・タイプ（STOP_WORD、STOP_CLASS または STOP_THEME のいずれか）
SPW_WORD	VARCHAR2(80)	ストップワード
SPW_LANGUAGE	VARCHAR2(30)	ストップワードの言語

CTX_USER_SUB_LEXERS

このビューは、カレント・ユーザーに対して、マルチレクサーおよびそれに含まれるサブレクサー・プリファレンスの情報を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
SLX_NAME	VARCHAR2(30)	マルチレクサー・プリファレンスの名前
SLX_LANGUAGE	VARCHAR2(30)	参照されるレクサーの言語（略称ではなくフル・ネーム）
SLX_ALT_VALUE	VARCHAR2(30)	言語の代替値
SLX_SUB_OWNER	VARCHAR2(30)	サブレクサーの所有者
SLX_SUB_NAME	VARCHAR2(30)	サブレクサーの名前

CTX_USER_THESAURI

このビューは、カレント・ユーザーによってシステムに作成されたすべてのシソーラスに関する情報を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
THS_NAME	VARCHAR2(30)	シソーラス名

CTX_USER_THES_PHRASES

このビューは、カレント・ユーザーが所有するすべてのシソーラス内の句に関する情報を表示します。すべてのユーザーが、このビューを問い合せることができます。

列名	型	説明
THP_THESAURUS	VARCHAR2(30)	シソーラス名
THP_PHRASE	VARCHAR2(256)	シソーラスの句
THP_QUALIFIER	VARCHAR2(256)	句の修飾子
THP_SCOPE_NOTE	VARCHAR2(2000)	句のスコープ・ノート

CTX_VERSION

このビューは、ctxsys データ・ディクショナリとコードのバージョン・ナンバー情報を表示します。

列名	型	説明
VER_DICT	CHAR(9)	CTXSYS データ・ディクショナリのバージョン・ナンバー
VER_CODE	VARCHAR2(9)	Oracle シャドウ・プロセスにリンクするコードのバージョン・ナンバー この列は、信頼できるコールアウトをコールし、リンクされているコードのバージョン・ナンバーをフェッチします。このため、パッチ・リリースの検出および確認には、この列を使用します。

H

ストップワード変換

この付録では、ストップワード変換について説明します。この付録の内容は次のとおりです。

- ストップワード変換の理解

ストップワード変換の理解

ストップワード単独、またはストップワードのみからなる句を問合せ演算子のオペランドとして使用すると、Oracle は式を書き換えてストップワードまたはストップワードのみからなる句を排除した後、問合せを実行します。

次の項では、それぞれの演算子に対するストップワードの書換え（変換）について説明します。すべての表のストップワード式の列では、問合せ式または問合せ式のコンポーネントを示します。また、右側の列では、Oracle による問合せの書換えを示します。

トークン *stopword* は、ストップワードまたはストップワードのみからなる句を表します。

トークン *non_stopword* は、ストップワード以外のワード、ストップワード以外のワードのみからなる句、ストップワード以外のワードとストップワードの両方で構成される句のいずれかを表します。

トークン *no_lex* は、ストップワードでもなく、索引が作成されるワードでもない、単一文字または文字列を表します。たとえば、単独の文字 + などです。

ストップワード式の列で問合せ式を示している場合、変換後の式 *no_token* は、この問合せを入力するとヒットが戻らないことを意味します。

ストップワード式の列で複数の演算子のある問合せ式のコンポーネントを示している場合、変換後の式 *no_token* は、*no_token* の値が変換後の式の次の段階に渡されることを意味します。

ストップワード式の列でオペランドとして *no_token* が指定されている変換は、*no_token* が前の変換の結果となる、中間の変換を示します。この中間の変換は、元の問合せ式に少なくとも 1 つのストップワードおよび複数の演算子がある場合に適用されます。

たとえば、次のコンパウンド問合せ式について考えます。

```
'(this NOT dog) AND cat'
```

この式でのストップワードが *this* のみの場合、Oracle は次の順に次の変換を適用します。

stopword NOT non-stopword => no_token

no_token AND non_stopword => non_stopword

結果式は次のとおりです。

```
'cat'
```

ワード変換

ストップワード式	変換後の式
stopword	no_token
no_lex	no_token

最初の変換は、問合せ式でのストップワードまたはストップワードのみからなる句自体はヒットを戻さないことを意味します。

2 番目の変換は、+ のように索引が作成されない問合せ語句は、ヒットを戻さないことを表しています。

AND 変換

ストップワード式	変換後の式
non_stopword AND *stopword*	non_stopword
non_stopword AND *no_token*	non_stopword
stopword AND *non_stopword*	non_stopword
no_token AND *non_stopword*	non_stopword
stopword AND *stopword*	no_token
no_token AND *stopword*	no_token
stopword AND *no_token*	no_token
no_token AND *no_token*	no_token

OR 変換

ストップワード式	変換後の式
non_stopword OR *stopword*	non_stopword
non_stopword OR *no_token*	non_stopword
stopword OR *non_stopword*	non_stopword
no_token OR *non_stopword*	non_stopword
stopword OR *stopword*	no_token
no_token OR *stopword*	no_token
stopword OR *no_token*	no_token
no_token OR *no_token*	no_token

ACCUMulate 変換

ストップワード式	変換後の式
non_stopword ACCUM *stopword*	non_stopword
non_stopword ACCUM *no_token*	non_stopword
stopword ACCUM *non_stopword*	non_stopword
no_token ACCUM *non_stopword*	non_stopword
stopword ACCUM *stopword*	no_token
no_token ACCUM *stopword*	no_token
stopword ACCUM *no_token*	no_token
no_token ACCUM *no_token*	no_token

MINUS 変換

ストップワード式	変換後の式
non_stopword MINUS *stopword*	non_stopword
non_stopword MINUS *no_token*	non_stopword
stopword MINUS *non_stopword*	no_token
no_token MINUS *non_stopword*	no_token
stopword MINUS *stopword*	no_token
no_token MINUS *stopword*	no_token
stopword MINUS *no_token*	no_token
no_token MINUS *no_token*	no_token

NOT 変換

ストップワード式	変換後の式
non_stopword NOT *stopword*	non_stopword
non_stopword NOT *no_token*	non_stopword
stopword NOT *non_stopword*	no_token
no_token NOT *non_stopword*	no_token
stopword NOT *stopword*	no_token
no_token NOT *stopword*	no_token
stopword NOT *no_token*	no_token
no_token NOT *no_token*	no_token

EQUIValence 変換

ストップワード式	変換後の式
non_stopword EQUIV *stopword*	non_stopword
non_stopword EQUIV *no_token*	non_stopword
stopword EQUIV *non_stopword*	non_stopword
no_token EQUIV *non_stopword*	non_stopword
stopword EQUIV *stopword*	no_token
no_token EQUIV *stopword*	no_token
stopword EQUIV *no_token*	no_token
no_token EQUIV *no_token*	no_token

注意： 問合せ実行計画を使用する場合、すべての EQUIVALENCE 変換が EXPLAIN 表に表示されるわけではありません。

NEAR 変換

ストップワード式	変換後の式
non_stopword NEAR *stopword*	non_stopword
non_stopword NEAR *no_token*	non_stopword
stopword NEAR *non_stopword*	non_stopword
no_token NEAR *non_stopword*	non_stopword
stopword NEAR *stopword*	no_token
no_token NEAR *stopword*	no_token
stopword NEAR *no_token*	no_token
no_token NEAR *no_token*	no_token

WEIGHT 変換

ストップワード式	変換後の式
stopword * n	no_token
no_token * n	no_token

THRESHOLD 変換

ストップワード式	変換後の式
stopword > n	no_token
no_token > n	no_token

WITHIN 変換

ストップワード式	変換後の式
stopword WITHIN *section*	no_token
no_token WITHIN *section*	no_token

英語のナレッジ・ベースのカテゴリ階層

この付録では、カテゴリとなるナレッジ・ベースのすべての概念をリストで示します。

この付録は 6 つの項で構成され、それぞれの項がナレッジ・ベースの 6 つの主なブランチに対応しています。

- ブランチ 1: science and technology
- ブランチ 2: business and economics
- ブランチ 3: government and military
- ブランチ 4: social environment
- ブランチ 5: geography
- ブランチ 6: abstract ideas and concepts

カテゴリは逆ツリー階層で表示され、サブカテゴリは各カテゴリ内でアルファベット順に表示されます。

注意： この付録には、ナレッジ・ベースにあるすべての概念が含まれているわけではありません。この付録に含まれているのは、カテゴリとなる概念（階層の親ノード）のみです。

ブランチ 1: science and technology

[1] communications

[2] journalism
    [3] broadcast journalism
    [3] photojournalism
    [3] print journalism
        [4] newspapers
[2] public speaking
[2] publishing industry
    [3] desktop publishing
    [3] periodicals
        [4] business publications
    [3] printing
[2] telecommunications industry
    [3] computer networking
        [4] Internet technology
            [5] Internet providers
            [5] Web browsers
            [5] search engines
    [3] data transmission
    [3] fiber optics
    [3] telephone service

[1] formal education

[2] colleges and universities
    [3] academic degrees
    [3] business education
[2] curricula and methods
[2] library science
[2] reference books
[2] schools
[2] teachers and students

[1] hard sciences

[2] aerospace industry
    [3] satellite technology
    [3] space exploration
        [4] Mars exploration
        [4] lunar exploration
        [4] space explorers
        [4] spacecraft and space stations
[2] chemical industry
    [3] chemical adhesives
    [3] chemical dyes
    [3] chemical engineering
    [3] materials technology
        [4] industrial ceramics
        [4] metal industry
            [5] aluminum industry
            [5] metallurgy
            [5] steel industry
        [4] plastics
        [4] rubber
        [4] synthetic textiles
    [3] paints and finishing materials
    [3] pesticides
        [4] fungicides
        [4] herbicides
[2] chemistry
    [3] chemical properties
    [3] chemical reactions
    [3] chemicals
        [4] chemical acids
        [4] chemical elements
        [4] molecular reactivity
        [4] molecular structure
    [3] chemistry tools
        [4] chemical analysis
        [4] chemistry glassware
        [4] purification and isolation of chemicals
    [3] organic chemistry
    [3] theory and physics of chemistry
[2] civil engineering
    [3] building architecture
    [3] construction industry
        [4] building components
            [5] exterior structures
                [6] entryways and extensions
                [6] landscaping
                [6] ornamental architecture
                [6] roofs and towers
                [6] walls
                [6] windows

[5] interior structures
[6] building foundations
[6] building systems
[7] electrical systems
[7] fireproofing and insulation
[7] plumbing
[6] rooms
[4] buildings and dwellings
[5] outbuildings
[4] carpentry
[4] construction equipment
[4] construction materials
[5] paneling and composites
[5] surfaces and finishing

[2] computer industry
[3] computer hardware industry
[4] computer components
[5] computer memory
[5] microprocessors
[4] computer peripherals
[5] data storage devices
[4] hand-held computers
[4] laptop computers
[4] mainframes
[4] personal computers
[4] workstations
[3] computer science
[4] artificial intelligence
[3] computer security and data encryption
[4] computer viruses and protection
[3] computer software industry
[4] CAD-CAM
[4] client-server software
[4] computer programming
[5] programming development tools
[5] programming languages
[4] operating systems
[3] computer standards
[3] cyberculture
[3] human-computer interaction
[3] information technology
[4] computer multimedia
[5] computer graphics
[5] computer sound
[5] computer video
[4] databases
[4] document management
[4] natural language processing
[4] spreadsheets
[3] network computing
[3] supercomputing and parallel computing
[3] virtual reality

[2] electrical engineering

[2] electronics
[3] consumer electronics
[4] audio electronics
[4] video electronics
[3] electronic circuits and components
[4] microelectronics
[4] semiconductors and superconductors
[3] radar technology

[2] energy industry
[3] electric power industry
[3] energy sources
[4] alternative energy sources
[4] fossil fuels industry
[5] coal industry
[5] petroleum products industry
[4] nuclear power industry

[2] environment control industries
[3] heating and cooling systems
[3] pest control
[3] waste management

[2] explosives and firearms
[3] chemical explosives
[3] firearm parts and accessories
[3] recreational firearms

[2] geology
[3] geologic formations
[3] geologic substances
[4] mineralogy
[5] gemstones
[5] igneous rocks
[5] metamorphic rocks
[5] sedimentary rocks

```
      [3] hydrology
      [3] meteorology
          [4] atmospheric science
          [4] clouds
          [4] storms
          [4] weather modification
          [4] weather phenomena
          [4] winds
      [3] mining industry
      [3] natural disasters
      [3] oceanography
      [3] seismology
      [3] speleology
      [3] vulcanology
  [2] inventions
  [2] life sciences
      [3] biology
          [4] biochemistry
              [5] biological compounds
                  [6] amino acids
                  [6] enzymes
                  [6] hormones
                      [7] androgens and anabolic
steroids
                      [7] blood sugar hormones
                      [7] corticosteroids
                      [7] estrogens and progestins
                      [7] gonadotropins
                      [7] pituitary hormones
                      [7] thyroid hormones
                  [6] lipids and fatty acids
                  [6] nucleic acids
                  [6] sugars and carbohydrates
                  [6] toxins
                  [6] vitamins
              [5] cell reproduction
              [5] cell structure and function
              [5] molecular genetics
          [4] botany
              [5] algae
              [5] fungi
              [5] plant diseases
              [5] plant kingdom
                  [6] ferns
                  [6] flowering plants
                      [7] cacti
                      [7] grasses
                  [6] mosses
                  [6] trees and shrubs
                      [7] conifers
                      [7] deciduous trees
                      [7] palm trees
              [5] plant physiology
                  [6] plant development
                  [6] plant parts
          [4] lower life forms
              [5] bacteria
              [5] viruses
          [4] paleontology
              [5] dinosaurs
          [4] physiology
              [5] anatomy
                  [6] cardiovascular systems
                  [6] digestive systems
                  [6] extremities and appendages
                  [6] glandular systems
                  [6] head and neck
                      [7] ear anatomy
                      [7] eye anatomy
                      [7] mouth and teeth
                  [6] immune systems
                      [7] antigens and antibodies
                  [6] lymphatic systems
                  [6] muscular systems
                  [6] nervous systems
                  [6] reproductive systems
                  [6] respiratory systems
                  [6] skeletal systems
                  [6] tissue systems
                  [6] torso
                  [6] urinary systems
              [5] reproduction and development
          [4] populations and vivisystems
              [5] biological evolution
              [5] ecology
                  [6] ecological conservation
                  [6] environmental pollution
              [5] genetics and heredity
```

```
        [4] zoology
            [5] invertebrates
                [6] aquatic sponges
                [6] arthropods
                    [7] arachnids
                        [8] mites and ticks
                        [8] scorpions
                        [8] spiders
                    [7] crustaceans
                    [7] insects
                [6] coral and sea anemones
                [6] jellyfish
                [6] mollusks
                    [7] clams, oysters, and
mussels
                    [7] octopi and squids
                    [7] snails and slugs
                [6] starfish and sea urchins
                [6] worms
            [5] vertebrates
                [6] amphibians
                [6] birds
                    [7] birds of prey
                        [8] owls
                    [7] game birds
                    [7] hummingbirds
                    [7] jays, crows, and magpies
                    [7] parrots and parakeets
                    [7] penguins
                    [7] pigeons and doves
                    [7] warblers and sparrows
                    [7] water birds
                        [8] ducks, geese, and
swans
                        [8] gulls and terns
                        [8] pelicans
                    [7] woodpeckers
                    [7] wrens
                [6] fish
                    [7] boneless fish
                        [8] rays and skates
                        [8] sharks
                    [7] bony fish
                        [8] deep sea fish
                        [8] eels
                        [8] tropical fish
                    [7] jawless fish
                [6] mammals
                    [7] anteaters and sloths
                        [8] aardvarks
                    [7] carnivores
                        [8] canines
                        [8] felines
                    [7] chiropterans
                    [7] elephants
                    [7] hoofed mammals
                        [8] cattle
                        [8] goats
                        [8] horses
                        [8] pigs
                        [8] sheep
                    [7] hyraxes
                    [7] marine mammals
                        [8] seals and walruses
                            [9] manatees
                        [8] whales and porpoises
                    [7] marsupials
                    [7] monotremes
                    [7] primates
                        [8] lemurs
                    [7] rabbits
                    [7] rodents
                [6] reptiles
                    [7] crocodilians
                    [7] lizards
                    [7] snakes
                    [7] turtles
    [3] biotechnology
        [4] antibody technology
            [5] immunoassays
        [4] biometrics
            [5] voice recognition technology
        [4] genetic engineering
        [4] pharmaceutical industry
            [5] anesthetics
                [6] general anesthetics
                [6] local anesthetics
            [5] antagonists and antidotes
            [5] antibiotics, antimicrobials, and
                antiparasitics
                [6] anthelmintics
                [6] antibacterials
                    [7] antimalarials
                    [7] antituberculars and
```

```
antileprotics
                [6] antifungals
                [6] antivirals
                [6] local anti-infectives
            [5] antigout agents
            [5] autonomic nervous system drugs
                [6] neuromuscular blockers
                [6] skeletal muscle relaxants
            [5] blood drugs
            [5] cardiovascular drugs
                [6] antihypertensives
            [5] central nervous system drugs
                [6] analgesics and antipyretics
                [6] antianxiety agents
                [6] antidepressants
                [6] antipsychotics
                [6] narcotic and opioid analgesics
                [6] nonsteroidal anti-inflammatory
drugs
                [6] sedative-hypnotics
            [5] chemotherapeutics, antineoplastic
agents
            [5] dermatomucosal agents
                [6] topical corticosteroids
            [5] digestive system drugs
                [6] antacids, adsorbents, and
                      antiflatulents
                [6] antidiarrheals
                [6] antiemetics
                [6] antiulcer agents
                [6] digestants
                [6] laxatives
            [5] eye, ear, nose, and throat drugs
                [6] nasal agents
                [6] ophthalmics
                    [7] ophthalmic
vasoconstrictors
                [6] otics, ear care drugs
            [5] fluid and electrolyte balance
drugs
                [6] diuretics
            [5] hormonal agents
            [5] immune system drugs
                [6] antitoxins and antivenins
                [6] biological response modifiers
                [6] immune serums
                [6] immunosuppressants
                [6] vaccines and toxoids
            [5] oxytocics
            [5] respiratory drugs
                [6] antihistamines
                [6] bronchodilators
                [6] expectorants and antitussives
            [5] spasmolytics
            [5] topical agents
        [3] health and medicine
            [4] healthcare industry
                [5] healthcare providers and practices
                [5] medical disciplines and
specialties
                    [6] cardiology
                    [6] dentistry
                    [6] dermatology
                    [6] geriatrics
                    [6] neurology
                    [6] obstetrics and gynecology
                    [6] oncology
                    [6] ophthalmology
                    [6] pediatrics
                [5] medical equipment
                    [6] artificial limbs and organs
                    [6] dressings and supports
                [5] medical equipment manufacturers
                [5] medical facilities
            [4] medical problems
                [5] blood disorders
                [5] cancers and tumors
                    [6] carcinogens
                [5] cardiovascular disorders
                [5] developmental disorders
                [5] environment-related afflictions
                [5] gastrointestinal disorders
                [5] genetic and hereditary disorders
                [5] infectious diseases
                    [6] communicable diseases
                        [7] sexually transmitted
diseases
                [5] injuries
                [5] medical disabilities
                [5] neurological disorders
                [5] respiratory disorders
                [5] skin conditions
            [4] nutrition
            [4] practice of medicine
                [5] alternative medicine
                [5] medical diagnosis
                    [6] medical imaging
                [5] medical personnel
```

[5] medical procedures
[6] physical therapy
[6] surgical procedures
[7] cosmetic surgery
[4] veterinary medicine
[2] machinery
[3] machine components
[2] mathematics
[3] algebra
[4] linear algebra
[4] modern algebra
[3] arithmetic
[4] elementary algebra
[3] calculus
[3] geometry
[4] mathematical topology
[4] plane geometry
[4] trigonometry
[3] math tools
[3] mathematical analysis
[3] mathematical foundations
[4] number theory
[4] set theory
[4] symbolic logic
[3] statistics
[2] mechanical engineering
[2] physics
[3] acoustics
[3] cosmology
[4] astronomy
[5] celestial bodies
[6] celestial stars
[6] comets
[6] constellations
[6] galaxies
[6] moons
[6] nebulae
[6] planets
[5] celestial phenomena
[3] electricity and magnetism
[3] motion physics
[3] nuclear physics
[4] subatomic particles
[3] optical technology
[4] holography
[4] laser technology
[5] high-energy lasers
[5] low-energy lasers
[3] thermodynamics
[2] robotics
[2] textiles
[2] tools and hardware
[3] cements and glues
[3] hand and power tools
[4] chisels
[4] drills and bits
[4] gauges and calipers
[4] hammers
[4] machine tools
[4] planes and sanders
[4] pliers and clamps
[4] screwdrivers
[4] shovels
[4] trowels
[4] wrenches
[3] knots

[1] social sciences

[2] anthropology
[3] cultural identities
[4] Native Americans
[3] cultural studies
[4] ancient cultures
[3] customs and practices
[2] archeology
[3] ages and periods
[3] prehistoric humanoids
[2] history
[3] U.S. history
[4] slavery in the U.S.
[3] ancient Rome
[4] Roman emperors
[3] ancient history
[3] biographies
[3] historical eras
[2] human sexuality
[3] homosexuality
[3] pornography
[3] prostitution
[3] sexual issues
[2] linguistics
[3] descriptive linguistics
[4] grammar
[5] parts of speech
[4] phonetics and phonology
[3] historical linguistics
[3] languages

```
    [3] linguistic theories
    [3] rhetoric and figures of speech
    [3] sociolinguistics
        [4] dialects and accents
    [3] writing and mechanics
        [4] punctuation and diacritics
        [4] writing systems
[2] psychology
    [3] abnormal psychology
        [4] anxiety disorders
        [4] childhood onset disorders
        [4] cognitive disorders
        [4] dissociative disorders
        [4] eating disorders
        [4] impulse control disorders
        [4] mood disorders
        [4] personality disorders
        [4] phobias
        [4] psychosomatic disorders
        [4] psychotic disorders
        [4] somatoform disorders
        [4] substance related disorders
    [3] behaviorist psychology
    [3] cognitive psychology
    [3] developmental psychology
    [3] experimental psychology
    [3] humanistic psychology
    [3] neuropsychology
    [3] perceptual psychology
    [3] psychiatry
    [3] psychoanalytic psychology
    [3] psychological states and behaviors
    [3] psychological therapy
    [3] psychological tools and techniques
    [3] sleep psychology
        [4] sleep disorders
[2] sociology
    [3] demographics
    [3] social identities
        [4] gender studies
        [4] senior citizens
    [3] social movements and institutions
    [3] social structures
```

[1] transportation

```
[2] aviation
    [3] aircraft
    [3] airlines
    [3] airports
    [3] avionics
[2] freight and shipping
    [3] package delivery industry
    [3] trucking industry
[2] ground transportation
    [3] animal powered transportation
    [3] automotive industry
        [4] automobiles
        [4] automotive engineering
            [5] automotive parts
            [5] internal combustion engines
        [4] automotive sales
        [4] automotive service and repair
        [4] car rentals
        [4] motorcycles
        [4] trucks and buses
    [3] human powered vehicles
    [3] rail transportation
        [4] subways
        [4] trains
    [3] roadways and driving
[2] marine transportation
    [3] boats and ships
    [3] seamanship
    [3] waterways
[2] travel industry
    [3] hotels and lodging
    [3] tourism
        [4] cruise lines
        [4] places of interest
        [4] resorts and spas
```

ブランチ 2: business and economics

[1] business services industry

[1] commerce and trade

[2] electronic commerce
[2] general commerce
[2] international trade and finance
[2] mail-order industry
[2] retail trade industry
    [3] convenience stores
    [3] department stores
    [3] discount stores
    [3] supermarkets
[2] wholesale trade industry

[1] corporate business

[2] business enterprise
    [3] entrepreneurship
[2] business fundamentals
[2] consulting industry
[2] corporate finance
    [3] accountancy
[2] corporate management
[2] corporate practices
[2] diversified companies
[2] human resources
    [3] employment agencies
[2] office products
[2] quality control
    [3] customer support
[2] research and development
[2] sales and marketing
    [3] advertising industry

[1] economics

[1] financial institutions

[2] banking industry
[2] insurance industry
[2] real-estate industry

[1] financial investments

[2] commodities market
    [3] money
        [4] currency market
    [3] precious metals market
[2] general investment
[2] personal finance
    [3] retirement investments
[2] securities market
    [3] bond market
    [3] mutual funds
    [3] stock market

[1] financial lending

[2] credit cards

[1] industrial business

[2] industrial engineering
    [3] production methods
[2] industrialists and financiers
[2] manufacturing
    [3] industrial goods manufacturing

[1] public sector industry

[1] taxes and tariffs

[1] work force

[2] organized labor

ブランチ 3: government and military

[1] government

[2] county government
[2] forms and philosophies of government
[2] government actions
[2] government bodies and institutions
    [3] executive branch
        [4] U.S. presidents
        [4] executive cabinet
    [3] judiciary branch
        [4] Supreme Court
            [5] chief justices
    [3] legislative branch
        [4] house of representatives
        [4] senate
[2] government officials
    [3] royalty and aristocracy
    [3] statesmanship
[2] government programs
    [3] social programs
        [4] welfare
[2] international relations
    [3] Cold War
    [3] diplomacy
    [3] immigration
[2] law
    [3] business law
    [3] courts
    [3] crimes and offenses
        [4] controlled substances
            [5] substance abuse
        [4] criminals
        [4] organized crime
    [3] law enforcement
    [3] law firms
    [3] law systems
        [4] constitutional law
    [3] legal bodies
    [3] legal customs and formalities
    [3] legal judgments
    [3] legal proceedings
    [3] prisons and punishments
[2] municipal government
    [3] municipal infrastructure
    [3] urban areas
        [4] urban phenomena
        [4] urban structures
[2] politics
    [3] civil rights
    [3] elections and campaigns
    [3] political activities
    [3] political advocacy
        [4] animal rights
        [4] consumer advocacy
    [3] political parties
    [3] political principles and philosophies
        [4] utopias
    [3] political scandals
    [3] revolution and subversion
        [4] terrorism
[2] postal communications
[2] public facilities
[2] state government

[1] military

[2] air force
[2] armored clothing
[2] army
[2] cryptography
[2] military honors
[2] military intelligence
[2] military leaders
[2] military ranks
    [3] army, air force, and marine ranks
    [3] navy and coast guard ranks
[2] military wars
    [3] American Civil War
    [3] American Revolution
    [3] World War I
    [3] World War II
    [3] warfare
[2] military weaponry
    [3] bombs and mines
    [3] chemical and biological warfare
    [3] military aircraft
    [3] missiles, rockets, and torpedoes
    [3] nuclear weaponry
    [3] space-based weapons

[2] navy
    [3] warships
[2] service academies

ブランチ 4: social environment

[1] belief systems

[2] folklore
[2] mythology
    [3] Celtic mythology
    [3] Egyptian mythology
    [3] Greek mythology
    [3] Japanese mythology
    [3] Mesopotamian and Sumerian mythology
    [3] Norse and Germanic mythology
    [3] Roman mythology
    [3] South and Central American mythology
    [3] mythological beings
    [3] myths and legends
[2] paranormal phenomena
    [3] astrology
    [3] occult
    [3] superstitions
[2] philosophy
    [3] epistemology
    [3] ethics and aesthetics
    [3] metaphysics
    [3] philosophical logic
    [3] schools of philosophy
[2] religion
    [3] God and divinity
    [3] doctrines and practices
    [3] history of religion
    [3] religious institutions and structures
    [3] sacred texts and objects
        [4] Bible
        [4] liturgical garments
    [3] world religions
        [4] Christianity
            [5] Christian denominations
            [5] Christian heresies
            [5] Christian theology
            [5] Mormonism
            [5] Roman Catholicism
                [6] popes
                [6] religious orders
            [5] evangelism
            [5] protestant reformation
        [4] Islam
        [4] Judaism
        [4] eastern religions
            [5] Buddhism
            [5] Hinduism
                [6] Hindu deities

[1] clothing and appearance

[2] clothing
    [3] clothing accessories
        [4] belts
        [4] functional accessories
        [4] gloves
    [3] fabrics
        [4] laces
        [4] leather and fur
    [3] footwear
    [3] garment parts
        [4] garment fasteners
        [4] garment trim
    [3] headgear
        [4] hats
        [4] helmets
    [3] laundry
    [3] neckwear
    [3] outer garments
        [4] dresses
        [4] formalwear
        [4] jackets
        [4] pants
        [4] shirts
        [4] skirts
        [4] sporting wear
        [4] sweaters
    [3] sewing
    [3] undergarments
        [4] deshabille
        [4] hosiery
        [4] lingerie
        [4] men's underwear
[2] cosmetics
    [3] facial hair
    [3] hair styling
[2] fashion industry
    [3] supermodels

[2] grooming
    [3] grooming aids
[2] jewelry

[1] emergency services

[2] emergency dispatch
[2] emergency medical services
[2] fire prevention and suppression
[2] hazardous material control
[2] heavy rescue

[1] family

[2] death and burial
    [3] funeral industry
[2] divorce
[2] infancy
[2] kinship and ancestry
[2] marriage
[2] pregnancy
    [3] contraception
[2] upbringing

[1] food and agriculture

[2] agribusiness
[2] agricultural equipment
[2] agricultural technology
    [3] soil management
        [4] fertilizers
[2] aquaculture
[2] cereals
[2] condiments
[2] crop grain
[2] dairy products
    [3] cheeses
[2] drinking and dining
    [3] alcoholic beverages
        [4] beers
        [4] liqueurs
        [4] liquors
        [4] mixed drinks
        [4] wines
            [5] wineries
    [3] cooking
    [3] meals and dishes
        [4] sandwiches
    [3] non-alcoholic beverages
        [4] coffee
        [4] soft drinks
        [4] tea
[2] farming
[2] fats and oils
    [3] butter and margarine
[2] food and drink industry
    [3] foodservice industry
    [3] meat packing industry
[2] forestry
    [3] forest products
[2] fruits and vegetables
    [3] legumes
[2] leavening agents
[2] mariculture
[2] meats
    [3] beef
    [3] pate and sausages
    [3] pork
    [3] poultry
[2] nuts and seeds
[2] pasta
[2] prepared foods
    [3] breads
    [3] candies
    [3] crackers
    [3] desserts
        [4] cakes
        [4] cookies
        [4] pies
    [3] pastries
    [3] sauces
    [3] soups and stews
[2] ranching
[2] seafood
[2] spices and flavorings
    [3] sweeteners

[1] housekeeping and butlery

[1] housewares

[2] beds
[2] candles
[2] carpets and rugs
[2] cases, cabinets, and chests
[2] chairs and sofas
[2] curtains, drapes, and screens
[2] functional wares
    [3] cleaning supplies
[2] home appliances
[2] kitchenware
    [3] cookers
    [3] fine china
    [3] glassware
    [3] kitchen appliances
    [3] kitchen utensils
        [4] cutting utensils
    [3] pots and pans
    [3] serving containers
    [3] tableware
[2] lamps
[2] linen
[2] mirrors
[2] ornamental objects
[2] stationery
[2] stools and stands
[2] tables and desks
[2] timepieces

[1] leisure and recreation

[2] arts and entertainment
    [3] broadcast media
        [4] radio
            [5] amateur radio
        [4] television
    [3] cartoons, comic books, and superheroes
    [3] cinema
        [4] movie stars
        [4] movie tools and techniques
        [4] movies
    [3] entertainments and spectacles
        [4] entertainers
    [3] humor and satire
    [3] literature
        [4] children's literature
        [4] literary criticism
        [4] literary devices and techniques
        [4] poetry
            [5] classical poetry
        [4] prose
            [5] fiction
                [6] horror fiction
                [6] mystery fiction
        [4] styles and schools of literature
    [3] performing arts
        [4] dance
            [5] ballet
            [5] choreography
            [5] folk dances
            [5] modern dance
        [4] drama
            [5] dramatic structure
            [5] stagecraft
        [4] music
            [5] blues music
            [5] classical music
            [5] composition types
            [5] folk music
            [5] jazz music
            [5] music industry
            [5] musical instruments
                [6] keyboard instruments
                [6] percussion instruments
                [6] string instruments
                [6] wind instruments
                    [7] brass instruments
                    [7] woodwinds
            [5] opera and vocal
            [5] popular music and dance
            [5] world music
    [3] science fiction
    [3] visual arts
        [4] art galleries and museums
        [4] artistic painting
            [5] painting tools and techniques
            [5] styles and schools of art
        [4] graphic arts

```
        [4] photography
            [5] cameras
            [5] photographic lenses
            [5] photographic processes
            [5] photographic techniques
            [5] photographic tools
        [4] sculpture
            [5] sculpture tools and techniques
[2] crafts
[2] games
    [3] indoor games
        [4] board games
        [4] card games
        [4] video games
    [3] outdoor games
[2] gaming industry
    [3] gambling
[2] gardening
[2] hobbies
    [3] coin collecting
    [3] stamp collecting
[2] outdoor recreation
    [3] hunting and fishing
[2] pets
[2] restaurant industry
[2] sports
    [3] Olympics
    [3] aquatic sports
        [4] canoeing, kayaking, and rafting
        [4] swimming and diving
        [4] yachting
    [3] baseball
    [3] basketball
    [3] bicycling
    [3] bowling
    [3] boxing
    [3] equestrian events
        [4] horse racing
        [4] rodeo
    [3] fantasy sports
    [3] fitness and health
        [4] fitness equipment
    [3] football
    [3] golf
    [3] gymnastics
    [3] martial arts
    [3] motor sports
        [4] Formula I racing
        [4] Indy car racing
        [4] NASCAR racing
        [4] drag racing
        [4] motorcycle racing
        [4] off-road racing
    [3] soccer
    [3] sports equipment
    [3] tennis
    [3] track and field
    [3] winter sports
        [4] hockey
        [4] ice skating
        [4] skiing
[2] tobacco industry
[2] toys
```

ブランチ 5: geography

[1] cartography

```
[2] explorers
```

[1] physical geography

```
[2] bodies of water
    [3] lakes
    [3] oceans
    [3] rivers
[2] land forms
    [3] coastlands
    [3] continents
    [3] deserts
    [3] highlands
    [3] islands
    [3] lowlands
    [3] mountains
    [3] wetlands
```

[1] political geography

[2] Africa
- [3] Central Africa
  - [4] Angola
  - [4] Burundi
  - [4] Central African Republic
  - [4] Congo
  - [4] Gabon
  - [4] Kenya
  - [4] Malawi
  - [4] Rwanda
  - [4] Tanzania
  - [4] Uganda
  - [4] Zaire
  - [4] Zambia
- [3] North Africa
  - [4] Algeria
  - [4] Chad
  - [4] Djibouti
  - [4] Egypt
  - [4] Ethiopia
  - [4] Libya
  - [4] Morocco
  - [4] Somalia
  - [4] Sudan
  - [4] Tunisia
- [3] Southern Africa
  - [4] Botswana
  - [4] Lesotho
  - [4] Mozambique
  - [4] Namibia
  - [4] South Africa
  - [4] Swaziland
  - [4] Zimbabwe
- [3] West Africa
  - [4] Benin
  - [4] Burkina Faso
  - [4] Cameroon
  - [4] Equatorial Guinea
  - [4] Gambia
  - [4] Ghana
  - [4] Guinea
  - [4] Guinea-Bissau
  - [4] Ivory Coast
  - [4] Liberia
  - [4] Mali
  - [4] Mauritania
  - [4] Niger
  - [4] Nigeria
  - [4] Sao Tome and Principe
  - [4] Senegal
  - [4] Sierra Leone
  - [4] Togo

[2] Antarctica

[2] Arctic
- [3] Greenland
- [3] Iceland

[2] Asia
- [3] Central Asia
  - [4] Afghanistan
  - [4] Bangladesh
  - [4] Bhutan
  - [4] India
  - [4] Kazakhstan
  - [4] Kyrgyzstan
  - [4] Nepal
  - [4] Pakistan
  - [4] Tajikstan
  - [4] Turkmenistan
  - [4] Uzbekistan
- [3] East Asia
  - [4] China
  - [4] Hong Kong
  - [4] Japan
  - [4] Macao
  - [4] Mongolia
  - [4] North Korea
  - [4] South Korea
  - [4] Taiwan
- [3] Southeast Asia
  - [4] Brunei
  - [4] Cambodia
  - [4] Indonesia
  - [4] Laos
  - [4] Malaysia
  - [4] Myanmar
  - [4] Papua New Guinea
  - [4] Philippines
  - [4] Singapore
  - [4] Thailand
  - [4] Vietnam

[2] Atlantic area
- [3] Azores
- [3] Bermuda
- [3] Canary Islands
- [3] Cape Verde
- [3] Falkland Islands

[2] Caribbean
- [3] Antigua and Barbuda
- [3] Bahamas
- [3] Barbados
- [3] Cuba
- [3] Dominica
- [3] Dominican Republic
- [3] Grenada
- [3] Haiti
- [3] Jamaica
- [3] Netherlands Antilles
- [3] Puerto Rico
- [3] Trinidad and Tobago

[2] Central America
- [3] Belize
- [3] Costa Rica
- [3] El Salvador
- [3] Guatemala
- [3] Honduras
- [3] Nicaragua
- [3] Panama

[2] Europe
- [3] Eastern Europe
  - [4] Albania
  - [4] Armenia
  - [4] Azerbaijan
  - [4] Belarus
  - [4] Bulgaria
  - [4] Czech Republic
  - [4] Czechoslovakia
  - [4] Estonia
  - [4] Greece
  - [4] Hungary
  - [4] Latvia
  - [4] Lithuania
  - [4] Moldava
  - [4] Poland
  - [4] Republic of Georgia
  - [4] Romania
  - [4] Russia
    - [5] Siberia
  - [4] Slovakia
  - [4] Soviet Union
  - [4] Ukraine
  - [4] Yugoslavia
    - [5] Bosnia and Herzegovina
    - [5] Croatia
    - [5] Macedonia
    - [5] Montenegro
    - [5] Serbia
    - [5] Slovenia
- [3] Western Europe
  - [4] Austria
  - [4] Belgium
  - [4] Denmark
  - [4] Faeroe Island
  - [4] Finland
  - [4] France
  - [4] Germany
  - [4] Iberia
    - [5] Andorra
    - [5] Portugal
    - [5] Spain
  - [4] Ireland
  - [4] Italy
  - [4] Liechtenstein
  - [4] Luxembourg
  - [4] Monaco
  - [4] Netherlands
  - [4] Norway
  - [4] San Marino
  - [4] Sweden
  - [4] Switzerland
  - [4] United Kingdom
    - [5] England
    - [5] Northern Ireland
    - [5] Scotland
    - [5] Wales

[2] Indian Ocean area
- [3] Comoros
- [3] Madagascar
- [3] Maldives
- [3] Mauritius
- [3] Seychelles
- [3] Sri Lanka

[2] Mediterranean
- [3] Corsica
- [3] Cyprus
- [3] Malta
- [3] Sardinia

[2] Middle East
- [3] Bahrain
- [3] Iran
- [3] Iraq
- [3] Israel
- [3] Jordan
- [3] Kuwait
- [3] Lebanon
- [3] Oman
- [3] Palestine
- [3] Qatar
- [3] Saudi Arabia
- [3] Socotra
- [3] Syria
- [3] Turkey
- [3] United Arab Emirates
- [3] Yemen

[2] North America
- [3] Canada
- [3] Mexico
- [3] United States
  - [4] Alabama
  - [4] Alaska
  - [4] Arizona
  - [4] Arkansas
  - [4] California
  - [4] Colorado
  - [4] Delaware
  - [4] Florida
  - [4] Georgia
  - [4] Hawaii
  - [4] Idaho
  - [4] Illinois
  - [4] Indiana
  - [4] Iowa
  - [4] Kansas
  - [4] Kentucky
  - [4] Louisiana
  - [4] Maryland
  - [4] Michigan
  - [4] Minnesota
  - [4] Mississippi
  - [4] Missouri
  - [4] Montana
  - [4] Nebraska
  - [4] Nevada
  - [4] New England
    - [5] Connecticut
    - [5] Maine
    - [5] Massachusetts
    - [5] New Hampshire
    - [5] Rhode Island
    - [5] Vermont
  - [4] New Jersey
  - [4] New Mexico
  - [4] New York
  - [4] North Carolina
  - [4] North Dakota
  - [4] Ohio
  - [4] Oklahoma
  - [4] Oregon
  - [4] Pennsylvania
  - [4] South Carolina
  - [4] South Dakota
  - [4] Tennessee
  - [4] Texas
  - [4] Utah
  - [4] Virginia
  - [4] Washington
  - [4] Washington D.C.
  - [4] West Virginia
  - [4] Wisconsin
  - [4] Wyoming

[2] Pacific area
- [3] American Samoa
- [3] Australia
  - [4] Tasmania
- [3] Cook Islands
- [3] Fiji
- [3] French Polynesia
- [3] Guam
- [3] Kiribati
- [3] Mariana Islands
- [3] Marshall Islands
- [3] Micronesia
- [3] Nauru
- [3] New Caledonia
- [3] New Zealand
- [3] Palau
- [3] Solomon Islands
- [3] Tonga
- [3] Tuvalu
- [3] Vanuatu
- [3] Western Samoa

```
[2] South America
    [3] Argentina
    [3] Bolivia
    [3] Brazil
    [3] Chile
    [3] Colombia
    [3] Ecuador
    [3] French Guiana
    [3] Guyana
    [3] Paraguay
    [3] Peru
    [3] Suriname
    [3] Uruguay
    [3] Venezuela
```

ブランチ 6: abstract ideas and concepts

[1] dynamic relations

```
[2] activity
    [3] attempts
        [4] achievement
        [4] difficulty
        [4] ease
        [4] extemporaneousness
        [4] failure
        [4] preparation
        [4] success
    [3] inertia
    [3] motion
        [4] agitation
        [4] directional movement
            [5] ascent
            [5] convergence
            [5] departure
            [5] descent
            [5] divergence
            [5] entrance
            [5] inward motion
            [5] jumps
            [5] motions around
            [5] outward motion
            [5] progression
            [5] withdrawal
        [4] forceful motions
            [5] friction
            [5] pulls
            [5] pushes
            [5] throws
        [4] haste
        [4] slowness
        [4] transporting
    [3] rest
    [3] violence
[2] change
    [3] exchanges
    [3] gradual change
    [3] major change
    [3] reversion
[2] time
    [3] future
    [3] longevity
    [3] past
    [3] regularity of time
    [3] relative age
        [4] stages of development
    [3] simultaneity
    [3] time measurement
        [4] instants
    [3] timeliness
        [4] earliness
        [4] lateness
    [3] transience
```

[1] human life and activity

```
[2] communication
    [3] announcements
    [3] conversation
    [3] declarations
    [3] disclosure
    [3] identifiers
    [3] implication
    [3] obscene language
    [3] representation
        [4] interpretation
    [3] secrecy
    [3] shyness
    [3] speech
```

[3] styles of expression
[4] boasting
[4] clarity
[4] eloquence
[4] intelligibility
[4] nonsense
[4] plain speech
[4] wordiness
[2] feelings and sensations
[3] calmness
[3] composure
[3] emotions
[4] anger
[4] contentment
[4] courage
[4] cowardice
[4] happiness
[4] humiliation
[4] ill humor
[4] insolence
[4] nervousness
[4] pickiness
[4] regret
[4] relief
[4] sadness
[4] vanity
[3] excitement
[3] five senses
[4] audiences
[4] hearing
[5] faintness of sound
[5] loudness
[5] silence
[5] sound
[6] cries
[6] dissonant sound
[6] harmonious sound
[6] harsh sound
[6] repeated sounds
[4] sight
[5] appearance
[5] fading
[5] visibility
[4] smelling
[5] odors
[4] tasting
[5] flavor
[6] sweetness
[4] touching
[3] numbness
[3] pleasure
[3] suffering
[2] gender
[2] intellect
[3] cleverness
[3] foolishness
[3] ignorance
[3] intelligence and wisdom
[3] intuition
[3] knowledge
[3] learning
[3] teaching
[3] thinking
[4] conclusion
[5] discovery
[5] evidence
[5] rebuttal
[4] consideration
[5] analysis
[5] questioning
[5] tests
[4] faith
[5] ideology
[5] sanctimony
[4] judgment
[4] rationality
[4] skepticism
[4] sophistry
[4] speculation
[2] social attitude, custom
[3] behavior
[4] approval
[4] courtesy
[4] criticism
[4] cruelty
[4] flattery
[4] forgiveness
[4] friendliness
[4] generosity
[4] gratitude
[4] hatred
[4] jealousy
[4] kindness
[4] love
[5] adoration
[4] respect
[4] rudeness

```
    [4] ruthlessness
    [4] stinginess
    [4] sympathy
[3] morality and ethics
    [4] evil
    [4] goodness
    [4] moral action
        [5] asceticism
        [5] decency
        [5] deception
        [5] integrity
        [5] lewdness
        [5] self-indulgence
    [4] moral consequences
        [5] allegation
        [5] entitlement
        [5] excuses
        [5] punishment
        [5] reparation
    [4] moral states
        [5] fairness
        [5] guilt
        [5] innocence
        [5] partiality
    [4] responsibility
[3] reputation
    [4] acclaim
    [4] notoriety
[3] social activities
    [4] enjoyment
    [4] monotony
[3] social conventions
    [4] conventionalism
    [4] formality
    [4] trends
[3] social transactions
    [4] debt
    [4] offers
    [4] payments
    [4] petitions
    [4] promises and contracts
```

```
[2] states of mind
    [3] anticipation
        [4] fear
        [4] frustration
        [4] hopefulness
        [4] hopelessness
        [4] prediction
        [4] surprise
        [4] warnings
    [3] boredom
    [3] broad-mindedness
    [3] carelessness
    [3] caution
    [3] confusion
    [3] creativity
    [3] curiosity
    [3] forgetfulness
    [3] patience
    [3] prejudice
    [3] remembering
    [3] seriousness
[2] volition
    [3] assent
    [3] choices
        [4] denial
    [3] decidedness
    [3] dissent
    [3] eagerness
    [3] enticement
    [3] evasion
        [4] abandonment
        [4] escape
    [3] impulses
    [3] indecision
    [3] indifference
    [3] inevitability
    [3] motivation
    [3] obstinacy
    [3] tendency
```

[1] potential relations

[2] ability, power
- [3] competence, expertise
- [3] energy, vigor
- [3] ineptness
- [3] productivity
- [3] provision
- [3] strength
- [3] weakness

[2] conflict
- [3] attacks
- [3] competition
- [3] crises
- [3] retaliation

[2] control
- [3] anarchy
- [3] command
  - [4] cancelations
  - [4] delegation
  - [4] permission
  - [4] prohibiting
- [3] defiance
- [3] influence
- [3] leadership
- [3] modes of authority
  - [4] confinement
  - [4] constraint
  - [4] discipline
  - [4] freedom
  - [4] leniency
  - [4] liberation
- [3] obedience
- [3] regulation
- [3] servility

[2] possession
- [3] giving
- [3] keeping
- [3] losing
- [3] receiving
- [3] sharing
- [3] taking

[2] possibility
- [3] chance
- [3] falseness
- [3] truth

[2] purpose
- [3] abuse
- [3] depletion
- [3] obsolescence

[2] support
- [3] cooperation
- [3] mediation
- [3] neutrality
- [3] peace
- [3] protection
- [3] sanctuary
- [3] security

[1] relation

[2] agreement

[2] cause and effect
- [3] causation
- [3] result

[2] difference

[2] examples

[2] relevance

[2] similarity
- [3] duplication

[2] uniformity

[2] variety

[1] static relations

[2] amounts
- [3] fewness
- [3] fragmentation
- [3] large quantities
- [3] majority
- [3] mass quantity
- [3] minority
- [3] numbers
- [3] quantity modification
  - [4] combination
  - [4] connection
  - [4] decrease
  - [4] increase
  - [4] remainders
  - [4] separation
- [3] required quantity
  - [4] deficiency
  - [4] excess
  - [4] sufficiency
- [3] wholeness
  - [4] omission
  - [4] thoroughness

[2] existence
- [3] creation
- [3] life

[2] form
- [3] defects
- [3] effervescence
- [3] physical qualities
  - [4] brightness and color
    - [5] color
      - [6] variegation
    - [5] colorlessness
    - [5] darkness
    - [5] lighting
      - [6] opaqueness
      - [6] transparency
  - [4] dryness
  - [4] fragility
  - [4] heaviness
  - [4] mass and weight measurement
  - [4] moisture
  - [4] pliancy
  - [4] rigidity
  - [4] softness
  - [4] temperature
    - [5] coldness
    - [5] heat
  - [4] texture
    - [5] fluids
    - [5] gaseousness
    - [5] jaggedness
    - [5] powderiness
    - [5] semiliquidity
    - [5] smoothness
  - [4] weightlessness
- [3] shape
  - [4] angularity
  - [4] circularity
  - [4] curvature
  - [4] roundness
  - [4] straightness
- [3] symmetry
- [3] tangibility
- [3] topological form
  - [4] concavity
  - [4] convexity
  - [4] covering
  - [4] folds
  - [4] openings

[2] nonexistence
- [3] death
- [3] destruction

[2] quality
- [3] badness
- [3] beauty
- [3] cleanness
- [3] complexity
- [3] correctness
- [3] deterioration
- [3] dirtiness
- [3] good quality
- [3] improvement
- [3] mediocrity
- [3] mistakes
- [3] normality
- [3] perfection
- [3] remedy
- [3] simplicity
- [3] stability
  - [4] resistance to change
- [3] strangeness
- [3] ugliness
- [3] value

[2] range
- [3] areas
  - [4] area measurement
  - [4] regions
  - [4] storage
  - [4] volume measurement
- [3] arrangement
  - [4] locations
    - [5] anteriors
    - [5] compass directions
    - [5] exteriors
    - [5] interiors
    - [5] left side
    - [5] posteriors
    - [5] right side
    - [5] topsides
    - [5] undersides

```
    [4] positions
        [5] disorder
        [5] groups
            [6] dispersion
            [6] exclusion
            [6] inclusion
            [6] itemization
            [6] seclusion
            [6] togetherness
        [5] hierarchical relationships
            [6] downgrades
            [6] ranks
            [6] upgrades
        [5] sequence
            [6] beginnings
            [6] continuation
            [6] ends
            [6] middles
            [6] preludes
[3] boundaries
[3] dimension
    [4] contraction
    [4] depth
    [4] expansion
    [4] flatness
    [4] height
    [4] largeness
    [4] length
    [4] linear measurement
    [4] narrowness
    [4] shallowness
    [4] shortness
    [4] slopes
    [4] smallness
    [4] steepness
    [4] thickness
[3] essence
[3] generalization
[3] nearness
[3] obstruction
[3] remoteness
[3] removal
[3] significance
[3] trivialness
[3] uniqueness
[3] ways and methods
```

索引

記号

! 演算子，3-40
$ 演算子，3-41
% ワイルド・カード，3-53
* 演算子，3-51
- 演算子，3-28
, 演算子，3-9
= 演算子，3-15
\> 演算子，3-45
? 演算子，3-16
_ ワイルド・カード，3-53
{} エスケープ文字，4-2

A

ABOUT 問合せ，3-5
　ハイライト・オフセット，8-10
　ハイライト・マークアップ，8-14
　表示の拡張，10-6
　例，3-7
ACCUMULATE 演算子，3-9
　スコア付け，3-9
　ストップワード変換，H-4
ADD_ATTR_SECTION プロシージャ，7-3
ADD_EVENT プロシージャ，9-2
ADD_FIELD_SECTION プロシージャ，7-5
ADD_SPECIAL_SECTION プロシージャ，7-11
ADD_STOP_SECTION プロシージャ，7-14
ADD_STOPCLASS プロシージャ，7-13
ADD_STOPTHEME プロシージャ，7-16
ADD_STOPWORD プロシージャ，7-17
ADD_SUB_LEXER プロシージャ，7-19
　例，2-44
ADD_ZONE_SECTION プロシージャ，7-22
ALTER INDEX，1-2
　REBUILD 構文，1-3
　RENAME 構文，1-2
　例，1-9
ALTER TABLE，1-12
ALTER_PHRASE プロシージャ，12-3
ALTER_THESAURUS プロシージャ，12-5
alternate_spelling 属性，2-43
AND 演算子，3-11
　ストップワード変換，H-3
AUTO_SECTION_GROUP オブジェクト，1-42，2-79，7-34
AUTO_SECTION_GROUP システム定義プリファレンス，2-86
AUTO_SECTION_GROUP の例，2-80
AUTO ステミング，2-69

B

base_letter 属性，2-40
BASIC_LEXER オブジェクト，2-35
　キャラクタ・セットのサポート，2-36
BASIC_LEXER 型
　例，2-43
BASIC_LEXER システム定義プリファレンス，2-86
BASIC_SECTION_GROUP オブジェクト，1-42，2-78，7-33
BASIC_STOPLIST 型，7-36
BASIC_STORAGE オブジェクト
　属性，2-76
　デフォルト，2-77
　例，2-77
BASIC_WORDLIST オブジェクト
　属性，2-69
　例，2-74

BFILE 列
索引付け，1-29
BINARY
フォーマット列の値，1-33
binary 属性，2-7，2-19
BLOB 列
索引付け，1-29
ロード例，C-2
BROADER TERM 演算子
例，3-12
BROWSE_WORDS プロシージャ，10-2
BTG 演算子，3-12
BTG ファンクション，12-10
BTI 演算子，3-12
BTI ファンクション，12-12
BTP 演算子，3-12
BTP ファンクション，12-14
BT 演算子，3-12
BT ファンクション，12-7

C

CATSEARCH 演算子，1-16
CHARSET_FILTER
属性，2-22
複合キャラクタ・セットの例，2-23
charset 属性，2-22
CHAR 列
索引付け，1-29
CHINESE_VGRAM_LEXER オブジェクト，2-45
CLOB 列
索引付け，1-29
command 属性，2-28
composite 属性
BASIC_LEXER，2-41
KOREAN_MORP_LEXER，2-50
CONTAINS 演算子
構文，1-23
例，1-25
CONTEXT 索引
概要，1-28
構文，1-29
デフォルトのパラメータ，2-89
CONTEXT 索引タイプ，1-28
continuation 属性，2-37
COUNT_HITS プロシージャ，10-5
CREATE INDEX，1-28
CONTEXT，1-29
CTXCAT，1-38
CTXRULE，1-41
CTXXPATH，1-42
デフォルトのパラメータ，2-89
CREATE_INDEX_SET プロシージャ，7-26，7-56
CREATE_PHRASE プロシージャ，12-16
CREATE_POLICY プロシージャ，7-27
CREATE_PREFERENCE プロシージャ，7-30
CREATE_RELATION プロシージャ，12-18
CREATE_SECTION_GROUP プロシージャ，7-33
CREATE_STOPLIST プロシージャ，7-36
CREATE_THESAURUS ファンクション，12-20
CREATE_TRANSLATION プロシージャ，12-21
CTX_ADM パッケージ
RECOVER，5-2
SET_PARAMETER，5-3
CTX_CLASSES ビュー，G-3
CTX_CLS
TRAIN，6-2
CTX_DDL パッケージ
ADD_ATTR_SECTION，7-3
ADD_FIELD_SECTION，7-5
ADD_SPECIAL_SECTION，7-11
ADD_STOP_SECTION，7-14
ADD_STOPCLASS，7-13
ADD_STOPTHEME，7-16
ADD_STOPWORD，7-17
ADD_SUB_LEXER，7-19
ADD_ZONE_SECTION，7-22
CREATE_INDEX_SET，7-26，7-56
CREATE_POLICY，7-27
CREATE_PREFERENCE，7-30
CREATE_SECTION_GROUP，7-33
CREATE_STOPLIST，7-36
DROP_POLICY，7-39
DROP_PREFERENCE，7-40
DROP_STOPLIST，7-42
OPTIMIZE_INDEX プロシージャ，7-43
REMOVE_SECTION，7-47
REMOVE_STOPCLASS，7-49
REMOVE_STOPTHEME，7-50
REMOVE_STOPWORD，7-51
SET_ATTRIBUTE，7-52
SYNC_INDEX プロシージャ，7-53
UNSET_ATTRIBUTE，7-55

CTX_DOC_KEY_TYPE システム・パラメータ，2-88
CTX_DOC パッケージ，8-1
FILTER，8-2
GIST，8-5
HIGHLIGHT，8-10
IFILTER，8-13
MARKUP，8-14
PKENCODE，8-19
SET_KEY_TYPE，8-20
THEMES，8-21
TOKENS，8-24
結果表，A-7
CTX_FEEDBACK_ITEM_TYPE 型，A-6
CTX_FEEDBACK_TYPE 型，10-11，A-6
CTX_INDEX_ERRORS ビュー，G-4
例，1-38
CTX_INDEX_OBJECTS ビュー，G-4
CTX_INDEX_SET_INDEXES ビュー，G-6
CTX_INDEX_SUB_LEXERS_VALUES ビュー，G-7
CTX_INDEX_SUB_LEXERS ビュー，G-6，G-18
CTX_INDEX_VALUES ビュー，G-7
CTX_INDEXES ビュー，G-3
CTX_OBJECT_ATTRIBUTE_LOV ビュー，G-9
CTX_OBJECT_ATTRIBUTES ビュー，G-8
CTX_OBJECTS ビュー，G-8
CTX_OUTPUT パッケージ，9-1
ADD_EVENT，9-2
END_LOG，9-3
LOGFILENAME，9-4
REMOVE_EVENT，9-5
START_LOG，9-6
CTX_PARAMETERS ビュー，2-88，G-9
CTX_PENDING ビュー，G-11
CTX_PREFERENCE_VALUES ビュー，G-12
CTX_PREFERENCES ビュー，G-11
CTX_QUERY パッケージ
BROWSE_WORDS，10-2
COUNT_HITS，10-5
EXPLAIN，10-6
HFEEDBACK，10-10
REMOVE_SQE，10-14
STORE_SQE，10-15
結果表，A-2
CTX_SECTION_GROUPS ビュー，G-13
CTX_SECTIONS ビュー，G-12
CTX_SQES ビュー，G-13
CTX_STOPLISTS ビュー，G-13
CTX_STOPWORDS ビュー，G-14
CTX_SUB_LEXERS ビュー，G-14
CTX_THESAURI ビュー，G-14
CTX_THES.CREATE_TRANSLATION，12-21
CTX_THES.DROP_TRANSLATION，12-28
CTX_THES.UPDATE_TRANSLATION，12-55
CTX_THES パッケージ，12-1
ALTER_PHRASE，12-3
ALTER_THESAURUS，12-5
BT，12-7
BTG，12-10
BTI，12-12
BTP，12-14
CREATE_PHRASE，12-16
CREATE_RELATION，12-18
CREATE_THESAURUS，12-20
DROP_PHRASE，12-22
DROP_RELATION，12-24
DROP_THESAURUS，12-27
NT，12-30
NTG，12-33
NTI，12-35
NTP，12-37
OUTPUT_STYLE，12-39
PT，12-40
RT，12-42
SN，12-44
SYN，12-45
THES_TT，12-48
TR，12-49
TRSYN，12-51
TT，12-53
結果表，A-11
CTX_ULEXER パッケージ，13-1
CTX_USER_INDEX_ERRORS ビュー，G-16
例，1-38
CTX_USER_INDEX_OBJECTS ビュー，G-16
CTX_USER_INDEX_SET_INDEXES ビュー，G-17
CTX_USER_INDEX_SETS ビュー，G-17
CTX_USER_INDEX_SUB_LEXERS ビュー，G-18
CTX_USER_INDEX_VALUES ビュー，G-18
CTX_USER_INDEXES ビュー，G-15
CTX_USER_PENDING ビュー，G-19
CTX_USER_PREFERENCE_VALUES ビュー，G-19
CTX_USER_PREFERENCES ビュー，G-19
CTX_USER_SECTION_GROUPS ビュー，G-20
CTX_USER_SECTIONS ビュー，G-20

CTX_USER_SQES ビュー，G-21
CTX_USER_STOPLISTS ビュー，G-21
CTX_USER_STOPWORDS ビュー，G-21
CTX_USER_SUB_LEXERS ビュー，G-22
CTX_USER_THES_PHRASES ビュー，G-22
CTX_USER_THESAURI ビュー，G-22
CTX_VERSION ビュー，G-23
CTXCAT 索引
　概要，1-28
　構文，1-38
　サポートされているプリファレンス，1-39
　サポートされないプリファレンス，1-40
　デフォルトのパラメータ，2-91
ctxkbtc コンパイラ，14-6
ctxload，14-2
　インポート・ファイルの構造，C-4
　例，14-5
CTXRULE 索引
　概要，1-28
　構文，1-41
　デフォルトのパラメータ，2-92
CTXXPATH 索引
　概要，1-28
　構文，1-42
CTXXPATH 索引タイプ
　作成，1-43

D

DATE 列，1-29
DEFAULT_CTXCAT_INDEX_SET システム・パラメータ，2-91
DEFAULT_CTXCAT_LEXER システム・パラメータ，2-91
DEFAULT_CTXCAT_STOPLIST システム・パラメータ，2-91
DEFAULT_CTXCAT_STORAGE システム・パラメータ，2-91
DEFAULT_CTXCAT_WORDLIST システム・パラメータ，2-91
DEFAULT_CTXRULE_LEXER システム・パラメータ，2-92
DEFAULT_CTXRULE_STOPLIST システム・パラメータ，2-92
DEFAULT_CTXRULE_WORDLIST システム・パラメータ，2-92
DEFAULT_DATASTORE システム定義索引付けプリファレンス，2-84
DEFAULT_DATASTORE システム・パラメータ，2-89
DEFAULT_FILTER_BINARY システム・パラメータ，2-89
DEFAULT_FILTER_FILE システム・パラメータ，2-89
DEFAULT_FILTER_TEXT システム・パラメータ，2-89
DEFAULT_INDEX_MEMORY システム・パラメータ，2-88
DEFAULT_LEXER システム定義索引付けプリファレンス，2-85
DEFAULT_LEXER システム・パラメータ，2-90
DEFAULT_RULE_STORAGE システム・パラメータ，2-92
DEFAULT_SECTION_HTML システム・パラメータ，2-89
DEFAULT_SECTION_TEXT システム・パラメータ，2-90
DEFAULT_STOPLIST システム定義プリファレンス，2-87
DEFAULT_STOPLIST システム・パラメータ，2-90
DEFAULT_STORAGE システム定義プリファレンス，2-87
DEFAULT_STORAGE システム・パラメータ，2-90
DEFAULT_WORDLIST システム定義プリファレンス，2-87
DEFAULT_WORDLIST システム・パラメータ，2-90
DEFAULT シソーラス，3-13，3-30
DETAIL_DATASTORE オブジェクト，2-7
　例，2-8
detail_key 属性，2-7
detail_lineno 属性，2-7
detail_table 属性，2-7
detail_text 属性，2-7
DIRECT_DATASTORE オブジェクト，2-3
　例，2-3
DML
　スコア付けへの影響，F-3
DML エラー
　表示，G-4
DML キュー
　表示，G-11
DML 処理
　バッチ，1-3
DROP INDEX，1-45
DROP_PHRASE プロシージャ，12-22
DROP_POLICY プロシージャ，7-39

DROP_PREFERENCE プロシージャ，7-40
DROP_RELATION プロシージャ，12-24
DROP_STOPLIST プロシージャ，7-42
DROP_THESAURUS プロシージャ，12-27
DROP_TRANSLATION プロシージャ，12-28

E

EMPTY_STOPLIST システム定義プリファレンス，2-87
END_LOG プロシージャ，9-3
endjoin 属性，2-40
english 属性（韓国語レクサー），2-51
EQUIVALENCE 演算子，3-15
　NEAR，3-33
　ストップワード変換，H-6
EXP_TAB 表型，A-11
EXPLAIN プロシージャ，10-6
　結果表，A-2
　例，10-8

F

FILE_DATASTORE オブジェクト，2-10
　例，2-10
FILE_DATASTORE システム定義プリファレンス，2-84
FILTER プロシージャ，8-2
　結果表，A-8
　メモリー内の例，8-4
　例，8-4
ftp_proxy 属性，2-13
fuzzy_match 属性，2-71
fuzzy_numresults 属性，2-71
fuzzy_score 属性，2-71
FUZZY 演算子，3-16

G

GIST プロシージャ
　結果表，A-8
　更新済みの構文，8-5
　例，8-8

H

hanja 属性，2-51
HASPATH 演算子，3-18
HFEEDBACK 表の OPERATION 列
　値，A-5
HFEEDBACK プロシージャ，10-10
　結果表，A-4
　例，10-11
HIGHLIGHT プロシージャ，8-10
　結果表，A-9
　例，8-12
HTML
　索引付け，1-42，2-28，2-78，7-33
　ゾーン・セクションの例，7-23
　ハイライト・オフセットの生成，8-10
　ハイライト・マークアップ，8-14
　ハイライト例，8-18
　フィルタ処理，8-2
　フィルタ処理のバイパス，2-25
HTML_SECTION_GROUP
　例，2-80
HTML_SECTION_GROUP オブジェクト，1-42，2-78，7-23，7-33
　NULL_FILTER，2-28
HTML_SECTION_GROUP システム定義プリファレンス，2-86
http_proxy 属性，2-13

I

i_index_clause 属性，2-77
i_table_clause 属性，2-76
IFILTER プロシージャ，8-13
IGNORE
　フォーマット列の値，1-33
INDEX_PROCEDURE user_lexer 属性，2-54
index_stems 属性，2-42
index_text 属性，2-42
index_themes 属性，2-41
INPATH 演算子，3-20
INPUT_TYPE user_lexer 属性，2-54
INSERT 文
　ロード例，C-2
INSO_FILTER オブジェクト，2-24
　キャラクタ・セット変換，2-26
INSO_FILTER システム定義プリファレンス，2-85

Inso フィルタ
索引プリファレンス・オブジェクト，2-24
サポートされていない形式，B-14
サポートされている形式，B-4
サポートされているプラットフォーム，B-2
設定，B-2

J

JA16EUC キャラクタ・セット，2-46，2-47
JA16SJIS キャラクタ・セット，2-46，2-47
JAPANESE_LEXER，2-47
JAPANESE_VGRAM_LEXER オブジェクト，2-46
japanese 属性（韓国語レクサー），2-51
JOB_QUEUE_PROCESSES 初期化パラメータ，1-4，1-30

K

k_table_clause 属性，2-76
KO16KSC5601 キャラクタ・セット，2-48，2-50
KOREAN_LEXER オブジェクト，2-48
KOREAN_MORP_LEXER，2-49
composite の例，2-51
提供されるディクショナリ，2-49

L

LOB 列
ロード，C-2
LOG_DIRECTORY システム・パラメータ，2-88，9-4
LOGFILENAME プロシージャ，9-4
long_word 属性，2-51
LONG 列
索引付け，1-29

M

MARKUP プロシージャ，8-14
HTML ハイライト例，8-18
結果表，A-9
例，8-18
MAX_INDEX_MEMORY システム・パラメータ，2-88
maxdocsize 属性，2-13
maxthreads 属性，2-12
maxurls 属性，2-13
META タグ
ゾーン・セクションの作成，7-23
フィールド・セクションの作成，7-7
MINUS 演算子，3-28
ストップワード変換，H-5
mixed_case 属性，2-40
morpheme 属性，2-50
MULTI_LEXER オブジェクト
CREATE INDEX の例，1-36
例，2-44
MULTI_LEXER 型，2-43
MULTI_STOPLIST 型，7-36

N

n_table_clause 属性，2-77
NARROWER TERM 演算子
例，3-29
NEAR 演算子，3-32
WITHIN，3-56
下位互換性，3-34
スコア付け，3-33
ストップワード変換，H-6
その他の演算子，3-33
ハイライト表示，3-34
NEAR 演算子の max_span パラメータ，3-32
NEAR 演算子のクランプ・サイズ，3-32
nested_column 属性，2-18
NESTED_DATASTORE オブジェクト，2-18
NESTED_DATASTORE 属性，2-19
nested_lineno 属性，2-18
nested_text 属性，2-19
nested_type 属性，2-18
newline 属性，2-40
NEWS_SECTION_GROUP オブジェクト，2-79，7-34
no_proxy 属性，2-14
NOT 演算子，3-36
ストップワード変換，H-5
NTG 演算子，3-29
NTG ファンクション，12-33
NTI 演算子，3-29
NTI ファンクション，12-35
NTP 演算子，3-29
NTP ファンクション，12-37
NT 演算子，3-29
NT ファンクション，12-30
NULL_FILTER オブジェクト，2-28

NULL_FILTER システム定義プリファレンス，2-85
NULL_SECTION_GROUP オブジェクト，2-78，7-33
NULL_SECTION_GROUP システム定義プリファレンス，2-86
number 属性，2-50
NUMBER 列，1-29
numgroup 属性，2-37
numjoin 属性，2-38

O

one_char_word 属性，2-50
OPTIMIZE_INDEX プロシージャ，7-43
OPTIONS 列
  HFEEDBACK 表，A-6
  実行計画表，A-4
OR 演算子，3-37
  ストップワード変換，H-4
OUTPUT_STYLE プロシージャ，12-39
output_type 属性，2-16

P

p_table_clause，2-77
PARAGRAPH キーワード，3-58
PATH_SECTION_GROUP
  問合せ，3-20
PATH_SECTION_GROUP オブジェクト，2-79，7-34
PATH_SECTION_GROUP システム定義プリファレンス，2-86
path 属性，2-10
PKENCODE ファンクション，8-19
PREFERRED TERM 演算子
  例，3-38
prefix_index 属性，2-72
prefix_length_max 属性，2-73
prefix_length_min 属性，2-73
printjoin 属性，2-38
procedure 属性，2-15
prove_themes 属性，2-42
PT 演算子，3-38
PT ファンクション，12-40
punctuation 属性，2-39

Q

QUERY_PROCEDURE user_lexer 属性，2-57

R

r_table_clause 属性，2-76
RECOVER プロシージャ，5-2
RELATED TERM 演算子，3-39
REMOVE_EVENT プロシージャ，9-5
REMOVE_SECTION プロシージャ，7-47
REMOVE_SQE プロシージャ，10-14
REMOVE_STOPCLASS プロシージャ，7-49
REMOVE_STOPTHEME プロシージャ，7-50
REMOVE_STOPWORD プロシージャ，7-51
RFC 1738 URL 仕様，2-11
RT 演算子，3-39
RT ファンクション，12-42
RULE_CLASSIFIER 型，2-81

S

SCORE 演算子，1-47
SENTENCE キーワード，3-58
SET_ATTRIBUTE プロシージャ，7-52
SET_KEY_TYPE プロシージャ，8-20
SET_PARAMETER プロシージャ，2-88，5-3
ShiftJis，2-47
skipjoin 属性，2-39
SN プロシージャ，12-44
SOUNDEX 演算子，3-40
SQE 演算子，3-42
SQL*Loader
  制御ファイル例，C-3
  データ・ファイル例，C-4
  例，C-2
sqlldr 例，C-3
SQL 演算子
  CONTAINS，1-23
  SCORE，1-47
SQL コマンド
  ALTER INDEX，1-2
  CREATE INDEX，1-28
  DROP INDEX，1-45
START_LOG プロシージャ，9-6
startjoin 属性，2-40
stemmer 属性，2-71

STEM 演算子，3-41
stop_dic 属性，2-50
STORE_SQE プロシージャ
　構文，10-15
　例，3-42
substring_index 属性，2-72
SYNC_INDEX プロシージャ，7-53
SYNONYM 演算子，3-43
SYN 演算子，3-43
SYN ファンクション，12-45

T

TEXT
　フォーマット列の値，1-33
theme_language 属性，2-42
THEMES プロシージャ
　結果表，A-10
　構文，8-21
THES_TT プロシージャ，12-48
THRESHOLD 演算子，3-45
　ストップワード変換，H-7
timeout 属性，2-12
to_upper 属性，2-50
TOKENS プロシージャ
　結果表，A-10
　構文，8-24
TOP TERM 演算子，3-50
TRAIN プロシージャ，6-2
TRANSLATION TERM SYNONYM 演算子，3-48
TRANSLATION TERM 演算子，3-46
TRSYN 演算子，3-48
TRSYN ファンクション，12-51
TR 演算子，3-46
TR ファンクション，12-49
TT 演算子，3-50
TT ファンクション，12-53

U

UNIX プラットフォーム
　Inso の変数の設定，B-3
UNSET_ATTRIBUTE プロシージャ，7-55
UPDATE_TRANSLATION プロシージャ，12-55
URL_DATASTORE オブジェクト
　属性，2-11
　例，2-14
URL_DATASTORE システム定義プリファレンス，
　2-84
urlsize 属性，2-13
URL の構文，2-11
USER_DATASTORE オブジェクト，2-15
　例，2-16
USER_DATSTORE
　バイナリ・ドキュメントのフィルタ処理，8-13
user_dic 属性，2-50
USER_FILTER オブジェクト，2-28
　例，2-29
USER_LEXER オブジェクト，2-53
UTF-16 endian 自動識別，2-22
UTF8，2-47
UTF8 キャラクタ・セット，2-36，2-45，2-46，2-47，
　2-48，2-50

V

VARCHAR2 列
　索引付け，1-29
verb_adjective 属性，2-50

W

WEIGHT 演算子，3-51
　ストップワード変換，H-7
whitespace 属性，2-40
wildcard_maxterms 属性，2-74
WILDCARD_TAB 型，13-1
WITHIN 演算子，3-55
　ストップワード変換，H-7
　制限事項，3-61
　属性セクション，3-58
　ネスト，3-57
　優先順位，3-4

X

XML_SECTION_GROUP
　例，2-80
XML_SECTION_GROUP オブジェクト，1-42，2-78，
　7-34
XMLType 列
　索引付け，1-43

XML ドキュメント
索引付け，1-42，2-79，7-34
属性セクション，7-3
問合せ，3-58
ドキュメント・タイプを区別するセクション，7-24
XML のセクション化，2-80

Z

ZHS16CGB231280 キャラクタ・セット，2-45
ZHS16GBK キャラクタ・セット，2-45
ZHT16BIG5 キャラクタ・セット，2-45
ZHT32EUC キャラクタ・セット，2-45
ZHT32TRIS キャラクタ・セット，2-45

あ

アメリカ英語
索引デフォルト，2-85

い

イタリア語
ステミング，2-69
提供されるストップリスト，D-10
ファジー・マッチング，2-70
意味を限定する語
シソーラス・インポート・ファイル内，C-8
シソーラスの問合せ，3-12
インポート・ファイル
構造，C-4
例，C-10

え

英語
索引デフォルト，2-85
提供されるストップリスト，D-2
ファジー・マッチング，2-70
エスケープ文字，4-2
エラー
索引付け，1-38
演算子，3-1
ABOUT，3-5
ACCUMULATE，3-9
BROADER TERM，3-12
EQUIVALENCE，3-15
FUZZY，3-16
HASPATH，3-18
INPATH，3-20
MINUS，3-28
NARROWER TERM，3-29
NEAR，3-32
NOT，3-36
OR，3-37
PREFERRED TERM，3-38
RELATED TERM，3-39
SOUNDEX，3-40
SQE，3-42
STEM，3-41
SYNONYM，3-43
THRESHOLD，3-45
TOP TERM，3-50
TRANSLATION TERM，3-46
TRANSLATION TERM SYNONYM，3-48
WEIGHT，3-51
WITHIN，3-55
演算子の拡張
表示，10-6
演算子の優先順位，3-3
EQUIVALENCE 演算子，3-15
表示，10-6
変更，3-4，4-2
例，3-4

お

大 / 小文字が区別される索引
作成，2-40
大 / 小文字区別
ABOUT 問合せ，3-6
オブジェクト
索引の表示，G-8
オブジェクト値
表示，G-7
オランダ語
コンポジット・ワードの索引付け，2-41
索引デフォルト，2-85
ステミング，2-69
提供されるストップリスト，D-5
ファジー・マッチング，2-70

か

下位語問合せフィードバック，10-10
階層関係
　シソーラス・インポート・ファイル内，C-7
階層問合せフィードバック情報
　生成，10-10
外部フィルタ
　指定，2-28
拡張演算子
　SOUNDEX，3-40
　STEM，3-41
　表示，10-6
カッコ
　グループ化文字，4-2
　優先順位の変更，3-4，4-2
空の索引
　作成，1-35，1-43
環境変数
　Inso フィルタの設定，B-3
環境変数 HOME
　Inso の設定，B-3
環境変数 PATH
　Inso の設定，B-3
韓国語
　索引デフォルト，2-86
　ファジー・マッチング，2-70
韓国語キャラクタ・セットのサポート，2-48，2-50
韓国語テキスト
　索引付け，2-49
関連語問合せフィードバック，10-10
関連ランク付け
　ワード問合せ，F-2

き

記憶域オブジェクト，2-76
記憶域索引プリファレンス
　例，7-32
記憶域デフォルト，2-77
機能
　新規，xxi
逆頻度スコア付け，F-2
キャラクタ・セット
　韓国語，2-48，2-50
　中国語，2-45
　日本語，2-46
　複合列の索引付け，2-23
キャラクタ・セット変換
　INSO_FILTER，2-26
キャラクタ・セット列，1-33
近接演算子，「NEAR 演算子」を参照

く

繰返しフィールド・セクション
　問合せ，3-58

け

形式設定されたドキュメント
　フィルタ処理，2-24
結果表，A-1
　CTX_DOC，A-7
　CTX_QUERY，A-2
　CTX_THES，A-11
　TOKENS，A-10
権限
　索引付けに必要，1-28
言語
　設定，2-35
言語列，1-34

こ

語幹索引付け，2-42
語形変化のステミング
　使用可能，2-71
コンポジット・テキストキー
　エンコーディング，8-19
コンポジット・ワード
　表示，10-6
コンポジット・ワード・ディクショナリ，2-41
コンポジット・ワードの索引付け
　ドイツ語またはオランダ語テキストの作成，2-41

さ

索引
　改名，1-2
　作成，1-28
　登録の表示，G-3

索引エラー
　削除，1-38
　表示，1-38
索引オブジェクト，2-1
　表示，G-4，G-8
索引最適化，1-5
索引作成パラメータ
　例，2-77
索引タイプ CONTEXT，1-28
索引付け
　テーマ，2-41
　パラレル，1-4，1-30
　マスター / ディテールの例，2-9
索引付けのデフォルト
　表示，G-9
索引トークン
　ドキュメントに対しての生成，8-24
索引内のワードのブラウズ，10-2
索引の改名，1-2
索引の再構築
　構文，1-3
　例，1-9
索引の最適化，1-5
索引の作成
　カスタム・プリファレンスの例，1-36
　デフォルトの例，1-36
索引の断片化，1-35
索引のメンテナンス，1-2
索引パラメータの移入，1-35，1-43
索引パラメータの非移入，1-35，1-43
索引表領域パラメータ
　指定，2-76
索引プリファレンス
　概要，2-2
　作成，2-2，7-30
索引メンテナンス，1-2
索引要求
　ロギング，9-6
索引要求のロギング，9-6
サブストリング索引
　作成の例，2-74
サブレクサー
　表示，G-6，G-14，G-18
サブレクサーの値
　表示，G-7
左右切捨て検索
　パフォーマンスの改善，2-72
左右切捨て問合せ，3-53

し

システム定義プリファレンス，2-84
システムのリカバリ
　手動，5-2
システム・パラメータ，2-88
　索引付けのデフォルト，2-89
シソーラス
　DEFAULT，3-13
　インポート / エクスポート，14-2
　インポート / エクスポート例，14-5
　改名および切捨て，12-5
　関係の作成，12-16
　コンパイル，14-6
　削除，12-27
　作成，12-20
　参照用プロシージャ，12-1
　情報の表示，G-14
シソーラス・インポート・ファイル
　構造，C-4
　例，C-10
シソーラス句
　削除，12-22
　変更，12-3
シソーラスの最上位語
　すべて検索，12-48
シソーラスのスコープ・ノート
　検索，12-44
シソーラスのリレーション
　削除，12-24
　作成，12-18
シソーラスのロード，14-2
実行計画表
　構造，A-2
　作成，10-8
　データの取出し例，10-8
実行計画表の OPERATION 列
　値，A-3
失敗した索引操作
　再開，1-5
失敗した索引の再開，1-5
　例，1-9
上位語問合せフィードバック，10-10

新機能，xxi
シングルバイト言語
索引付け，2-35

す

スウェーデン語
索引デフォルト，2-85
代替スペル，E-4
提供されるストップリスト，D-13
スコア付け
ACCUMULATE，3-9
DML の影響，F-3
NEAR 演算子，3-33
スコア付けの Salton の式，F-2
スコア付けのアルゴリズム
ワード問合せ，F-2
スコープ・ノート
検索，12-44
ステミング，2-71
自動，2-69
使用可能の例，2-74
ストアド・クエリー，3-42
ストアド・クエリー式
削除，10-14
作成，10-15
定義の表示，G-13
表示，G-21
ストップクラス
削除，7-49
定義，7-13
ストップテーマ
削除，7-50
定義，7-16
ストップリスト
イタリア語，D-10
英語，D-2
オランダ語，D-5
概要，2-82
削除，7-42
作成，2-83，7-36
スウェーデン語，D-13
スペイン語，D-12
デンマーク語，D-5
ドイツ語，D-9
表示，G-13
フィンランド語，D-7
フランス語，D-8
変更，2-83
ポルトガル語，D-11
マルチ言語，2-44，2-82
ストップワード
削除，2-83
動的な追加，2-83
削除，7-51
ストップリストへのすべての表示，G-14
定義，7-17
動的な追加，1-3，1-7
ストップワード変換，H-2
表示，10-6
スペイン語
ステミング，2-69
提供されるストップリスト，D-12
ファジー・マッチング，2-70

せ

制御ファイル例
SQL*Loader，C-3
セクション
オーバーラップ，7-24
繰返しゾーン，7-24
繰返しフィールド，7-7
削除，7-47
情報の表示，G-12
ゾーンの作成，7-22
属性の作成，7-3
動的な追加，1-3
動的な追加の制約，1-9
ネスト，7-25
フィールドの作成，7-5
セクション・グループ
作成，7-33
情報の表示，G-13
セクション・グループのタイプ，2-78，7-33
セクション・グループの例，2-80
セクション検索，3-55
ネスト，3-57

そ

ゾーン・セクション
　繰返し，7-24
　作成，7-22
　問合せ，3-55
　動的な追加，1-7
　動的な追加の例，1-11
ゾーン・セクションのオーバーラップ，7-24
属性
　alternate_spelling，2-43
　base_letter，2-40
　binary，2-7
　charset，2-22
　command，2-28
　composite，2-41
　continuation，2-37
　detail_key，2-7
　detail_lineno，2-7
　detail_table，2-7
　detail_text，2-7
　endjoin，2-40
　ftp_proxy，2-13
　fuzzy_match，2-71
　fuzzy_numresults，2-71
　fuzzy_score，2-71
　http_proxy，2-13
　i_index_clause，2-77
　i_table_clause，2-76
　index_text，2-42
　index_themes，2-41
　k_table_clause，2-76
　maxdocsize，2-13
　maxthreads，2-12
　maxurls，2-13
　mixed_case，2-40
　n_table_clause，2-77
　newline，2-40
　no_proxy，2-14
　numgroup，2-37
　numjoin，2-38
　output_type，2-16
　p_table_clause，2-77
　path，2-10
　printjoin，2-38
　procedure，2-15
　punctuation，2-39
　r_table_clause，2-76
　skipjoin，2-39
　startjoin，2-40
　stemmer，2-71
　timeout，2-12
　urlsize，2-13
　whitespace，2-40
　可能な値の表示，G-9
　使用禁止，7-55
　設定，7-52
　表示，G-8
属性セクション
　WITHIN 例，3-58
　定義，7-3
　問合せ，3-55
　動的な追加，1-8，1-11

た

大カッコ
　グループ化文字，4-2
　優先順位の変更，3-4，4-2
代替スペル
　概要，E-2
　使用可能の例，E-2
　使用禁止の例，7-55，E-2
　スウェーデン語，E-4
　デンマーク語，E-3
　ドイツ語，E-3
タグ付きテキスト
　検索，3-55
段落セクション
　定義，7-11
　問合せ，3-55

ち

中カッコエスケープ文字，4-2
中国語
　ファジー・マッチング，2-70
中国語（簡体字）
　索引デフォルト，2-86
中国語キャラクタ・セットのサポート，2-45
中国語テキスト
　索引付け，2-45

て

提供されるストップリスト，D-1
ディクショナリ
　韓国語，2-49
　ユーザー，2-41
停止セクション
　追加，7-14
　動的な追加，1-8
　動的な追加の例，1-11
データ記憶域
　URL，2-11
　外部，2-10
　ダイレクト，2-3
　プロシージャ的に定義，2-15
　マスター / ディテール，2-7
　例，7-31
データストア型，2-3
テーマ
　索引付け，2-41
　トップ n の取得，8-23
　ドキュメントに対しての生成，8-21
　ハイライト・オフセット例，8-12
　ハイライト・マークアップの生成，8-14
テーマ機能
　サポート対象言語，14-11
テーマ索引
　英語のデフォルト，2-85
　作成，2-41
テーマ・サマリー
　生成，8-5
　トップ n の生成，8-8
テーマの検証
　使用可能，2-42
テーマ・ハイライト
　HTML オフセット例，8-12
　HTML マークアップ例，8-18
　オフセットの生成，8-10
　マークアップの生成，8-14
テーマ表
　構造，A-10
テーマ・ベース，I-1
テキスト・データ・ディクショナリ
　クリーン・アップ，5-2
テキストのみの索引
　使用可能，2-42
　例，7-30
テキストのロード
　SQL*Loader 例，C-2
　SQL の INSERT 例，C-2
テキスト列
　サポートされているタイプ，1-29
デフォルトの索引
　例，1-36
デフォルトのパラメータ
　CONTEXT 索引，2-89
　CTXCAT 索引，2-91
　CTXRULE 索引，2-92
　表示，2-92
　変更，2-92
テンプレートによる問合せ，1-18，1-23
デンマーク語
　索引デフォルト，2-85
　代替スペル，E-3
　提供されるストップリスト，D-5

と

問合せ
　ACCUMULATE，3-9
　AND，3-11
　BROADER TERM，3-12
　EQUIVALENCE，3-15
　MINUS，3-28
　NARROWER TERM，3-29
　NOT，3-36
　OR，3-37
　PREFERRED TERM，3-38
　RELATED TERM，3-39
　SYNONYM，3-43
　THRESHOLD，3-45
　TOP TERM，3-50
　TRANSLATION TERM，3-46
　TRANSLATION TERM SYNONYM，3-48
　WEIGHT，3-51
　階層フィードバック，10-10
　ストアド，3-42
　例，1-25
問合せテンプレート，1-18，1-23
ドイツ語
　alternate spelling 属性，2-43
　コンポジット・ワードの索引付け，2-41
　索引デフォルト，2-85
　ステミング，2-69

代替スペルの規則，E-3
提供されるストップリスト，D-9
ファジー・マッチング，2-70
同形異義語
BROADER TERM 問合せ，3-13
シソーラス・インポート・ファイル内，C-8
問合せ内，3-12
トークン索引最適化，1-5
ドキュメント
HTML およびプレーン・テキストに対するフィルタ処理，8-2
ドキュメント形式
サポートされていない，B-14
サポートされている，B-4
ドキュメント・サービス
要求のロギング，9-6
ドキュメントの表示方法
プロシージャ，8-1
ドキュメントの分類，6-2
ドキュメントのロード
SQL*Loader，C-2
ドキュメント・フィルタ処理
Inso，B-2
特殊セクション
定義，7-11
問合せ，3-55
特殊文字のエスケープ，4-2

な

ナレッジ・カタログ
カテゴリの階層，I-1
ナレッジ・カタログでの概念，I-1
ナレッジ・カタログ内のカテゴリ，I-1
ナレッジ・ベース
キャラクタ・セットのサポート，14-6
ユーザー定義，14-11
ナレッジ・ベース拡張コンパイラ，14-6
ナレッジ・ベースの拡張，14-6

に

日本語
索引付け，2-46
索引デフォルト，2-86
ファジー・マッチング，2-70
日本語 EUC キャラクタ・セット，2-47
日本語キャラクタ・セットのサポート，2-46

ね

ネストされたセクション検索，3-57
ネストされたゾーン・セクション，7-25

の

ノルウェー語
索引デフォルト，2-85

は

バージョン・ナンバー
表示，G-23
パーティション索引
再構築の例，1-10
例，1-37
パーティション索引の作成
例，1-37
パーティション表
変更，1-12
ハイライト表
構造，A-9
例，8-12
ハイライト表示
NEAR 演算子，3-34
オフセットの生成，8-10
マークアップの生成，8-14
ハイライト表示のオフセット，8-10
派生語のステミング
英語での使用可能，2-71
バックスラッシュ・エスケープ文字，4-2
パフォーマンス
ワイルド・カード検索，3-54
パラメータ
システム定義の表示，G-9
設定，5-3
パラレル索引付け，1-4，1-30
例，1-37
パラレル索引の作成，1-37

ひ

左側切捨て検索
　パフォーマンスの改善，2-72
ヒット数のカウント，10-5
ビュー，G-1
表構造
　HFEEDBACK，A-4
　実行計画，A-2
　テーマ，A-10
　ハイライト，A-9
　フィルタ，A-8
　マークアップ，A-9
　要点，A-8
表示
　演算子の拡張，10-6
　演算子の優先順位，10-6

ふ

ファイル・データ記憶域
　例，7-31
ファジー・マッチング
　言語の指定，2-71
　自動言語識別，2-70
　使用可能の例，2-74
フィールド・セクション
　WITHIN 例，3-57
　繰返し，3-58
　制限事項，7-7
　定義，7-5
　問合せ，3-55
　動的な追加，1-7
フィールド・セクションに対する visible フラグ，7-6
フィールド・セクションの visible フラグ，3-57
フィルタ
　Inso，2-24，B-2
　キャラクタ・セット，2-22
　ユーザー，2-28
フィルタ型，2-21
フィルタ形式
　サポートされている，B-4
フィルタ処理
　プレーン・テキスト，8-13
　プレーン・テキストおよび HTML，8-2
フィルタ表
　構造，A-8
フィンランド語
　索引デフォルト，2-85
　提供されるストップリスト，D-7
フォーマット列，1-33
複合キャラクタ・セット列
　索引付け，2-23
複合フォーマット列
　索引付け，2-25
　サポートされている形式，B-4
　フィルタ処理，2-24
フランス語
　提供されるストップリスト，D-8
　ファジー・マッチング，2-70
フランス語ステミング，2-69
プリファレンス
　概要，2-2
　索引付けの指定，1-32
　削除，7-40
　作成，7-30
　システム定義，2-84
　置換，1-3
　表示，G-11
プリファレンス・クラス
　表示，G-3
プリファレンス値
　表示，G-12
プリファレンスの置換，1-3
プレーン・テキスト
　NULL_FILTER による索引付け，2-28
　索引付け，2-27
　ハイライト表示のオフセット，8-10
　ハイライト・マークアップ，8-14
　フィルタ処理，8-2，8-13
　フィルタ処理のバイパス，2-25
文セクション
　定義，7-11
　問合せ，3-55

へ

変換
　ストップワード，H-2

ほ

保留中の DML
　表示，G-11
ポルトガル語
　提供されるストップリスト，D-11
翻訳
　更新，12-55
　削除，12-28
　シソーラスに追加，12-21

ま

マークアップ表
　構造，A-9
　例，8-18
マスター表 / ディテール表
　索引付けの例，2-9
マスター表 / ディテール表のデータ記憶域，2-7
　例，2-8，7-31
マルチ言語の索引付け，7-19
マルチ言語のストップリスト，2-44，2-82
マルチ言語表
　問合せ，1-25，2-45

め

メモリー
　索引付け，1-6，1-35，1-43，7-53
　索引の同期化用，1-6

も

文字
　continuation，2-37
　newline の指定，2-40
　numgroup，2-37
　numjoin，2-38
　printjoin，2-38
　punctuation，2-39
　skipjoin，2-39
　startjoin および endjoin，2-40
　whitespace の指定，2-40

ゆ

ユーティリティ
　ctxload，14-2

よ

要点
　生成，8-5
要点表
　構造，A-8
予約語，4-3
　エスケープ，4-2

る

ルール
　生成，6-2

れ

例，1-37
レクサー型，2-35
列型
　CTXCAT 索引にサポートされている，1-38，1-41
　CTXXPATH 索引にサポートされている，1-42
　索引付けにサポートされている，1-29

ろ

論理演算子
　NEAR，3-33

わ

ワイルド・カード検索，3-53
　パフォーマンスの改善，3-54
ワイルド・カード問合せ
　パフォーマンスの改善，2-72