

Oracle8i

interMedia Text リファレンス

リリース 8.1

2000 年 2 月

部品番号 : J00966-01

ORACLE®

Oracle8i *interMedia* Text リファレンス , リリース 8.1

部品番号 : J00966-01

原本名 : Oracle8i *interMedia* Text Reference, Release 2 (8.1.6)

原本部品番号 : A77063-01

原本著者 : Colin McGregor

原本協力者 : Shamim Alpha, Steve Buxton, Chung-Ho Chen, Yun Cheng, Paul Dixon, Mohammad Faisal, Elena Huang, Jie Bai, Garret Kaminaga, Jacqueline Kud, Bryn Llewellyn, Wesley Lin, Kavi Mahesh, Yasuhiro Matsuda, Gerda Shank, Steve Yang

Copyright © 1996, 1999, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム (ソフトウェアおよびドキュメントを含む) の使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されています。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、Oracle Corporation (米国オラクル) または日本オラクル株式会社 (日本オラクル) を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation (米国オラクル) およびその関連会社は一切責任を負いかねます。当プログラムを米国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Notice が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに	xiii
対象読者	xiv
前提条件	xiv
関連マニュアル	xiv
このマニュアルの構成	xiv
このマニュアルの表記規則	xv
1 interMedia Text の概要	
Oracle8i <i>interMedia Text</i> の概要	1-2
テキスト・アプリケーションの作成	1-3
サンプル・コードのディレクトリ	1-3
<i>interMedia Text</i> のユーザーおよびロール	1-4
CTXSYS ユーザー	1-4
CTXAPP ロール	1-4
ドキュメントのロード	1-5
サポートされている列型	1-5
サポートされているドキュメント形式	1-5
ロード方法	1-5
索引付け	1-7
すべての言語に対する汎用デフォルト	1-7
言語固有のデフォルト	1-8
索引のメンテナンス	1-9
問合せ	1-10
ワード問合せの例	1-10
ABOUT 問合せの例	1-10

応答時間短縮のための問合せの最適化	1-11
その他の問合せ機能	1-11
ドキュメントの表示方法およびハイライト表示	1-13

2 SQL コマンド

ALTER INDEX	2-2
DROP INDEX	2-11
CONTAINS	2-12
CREATE INDEX	2-14
SCORE	2-22

3 索引付け

概要	3-2
プリファレンスの作成	3-2
データストア・オブジェクト	3-3
DIRECT_DATASTORE	3-3
DETAIL_DATASTORE	3-4
FILE_DATASTORE	3-7
URL_DATASTORE	3-8
USER_DATASTORE	3-12
NESTED_DATASTORE	3-15
フィルタ・オブジェクト	3-18
CHARSET_FILTER	3-19
INSO_FILTER	3-21
NULL_FILTER	3-24
USER_FILTER	3-25
レクサー・オブジェクト	3-27
BASIC_LEXER	3-28
MULTI_LEXER	3-35
CHINESE_VGRAM_LEXER	3-36
JAPANESE_VGRAM_LEXER	3-37
KOREAN_LEXER	3-37
ワードリスト・オブジェクト	3-39
BASIC_WORDLIST	3-39
記憶域オブジェクト	3-43

BASIC_STORAGE	3-43
セクション・グループのタイプ	3-46
セクション・グループの例	3-47
ストップリスト	3-48
ストップリストの作成	3-48
デフォルトのストップリストの変更	3-48
システム定義プリファレンス	3-50
データ記憶域	3-50
フィルタ	3-51
レクサー	3-51
セクション・グループ	3-52
ストップリスト	3-53
記憶域	3-53
ワードリスト	3-53
システム・パラメータ	3-54
概要	3-54
デフォルトの索引付けパラメータ	3-55

4 問合せ演算子

演算子の優先順位	4-3
グループ 1 演算子	4-3
グループ 2 の演算子および文字	4-3
プロシージャ型演算子	4-4
優先順位の例	4-4
優先順位の変更	4-5
ABOUT	4-6
ACCUMulate (.)	4-9
AND (&)	4-11
BROADER TERM (BT, BTG, BTP, BTI)	4-12
EQUIValence (=)	4-15
FUZZY (?)	4-16
MINUS (-)	4-17
NARROWER TERM (NT, NTG, NTP, NTI)	4-18
NEAR (:)	4-20
NOT (~)	4-24
OR (I)	4-25

PREFERRED TERM (PT)	4-26
RELATED TERM (RT)	4-27
SOUNDEX (!)	4-28
STEM (\$)	4-29
ストアド・クエリー式 (SQE)	4-30
SYNonym (SYN)	4-31
THRESHOLD (>)	4-33
TRANSLATION TERM (TR)	4-34
TRANSLATION TERM SYNONYM (TRSYN)	4-36
TOP TERM (TT)	4-38
WEIGHT (*)	4-39
ワイルド・カード (% _)	4-41
ライト・トランケーテッド問合せ	4-41
レフト・トランケーテッド問合せおよびダブル・トランケーテッド問合せ	4-41
WITHIN	4-43

5 問合せの特殊文字

グループ化文字	5-2
エスケープ文字	5-3
エスケープ文字の問合せ	5-3
予約語	5-4

6 CTX_ADM パッケージ

RECOVER	6-2
SET_PARAMETER	6-3
SHUTDOWN	6-5

7 CTX_DDL パッケージ

ADD_ATTR_SECTION	7-3
ADD_FIELD_SECTION	7-5
ADD_SPECIAL_SECTION	7-9
ADD_STOPCLASS	7-11
ADD_STOP_SECTION	7-12
ADD_STOPTHEME	7-14
ADD_STOPWORD	7-15
ADD_SUB_LEXER	7-17

ADD_ZONE_SECTION	7-20
CREATE_PREFERENCE	7-24
CREATE_SECTION_GROUP	7-27
CREATE_STOPLIST	7-30
DROP_PREFERENCE	7-31
DROP_SECTION_GROUP	7-32
DROP_STOPLIST	7-33
OPTIMIZE_INDEX	7-34
REMOVE_SECTION	7-36
REMOVE_STOPCLASS	7-38
REMOVE_STOPTHEME	7-39
REMOVE_STOPWORD	7-40
SET_ATTRIBUTE	7-41
SYNC_INDEX	7-42
UNSET_ATTRIBUTE	7-43

8 CTX_DOC パッケージ

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

9 CTX_OUTPUT パッケージ

END_LOG	9-2
LOGFILENAME	9-3
START_LOG	9-4

10 CTX_QUERY パッケージ

BROWSE_WORDS	10-2
COUNT_HITS	10-5
EXPLAIN	10-6
HFEEDBACK	10-9
REMOVE_SQE	10-13

STORE_SQE	10-14
-----------------	-------

11 CTX_THES パッケージ

ALTER_PHRASE	11-3
ALTER_THESAURUS	11-5
BT	11-7
BTG	11-10
BTI	11-12
BTP	11-14
CREATE_PHRASE	11-16
CREATE_RELATION	11-18
CREATE_THESAURUS	11-20
DROP_PHRASE	11-21
DROP_RELATION	11-23
DROP_THESAURUS	11-26
NT	11-27
NTG	11-30
NTI	11-32
NTP	11-34
OUTPUT_STYLE	11-36
PT	11-37
RT	11-39
SN	11-41
SYN	11-42
THES_TT	11-45
TR	11-46
TRSYN	11-48
TT	11-50

12 実行可能ファイル

ctxsrv	12-2
構文	12-2
例	12-3
注意	12-4
関連項目	12-4
ctxload	12-6

シソーラスのインポートおよびエクスポート	12-6
テキストのロード	12-6
ドキュメントの更新 / エクスポート	12-7
ctxload 構文	12-7
例	12-11
知識ベース拡張コンパイラ (ctxkbtcl)	12-12
構文	12-12
使用上の注意	12-13
シソーラスの語句の制約	12-13
シソーラス関係の制約	12-13
新規語句の既存語句へのリンク	12-14
複数のシソーラスの優先順位	12-15
サイズ制限	12-15

A 問合せのチューニング

統計を使用した問合せの最適化	A-2
統計の収集	A-2
統計の再収集	A-3
統計の削除	A-4
拡張可能問合せオプティマイザの使用禁止および使用可能	A-4
応答時間短縮のための問合せの最適化	A-5
FIRST_ROWS による応答時間の短縮	A-5
CHOOSE による応答時間の短縮	A-7
スループット向上のための問合せの最適化	A-8
CHOOSE モードおよび ALL ROWS モード	A-8
FIRST_ROWS モード	A-8
ブロック操作による問合せのチューニング	A-9

B 結果表

CTX_QUERY 結果表	B-2
実行計画表	B-2
HFEEDBACK 表	B-5
CTX_DOC 結果表	B-8
フィルタ表	B-8
要旨表	B-8

ハイライト表	B-10
マークアップ表	B-10
テーマ表	B-11
CTX_THES 結果表およびデータ型	B-12
EXP_TAB 表型	B-12

C サポートされているフィルタ形式

Inso フィルタ処理テクノロジー	C-2
サポートされているプラットフォーム	C-2
環境変数の場所	C-2
UNIX プラットフォームに対する考慮点	C-3
OLE2 オブジェクトのサポート	C-4
サポートされているドキュメント形式	C-5
ワード処理 - 共通	C-5
ワード処理 - DOS	C-5
ワード処理 - インターナショナル	C-7
ワード処理 - Windows	C-7
ワード処理 - Macintosh	C-8
スプレッドシート形式	C-8
データベース形式	C-9
表示形式	C-10
プレゼンテーション形式	C-10
標準グラフィック形式	C-11
その他	C-12
サポートされていない形式	C-13

D ロード例

SQL の INSERT 例	D-2
SQL*Loader 例	D-3
表の作成	D-3
SQL*Loader コマンドの発行	D-3
ctxload シソーラス・インポート・ファイルの構造	D-6
代替階層構造	D-8
インポート・ファイル内の語句の使用上の注意	D-9
インポート・ファイル内の関連語句の使用上の注意	D-10

インポート・ファイル例	D-11
ctxload テキスト・ロード・ファイルの構造	D-13
ロード・ファイルの構造	D-14
ロード・ファイル構文	D-14
ロード・ファイルの埋込みテキスト例	D-15
ロード・ファイルのファイル名ポインタ例	D-15

E 提供されるストップリスト

英語	E-2
デンマーク語 (DA)	E-3
オランダ語 (NL)	E-4
フィンランド語 (FI)	E-5
フランス語 (FR)	E-6
ドイツ語 (DE)	E-7
イタリア語 (IT)	E-8
ポルトガル語 (PR)	E-9
スペイン語 (ES)	E-10
スウェーデン語 (SE)	E-11

F 代替スペルの規則

概要	F-2
代替スペルの使用可能	F-2
代替スペルの使用禁止	F-2
ドイツ語	F-3
デンマーク語	F-4
スウェーデン語	F-5

G 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	G-2
例	G-3
DML およびスコア付け	G-3

H ビュー

概要	H-2
CTX_CLASSES	H-4

CTX_INDEXES	H-4
CTX_INDEX_ERRORS	H-5
CTX_INDEX_OBJECTS	H-5
CTX_INDEX_VALUES	H-6
CTX_OBJECTS	H-6
CTX_OBJECT_ATTRIBUTES	H-7
CTX_OBJECT_ATTRIBUTE_LOV	H-7
CTX_PARAMETERS	H-8
CTX_PENDING	H-9
CTX_PREFERENCES	H-9
CTX_PREFERENCE_VALUES	H-9
CTX_SECTIONS	H-10
CTX_SECTION_GROUPS	H-10
CTX_SERVERS	H-11
CTX_SQES	H-11
CTX_STOPLISTS	H-12
CTX_STOPWORDS	H-12
CTX_SUB_LEXERS	H-13
CTX_THESAURI	H-13
CTX_USER_INDEXES	H-14
CTX_USER_INDEX_ERRORS	H-15
CTX_USER_INDEX_OBJECTS	H-15
CTX_USER_INDEX_VALUES	H-15
CTX_USER_PENDING	H-16
CTX_USER_PREFERENCES	H-16
CTX_USER_PREFERENCE_VALUES	H-16
CTX_USER_SECTIONS	H-17
CTX_USER_SECTION_GROUPS	H-17
CTX_USER_SQES	H-17
CTX_USER_STOPLISTS	H-18
CTX_USER_STOPWORDS	H-18
CTX_USER_SUB_LEXERS	H-19
CTX_USER_THESAURI	H-19

I ストップワード変換

ストップワード変換の理解	I-2
ワード変換	I-3

AND 変換	I-3
OR 変換	I-3
ACCUMulate 変換	I-4
MINUS 変換	I-5
NOT 変換	I-5
EQUIValence 変換	I-6
NEAR 変換	I-6
WEIGHT 変換	I-7
THRESHOLD 変換	I-7
WITHIN 変換	I-7

J 知識ベース - カテゴリの階層

Branch 1: science and technology	J-2
Branch 2: business and economics	J-8
Branch 3: government and military	J-9
Branch 4: social environment	J-10
Branch 5: geography	J-14
Branch 6: Abstract Ideas and Concepts	J-17

索引

はじめに

このマニュアルでは、Oracle8i *interMedia Text* の参照情報を記載しています。*interMedia Text* の索引の作成、テキスト問合せの発行、ドキュメントの表示および *interMedia Text* PL/SQL パッケージの使用に関する参照としてご使用ください。

対象読者

このマニュアルは、*interMedia Text* アプリケーション開発者または *interMedia Text* システムのメンテナンスに責任のあるシステム管理者を対象としています。

前提条件

このマニュアルは、ユーザーが 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 の経験者であることを前提とします。詳細は、ご使用の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 Tools に慣れていない場合は、『Oracle8i 概要』の第 1 章「Oracle Server の基礎知識」をお読みください。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。

関連マニュアル

interMedia Text の詳細は、次のマニュアルを参照してください。

- 『Oracle8i *interMedia Text* 移行ガイド』

Oracle8i の詳細は、次のマニュアルを参照してください。

- 『Oracle8i 概要』
- 『Oracle8i 管理者ガイド』
- 『Oracle8i ユーティリティ・ガイド』
- 『Oracle8i パフォーマンスのための設計およびチューニング』
- 『Oracle8i SQL リファレンス』
- 『Oracle8i リファレンス・マニュアル』
- 『Oracle8i アプリケーション開発者ガイド 基礎編』

PL/SQL の詳細は、次のマニュアルを参照してください。

- 『Oracle8i PL/SQL ユーザーズ・ガイドおよびリファレンス』

このマニュアルの構成

目次を参照してください。

このマニュアルの表記規則

このマニュアルでは、次の表記規則を使用します。

表記	意味
大文字	大文字は、Oracle コマンド、標準データベース・オブジェクトと定数、および Oracle PL/SQL の標準プロシージャを示します。
イタリック体	イタリック体は、CONTAINS 問合せの問合せ語句を示します。
固定幅フォント	固定幅フォントは、SQL*Plus コマンドの例および PL/SQL コードの例を示します。コマンドまたはコードは、記載されているとおり正確に入力してください。

interMedia Text の概要

この章では、Oracle8i *interMedia Text* の主な機能について説明します。この章の内容は、索引付け、問合せおよびドキュメント表示を始める場合に有効です。

この章の内容は次のとおりです。

- [Oracle8i *interMedia Text* の概要](#)
- [テキスト・アプリケーションの作成](#)
- [interMedia Text のユーザーおよびロール](#)
- [ドキュメントのロード](#)
- [索引付け](#)
- [索引のメンテナンス](#)
- [問合せ](#)
- [ドキュメントの表示方法およびハイライト表示](#)

Oracle8i *interMedia Text* の概要

Oracle8i *interMedia Text* を使用して、テキスト問合せアプリケーションを作成できます。Oracle8i *interMedia Text* では、テキストに対する検索、取出しおよび表示の機能が提供されています。また、英語ドキュメントの概念検索およびテーマの分析も提供されています。

索引付けおよび問合せを行うには、`context` 型のドメイン・インデックスとともに標準 SQL を使用します。また、提供されている *interMedia Text* PL/SQL パッケージを使用して、ドキュメントの表示やシソーラスのメンテナンスなどの高度な機能も利用できます。

テキスト・アプリケーションの作成

この章では、問合せアプリケーションの設計および作成に有効な *interMedia Text* の主な機能について説明します。この章で示す例は、出荷時のデフォルト動作に基づいています。

問合せアプリケーションを作成するには、*interMedia Text* に対応付けられたユーザーおよびロールを理解する必要があります。詳細は、次の「[interMedia Text のユーザーおよびロール](#)」を参照してください。

問合せアプリケーションを作成する一般的なステップは次のとおりです。

1. ドキュメントをロードします。この章の「[ドキュメントのロード](#)」を参照してください。
2. ドキュメントを索引付けします。この章の「[索引付け](#)」を参照してください。
3. 問合せを発行します。この章の「[問合せ](#)」を参照してください。
4. 問合せを満たすドキュメントを表示します。この章の「[ドキュメントの表示方法およびハイライト表示](#)」を参照してください。

後続の項では、これらの作業についての一般的な概要を説明します。

サンプル・コードのディレクトリ

サンプル・コードは、Oracle8i *interMedia Text* のインストレーションに含まれています。ブラウザで次のファイルをポイントすると、これらの例を表示できます。

```
$ORACLE_HOME/ctx/sample/api/index.html
```

これらのサンプル・コードは、*interMedia Text* での SQL および PL/SQL インタフェースを示します。これらの例は、*interMedia Text* の概念を十分に理解しているプログラマを対象としています。

この例には、SQL*Plus で実行する .sql ファイルおよびコマンドラインで実行する .sh ファイルが含まれます。

通常、任意の機能を説明するために、複数のファイルを順に実行します。それぞれのファイルには、HTML ページからのリンクを介してアクセスできます。これらのページには、必要に応じて詳しい説明が記載されています。

interMedia Text のユーザーおよびロール

すべてのユーザーは、*interMedia Text* の索引の作成および CONTAINS 問合せの発行を行うことができます。Oracle8i *interMedia Text* では、管理用の CTXSYS ユーザーおよびアプリケーション開発者用の CTXAPP ロールが提供されています。

CTXSYS ユーザー

CTXSYS ユーザーは、インストール時に作成されます。*interMedia Text* ユーザーをこのユーザーとして管理します。

CTXSYS ユーザーは、次のことを実行できます。

- システム定義プリファレンスの変更
- 他のユーザー設定項目の削除および変更
- PL/SQL パッケージ CTX_ADM にあるプロシージャのコールによる、サーバーの起動およびシステム・パラメータの設定
- ctxsrv サーバーの起動
- すべてのシステム定義ビューへの問合せ
- CTXAPP ロールを持つユーザーのすべての作業の実行

CTXAPP ロール

CTXAPP ロールは、システム定義のロールです。ユーザーは次のことを実行できます。

- *interMedia Text* のプリファレンスの作成および削除
- *interMedia Text* PL/SQL パッケージの使用

すべてのユーザーは、*interMedia Text* の索引の作成およびテキスト問合せの発行を行うことができます。CTXAPP ロールによって、ユーザーはプリファレンスを作成し、PL/SQL パッケージを使用できます。

ドキュメントのロード

デフォルトの索引付けの動作では、テキスト列にロードされたドキュメントを想定しています。

注意： システムは、ドキュメントがテキスト列にロードされることを想定していますが、ファイル・システムや URL など、他の方法で格納することもできます。

データ記憶域の詳細は、[第3章の「データストア・オブジェクト」](#)を参照してください。

サポートされている列型

デフォルトでは、システムはドキュメントがテキスト列にロードされることを想定しています。テキスト列には、VARCHAR2、CLOB、BLOB、CHAR または BFILE が使用できます。

注意： 使用すべきでない列型である LONG および LONG RAW へのデータの格納は、Oracle7 システムから Oracle8 へ移行する場合にのみサポートされています。

列型 NCLOB、DATE および NUMBER は索引付けされません。

サポートされているドキュメント形式

システムは HTML、PDF、Microsoft Word およびプレーン・テキストを含むほとんどのドキュメント形式を索引付けできるため、ユーザーはこれらのすべてのドキュメント・タイプをテキスト列へロードできます。

参照： サポートされるドキュメント形式の詳細は、[付録 C「サポートされているフィルタ形式」](#)を参照してください。

ロード方法

INSERT 文

SQL の INSERT 文を使用し、ドキュメントを表にロードできます。

次の例では、CREATE TABLE を使用して、id 列および text 列を持つ表を作成します。この例では、INSERT 文を使用して表を移入します。この例では id 列を主キーとしています。これは、テキスト表に必要な制約です。text 列は、VARCHAR2 です。

```
create table docs (id number primary key, text varchar2(80));
```

この表を移入するには、次の 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');
```

ファイル・システムからのテキスト・データのロード

INSERT 文に加え、Oracle では次のような自動化された方法を使用して、ファイル・システムから表にテキスト・データ（ドキュメント、ドキュメントに対するポインタおよび URL を含む）をロードできます。

- SQL*Loader
- ctxload
- BFILE から LOB をロードするための PL/SQL プロシージャ
DBMS_LOB.LOADFROMFILE()
- Oracle Call Interface

参照： SQL*Loader の使用方法を含むロードの例は、[付録 D「ロード例」](#)を参照してください。

ctxload の詳細は、[第 12 章の「ctxload」](#)を参照してください。

DBMS_LOB パッケージの詳細は、『Oracle8i PL/SQL パッケージ・プロシージャ リファレンス』を参照してください。

LOB の使用方法の詳細は、『Oracle8i アプリケーション開発者ガイド ラージ・オブジェクト』を参照してください。

Oracle Call Interface の詳細は、『Oracle8i コール・インタフェース・プログラマーズ・ガイド』を参照してください。

索引付け

テキスト・データが一度表にロードされ、表に主キーが含まれると、コマンドを実行してテキスト索引を作成できます。

たとえば、次のコマンドによって、docs 表内の text 列に myindex というテキスト索引を作成します。

```
create index myindex on docs(text) indextype is ctxsys.context;
```

すべての言語に対する汎用デフォルト

パラメータを明示的に指定せずに **CREATE INDEX** を使用すると、システムはデフォルトですべての言語に対して次のことを行います。

- 索引付けされるテキストは、テキスト列に直接格納されると想定します。テキストの列型は、CLOB、BLOB、BFILE、VARCHAR2 または CHAR です。列型 LONG および LONG RAW は、Oracle7 システムから Oracle8i へ移行する場合はサポートされます。列型 NCLOB、DATE および NUMBER は索引付けできません。
- 列型を検出し、バイナリの列型に対してフィルタ処理を使用します。ほとんどのドキュメント形式はフィルタ処理でサポートされています。列がプレーン・テキストの場合、システムはフィルタ処理を使用しません。

注意： システムで正しくドキュメントをフィルタ処理するために、ご使用の環境が Inso フィルタをサポートするように正しく設定されていることを確認してください。

Inso フィルタを使用するための環境を構成する方法は、付録 C の「[Inso フィルタ処理テクノロジー](#)」を参照してください。

- 索引付けするテキストの言語は、データベース設定で指定した言語であるとみなします。
- データベース設定で指定した言語に対するデフォルトのストップリストを使用します。ストップリストは、索引付け時にシステムが無視するワードを識別します。
- ご使用の言語においてファジーおよびステミング問合せを使用可能にします（ご使用の言語にこの機能が使用可能な場合）。

ユーザー独自のプリファレンスを作成し、これらのカスタム・プリファレンスを CREATE INDEX のパラメータ文字列に指定することによって、いつでもデフォルトの索引付け動作を変更できます。

参照： ユーザー独自のカスタム・プリファレンスの作成の詳細は、[第3章「索引付け」](#)を参照してください。

[第7章](#)の「CTX_DDL.CREATE_PREFERENCE」も参照してください。

[CREATE INDEX](#)の使用の詳細は、[第2章](#)を参照してください。

言語固有のデフォルト

英語のデフォルト

汎用デフォルトに加えて、英語テキストでは、次のオプションが使用可能です。

- ドキュメント・テーマ情報の索引付け。テーマ情報が索引付けされた場合、ABOUT 問合せの精度が高くなります。

デンマーク語のデフォルト

汎用デフォルトに加えて、デンマーク語テキストでは、次のオプションが使用可能です。

- 代替スペル

オランダ語のデフォルト

汎用デフォルトに加えて、オランダ語テキストでは、次のオプションが使用可能です。

- コンポジット索引付け
- 代替スペル

フィンランド語、ノルウェイ語およびスウェーデン語のデフォルト

汎用デフォルトに加えて、フィンランド語、ノルウェイ語およびスウェーデン語のテキストでは、次のオプションが使用可能です。

- 代替スペル

ドイツ語およびドイツ工業規格語のデフォルト

汎用デフォルトに加えて、ドイツ語テキストでは、次のオプションが使用可能です。

- 大 / 小文字が区別される索引付け

- コンポジット索引付け
- 代替スペル

参照： これらのオプションの詳細は、[第 3 章の「BASIC_LEXER」](#)を参照してください。

索引のメンテナンス

索引のメンテナンスは、アプリケーションがベース表でドキュメントを挿入、更新または削除した後が必要です。

ベース表が静的である場合（初期の索引付け以降にドキュメントを更新、挿入または削除していない場合）、索引をメンテナンスする必要はありません。

ただし、ベース表上で DML（挿入、更新または削除）を実行した場合は、索引を更新する必要があります。[ALTER INDEX](#) を使用して、索引を手動で同期化できます。また、ctxsrv サーバーをバックグラウンドで実行し、一定の間隔で自動的に索引を同期化することもできます。

参照： 索引の同期化の詳細は、[第 2 章の「ALTER INDEX」](#)を参照してください。

ctxsrv の詳細は、[第 12 章の「ctxsrv」](#)を参照してください。

問合せ

SELECT 文で CONTAINS 演算子を使用して、テキスト問合せを発行します。CONTAINS を使用すると、次の 2 種類の問合せを発行できます。

- ワード問合せ
- ABOUT 問合せ

また、応答時間短縮のために問合せを最適化することもできます。次の項で、これらの問合せについての概要を説明します。

ワード問合せの例

ワード問合せでは、CONTAINS 演算子で、一重引用符で囲んで入力したワードまたは句の完全一致を問い合わせます。

次の例は、ワード *oracle* を含む *text* 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 news
       WHERE CONTAINS(text, 'oracle', 1) > 0;
```

問合せ式では、異なる結果を得るために、AND や OR のようなテキスト演算子を使用できます。また、構造化述語を WHERE 句に追加することもできます。

参照： 問合せに使用可能な異なる演算子の詳細は、[第 4 章「問合せ演算子」](#)を参照してください。

COUNT(*) または CTX_QUERY.COUNT_HITS を使用して、問合せのヒット数をカウントできます。

ABOUT 問合せの例

すべての言語では、ABOUT 問合せによって、問合せから戻される関連ドキュメント数が増加します。

英語では、ABOUT 問合せは、デフォルトで作成された索引のテーマ・コンポーネントを使用できます。したがって、この演算子は指定したワードまたは句の完全一致のみでなく、問合せの概念に基づいてドキュメントを戻します。

たとえば、次の問合せでは、単に *politics* というワードを含むドキュメントのみでなく、*text* 列にある、*politics* という主題に関するすべてのドキュメントを検索します。

```
SELECT SCORE(1) title from news
```

```
WHERE CONTAINS(text, 'about(politics)', 1) > 0;
```

参照： ABOUT 演算子の詳細は、[第 4 章「問合せ演算子」](#)の「[ABOUT](#)」を参照してください。

応答時間短縮のための問合せの最適化

すべての CONTAINS 問合せ（ワードまたは ABOUT での）は、結果セット内で最もランクが高いヒットを最短時間で取り出せるように、応答時間について最適化できます。応答時間についての最適化は、Web ベースの検索アプリケーションで有効です。

応答時間について最適化するには、FIRST_ROWS ヒントを使用します。たとえば、次の PL/SQL ブロックでは、カーソルを使用して問合せの最初の 20 ヒットを取り出し、FIRST_ROWS ヒントを使用して応答時間を短縮するように最適化しています。

```
declare
cursor c is
select /*+ FIRST_ROWS */ pk, score(1), col from ctx_tab
      where contains(txt_col, 'oracle', 1) > 0 order by score(1) desc;
begin
for i in c
loop
insert into t_s values(i.pk, i.col);
exit when c%rowcount > 21;
end loop;
end;
```

参照： 問合せの最適化の詳細は、[付録 A「問合せのチューニング」](#)を参照してください。

その他の問合せ機能

問合せアプリケーションでは、セクション検索などのその他の問合せ機能を使用できます。次の表に、機能およびその詳細のこのマニュアルでの記載箇所を示します。

機能	詳細の記載箇所
セクション検索	セクションの定義については、 第 7 章「CTX_DDL パッケージ」 を参照してください。 セクション内の検索については、 第 4 章の「WITHIN」 演算子を参照してください。
近接検索	第 4 章の「NEAR (;)」 演算子を参照してください。

機能	詳細の記載箇所
STEM および FUZZY 検索	問合せの発行については、第 4 章の「 STEM (\$) 」演算子および「 FUZZY (?) 」演算子を参照してください。 言語オプションの設定については、第 3 章の「 ワードリスト・オブジェクト 」を参照してください。
シソーラス問合せ	問合せでのシソーラス演算子の使用方法は、第 4 章「 問合せ演算子 」を参照してください。シソーラス演算子には、SYN、BT および NT が含まれます。 シソーラスのブラウズ方法および変更方法については、第 11 章「 CTX_THES パッケージ 」を参照してください。 シソーラスのロードについては、第 12 章の「 ctxload 」を参照してください。
大 / 小文字を区別する検索 基本文字変換 ワード分割処理（ドイツ語およびオランダ語） 代替スペル（ドイツ語、オランダ語およびスウェーデン語）	これらの機能を使用可能にする方法については、第 3 章の「 レクサー・オブジェクト 」を参照してください。
問合せの最適化	付録 A「 問合せのチューニング 」を参照してください。
問合せ実行計画	第 10 章の「 CTX_QUERY.EXPLAIN 」プロシージャを参照してください。
階層問合せフィードバック	第 10 章の「 CTX_QUERY.HFEEDBACK 」プロシージャを参照してください。

ドキュメントの表示方法およびハイライト表示

通常、テキスト問合せアプリケーションによって、問合せから戻されるドキュメントを表示できます。ユーザーがヒットリストからドキュメントを選択すると、アプリケーションがドキュメントをある形式で表示します。

interMedia Text を使用すると、ドキュメントを異なる方法で表示できます。たとえば、問合せ語句をハイライト表示させてドキュメントを表示できます。ハイライト表示される問合せ語句は、英語でのワード問合せのワードまたは ABOUT 問合せのテーマのいずれかです。

また、CTX_DOC PL/SQL パッケージを使用して、ドキュメントからテーマ情報を取得することもできます。

表 1-1 に、ユーザーが取得可能な異なる出力、および出力を得るためのプロシージャを示します。

表 1-1

出力	プロシージャ
ハイライト表示されたドキュメント、プレーン・テキスト形式	CTX_DOC. MARKUP
ハイライト表示されたドキュメント、HTML 形式	CTX_DOC. MARKUP
プレーン・テキスト形式のハイライト・オフセット情報	CTX_DOC. HIGHLIGHT
HTML 形式のハイライト・オフセット情報	CTX_DOC. HIGHLIGHT
プレーン・テキスト形式、ハイライトなし	CTX_DOC. FILTER
ドキュメントの HTML 形式、ハイライトなし	CTX_DOC. FILTER
ドキュメントのテーマ・サマリーおよび要旨	CTX_DOC. GIST
ドキュメントにおけるテーマのリスト	CTX_DOC. THEMES

参照： これらの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。

SQL コマンド

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 コマンドについて説明します。

この章で説明するコマンドは、次のとおりです。

- [ALTER INDEX](#)
- [DROP INDEX](#)
- [CONTAINS](#)
- [CREATE INDEX](#)
- [SCORE](#)

ALTER INDEX

注意： この項では、テキスト・ドメイン・インデックスの管理に関連する ALTER INDEX コマンドについて説明します。

ALTER INDEX コマンドの詳細は、『Oracle8i 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、テキスト索引に対して次のメンテナンス作業を実行します。

- 索引の改名（「[RENAME 構文](#)」を参照）
- 異なるプリファレンスを使用した索引の再構築（「[REBUILD 構文](#)」を参照）
- 失敗した索引処理（作成 / 最適化）の再開（「[REBUILD 構文](#)」を参照）
- バッチでの DML 処理（同期化）（「[REBUILD 構文](#)」を参照）
- 索引の最適化（「[REBUILD 構文](#)」を参照）
- 索引へのストップワードの追加（「[REBUILD 構文](#)」を参照）
- 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）

RENAME 構文

次の構文は、索引の改名に使用します。

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

schema.index_name

改名する索引名を指定します。

new_index_name

schema.index に、新しい名前を指定します。*new_index_name* パラメータは、25 文字以内で設定します。26 文字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* の文字数が 26 文字以上 30 文字未満の場合、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 コマンドで *new_index_name* を削除した後、*index_name* を再作成して削除してください。

REBUILD 構文

次の構文を使用して、索引の再構築、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、または索引の最適化を実行します。

```
ALTER INDEX [schema.]index REBUILD [online] [parameters (paramstring)];
```

[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 の同期化操作または最適化操作中に索引を問い合わせることができます。replace、resume、またはストップワードや停止セクションの追加時には、*online* は指定できません。

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]

                | resume [memory memsize]
                | optimize [fast | full [maxtime (time | unlimited)]]
                | sync [memory memsize]
                | add stopword word
                | 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。

参照： 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および設定の詳細は、[第 3 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。

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

索引を最適化します。*fast*（高速）最適化または *full*（完全）最適化のいずれかを指定します。

高速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、**高速モード**では、古いデータは削除されません。

完全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、行は圧縮され、古いデータは削除されます（ガベージ・コレクション）。

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。

sync [memory memsize]

索引を同期化します。オプションで、使用する実行時メモリー量を *memsize* で指定できます。

add stopword word

索引にストップワードの *word* を動的に追加します。

このコマンド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。

このコマンドによって索引が再構築されることはありません。

参照： セクションの追加に関する制約の詳細は、この項の「[セクション追加時の制約](#)」を参照してください。

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・セクションが参照できます。デフォルトは INVISIBLE です。

参照： VISIBLE なフィールド・セクションおよび INVISIBLE な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。索引付け済の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。

このコマンドによって索引が再構築されることはありません。

参照： セクションの追加に関する制約の詳細は、この項の「[セクション追加時の制約](#)」を参照してください。

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。

このコマンドによって索引が再構築されることはありません。

参照： セクションの追加に関する制約の詳細は、この項の「[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、かつ大 / 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。索引付け済の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。

このコマンドによって索引が再構築されることはありません。

参照： セクションの追加に関する制約の詳細は、この項の「[セクション追加時の制約](#)」を参照してください。

例

失敗した索引の再開

次のコマンドは、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

次のコマンドは、索引を再構築し、ストップリスト・プリファレンスを new_stop と置き換えます。

```
ALTER INDEX newsindex rebuild parameters('replace stoplist new_stop');
```

高速最適化

次のコマンドは、newsindex を高速モードで最適化します。

```
ALTER INDEX newsindex rebuild parameters('optimize fast');
```

完全最適化

最適化操作を 3 時間（180 分）継続するように指定するには、次のコマンドを発行します。

```
ALTER INDEX newsindex rebuild parameters('optimize full maxtime 180');
```

時間を考慮せず索引全体を最適化するには、次のコマンドを発行します。

```
ALTER INDEX newsindex rebuild parameters('optimize full maxtime unlimited');
```

索引全体を最適化し、最適化処理中に問合せを発行できるようにするには、次のコマンドを発行します。

```
ALTER INDEX newsindex rebuild online parameters('optimize full maxtime unlimited');
```

索引の同期化

次の例では、2MB の実行時メモリーで索引を同期化します。

```
ALTER INDEX newsindex rebuild PARAMETERS('sync memory 2M');
```

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、次のコマンドを発行します。

```
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、次のコマンドを発行します。

```
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、次のコマンドを発行します。

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

注意

メモリー考慮点

メモリー・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、または実行時メモリーが不足している場合に有効であることがあります。

索引エラーの表示

iMT ビューで、索引エラーを表示できます。ご使用の索引のエラーを表示するには、[CTX_USER_INDEX_ERRORS](#) を使用します。すべての索引のエラーを CTXSYS として表示するには、[CTX_INDEX_ERRORS](#) を使用します。

たとえば、索引に最後に発生したエラーを表示するには、次の文を発行します。

```
select err_timestamp, err_text from CTX_USER_INDEX_ERRORS order by err_timestamp
desc;
```

ビューを消去するには、次の文を発行します。

```
delete from CTX_USER_INDEX_ERRORS;
```

保留中の DML の表示

[CTX_USER_PENDING](#) ビューを使用すると、索引の同期が必要な、新しく更新、削除または挿入した行を表示できます。たとえば、すべての索引で保留中の DML を表示するには、次の文を発行します。

```
select pnd_index_name, pnd_rowid, to_char(pnd_timestamp, 'dd-mon-yyyy hh24:mi:ss')
timestamp from ctx_user_pending;
```

この文によって、次の形式の出力結果が戻されます。

PND_INDEX_NAME	PND_ROWID	TIMESTAMP
MYINDEX	AAADXnAABAAAS3SAAC	06-oct-1999 15:56:50

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 AUTO_SECTION_GROUP 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。

関連項目

第 7 章の「[CTX_DDL.CREATE_PREFERENCE](#)」

第 7 章の「[CTX_DDL.CREATE_STOPLIST](#)」

第 7 章の「[CTX_DDL.CREATE_SECTION_GROUP](#)」

[CREATE INDEX](#)

[DROP INDEX](#)

DROP INDEX

注意： この項では、テキスト・ドメイン・インデックスの削除に関連する DROP INDEX コマンドについて説明します。

DROP INDEX コマンドの詳細は、『Oracle8i 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、索引のステータス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](#)

CONTAINS

用途

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。

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

列の検索を定義する問合せ式を指定します。

参照： 問合せ式で利用できるテキスト演算子の詳細は、[第 4 章「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

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 *text* 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、**SCORE** 演算子でラベル 1 を使用して選択されます。

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

CONTAINS 演算子の後には、常に >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 演算子がコールされる時（たとえば **SELECT** 句で）、次の例に示すように、**SCORE** 演算子がラベル値を参照するようにしてください。

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

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、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、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。

関連項目

[SCORE](#)

[付録 A 「問合せのチューニング」](#)

CREATE INDEX

注意： この項では、テキスト・ドメイン・インデックスの作成に関連する CREATE INDEX コマンドについて説明します。

CREATE INDEX コマンドの詳細は、『Oracle8i 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、*interMedia Text* の索引を作成します。*interMedia Text* の索引は、拡張索引作成フレームワークを使用して作成された *conText* 型の Oracle ドメイン・インデックスです。

CONTAINS 問合せを発行するには、iMT 索引を作成する必要があります。

構文

```
CREATE INDEX [schema.]index on [schema.]table(column) INDEXTYPE IS ctxsys.context  
[PARAMETERS (paramstring)] [PARALLEL N];
```

[*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。表に主キー制約がある必要があります。これは主に、ドキュメント・サービス用にドキュメントを識別するために必要です。コンポジット主キーは、16 列までサポートされます。

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

注意： 使用すべきでない列型 LONG および LONG RAW の索引付けは、Oracle7 のシステムを Oracle8i に移行する処理のために、サポートされています。

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。

コンポジット索引はサポートされていません。列リストに指定できるのは1列のみです。

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。iMT でパラレル索引付けを使用するには、パーティション化されたテキスト表が必要です。各パラレル・インデクサーはそれぞれのパーティションを処理するため、パーティションの数が N を超えないようにする必要があります。

また、パラレル索引付けを使用するには、init.ora ファイル内の初期化パラメータ JOB_QUEUE_PROCESSES および JOB_QUEUE_INTERVAL を設定する必要があります。指定する並列度 N は、JOB_QUEUE_PROCESSES より小さい値である必要があります。

パラレルで索引を作成する方法の詳細は、後述の例を参照してください。

参照： これらの初期化パラメータの詳細は、『Oracle8i リファレンス・マニュアル』および『Oracle8i 管理者ガイド』を参照してください。

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。

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 3 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。[第 3 章の「データストア・オブジェクト」](#)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。[第 3 章の「フィルタ・オブジェクト」](#)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。charset 列は、text 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 / 小文字の区別はありませんが、JA16EUC のような NLS キャラクタ・セット文字列である必要があります。

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 オブジェクトおよび CHARSET フィルタ・オブジェクト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、この列を使用してください。

format column *format_column_name*

format 列の名前を指定します。format 列は、text 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 format 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。

format 列では、TEXT または BINARY のいずれかを指定できます。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 オブジェクトにサポートされた形式であることを示します。format 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。

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。第3章の「[レクサー・オブジェクト](#)」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、`language` 列の名前を指定します。第3章の「[MULTI_LEXER](#)」を参照してください。

この列は、ベース表内に必要があります。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。`language` 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： `language` 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。第3章の「[ワードリスト・オブジェクト](#)」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。第3章の「[記憶域オブジェクト](#)」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。第7章の「[CTX_DDL.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。第7章の「[CTX_DDL.CREATE_SECTION_GROUP](#)」を参照してください。

memory *memsize*

索引付けに使用する実行時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]
```

ここで、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](#) ビューで *max_index_memory* に指定された値の間である必要があります。デフォルトは、[CTX_PARAMETERS](#) で *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、索引付けの処理過程を追跡する場合、またはランタイム記憶域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、ベース表を更新またはベース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、ベース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旨の出力のみを必要とする場合にも作成します。

例

デフォルトのプリファレンスを使用した索引作成

次の例では、mytable 内の docs 列に、myindex というテキスト索引を作成します。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;
```

参照： デフォルト設定の詳細は、[第3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、[第1章の「索引付け」](#)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索引作成

次の例では、mytable 内の docs 列に、myindex というテキスト索引を作成します。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索引作成

マルチレクサーは、各行にどのレクサーを使用するかを、`language` 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 `language` 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');
```

参照： マルチレクサー・プリファレンスの作成の詳細は、[第3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。

1. ジョブ・キュー・パラメータを Oracle 初期化ファイルに設定し、パラレル索引付けを使用可能にします。

```
job_queue_processes = 8
job_queue_interval = 4
```

JOB_QUEUE_PROCESSES パラメータは、ご使用の並列度以上に設定する必要があります。

この例では、CREATE INDEX コマンドの発行から 4 秒以内に索引付けジョブを開始します。索引付け操作を即時に行う必要がない場合、JOB_QUEUE_INTERVAL に設定する秒数を大きくできます。

参照： これらの初期化パラメータの詳細は、『Oracle8i リファレンス・マニュアル』および『Oracle8i 管理者ガイド』を参照してください。

- パーティション化されたテキスト表を作成します。次の例では、pk 列内の値をハッシュ・キーとして使用して、ハッシュ・パーティション表を作成します。

```
create table mypart_tab(pk number primary key, text varchar2(80))
PARTITION BY HASH (pk) PARTITIONS 8;
```

- 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 mypart_tab(pk) indextype is ctxsys.context parallel 3;
```

注意

発行ユーザーには、索引を作成するために CTXAPP ロールは必要ありません。列に B-tree 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-tree 索引では、標準の動作です。

索引エラー

iMT ビューで、索引エラーを表示できます。ご使用の索引のエラーを表示するには、[CTX_USER_INDEX_ERRORS](#) を使用します。すべての索引のエラーを CTXSYS として表示するには、[CTX_INDEX_ERRORS](#) を使用します。

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;
```

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;
```

関連項目

[第 7 章の「CTX_DDL.CREATE_PREFERENCE」](#)

[第7章の「CTX_DDL.CREATE_STOPLIST」](#)

[第7章の「CTX_DDL.CREATE_SECTION_GROUP」](#)

[ALTER INDEX](#)

[DROP INDEX](#)

SCORE

SELECT 文で SCORE 演算子を使用して、テキスト問合せの [CONTAINS](#) 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。

注意

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

例

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 G 「スコア付けのアルゴリズム」](#)

この章では、*interMedia Text* の索引の作成に使用できる索引プリファレンス・オブジェクト、セクション・グループ・タイプおよびストップリスト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・オブジェクト
- フィルタ・オブジェクト
- レクサー・オブジェクト
- ワードリスト・オブジェクト
- 記憶域オブジェクト
- セクション・グループのタイプ
- 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、各プリファレンス設定用のオプションについて説明します。オプションを使用可能にするには、この章で説明するオブジェクトの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](#) オブジェクトを使用して、mydatastore というデータストア・プリファレンスを作成し、[CREATE INDEX](#) のパラメータ文字列にデータストア・プリファレンスとして mydatastore を指定できます。

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、この章で説明するオブジェクトの1つを指定します。いくつかのオブジェクトには、CTX_DDL.[SET_ATTRIBUTE](#) 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。

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。

データストア・オブジェクト

データストア・オブジェクトを使用して、テキストの格納方法を指定します。データ記憶域プリファレンスを作成するには、次のオブジェクトの1つを使用する必要があります。

オブジェクト	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 各列は単一のドキュメントとして索引付けされます。
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	データは、NESTED TABLE に格納されます。
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 ユニフォーム・リソース・ロケータ (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 オブジェクトを使用します。DIRECT_DATASTORE には、属性はありません。

DIRECT_DATASTORE 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・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');
```

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` オブジェクトを使用します。この場合、マスター表にテキストキー列があります。

`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 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

ディテール表 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	CLOB	ドキュメント・テキスト

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 で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。次の例では、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;
```

索引 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 / デティ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 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 オブジェクトは、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	<i>path1:path2:...:pathn</i>

path

ファイル・システムに格納されているテキスト・ファイルの位置を指定します。

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。*path* をファイルへのパスの指定に使用しない場合は、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必要です。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 オブジェクトを使用します。

- 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）。

[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。例を示します。

`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。ここで、*xx* は特殊文字の 16 進コードを表します。

注意： `login:password@` 構文は、URL 内ではサポートされません。

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 の数を指定します。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、 <i>hostname:port</i> 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、 <i>hostname:port</i> 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。カンマで区切っ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

接続や読み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 が最大長より長いと、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 オブジェクトで利用できるメモリー・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。

maxdocsize

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

interMedia 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

interMedia 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 オブジェクトを使用して、索引付け時にドキュメントを合成するストア・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名を指定します。 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。CLOB、BLOB または VARCHAR2 のいずれかを指定できます。デフォルトは CLOB 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。この仕様は、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パラメータが必要です。

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

1 つ目の引数 *r* は、ROWID 型である必要があります。2 つ目の引数 *c* は、*output_type* に定義されている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・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。カレント行の 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（型は *output_type* で指定）に書き込む必要があります。

制約 次の制約が、プロシージャに適用されます。

- プロシージャの所有者は、CTXSYS である必要があります。

- プロシージャは、索引所有者が実行できる必要があります。
- プロシージャは、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。
- プロシージャは、安全なコールアウトになることはできません。また、安全なコールアウトをコールすることもできません。

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 または VARCHAR2 のいずれかを指定できます。

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
    offset number := 1;
begin
    for c1 in (select author, title, text from articles
               where rowid = rid)
    loop
        append_varchar_to_lob(tlob, c1.title, offset);
        append_varchar_to_lob(tlob, 'by ' || c1.author, offset);
        dbms_lob.append(tlob, 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。append_varchar_to_lob プロシージャが連結を行うとします。

CTXSYS が所有するストアド・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。

```
ctx_ddl.create_preference('myud', 'user_datastore');
ctx_ddl.set_attribute('myud', 'procedure', 's_myproc');
```

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。

NESTED_DATASTORE

NESTED_DATASTORE 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含まれる表名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 <code>DETAIL_LINENO</code> に似ています。この属性は必須です。
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 <code>DETAIL_TEXT</code> に似ています。この属性は必須です。テキスト属性には、 <code>LONG</code> または <code>LONG RAW</code> 以外のどの索引付け可能な型でも指定できます。 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	<code>FALSE</code> 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。 <code>TRUE</code> 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 <code>FALSE</code> 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ので、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 がダミー列に伝播されるには、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 `lineno` と `text` に対して指定した属性が NESTED TABLE に存在するかどうかチェックされます。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 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 用の記憶表 myntab を作成します。

その後、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 tab  
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わせます。

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

フィルタ・オブジェクト

フィルタ・オブジェクトを使用して、索引付け用のテキストのフィルタ処理方法を決定する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、次のオブジェクトの 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・オブジェクト		説明
CHARSET_FILTER		キャラクタ・セット変換フィルタ
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
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フィルタ

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	ソース・キャラクタ・セットの NLS 名を指定します。 日本語キャラクタ・セットの自動検出には 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検出し、データベースのキャラクタ・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。

参照： サポートされている NLS キャラクタ・セットの詳細は、『Oracle8i NLS ガイド』を参照してください。

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、日本語テキスト列は、JA16EUC および JA16SJIS キャラクタ・セットのドキュメントを格納します。

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・セット列を持つベース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、データベースのキャラクタ・セットに変換され、索引付けされます。

キャラクタ・セット変換作業は、CHARSET_FILTER で機能します。charset 列が NULL、または認識されない場合、ソース・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： ドキュメント形式列が 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 `charset` 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・テキストの日本語ドキュメントを EUC で挿入し、キャラクタ・セットを指定します。

```
insert into hdocs values(1, 'text', 'JA16EUC', '/docs/tekusuto.euc');
insert in hdocs values (2, 'text', 'JA16SJIS', '/docs/tekusuto.sjs');
```

索引を作成し、`charset` 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 フィルタは、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。単一ドキュメント形式または複合ドキュメント形式の列の索引付けに使用します。INSO_FILTER には属性がありません。

参照： 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については、[付録 C「サポートされているフィルタ形式」](#)を参照してください。

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、パラメータ文字列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');
```

プレーン・テキストまたは HTML のフィルタ処理のバイパス

複合書式列は、1 つ以上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、複合書式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。

ベース表内の書式列では、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（charset 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、ベース表に書式列を作成する必要があります。

次に例を示します。

```
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、書式列に 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、書式列に 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');
```

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書式列名を指定します。

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

書式列に TEXT または BINARY を指定しない場合、BINARY が使用されます。

注意： INSO_FILTER を使用している場合は、CREATE INDEX に書式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 charset 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。

charset 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、charset 列を指定する必要はありません。

charset 列を指定して、書式列を指定しない場合、INSO_FILTER は [USER_FILTER](#) のように機能します。ただし、この場合は、日本語キャラクタ・セットの自動検出は行われません。

参照：「[USER_FILTER](#)」を参照してください。

プレーン・テキストの索引付けおよび INSO_FILTER

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、お薦めしません。

表全体にテキスト・ドキュメントが含まれている場合は、[INSO_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、書式列を作成し、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、`charset` 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、書式列および `charset` 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` テクノロジーのキャラクタ・セット制限が適用されます。特に、ご使用のアプリケーションで、次の条件のいずれかが真であ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・セットが次のうちの1つ。
 - `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かではなく、ドキュメントのキャラクタ・セットが `WE8ISO8859P1`。

NULL_FILTER

NULL_FILTER オブジェクトは、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、Inso フィルタを使用しないことをお勧めします。かわりに、フィルタ・プリファレンスに NULL_FILTER を使用します。

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');
```

参照： HTML ドキュメントの索引付けの詳細は、この章の「[セクション・グループのタイプ](#)」を参照してください。

USER_FILTER

USER_FILTER オブジェクトを使用して、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	フィルタ実行可能ファイル

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 ディレクトリに置く必要があります。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限り、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');
```


レクサー・オブジェクト

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・オブジェクトを作成するには、次のオブジェクトの1つを使用する必要があります。

オブジェクト	説明
BASIC_LEXER	シングルバイト・キャラクタ・セット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
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
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

BASIC_LEXER

BASIC_LEXER オブジェクトを使用して、英語およびサポートされているその他すべてのシングルバイト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。

BASIC_LEXER を使用して、拡張キャラクタ・セットを持つシングルバイト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： テキスト索引付け（文字の削除、基本文字変換など）の前にトークンに対して行われるすべての変更は、テキスト問合せ内の問合せ語句に実行されます。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、すべてのシングルバイト・キャラクタ・セットの他に、UTF8 もサポートします。

BASIC_LEXER には、次の属性があります。

属性	属性値
continuation	文字（文字列）
numgroup	文字（文字列）
numjoin	文字（文字列）
printjoin	文字（文字列）
punctuation	文字（文字列）
skipjoin	文字（文字列）
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）

属性	属性値
composite	DEFAULT (コンポジット・ワードの索引付けなし、デフォルト値)
	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_themes	YES (使用可能)
	NO (使用禁止)
index_text	YES (使用可能)
	NO (使用禁止)
theme_language	AUTO (デフォルト)
	ENGLISH
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)

注意： BASIC_LEXER オブジェクトの属性が文字列を使用する場合、複数の文字が指定できます。文字列の各文字は、属性のタイプに対する個別の文字として機能します。

たとえば、*printjoin* 属性に文字列 `*_.` を指定した場合、文字列の個々の文字 (`*`、`_`、`.`、`'`、`'`) は結合文字とみなされ、一致する文字が含まれているトークンの索引エントリに結合文字として組み込まれます。

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 NLS 初期化パラメータによって決定されます。

通常、BASIC_LEXER オブジェクトのレクサー・プリファレンスを作成する場合、*numjoin* または *numgroup* のいずれかに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、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、文章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、テキスト索引付けの前にトークンから削除されます。ただし、*punctuation* 文字が *printjoin* 文字としても定義されている場合、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、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* 文字と組み合わせて使用し、文章 / 段落検索用の文章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* 文字 *lendjoin* 文字は、トークンの最初 / 最後에만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、文章 / 段落検索用の文章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* 文字と組み合わせて使用します。

whitespace の事前定義済みのデフォルト値は、空白およびタブです。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、文章 / 段落検索用の文章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* 文字と組み合わせて使用します。

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

発音符号（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

レクサーによって、テキスト索引のエントリ内のトークンがすべて大文字に変換されるか、テキストに表記されているとおりに大 / 小文字を区別して格納されるかを指定します。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、テキスト問合せを、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のいずれかに対して、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。

注意： 基本文字の索引付けは、コンポジット・ワードの索引付けがドイツ語またはオランダ語に設定されている場合は、常にオフです。

大 / 小文字の区別は、*composite* がドイツ語に設定されている場合は、常にオンです。

大 / 小文字の区別は、*composite* がオランダ語に設定されている場合は、オンまたはオフにできます。

index_themes

英語のテーマ情報の索引付けに 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。

theme_language

index_themes が YES に設定されている場合、テーマ生成にどの知識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、このパラメータを設定しても、何も影響はありません。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。

注意： リリース 8.1 では、英語によるテーマの生成のみが、システムによってサポートされます。

index_text

ワード情報の索引付けに 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 YES 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。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。

参照： Oracle が使用する代替スペルの規則については、[付録 F「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 *printjoins* 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

このレクサーは、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。

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、CTX_DDL.ADD_SUB_LEXER を使用して言語固有の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。また、マルチ言語表を索引付けするには、ベース表に言語列がある必要があります。言語列は、CREATE INDEX を使用して索引付けするときに指定します。例を参照してください。

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、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 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、パラメータ文字列にマルチレクサー・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、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、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 オブジェクトは、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。

データベースのキャラクタ・セットが次のいずれかである場合、このレクサーを使用できます。

- ZHS16CGB231280
- ZHS16GBK
- ZHT32EUC
- ZHT16BIG5
- ZHT32TRIS
- AL24UTFFSS

- UTF8

JAPANESE_VGRAM_LEXER

JAPANESE_VGRAM_LEXER オブジェクトは、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。

データベースのキャラクタ・セットが次のいずれかである場合、このレクサーを使用できません。

- JA16SJIS
- JA16EUC
- UTF8

KOREAN_LEXER

KOREAN_LEXER オブジェクトは、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。

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場合、このレクサーを使用できません。

- KO16KSC5601
- UTF8

KOREAN_LEXER を使用する場合は、次のブール属性を指定します。

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。デフォルト値は 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。デフォルト値は 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。デフォルト値は 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。デフォルト値は 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。デフォルト値は 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。

属性	属性値
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。デフォルト値は 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。デフォルト値は TRUE です。
morpheme	語形分析に対して、TRUE または FALSE を指定します。デフォルト値は 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。デフォルト値は 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。デフォルト値は 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。

ワードリスト・オブジェクト

ワードリスト・プリファレンスを使用して、ご使用の言語に対するステミングやファジー・マッチングなどの拡張問合せオプションを使用可能にします。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。これが使用可能な唯一のオブジェクトです。

BASIC_WORDLIST

BASIC_WORDLIST オブジェクトを使用して、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ます。

参照： STEM および FUZZY 演算子の詳細は、[第 4 章「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 3-1

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検出）

表 3-1 (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検出）
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。ファジー・スコアを設定すると、この数値未満のスコアは生成されません。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 ～ 5000 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、サブ文字列索引が作成されます。サブ文字列索引を使用すると、 <i>%ing</i> や <i>%benz%</i> のようなレフト・トランケーテッドおよびダブル・トランケーテッド・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。

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、デフォルト値は NULL です。NULL ステマーを使用すると、\$ 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。

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。

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_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。ファジー・スコアを設定すると、この数値未満のスコアは生成されません。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接しているかという計測値であり、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、最も適合するワードの特定の数値に拡張の度合いを制限できます。

substring_index

TRUE を指定すると、サブ文字列索引が作成されます。サブ文字列索引を使用すると、*%ing* や *%benz%* のようなレフト・トランケーテッドまたはダブル・トランケーテッド・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。

サブ文字列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、次のような影響が及ぼされます。

- 索引作成および DML 処理が、最大 4 倍遅くなります。
- 作成するサブ文字列索引のサイズは、ワード表上の \$X 索引とほぼ同じサイズになります。
- *substring_index* を使用可能して索引を作成すると、サブ文字列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。索引作成時に、通常の 2 倍のロールバックを使用可能にするか、または索引メモリーを減らすことをお勧めします。

BASIC_WORDLIST の例

次の例では、英語に対するステミングおよびファジー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、ダブル・トランケートド検索のパフォーマンスが向上するように、サブ文字列索引がシステムによって作成されます。

```
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');
```


記憶域オブジェクト

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。このシステムには、BASIC_STORAGE という単一の記憶域オブジェクトがあります。

オブジェクト	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オブジェクト

BASIC_STORAGE

BASIC_STORAGE オブジェクトは、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せん。

ただし、索引構成表句は指定しないでください。

参照： CREATE TABLE 句および CREATE INDEX 句の指定方法については、『Oracle8i SQL リファレンス』のコマンド構文を参照してください。

BASIC_STORAGE には、次の属性があります。

BASIC_STORAGE 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。

BASIC_STORAGE 属性	属性値
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。
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。
p_table_clause	BASIC_WORDLIST で SUBSTRING_INDEX を使用可能にしている場合の、サブ文字列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、テキスト索引表は IUSER 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を持つ 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・グループのタイプ

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](#) のパラメータ文字列で指定します。

次のグループのタイプの 1 つを指定して、CTX_DDL.[CREATE_SECTION_GROUP](#) でセクション・グループを作成できます。

セクション・グループ・リファレンス	説明
NULL_SECTION_GROUP	これはデフォルトです。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に、このグループ・タイプを使用します。
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 という形式のセクションを定義します。
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。
AUTO_SECTION_GROUP	<p>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。</p> <p>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、<i>attribute@tag</i> という形式でネーミングされます。</p> <p>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。</p> <p>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。</p> <ul style="list-style-type: none">■ 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。■ 自動セクション化では、XML ドキュメント・タイプ（ルート要素）は索引付けされません。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。■ プレフィ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。
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

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;
```

CTX_DDL.ADD_SECTION 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

次のコマンドは、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 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');
```

ストップリスト

デフォルトでは、データベース設定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。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。

Oracle8i *interMedia* 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。

参照： 提供されるストップリストの詳細は、[付録 E「提供されるストップリスト」](#)を参照してください。

ストップリストの作成

CTX_DLL.CREATE_STOPLIST を使用して、独自のストップリストを作成できます。

独自のストップリストを作成する場合、このリストを CREATE INDEX のパラメータ文字列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、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` を使用してから再索引付けしてください。

システム定義プリファレンス

interMedia 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。これらのプリファレンスは [CREATE INDEX](#) のパラメータ文字列に使用でき、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（次の項を参照）は、この項で説明するシステム定義のプリファレンスのいくつかにマップされます。

参照： デフォルトの索引パラメータの詳細は、この章の「[デフォルトの索引付けパラメータ](#)」を参照してください。

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](#) オブジェクトを使用します。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、索引を作成するために使用します。

CTXSYS.FILE_DATASTORE

このプリファレンスは、[FILE_DATASTORE](#) オブジェクトを使用します。

CTXSYS.URL_DATASTORE

このプリファレンスは、[URL_DATASTORE](#) オブジェクトを使用します。

フィルタ

CTXSYS.NULL_FILTER

このプリファレンスは、[NULL_FILTER](#) オブジェクトを使用します。

CTXSYS.INSO_FILTER

このプリファレンスは、[INSO_FILTER](#) オブジェクトを使用します。

レクサー

CTXSYS.DEFAULT_LEXER

このプリファレンスは、データベース設定時に指定した言語に必要なレクサーをデフォルトで設定します。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 ご使用の言語が英語である場合、このプリファレンスは *index_themes* 属性を使用可能にして [BASIC_LEXER](#) を使用します。ドキュメント・テーマ情報は、アメリカ英語およびイギリス英語に対して索引付けされます。

デンマーク語での設定 ご使用の言語がデンマーク語である場合、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 ご使用の言語がオランダ語である場合、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)
- 代替スペル (*alternate_spelling* 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 ご使用の言語がドイツ語である場合、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 ご使用の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である場合、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 ご使用の言語が日本語である場合、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 ご使用の言語が韓国語である場合、このプリファレンス **KOREAN_LEXER** を使用します。KOREAN_LEXER 用のすべての属性が使用可能になります。

簡易中国語での設定 ご使用の言語が簡易中国語である場合、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。

参照： これらのオプションの詳細は、この章の「**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 オブジェクトを使用します。

CTXSYS.HTML_SECTION_GROUP

このプリファレンスは、HTML_SECTION_GROUP オブジェクトを使用します。

CTXSYS.AUTO_SECTION_GROUP

このプリファレンスは、AUTO_SECTION_GROUP オブジェクトを使用します。

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、データベース設定時に指定した言語のストップリストをデフォルトで設定します。

参照： 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 E「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

この記憶域プリファレンスは、[BASIC_STORAGE](#) オブジェクトを使用します。

ワードリスト

CTXSYS.DEFAULT_WORDLIST

このプリファレンスは、データベース設定時に指定した言語に対して言語ステマーを使用します。ご使用の言語が[表 3-1](#)に記載されていない場合、このプリファレンスのデフォルトは、NULL ステマーおよび GENERIC ファジー・マッチング属性です。

システム・パラメータ

概要

interMedia 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 のパラメータ文字列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 に設定します。 「CTX_DOC. SET_KEY_TYPE 」も参照してください。

システムのデフォルト値は、[CTX_PARAMETERS](#) ビューで表示できます。
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

[CREATE INDEX](#) のパラメータ文字列でプリファレンスを指定しない場合、次のデフォルトのパラメータが使用されます。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 のパラメータ文字列に、データストア・プリファレンスが指定されない場合	CTXSYS.DEFAULT_DATASTORE
DEFAULT_FILTER_FILE	CREATE INDEX のパラメータ文字列にフィルタ・プリファレンスが指定されず、次の条件のいずれかが真となる場合 <ul style="list-style-type: none"> ■ ファイルが外部ファイル (BFILES) 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 のパラメータ文字列に、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 のパラメータ文字列に、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 のパラメータ文字列に、セクション・グループが指定されず、次の条件のいずれかが真となる場合 <ul style="list-style-type: none"> ■ 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
DEFAULT_SECTION_TEXT	CREATE INDEX のパラメータ文字列に、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 のパラメータ文字列に、記憶域プリファレンスが指定されていない場合	CTXSYS.DEFAULT_STORAGE

システム・パラメータ	使用する場合	デフォルト値
DEFAULT_LEXER	CREATE INDEX のパラメータ文字列に、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 のパラメータ文字列に、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 のパラメータ文字列に、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](#) ビューで表示できます。

デフォルト値の変更

[CTX_ADM.SET_PARAMETER](#) プロシージャを使用してデフォルト値を変更し、デフォルトとして使用する別のプリファレンス（カスタムまたはシステム定義）に名前を付けることができます。

問合せ演算子

この章では、演算子の優先順位を説明し、テキスト問合せ演算子の説明、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 (?)
- 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* が含まれているすべてのドキュメント、および *w2* と *w3* の両方が含まれている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* の両方が含まれているすべてのドキュメント、および *w3* が含まれている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 演算子より高い優先順位を持つことを示しています。
- 5 番目の例は、AND 演算子が WITHIN 演算子より低い優先順位を持つ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限り、大カッコ内の式には適用されません。

参照： 詳細は、5-2 ページの「[グループ化文字](#)」を参照してください。

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。

英語での動作

英語では、ABOUT 演算子を使用して概念を問い合わせます。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 は、問合せワードまたは句に一致する概念を含むドキュメントを取り出します。

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、ワードまたは句を自動的に正規化します。

構文

構文	説明
about(<i>phrase</i>)	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。 <i>phrase</i> パラメータは、1 つのワード、句またはフリー・テキスト形式のワード文字列です。 英語では、 <i>phrase</i> 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。

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わ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に結合できます。

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) '
```

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 問合せを入力し、これをシステムがナレッジ・カタログで検索しなければ、システム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。

注意

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。

phrase に組み込まれた問合せ演算子は無視されます。

ACCUMulate (,)

ACCUM 演算子を使用して、問合せ語句のいずれかが 1 つ以上出現するドキュメントを検索します。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
<i>term1,term2</i>	<i>term1</i> または <i>term2</i> が含まれている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。
<i>term1 accum term2</i>	

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, 3*Brazil, 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, 3*y, z	100	0	100	2	40
F	x, 3*y, 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、3*y、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します。

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 (&)

AND 演算子を使用して、各問合せ語句が 1 つ以上出現するドキュメントを検索します。

構文

構文	説明
<i>term1</i> & <i>term2</i>	<i>term1</i> および <i>term2</i> 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。
<i>term1</i> and <i>term2</i>	

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'
```

注意

AND 問合せでは、問合せ語句の最も低いスコアが戻ります。前述の例では、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(<i>term</i> [(<i>qualifier</i>)][, <i>level</i>][, <i>thes</i>])	シソーラスで問合せ語句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(<i>term</i> [(<i>qualifier</i>)][, <i>level</i>][, <i>thes</i>])	シソーラスで問合せ語句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(<i>term</i> [(<i>qualifier</i>)][, <i>level</i>][, <i>thes</i>])	シソーラスで問合せ語句に対する上位部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(<i>term</i> [(<i>qualifier</i>)][, <i>level</i>][, <i>thes</i>])	シソーラスで問合せ語句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* によって指定されたシソーラスで、*term* に対して定義された上位語のエントリを含むように拡張されます。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、*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、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)
```

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。

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 BROADER TERM 演算子に対応します。BROADER TERM 問合せでは、指定された BROADER 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。

参照： シソーラスの上位語のブラウズについては、[第 11 章](#)の「CTX_THES.BT」を参照してください。

EQUIValence (=)

EQUIV 演算子を使用して、問合せ時に条件を満たすワードの置換を指定します。

構文

構文	説明
<i>term1=term2</i>	<i>term2</i> が <i>term1</i> 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、
<i>term1 equiv term2</i>	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'
```

注意

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を含む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、データベースのドキュメントにスペル・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。

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、索引内のワードに左右されるため、結果は索引の内容によってかなり変更されます。

構文

構文	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を拡張します。

例

入力	拡張結果
?cat	cat cats calc case
?feline	feline defined filtering
?apply	apply apple applied April
?read	lead real

注意

FUZZY 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。日本語レクサーの場合、ファジー・マッチングは制限されます。

さらに、FUZZY 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ないか、CTX_DOC.HIGHLIGHT または CTX_DOC.MARKUP でハイライト表示されません。

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。

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
$term1-term2$	$term1$ が含まれているドキュメントを戻します。 $term1$ の出現回数から
$term1 \text{ minus } term2$	$term2$ の出現回数を引いて、スコアを計算します。

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てします。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* の場合もある) 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* の出現回数から *Ford* の出現回数を差し引いた値になります。語句 *Ford* がドキュメントに出現しない場合は、スコアからは何も差し引かれません。

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(<i>term</i> [(<i>qualifier</i>)][, <i>level</i>][, <i>thes</i>])	シソーラスで問合せ語句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(<i>term</i> [(<i>qualifier</i>)][, <i>level</i>][, <i>thes</i>])	シソーラスで問合せ語句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(<i>term</i> [(<i>qualifier</i>)][, <i>level</i>][, <i>thes</i>])	シソーラスで問合せ語句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(<i>term</i> [(<i>qualifier</i>)][, <i>level</i>][, <i>thes</i>])	シソーラスで問合せ語句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 *term* に対して定義された下位語エントリを含む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。

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、*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、すべての同形異義語の下位語すべて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)'
```

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。

参照： シソーラスの下位語のブラウズについては、[第 11 章](#)の「CTX_THES.NT」を参照してください。

NEAR (;)

NEAR 演算子を使用して、2 つ以上の問合せ語句の出現位置の近さを基にしたスコアを戻します。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* について問い合わせるには、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わせて行われます。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します。多くの語句（7 個程度）で問い合わせると、少ない語句（2 個程度）で問い合わせたときよりも、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。前述の例では、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のすべてのオカレンス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.
```

参照： 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。

前述の例では、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
<i>term1</i> ~ <i>term2</i>	<i>term1</i> が含まれるが、 <i>term2</i> が含まれていないドキュメントを
<i>term1</i> not <i>term2</i>	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しません。

OR (|)

OR 演算子を使用して、問合せ語句が 1 つ以上出現するドキュメントを検索します。

構文

構文	説明
<i>term1</i> <i>term2</i>	<i>term1</i> または <i>term2</i> が含まれているドキュメントを戻します。
<i>term1</i> or <i>term2</i>	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'
```

注意

OR 問合せでは、問合せ語句の最も高いスコアが戻ります。前述の例では、ドキュメント内の *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(<i>term</i> [, <i>thes</i>])	問合せで指定されたワードを問合せ語句に対する優先語句で置換します。

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、その語句に対して 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。

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、DEFAULT という名前のシソーラスがシソーラス演算子を使用する前に、シソーラス表に存在する必要があります。

例

automobile はシソーラスに *car* という優先語を持つとします。*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。

参照： シソーラスでの優先語句のブラウズについては、[第 11 章](#)の「CTX_THES.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(<i>term</i> [, <i>thes</i>])	シソーラスで問合せ語句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を含むように、*term* が拡張されます。

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。したがって、DEFAULT という名前のシソーラスがシソーラス演算子を使用する前に、シソーラス表に存在する必要があります。

例

dog という語句には、*wolf* いう関連語句があるとします。*dog* に対して RT 問合せを行うと、*dog* および *wolf* 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。

参照： シソーラスでの関連語句のブラウズについては、[第 11 章](#)の「CTX_THES.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）を含む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
<i>!term</i>	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..
```

注意

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。

interMedia のステミング機能は 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ます。

構文

構文	説明
<code>\$term</code>	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
<code>\$scream</code>	scream screaming screamed
<code>\$distinguish</code>	distinguish distinguished distinguishes
<code>\$guitars</code>	guitars guitar
<code>\$commit</code>	commit committed
<code>\$cat</code>	cat cats
<code>\$sing</code>	sang sung sing

注意

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 emp
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
<code>SYN(<i>term</i>[,<i>thes</i>])</code>	シソーラスで問合せ語句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を含むように、 *term* が拡張されます。

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する必要があります。

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 演算子内の複合句

シノニム問合せで問合せ語句の複合句の拡張問合せは、AND 接続詞によって戻ります。
たとえば、 *temperature + measurement + instruments* という複合句は、 *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。

参照： シソーラスのシノニム語句のブラウズについては、[第 11 章](#)の「CTX_THES.[SYN](#)」を参照してください。

THRESHOLD (>)

THRESHOLD 演算子 (>) は、次の 2 通りの方法で使います。

- 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
<i>expression</i> > <i>n</i>	結果セットでスコアがしきい値 <i>n</i> を超えるドキュメントのみを戻します。
<i>term</i> > <i>n</i>	式内で、スコアが最低 <i>n</i> 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、次の式を使います。

```
'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、次の式を使います。

```
'(lion > 30) and tiger'
```

TRANSLATION TERM (TR)

定義されたすべての外国語の等価語句を含む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(<i>term</i> [, <i>lang</i> , [<i>thes</i>]])	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を含むように、問合せ語句を拡張します。

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を含むように拡張されます。

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う前に、DEFAULT という名前のシソーラスがシソーラス表に存在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 について考えます。

```
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。

参照： シソーラスでの関連語句のブラウズについては、[第 11 章](#)の「CTX_THES.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を含む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
TR(<i>term</i> [, <i>lang</i> , [<i>thes</i>]])	問合せ語句の外国語の等価語、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を含むように、問合せ語句を拡張します。

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を含むように拡張されます。

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する必要があります。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う前に、DEFAULT という名前のシソーラスがシソーラス表に存在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 について考えます。

```
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。

参照： 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 11 章](#)の「CTX_THES.[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(<i>term</i> [, <i>thes</i>])	問合せで指定されたワードを、問合せ語句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。

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する必要があります。

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。

参照： シソーラスでの最上位語のブラウズについては、[第 11 章](#)の「CTX_THES.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、出現した回数が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、関心の高い語句がドキュメント全体のランク付けに反映されるようにできます。

構文

構文	説明
<i>term</i> * <i>n</i>	<i>term</i> が含まれているドキュメントを戻します。 <i>term</i> のロー・スコアに <i>n</i> を掛けてスコアを計算します。 <i>n</i> 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) に関心があるとしめます。*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 4-1 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 4-1

	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。

ライト・トランケーテッド問合せ

ライト・トランケーテッドでは、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%'

レフト・トランケーテッド問合せおよびダブル・トランケーテッド問合せ

レフト・トランケーテッドでは、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、レフト・トランケーテッド検索およびライト・トランケーテッド検索を組み合わせ、ダブル・トランケーテッド検索を作成できます。次の問合せでは、サブ文字列 `%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%'
```

ダブル・トランケーテッド問合せパフォーマンスの向上

ワイルド・カード問合せがレフト・トランケーテッドおよびダブル・トランケーテッドの場合、サブ文字列索引を作成すると、問合せパフォーマンスが向上します。サブ文字列索引によって、`%ed`、`_ing`、`%benz%` などの、あらゆる種類のレフト・トランケーテッド・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。

参照： サブ文字列索引作成の詳細は、[第3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
<i>expression</i> WITHIN <i>section</i>	<p>事前定義済のゾーン、フィールドまたは属性 <i>section</i> 内で <i>expression</i> を検索します。</p> <p>セクションがゾーンの場合、<i>expression</i> 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。</p> <p>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。</p>
<i>expression</i> WITHIN SENTENCE	<p>文中に <i>expression</i> を含むドキュメントを検索します。 <i>expression</i> には AND または NOT 問合せを指定します。</p> <p><i>expression</i> 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。</p>
<i>expression</i> WITHIN PARAGRAPH	<p>段落中に <i>expression</i> を含むドキュメントを検索します。 <i>expression</i> には AND または NOT 問合せを指定します。</p> <p><i>expression</i> 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。</p>

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'
```

前述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 演算子よりも高い優先順位を持つため、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、ドキュメント・セットがゾーン BOOK セクション内にネストされたゾーン・セクション AUTHOR を持つとします。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 セクションにある *scott* のすべてのオカレンスを検索できます。

```
'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。

- テキストは、フィールド・セクション内で表示されないようにできます。
- WITHIN 問合せは、繰り返しフィールド・セクションを識別できません。
- 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 が、FALSE に設定された *visible* フラグで定義されたフィールド・セクションであるとしてします。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。

参照： フィールド・セクション作成の詳細は、7-5 ページの「[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、ドキュメントの繰返しフィールド・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わせるには、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、次のように問い合わせます。

```
'(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、次のように問い合わせます。

```
'(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、次のように問い合わせます。

```
'(dog not cat) WITHIN SENTENCE'
```

属性セクション内の問合せ

セクション・グループ・タイプとして XML_SECTION_GROUP または AUTOMATIC_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わせることができます。

次のような XML ドキュメントを持つとします。

```
<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 ドキュメントを持つとします。

```
<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、次のドキュメントを持つとします。

```
<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* で属性セクション book@author のオカレンスが異なるためです。

注意

スコア付け

WITHIN 演算子は、スコアには影響しません。

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セクションの特定のオカレンスに完全に囲まれている必要があります。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><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。フィールド・セクション内の問合せはオカレンス間で区別できません。フィールド・セクションのすべてのオカレンスは、1つのセクションの一部とみなされます。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。

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、*interMedia 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わせるには、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、ハイフンの片側に空白がある *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 問合せで検索する場合に、エスケープする必要がある *interMedia 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
(なし)	?	FUZZY
(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
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

予約語（ワード）	予約語（文字）	演算子
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

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・プロシージャがあります。

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。
SHUTDOWN	単一の ctxsrv サーバーまたは現在稼働中のすべてのサーバーを停止します。

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

次のいずれかで、設定するパラメータの名前を指定します。

- max_index_memory (索引付けに使用可能な最大メモリ)
- default_index_memory (索引付けに割り当てられたデフォルトのメモリ)
- log_directory (ctx_output 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)
- 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_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)

参照： これらのパラメータに対するデフォルト値の詳細は、[第3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、指定する値には次の構文が必要です。

number [M|G|K]

ここで、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;
```

SHUTDOWN

SHUTDOWN プロシージャは、指定された ctxsrv サーバーを停止します。

構文

```
CTX_ADM.SHUTDOWN (name    IN VARCHAR2 DEFAULT 'ALL',  
                  sdmode IN NUMBER   DEFAULT NULL);
```

name

停止する ctxsrv サーバーの名前（内部識別子）を指定します。

デフォルト値は ALL です。

sdmode

サーバーの停止モードを指定します。

- 0 または NULL (Normal)
- 1 (Immediate)
- 2 (Abort)

デフォルト値は NULL です。

例

```
begin  
  ctx_adm.shutdown('DRSRV_3321', 1);  
end;
```

注意

停止する ctxsrv サーバーを指定しないと、現在稼働中のすべての ctxsrv サーバーが停止します。

CTX_SERVERS ビューを使用すると、現在稼働中のすべての ctxsrv サーバーの名前を取得できます。

関連項目

第 12 章の「[ctxsrv](#)」

CTX_DDL パッケージ

この章では、テキスト索引に必要なオブジェク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・グループへ追加します。
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・グループへ割り当てます。
ADD_SPECIAL_SECTION	特殊セクションをセクション・グループへ追加します。
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・グループへ追加します。
ADD_STOPTHEME	ストップテーマをストップリストへ追加します。
ADD_STOPWORD	ストップワードをストップリストへ追加します。
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・グループへ追加します。
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。
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_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。

名前	説明
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_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。

ADD_ATTR_SECTION

属性セクションを XML セクション・グループへ追加します。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。

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。

section_name

属性セクションの名前を指定します。この名前は、属性テキストでの WITHIN 問合せに使用されます。

指定するセクション名には、コロン (:) やドット (.) は使用できません。またセクション名は、*group_name* の中で一意である必要があります。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。

```
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わせできます。

```
'Cities within booktitle'
```

注意

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。

AUTO_SECTION_GROUP を使用すると、属性セクションは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。

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・セクションでの **WITHIN** 問合せは、通常、ゾーン・セクションでの **WITHIN** 問合せより高速です。

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 が追加されるセクション・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。

section_name

group_name へ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・セクション名およびフィールド・セクション名は同一にはできません。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。

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>** の場合には、**H1** を指定します。

group_name が HTML_SECTION_GROUP の場合、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、NAME 属性の CONTENT 属性は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。

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。Oracle は、フィールド・セクション内のテキストを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* を問い合わせると、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わせる場合は、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

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.add_zone_section('mygroup', '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わせることができます。

```
'ken WITHIN author'
```

注意

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。指定する開始タグは、セクション・グループ内で一意にしてください。

セクション名はタグ内で一意である必要はありません。詳細な検索を行うために、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。

1 つのセクション・グループ内では最大 64 のフィールド・セクションを定義できます。同じグループ内では、セクション・ゾーン名およびセクション・フィールド名は同一にはできません。

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で作業するには、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・セクションは許可されていますが、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のセクション内で発生した場合にも、ドキュメントを戻します。

WITHIN 問合せて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 4 章の「WITHIN」演算子](#)

[第 3 章の「セクション・グループのタイプ」](#)

[CREATE_SECTION_GROUP](#)

[ADD_ZONE_SECTION](#)

[ADD_SPECIAL_SECTION](#)

[REMOVE_SECTION](#)

[DROP_SECTION_GROUP](#)

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します。

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。テキスト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。

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;
```

注意

文および段落の境界は、レクサーが判断します。このため、レクサーが境界を認識できない場合は、SENTENCE セクションまたは PARAGRAPH セクションはどれも索引付けされません。

関連項目

[第 4 章の「WITHIN」演算子](#)

[第 3 章の「セクション・グループのタイプ」](#)

[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。

ストップクラス

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。

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;
```

注意

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。

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・グループへ追加します。停止セクションを追加すると、自動セクションが索引付け操作で、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、ドキュメントに低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に制限はありません。

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。このパラメータは、大 / 小文字を区別します。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に、これを停止します。

タグは、*(doctype)tag* という形式でドキュメント・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、`(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を停止します。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">
```

前述のコードも、属性セクション fluff@type を停止します。

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;
```

注意

停止セクションはセクション名を持ちません。したがって、セクション・ビューには記録されません。

関連項目

[第 2 章の「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わせます。

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。

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;
```

注意

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。

関連項目

[CREATE_STOPLIST](#)

[REMOVE_STOPTHEME](#)

[DROP_STOPLIST](#)

第 4 章の「[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  
);
```

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。

例

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;
```

注意： [ALTER INDEX](#) を使用して索引を作成した後、ストップワードを追加できます。

注意

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。

関連項目

[CREATE_STOPLIST](#)

[REMOVE_STOPWORD](#)

[DROP_STOPLIST](#)

[第 2 章の「ALTER INDEX」](#)

[付録 E「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、マルチレクサー・プリファレンスを使用します。

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 NLS 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 NLS 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。

ベース表内の言語の列が NULL、無効、またはサブレクサーにマップされていない場合、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には、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、NLS 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るためのマルチレクサーのセットアップ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_attritbue('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');
```

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');
```

注意

制限

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。
- デフォルトのレクサーは、マルチレクサーが索引で使用できるようになる前に定義する必要があります。
- 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。

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 `` および `` 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やオーバーラップが可能です。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 が追加されるセクション・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・セクション名およびフィールド・セクション名は同一にはできません。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。

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、*tag* に `H1` を指定します。

group_name が *HTML_SECTION_GROUP* の場合、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、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、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 セクションの作成

次のコードは、HTML_SECTION_GROUP のタイプで *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わ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

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わ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

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。指定する開始タグは、セクション・グループ内で一意にしてください。

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。

繰り返しセクション

ゾーン・セクションは繰り返しが可能です。各オカレンス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、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、 および <I> が 2 つの異なるゾーン・セクションを示す場合、これらはドキュメントで次のようにオーバーラップします。

```
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、問合せを書き込み、セクション内のセクションのテキストを検索できます。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 で次のように発生するとします。

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 4 章の「WITHIN」演算子](#)

[第 3 章の「セクション・グループのタイプ」](#)

[CREATE_SECTION_GROUP](#)

[ADD_FIELD_SECTION](#)

[ADD_SPECIAL_SECTION](#)

[REMOVE_SECTION](#)

[DROP_SECTION_GROUP](#)

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・オブジェクトの名前を指定します。

参照： プリファレンス・オブジェクトおよび関連する属性のすべてのリストは、[第3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て行います。その後、これは CTX_DDL.[SET_ATTRIBUTE](#) を2回コールします。初めに INDEX_TEXT 属性に対して Y を指定し、その後 INDEX_THEMES 属性に対して N 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、ディクショナリ /docs に対する PATH 属性を設定します。

```
begin
ctx_ddl.create_preference('mypref', 'FILE_DATASTORE');
ctx_ddl.set_attribute('mypref', 'PATH', '/docs');
end;
```

参照： データ記憶域の詳細は、第3章の「データストア・オブジェクト」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;
```

参照： マスター表 / ディテール表の詳細は、第3章の「DETAIL_DATASTORE」を参照してください。

記憶域属性の指定

次の例では、索引表を 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;
```

参照： 記憶域の詳細は、[第3章の「記憶域オブジェクト」](#)を参照してください。

属性のないプリファレンスの作成

属性のないオブジェクトを持つプリファレンスを作成する場合に必要なのは、次の例で示すように、フィルタを NULL_FILTER に設定するプリファレンスを作成することのみです。

```
begin
ctx_ddl.create_preference('my_null_filter', 'NULL_FILTER');
end;
```

関連項目

[SET_ATTRIBUTE](#)

[DROP_PREFERENCE](#)

[第2章の「CREATE INDEX」](#)

[第2章の「ALTER INDEX」](#)

[第3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、[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、[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わ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必要があります。

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	これはデフォルトです。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に、このグループ・タイプを使用します。
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 という形式のセクションを定義します。
HTML_SECTION_GROUP	このグループ・タイプを使用して、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 形式のタグ付き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	<p>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。</p> <p>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、<code>attribute@tag</code> という形式でネーミングされます。</p> <p>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。</p> <p>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。</p> <ul style="list-style-type: none">■ 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。■ プレフィ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。
NEWS_SECTION_GROUP	<p>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。</p>

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 4 章の「WITHIN」演算子](#)

第3章の「セクション・グループのタイプ」

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。

[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_name

作成するストップリストの名前を指定します。

例

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');
end;
```

注意

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。

関連項目

[ADD_STOPWORD](#)

[ADD_STOPCLASS](#)

[ADD_STOPTHEME](#)

[DROP_STOPLIST](#)

第 2 章の「[CREATE INDEX](#)」

第 2 章の「[ALTER INDEX](#)」

付録 E「[提供されるストップリスト](#)」

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。

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;
```

注意

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ありません。

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。

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;
```

注意

ストップリストを削除した場合、変更を有効にするために索引を再作成または再構築してください。

関連項目

[CREATE_STOPLIST](#)

OPTIMIZE_INDEX

索引を最適化します。高速モードまたは完全モードで最適化できます。これは [ALTER INDEX](#) を使用した最適化と同じです。

構文

```
CTX_DDL.OPTIMIZE_INDEX(  
    idx_name in varchar2,  
    optlevel in varchar2,  
    maxtime  in number   default null  
);
```

idx_name

索引の名前を指定します。

optlevel

FAST または FULL のいずれかの最適化レベルを指定します。

また、このパラメータは、CTX_DDL.OPTLEVEL_FAST または CTX_DDL.OPTLEVEL_FAST のいずれでも指定できます。

高速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、**高速**モードでは、古いデータは削除されません。

完全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、行は圧縮され、古いデータは削除されます（ガベージ・コレクション）。

maxtime

完全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。

maxtime に 0（ゼロ）を指定すると、最小限の最適化が行われます。

例

次の2つの例では、索引を高速モードで最適化しています。

```
begin
ctx_ddl.optimize_index('myidx','FAST');
end;

begin
ctx_ddl.optimize_index('myidx',CTX_DDL.OPTLEVEL_FAST);
end;
```

関連項目

[第2章の「ALTER INDEX」](#)

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 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_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。

ストップクラス

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  
);
```

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。

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、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。例では、CTX_DDL.SET_ATTRIBUTE を使用し、PATH 属性をディクショナリ /docs に設定します。

```
begin  
ctx_ddl.create_preference('filepref', 'FILE_DATASTORE');  
ctx_ddl.set_attribute('filepref', 'PATH', '/docs');  
end;
```

参照： データ記憶域の詳細は、第 3 章の「データストア・オブジェクト」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、挿入、更新および削除を処理します。これは、[ALTER INDEX](#) を使用した同期化と同じです。

構文

```
ctx_ddl.sync_index(idx_name in varchar2 default NULL);
```

idx_name

索引の名前を指定します。

例

```
begin
ctx_ddl.sync_index(myindex);
end;
```

関連項目

[第 2 章の「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](#)

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。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	ドキュメントの要旨またはテーマ・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。
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 プロシージャのドキュメント識別子を ROWID または主キーに設定します。
THEMES	ドキュメントのテーマのリストを生成します。

FILTER

CTX_DOC.FILTER プロシージャを使用し、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を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 CLOB,  
    query_id    IN NUMBER DEFAULT 0,  
    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。

- 単一系列の主キーの値
- コンポジット（複数列）主キーのコード化された仕様
- ドキュメントを含む行の ROWID

主キーと ROWID の識別をトグルするには、CTX_DOC.[SET_KEY_TYPE](#) を使用します。

restab

このプロシージャが、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。

参照： フィルタ結果表の構造の詳細は、付録 B の「フィルタ表」を参照してください。

メモリーに結果を格納するには、CLOB ロケータの名前を指定します。*restab* が NULL の場合、一時 CLOB が割り当てられて戻されます。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、操作の前に CLOB が切り捨てられます。

query_id

restab に挿入した行を識別するために使用する識別子を指定します。

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ます。

```
declare
mklob clob;
amt number := 40;
line varchar2(80);

begin
  ctx_doc.filter('myindex','1', mklob, '0', 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;
```

注意

CTX_DOC.FILTER をコールする前に、*restab* で指定された結果表が存在している必要があります。

textkey がコンポジット・テキストキーの場合、CTX_DOC.[PKENCODE](#) 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してください。

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。

GIST

CTX_DOC.GIST プロシージャを使用して、ドキュメントの要旨およびテーマ・サマリーを生成します。段落レベルまたは文レベルの要旨 / 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,  
    query_id        IN NUMBER DEFAULT 0,  
    glevel          IN VARCHAR2 DEFAULT 'P',  
    pov             IN VARCHAR2 DEFAULT NULL,  
    numParagraphs   IN NUMBER DEFAULT 16,  
    maxPercent      IN NUMBER DEFAULT 10);
```

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 16,  
    maxPercent      IN NUMBER DEFAULT 10);
```

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。

- 単一系列の主キーの値
- コンポジット（複数列）主キーのコード化された仕様

- ドキュメントを含む行の ROWID

主キーと ROWID の識別をトグルするには、CTX_DOC.SET_KEY_TYPE を使用します。

restab

このプロシージャが、要旨およびテーマ・サマリーを表またはメモリー内 CLOB 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。

参照： 要旨結果表の構造の詳細は、付録 B の「要旨表」を参照してください。

メモリーに結果を格納するには、CLOB ロケータの名前を指定します。*restab* が NULL の場合、一時 CLOB が割り当てられて戻されます。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、操作の前に CLOB が切り捨てられます。

query_id

restab に挿入した行を識別するために使用する識別子を指定します。

glevel

生成する要旨 / 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

要旨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。生成される要旨 / 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* に指定した値に依存します。

ドキュメント全体の要旨を生成するには、*pov* に「GENERIC」の値を指定します。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、全体の要旨に加えて最大 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、*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旨のみを生成します。

注意： *pov* パラメータでは、大 / 小文字が区別されます。ドキュメントの要旨を戻すには、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 CTX_DOC.THEMES で生成されたテーマのみが、*pov* の入力として使用できます。

numParagraphs

ドキュメント要旨 / 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 16 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旨 / 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旨 / テーマ・サマリーのサイズよりも小さいときのみです。

これは、システムが常に最小サイズの要旨 / テーマ・サマリーを戻すことを意味します。

maxPercent

ドキュメントの要旨 / 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 10 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旨 / 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旨 / テーマ・サマリーのサイズよりも小さいときのみです。

これは、システムが常に最小サイズの要旨 / テーマ・サマリーを戻すことを意味します。

例

メモリー内要旨

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旨を生成します。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。

```
declare
  gklob clob;
  amt number := 40;
  line varchar2(80);

begin
  ctx_doc.gist('newsindex','34','gklob',1,glevel => 'P',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;
```

結果表の要旨

次の例では、CTX_GIST という要旨表を作成します。

```
create table CTX_GIST (query_id number,
                       pov      varchar2(80),
                       gist      CLOB);
```

要旨 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旨およびすべてのテーマに対するテーマ・サマリーを戻します。

```
begin
  ctx_doc.gist('newsindex','34','CTX_GIST',1,glevel => 'P');
end;
```

次の例では、10 個以内の段落のデフォルト・サイズでない要旨表を生成します。

```
begin
  ctx_doc.gist('newsindex','34','CTX_GIST',1,glevel => 'P',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旨表を生成します。

```
begin
ctx_doc.gist('newsindex','34','CTX_GIST',1, glevel =>'P',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,glevel =>'P', pov => 'insects');
end;
```

注意

結果表記憶域を使用し、*pov* を指定しない場合、デフォルトでは、このプロシージャはドキュメントに対して最大 50 個までのテーマ・サマリーを生成します。したがって、CTX_DOC.GIST は、各ドキュメントに対して最大 51 個の要旨を作成します（各テーマに対して 1 つのテーマ・サマリーおよびドキュメント全体に対して 1 つの要旨表）。

メモリー内記憶域を使用する場合、CTX_DOC.GIST は 1 つの要旨のみを作成します。

textkey がコンポジット・テキストキーの場合、前述の 2 番目の例にあるように、CTX_DOC.PKENCODING 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してください。

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 DEFAULT NULL,  
    restab      IN OUT HIGHLIGHT_TAB,  
    query_id    IN NUMBER   DEFAULT 0,  
    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 DEFAULT NULL,  
    restab      IN VARCHAR2 DEFAULT NULL,  
    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。

- 単一行の主キーの値
- コンポジット（複数列）主キーのコード化された仕様
- ドキュメントを含む行の ROWID

主キーと ROWID の識別をトグルするには、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間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。

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。

参照： ハイライト結果表の構造の詳細は、付録 B 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、メモリー内の型 CTX_DOC.HIGHLIGHT_TAB の表の名前を指定します。データ型 HIGHLIGHT_TAB 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 に挿入した行を識別するために使用する識別子を指定します。

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。

注意

CTX_DOC.HIGHLIGHT がコールされる前に、*restab* で指定された結果表が存在している必要があります。

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してください。

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。

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。

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を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 CLOB,
    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);
```

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
- ドキュメントを含む行の ROWID

主キーと ROWID の識別をトグルするには、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。

restab

このプロシージャが、マークアップされたテキストを表またはメモリ内 CLOB 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。

参照： マークアップ結果表の構造の詳細は、付録 B の「マークアップ表」を参照してください。

メモリーに結果を格納するには、CLOB ロケータの名前を指定します。*restab* が NULL の場合、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、操作の前に CLOB が切り捨てられます。

query_id

restab に挿入した行を識別するために使用する識別子を指定します。

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	
	endtag	
	prevtag	
	nexttag	
HTML_NAVIGATE	starttag	
	endtag	
	prevtag	< ;
	nexttag	> ;

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。

```
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;
```

注意

CTX_DOC.MARKUP がコールされる前に、*restab* で指定された結果表が存在している必要があります。

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してください。

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。

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。

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、文字列を返します。

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

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;
```

この例で、*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します。

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。

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最大 50 個のテーマのリストを生成します。各テーマは、ユーザーが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されます。

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 THEME_TAB,
               query_id      IN NUMBER DEFAULT 0,
               full_themes   IN BOOLEAN DEFAULT FALSE);
```

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);
```

index_name

テーマのリストに対するドキュメントが格納されている列の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
- ドキュメントを含む行の ROWID

主キーと ROWID の識別をトグルするには、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。

参照： テーマ結果表の構造の詳細は、[付録 B](#) の「[テーマ表](#)」を参照してください。

結果をメモリー内の表に格納するには、メモリー内の型 THEME_TAB の表の名前を指定します。データ型 THEME_TAB 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 に挿入した行を識別するため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。

例

メモリー内テーマ

次の例は、ドキュメント 1 のテーマを生成し、それをメモリー内表に格納して、*the_themes* として宣言します。その後、表全体をループし、ドキュメントのテーマを表示します。

```
declare  
the_themes ctx_doc.theme_tab;  
  
begin  
    ctx_doc.themes('myindex','1',the_themes);  
    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テーマ・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);  
end;
```

全テーマ

リストの各要素が親テーマの階層リストであるテーマ・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);  
end;
```

注意

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してください。

CTX_OUTPUT パッケージ

この章では、PL/SQL パッケージ CTX_OUTPUT の使用に関する参照情報を示します。
CTX_OUTPUT には、次のストアド・プロシージャがあります。

名前	説明
END_LOG	索引およびドキュメント・サービスの要求のログ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。
START_LOG	索引およびドキュメント・サービスの要求のログギングを開始します。

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。

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

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;
```

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 に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;
```

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する PL/SQL パッケージ CTX_QUERY について説明します。

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ます。

名前	説明
BROWSE_WORDS	索引の中のシード・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 SQE 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、*interMedia Text* の索引の中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 は索引中のこれに関するワードおよび各ワードを含む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。次のものを識別できます。

- 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_word(  
  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  
);
```

構文 2: メモリーへの結果の格納

```
ctx_query.browse_word(  
  index_name in      varchar2,  
  seed       in      varchar2,  
  resarr     in out  browse_tab,  
  numwords   in      number default 10,  
  direction  in      varchar2 default BROWSE_AROUND  
);
```

index

索引の名前を指定します。schema.name を指定できます。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* を `schema.name` として入力できます。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

生成されたリストの長さをワード数で指定します。1 ～ 1000 の数を指定します。

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。

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

```
declare
    resarr ctx_query.browse_tab;
begin
    ctx_query.browse_words('myindex','dog',resarr,5,CTX_QUERY.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返します。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返します。予測モードは上限の見積りを返しますが、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  
) 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。上限を見積もるには FALSE を指定します。

注意

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。

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。そのかわり、問合せを発行する前に、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
- コンボジット・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すると、問合せの詳細化やデバッグに有効です。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。

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);
```

index_name

問い合わ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。

explain_table

text_query に対する解析ツリーの表現を格納するために使用する表の名前を指定します。

参照： 実行計画表の構造の詳細は、付録 B 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、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、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。

注意

EXPLAIN からの結果を格納するためには、表に対して少なくとも INSERT 権限および DELETE 権限が必要です。

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は、不用意な削除を考慮しません。

リモート問合せで EXPLAIN を使用することはできません。

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 4 章「問合せ演算子」
- 付録 I「ストップワード変換」

HFEEDBACK

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知識ベースから取得されます。ただし、索引中にも存在する知識ベース語句のみが、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・セットより高いヒットを生成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。

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,  
);
```

index_name

問い合わ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

フィードバック語を格納するために使用する表の名前を指定します。

参照： HFEEDBACK 表の構造の詳細は、[付録 B 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、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、HFEEDBACK へのコールが戻し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。

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。

参照： HFEEDBACK 表の構造の詳細は、[付録 B](#) 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 ||
                          ':');
  END LOOP;
```

```
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* について問い合わ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、索引および知識ベースの内容に依存するため、前述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 SQE 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 (SQE) を作成します。問合せ定義のみが格納されます。

構文

```
CTX_QUERY.STORE_SQE(query_name      IN VARCHAR2,  
                    text_query      IN VARCHAR2);
```

query_name

作成する SQE の名前を指定します。CTXSYS ユーザーの場合、これを `user.name` として指定できます。

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;
```

注意

SQE は、すべてのテキスト問合せ式演算子をサポートします。SQE は、すべての特殊文字、および他の SQE を含む問合せ式で利用できるすべてのコンポーネントもサポートします。

ユーザーは、所有している SQE を作成および削除できます。ユーザーは、誰が所有する SQE でも使用できます。CTXSYS ユーザーは、どのユーザーに対しても SQE を作成および削除できます。

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機能は、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すると、シソーラス演算子を使用した問合せの書込みに有効です。シソーラスを使用して、英語での ABOUT 問合せおよびドキュメント・テーマの生成に使用される知識ベースも拡張できます。

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 つの句の間の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、そのシソーラスの ID を戻します。
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
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	句の優先語句を戻します。
RT	句の関連語句を戻します。
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。

参照： シソーラス演算子の詳細は、[第 4 章「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。

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	オペランド
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する場合、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、句から修飾子を変更、追加または削除できます。
PT または CTX_THES.OP_MAKE_PT	句を優先語句にします。シノニム・リングにある既存の優先語句は、非優先シノニムになります。	ありません。
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。

オペランド
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。

注意

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。

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 を優先語句にします。

```
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 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	オペランド
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する場合、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。

オペランド
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。

注意

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および CTXSYS のみです。

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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数を指定します。たとえば、2 によって、句の上位語の上位語を取得します。

tname

シソーラス名を指定します。指定しない場合、システムのデフォルトのシソーラスが使用されます。

戻り

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。

```
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、次の出力を生成します。

```
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。

```
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(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 WHITE WOLF
  BT WOLF
    BT CANINE
      BT ANIMAL
```

関連項目

[OUTPUT_STYLE](#)

第4章の「[BROADER TERM \(BT、BTG、BTP、BTI\)](#)」演算子

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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数を指定します。たとえば、2 によって、句の上位語の上位語を取得します。

tname

シソーラス名を指定します。指定しない場合、システムのデフォルトのシソーラスが使用されます。

戻り

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。

```
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 4 章の「[BROADER TERM \(BT、BTG、BTP、BTI\)](#)」演算子

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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数を指定します。たとえば、2 によって、句の上位語の上位語を取得します。

tname

シソーラス名を指定します。指定しない場合、システムのデフォルトのシソーラスが使用されます。

戻り

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。

```
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 4 章の「[BROADER TERM \(BT, BTG, BTP, BTI\)](#)」演算子

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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数を指定します。たとえば、2 によって、句の上位語の上位語を取得します。

tname

シソーラス名を指定します。指定しない場合、システムのデフォルトのシソーラスが使用されます。

戻り

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 4 章の「[BROADER TERM \(BT、BTG、BTP、BTI\)](#)」演算子

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は作成できますが、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勧めします。

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、thesarus.SYN に新しいリレーションを作成します (*phrase* は *relname* に対するシノニム語です)。

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

エントリの ID です。

例

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勧めします。

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 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、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。関係は、 <i>phrase</i> から <i>relphrase</i> 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* を考えてみます。

```
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);
```

thes_name

作成するシソーラスの名前を指定します。

casesens

作成するシソーラスで大 / 小文字が区別され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;
```

注意

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する場合、CREATE_THESAURUS はエラーを戻し、シソーラスは作成されません。

シソーラスに句を入力する場合は、CTX_THES.[CREATE_PHRASE](#) を使用するか、またはシステム管理ツールにあるシソーラス・メンテナンス画面を使用してください。

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 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、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、このプロシージャによって例外が発生します。

注意

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に対して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 <i>relphrase</i> 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 <i>relphrase</i> 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	優先語句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
言語	翻訳を句から削除します。	オプションで <i>relphrase</i> 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、 <i>relphrase</i> を指定できます。また、1 つの翻訳のみでも削除できます。 <i>relphrase</i> が NULL の場合、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* を考えてみます。

```
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;
```

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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数を指定します。たとえば、2 によって、句の下位語の下位語を取得します。

tname

シソーラス名を指定します。指定しない場合、システムのデフォルトのシソーラスが使用されます。

戻り

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}||{wild cat}||{Persian
cat}||{Siamese 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4章の「[NARROWER TERM \(NT、NTG、NTP、NTI\)](#)」演算子

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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数を指定します。たとえば、2 によって、句の下位語の下位語を取得します。

tname

シソーラス名を指定します。指定しない場合、システムのデフォルトのシソーラスが使用されます。

戻り

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 4 章の「[NARROWER TERM \(NT, NTG, NTP, NTI\)](#)」演算子

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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数を指定します。たとえば、2 によって、句の下位語の下位語を取得します。

tname

シソーラス名を指定します。指定しない場合、システムのデフォルトのシソーラスが使用されます。

戻り

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 4 章の「NARROWER TERM \(NT、NTG、NTP、NTI\)」演算子](#)

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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数を指定します。たとえば、2 によって、句の下位語の下位語を取得します。

tname

シソーラス名を指定します。指定しない場合、システムのデフォルトのシソーラスが使用されます。

戻り

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 4 章の「[NARROWER TERM \(NT, NTG, NTP, NTI\)](#)」演算子

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。このプロセスは、CTX_THES 拡張ファンクションに対する表結果には影響しません。

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 * を持つ優先語句を示すには、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}
```

優先語句の標識は、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、句の優先語句を返します。

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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、システムのデフォルトのシソーラスが使用されます。

戻り

このファンクションは、優先語句を次の形式の文字列として返します。

{pt}

例

automobile に対して次の優先語句定義を持つシソーラス MY_THES があるとします。

AUTOMOBILE

PT CAR

automobile に対する優先語句を検索するには、次の文を発行します。

```
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.pt('AUTOMOBILE','MY_THES');
  dbms_output.put_line('The preferred term for automobile is: '||terms);
end;
```

関連項目

[OUTPUT_STYLE](#)

[第4章の「PREFERRED TERM \(PT\)」演算子](#)

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句を戻します。

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タイプ EXP_TAB 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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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、システムのデフォルトのシソーラスが使用されます。

戻り

このファンクションは、次の形式で関連語句の文字列を戻します。

```
{rt1}|{rt2}|{rt3}| ...
```

例

dog に対して次の関連語句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 に対する関連語句を検索するには、次の文を発行します。

```
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 4 章の「[RELATED TERM \(RT\)](#)」演算子

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。

SN

このファンクションは、指定された句のスコープ・ノートに戻します。

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、システムのデフォルトのシソーラスが使用されます。

戻り

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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、システムのデフォルトのシソーラスが使用されます。

戻り

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

次のコードで *canine* に対するシノニムを検索し、表で結果を取得します。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 4 章の「SYNonym (SYN)」演算子

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 は句を取り、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。

構文

```
CTX_THES.THES_TT(restab IN OUT NOCOPY EXP_TAB,  
                 tname IN VARCHAR2 DEFAULT 'DEFAULT');
```

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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、システムのデフォルトのシソーラスが使用されます。

戻り

このプロシージャは、*restab* で指定した表に対してのみ最上位語を戻します。

注意

このファンクションはすべてのシソーラスを検索するため、大きいソーラスを処理することもあります。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。

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。この TR の動作は、*interMedia Text* 固有のものであります。

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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* のすべての翻訳に対して「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、システムのデフォルトのシソーラスが使用されます。

戻り

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 について考えます。

```
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4章の「TRANSLATION TERM \(TR\)」演算子](#)

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返します。

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。この TRSYN の動作は、*interMedia Text* 固有のものです。

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タイプ EXP_TAB 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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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* のすべての翻訳に対して「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、システムのデフォルトのシソーラスが使用されます。

戻り

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 について考えます。

```
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4章の「TRANSLATION TERM SYNONYM \(TRSYN\)」演算子](#)

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、句の最上位語を返します。

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;
```

参照： EXP_TAB の詳細は、B-12 ページの「[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、システムのデフォルトのシソーラスが使用されます。

戻り

このファンクションは、次の形式の最上位語文字列として戻します。

```
{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、次の文を発行します。

```
declare
  terms varchar2(2000);
begin
  terms := ctx_thes.rt('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4章の「[TOP TERM \(TT\)](#)」演算子

12

実行可能ファイル

この章では、*interMedia 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。

- [ctxsrv](#)
- [ctxload](#)
- 知識ベース拡張コンパイラ ([ctxkbt](#))

ctxsrv

バックグラウンド DML 処理には ctxsrv サーバー・デーモンを使用します。コマンドラインから、または *interMedia Text Manager* 管理ツールで起動できます。

このサーバーは、定期的に **ALTER INDEX** によって索引を同期化します。

注意： ctxsrv を実行できるのは、Oracle ユーザーである CTXSYS のみです。

構文

```
ctxsrv [-user ctxsys/passwd[@sqlnet_address]]
       [-personality M]
       [-logfile log_name]
       [-sqltrace]
```

-user

Oracle ユーザー CTXSYS の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。

注意： ctxsrv のコマンドラインで -user を指定しないと、この必須情報を CTXSYS/password (password は CTXSYS のパスワード) の形式で入力するように求めるプロンプトが表示されます。

これは、サーバーが実行されているマシンの他のユーザーから CTXSYS のパスワードをマスクする場合に有効です。

-personality

ctxsrv によって起動されるサーバーのパーソナリティ・マスクを指定します。有効な値は M のみで、M がデフォルト値です。

-logfile

サーバーがすべてのセッション情報およびエラーを書き込むログ・ファイルの名前を指定します。

-sqltrace

サーバーが、USER_DUMP_DEST 初期化パラメータで指定したディレクトリにトレース・ファイルを書き込めるようにします。

参照： SQL トレースおよび USER_DUMP_DEST 初期化パラメータの詳細は、『Oracle8i 管理者ガイド』を参照してください。

例

次の例では、サーバーを起動し、すべてのサーバーのメッセージを ctx.log ファイルに書き込みます。

```
ctxsrv -user ctxsys/ctxsys -personality M -log ctx.log &
```

次の例では、サーバーを起動し、すべてのサーバーのメッセージを ctx.log ファイルに書き込みます。-user を指定していないため、次のように、ユーザーの入力を求めるプロンプトが表示されます。

```
ctxsrv -log ctx.log
```

```
...
```

```
Copyright (c) Oracle Corporation 1979, 1998. All rights reserved.
```

```
...
```

```
Enter user:
```

これに対し、CTXSYS/password（password は CTXSYS ユーザーに割り当てられたパスワード）を入力します。

UNIX のユーザー： この例では、プロセスはバックグラウンド・プロセスとして実行されません。

プロセスをバックグラウンドで実行できる環境でも、ctxsrv のコマンドラインに -user を指定しない場合は、サーバー・プロセスをフォアグラウンドで実行するか、ctxsrv にオペレーティング・システム・ファイルから -user の値を渡す必要があります。

次に例を示します。

```
ctxsrv -log ctx.log < pword.txt
```

このファイルには、CTXSYS/password という 1 行のテキストが格納されている必要があります。

ファイルから ctxsrv に値を渡すと、ctxsrv にユーザー名の入力を求めるプロンプトは表示されません。

注意

保留中の更新の表示

保留中の索引更新は、DML キューに格納されます。このキューを表示するには、[CTX_PENDING](#) または [CTX_USER_PENDING](#) ビューを使用できます。

Oracle Enterprise Manager の一部である *interMedia Text Manager* 管理ツールを使用することもできます。

DML エラーの表示

[CTX_INDEX_ERRORS](#) または [CTX_USER_INDEX_ERRORS](#) ビューで DML エラーを表示できます。

索引の断片化

ctxsrv によるバックグラウンド DML は、常に DML キューをポーリングすることによって DML をスキャンします。これによって、新規の追加が自動的にすばやく索引付けされます。ただし、バックグラウンド DML は、より小さいバッチでドキュメントを処理する傾向もあり、索引の断片化の度合いも大きくなります。

ただし、[ALTER INDEX](#) で手動によって索引を同期化すると、通常バッチは大きくなり、索引の断片化の度合いも小さくなります。

サーバーの停止

次のものを使用して ctxsrv を停止できます。

- [CTX_ADM.SHUTDOWN](#)
- Oracle Enterprise Manager で使用可能な *interMedia Text Manager* 管理ツール

関連項目

[ALTER INDEX](#)

第 6 章の「[CTX_ADM.SHUTDOWN](#)」

付録 H「[ビュー](#)」に記載されている次のビュー

- [CTX_PENDING](#)
- [CTX_USER_PENDING](#)
- [CTX_INDEX_ERRORS](#)

- [CTX_USER_INDEX_ERRORS](#)

interMedia Text Manager

管理ツールを使用したサーバーの起動については、*interMedia Text Manager* のオンライン・ヘルプを参照してください。この管理ツールは、Oracle Enterprise Manager に統合された Java アプリケーションです。

ctxload

ctxload を使用して、次の操作を実行します。

- シソーラスのインポートおよびエクスポート
- テキストのロード
- ドキュメントの更新 / エクスポート

シソーラスのインポートおよびエクスポート

ctxload を使用して、シソーラスをインポート・ファイルから、iMT のシソーラス表にロードできます。

インポート・ファイルは、ASCII フラット・ファイルで、テキスト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句のエントリを含みます。

ctxload を使用して、シソーラスの内容をユーザー指定のオペレーティング・システム・ファイルにダンプをとることによって、シソーラスのエクスポートもできます。

参照： 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 D](#) 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

ctxload を使用して、ロード・ファイルから表の LONG 列または LONG RAW 列にテキストをロードできます。

提案： ターゲット表に LONG 列も LONG RAW 列もない場合、または LONG 列にも LONG RAW 列にもテキストをロードしない場合は、SQL*Loader を使用して表にテキストを移入してください。

SQL*Loader を使用したロードについては、[付録 D](#) の「[SQL*Loader 例](#)」を参照してください。

ロード・ファイルは、ASCII フラット・ファイルで、テキスト表に格納されるドキュメントのプレーン・テキストおよび構造化データ（タイトル、作成者、日付など）を含みます。ロード・ファイルには、ドキュメントのテキストではなく、ドキュメントの実際のテキスト（形式設定済またはプレーン）が入っている別ファイルのポインタを格納することもできます。

注意： ctxload ユーティリティは、埋込みテキストおよびファイル・ポインタの両方が格納されているロード・ファイルはサポートしません。ロード・ファイルの作成時に、いずれかを選択する必要があります。

ctxload ユーティリティは、ロード・ファイル内でヘッダーで識別された各ドキュメントについて、表に 1 行を作成します。

参照： テキストのロードについてのロード・ファイルの例は、[付録 D](#) の「ctxload テキスト・ロード・ファイルの構造」を参照してください。

ドキュメントの更新 / エクスポート

ctxload ユーティリティは、特に、LONG RAW 列、LONG 列、BLOB 列および CLOB 列について、オペレーティング・システム・ファイルからのデータベース列の更新、およびファイルへのデータベース列のエクスポートをサポートします。

注意： 更新用バッファ / フェッチ用バッファにメモリーが大量（最大 2GB）に必要なのを防ぐため、データの更新 / エクスポートはセクション単位で実行されます。

このため、ドキュメントの更新 / エクスポートに必要なメモリーは、最小値の 16KB です。

ctxload 構文

```
ctxload -user username[/password] [@sqlnet_address]
        -name object_name
        -file file_name
        [-pk primary_key]
        [-export]
        [-update]
        [-thes]
        [-thescase y|n]
        [-thesdump]
        [-separate]
        [-longsize n]
        [-date date_mask]
        [-log file_name]
        [-trace]
        [-commitafter n]
```

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には、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

シソーラスをエクスポート / インポートするため ctxload を使用する場合、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に一致する場合、既存のシソーラスが上書きされることはなく、ctxload によってエラーが発生します。

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 は既存のファイルを上書きします。

ctxload を使用してテキスト列の単一行を更新する場合、テキスト列に挿入されるテキストを格納するファイルを指定するには、`file_name` を使用します。`-pk` で、宛先の行の主キーを指定します。

ctxload を使用してテキスト列の単一行をエクスポートする場合、テキストのエクスポート先ファイルを指定するには、`file_name` を使用します。`-pk` で、ソース行を識別します。

参照： ctxload インポート・ファイルの構造の詳細は、[付録 D「ロード例」](#)を参照してください。

オプションの引数

-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。

主キーがコンポジットの場合、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1つのセルの内容を、*-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 LONG 列、LONG RAW 列、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1つのセルの内容を、*-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 LONG 列、LONG RAW 列、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。

-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* 引数のみとともに使用した場合に有効です。

-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。

-separate

テキストのロードで、ロード・ファイルの各ドキュメントのテキストが、別のテキスト・ファイルのポインタであることを指定するには、このパラメータを挿入します。これによって、ctxload が、各テキスト・ファイルの内容を、指定された行の LONG 列または LONG RAW 列にロードします。

-longsize n

テキストのロードで、LONG 列または LONG RAW 列にロードする最大値を KB 単位で指定します。

最小値は 1 (1KB) です。最大値はマシンによって異なります。

注意： *longsize* の値には、数値のみ入力してください。KB を表す *K* または *k* を挿入しないでください。

-date

ctxload を使用して、ロードされた日付列に使用する TO_CHAR 日付書式を指定します。

参照： 使用可能な日付書式モデルの詳細は、『Oracle8i SQL リファレンス』を参照してください。

-log

ファイルのロード時に ctxload によって生成される、各国語対応メッセージ (NLS) の書き込み先ログ・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と、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 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・システム固有であり、USER_DUMP_DEST 初期化パラメータを使用して変更もできます。

参照： SQL トレースおよび USER_DUMP_DEST 初期化パラメータの詳細は、『Oracle8i 管理者ガイド』を参照してください。

-commitafter n

データベースが 1 回コミットされる前に、表に挿入される行 (ドキュメント) の数を指定します。デフォルトは 1 です。

例

この項では、ctxload が実行可能な操作の例を示します。

参照： ドキュメントのロード例の詳細は、[付録 D「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
```

単一テキスト・フィールドのエクスポート

次の例では、主キー値 1 によって識別される単一テキスト・フィールドをファイル myfile ヘエクスポートします。索引 myindex は、テキスト列を識別します。

```
ctxload -user scott/tiger -export -name myindex -file myfile -pk 1
```

コンポジット主キーによって識別される単一テキスト・フィールドをエクスポートするには、次のように、引用符で主キーを囲み、値をカンマで区切る必要があります。

```
ctxload -user scott/tiger -export -name myindex -file myfile -pk "Oracle,1"
```

単一テキスト・フィールドの更新

次の例では、主キー値 1 で識別される単一テキスト・フィールドを myfile の内容で更新します。索引 myindex は、テキスト列を識別します。

```
ctxload -user scott/tiger -update -name myindex -file myfile -pk 1
```

コンポジット主キーによって識別される単一テキスト・フィールドを更新するには、次のように、引用符で主キーを囲み、値をカンマで区切る必要があります。

```
ctxload -user scott/tiger -update -name myindex -file myfile -pk "Oracle,1"
```

知識ベース拡張コンパイラ (ctxkbtc)

ctxkbtc コンパイラは、1つ以上の指定されたシソーラスを取得し、*interMedia Text* 知識ベースでコンパイルして、拡張知識ベースを作成します。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。

拡張知識ベースは、知識ベースでオーバーラップする語句および関係をオーバーライドします。拡張知識ベースは、テーマの索引付け、英語での ABOUT 問合せの処理、ドキュメント・サービスでのドキュメント・テーマの抽出のように、知識ベースを使用する作業でアクセスされます。

参照： *interMedia Text* の知識ベースの詳細は、[付録 J「知識ベース - カテゴリの階層」](#)を参照してください。

ABOUT 演算子の詳細は、[第 4 章の「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。

構文

```
ctxkbtc -user uname/passwd  
        [-name thesname1 [thesname2 ... thesname16]]  
        [-revert]  
        [-verbose]  
        [-log filename]
```

-user

拡張知識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。

-name

拡張知識ベースを作成するために知識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、すでに ctxload でロードされている必要があります。

-revert

拡張知識ベースを、*interMedia Text* のデフォルトの知識ベースに回復します。

-verbose

NLS 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。

使用上の注意

テーマ索引付けの実行中は、知識ベースの拡張は実行できません。

また、拡張知識ベースを使用するには、*interMedia Text* の機能を使用しているすべての SQL セッションを終了させ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するため、管理者または技術マネージャのみが知識ベースを拡張してください。

ctxkbtc を 2 回実行すると、前の拡張は削除されます。

コンパイルされる前に、各シソーラスは、ctxload の "-thescase Y" オプションによって大 / 小文字が区別されて *interMedia Text* にロードされる必要があります。

シソーラスの語句の制約

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。

シソーラス関係の制約

次の制約が、シソーラス関係に適用されます。

- BTG および BTP は BT と同じです。NTG および NTP は NT と同じです。
- 優先語句のみが、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、独自の優先語句として扱われます。
- 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、そのうちの 1 つのみが優先語句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 つの優先語句および最大 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、その語句の先祖または子になることができません。優先語句は、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は許容します。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、既存の知識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。

新規語句の既存語句へのリンク

テーマを検証する場合に最適な結果を得るために、新規語句を知識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。

参照： 知識ベースの詳細は、[付録 J「知識ベース - カテゴリの階層」](#)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 問合せでコールし直す必要があります。また、要旨およびテーマ・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。

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知識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-Inflammatory Agents, Antirheumatic Agents, and Inflammation Mediators

NT Antineoplastic and Immunosuppressive Agents

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 を知識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
```

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た知識ベースに対して優先されます。

サイズ制限

次の表に、拡張知識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
拡張知識ベースの新規語句数	100 万
知識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

問合せのチューニング

この付録では、応答時間を短縮するための問合せのチューニング方法について説明します。
この付録の内容は次のとおりです。

- 統計を使用した問合せの最適化
- 応答時間短縮のための問合せの最適化
- スループット向上のための問合せの最適化
- ブロック操作による問合せのチューニング

統計を使用した問合せの最適化

統計を使用して問合せを最適化する場合は、問合せの表および索引に関して収集された統計で、その問合せを最も効率的に処理できる実行計画が選択されます。オプティマイザは、次のパラメータに基づいて最適な実行計画を選択します。

- CONTAINS 述語の選択性
- 同じ問合せに含まれるその他の述語の選択性
- CONTAINS 述語を処理したときの CPU コストおよび I/O コスト

次の項では、統計を拡張可能問合せオプティマイザで使用方法について説明します。統計を使用して最適化を行うと、CONTAINS 述語の選択性およびコストをより正確に見積もることができるため、より適した実行計画が選択されます。

統計の収集

デフォルトでは、拡張可能問合せオプティマイザは使用可能です。拡張可能オプティマイザを使用するには、問い合わせる表の統計を計算する必要があります。そのためには、次の文を発行します。

```
ANALYZE TABLE <table_name> COMPUTE STATISTICS;
```

また、次のように表のサンプルに関する統計を見積もることができます。

```
ANALYZE TABLE <table_name> ESTIMATE STATISTICS 1000 ROWS;
```

または

```
ANALYZE TABLE <table_name> ESTIMATE STATISTICS 50 PERCENT;
```

これらの文は、*table_name* に対応付けられたその表の列およびすべての索引（B-tree、ビットマップまたはテキスト・ドメイン）を含め、その表に対応付けられたすべてのオブジェクトの統計を収集します。前述の ANALYZE コマンドは、必要に応じて何度でも発行して表中の統計を再収集できます。

参照： ANALYZE コマンドの詳細は、『Oracle8i SQL リファレンス』および『Oracle8i パフォーマンスのための設計およびチューニング』を参照してください。

テキスト・ドメイン・インデックスの統計を収集することによって、拡張可能問合せオプティマイザは次のことを行うことができます。

- CONTAINS 述語の選択性を見積り

- テキスト索引を使用したときの I/O コストおよび CPU コスト（ドメイン・インデックスを使用して CONTAINS を処理するときのコスト）の見積り
- CONTAINS() 関数を起動するごとの I/O コストおよび CPU コストの見積り

CONTAINS 述語の選択性を理解しておくことは、構造化問合せなどの、2 つ以上の述語で問い合わせる場合に効果的です。これによって、拡張可能問合せオプティマイザで、CONTAINS の評価にドメイン・インデックスを使用するか、または CONTAINS 述語を後フィルタとして適用するかを、より適切に判断できます。

例

次の構造化問合せについて考えます。

```
select score(1) from tab where contains(txt, 'freedom', 1) > 0 and author = 'King'
and year > 1960;
```

author 列は VARCHAR2 型で、year 列は NUMBER 型であるとします。また、author 列には B-tree 索引があるとします。

さらに、構造化された author 述語は、CONTAINS 述語および YEAR 述語よりも選択性が高いと想定します。つまり、構造化された述語 (author = 'King') が、YEAR 述語および CONTAINS 述語と比較して、author 述語はたとえばそれぞれ 1000 行および 1500 行に対して 5 行と、かなり少ない数の行を戻すと想定します。

この場合、まず構造化された述語 (author = 'King') に B-tree 索引のレンジ・スキャンを行い、ROWID で表アクセスし、次に B-tree の表アクセスから戻された行に他の 2 つの述語を適用することで、この問合せをより効率的に行うことができます。

拡張可能問合せオプティマイザは、統計タイプを索引タイプ context に対応付けず、テキスト・ドメイン・インデックスで CONTAINS() 述語を処理する方を常に選択します。

注意： テキスト索引用に統計が収集されていない場合は、拡張可能問合せオプティマイザを使用禁止にすると同じ動作になります。

統計の再収集

1 つの索引の統計を再収集するには、次の文のいずれかを発行します。

```
ANALYZE INDEX <index_name> COMPUTE STATISTICS;
```

または

```
ANALYZE INDEX <index_name> ESTIMATE STATISTICS SAMPLE 1000 ROWS;
```

または

```
ANALYZE INDEX <index_name> ESTIMATE STATISTICS SAMPLE 50 PERCENT;
```

統計の削除

表に対応付けられた統計を削除するには、次の文を発行します。

```
ANALYZE TABLE <table_name> DELETE STATISTICS;
```

1 つの索引の統計を削除するには、次の文を発行します。

```
ANALYZE INDEX <index_name> DELETE STATISTICS;
```

拡張可能問合せオブティマイザの使用禁止および使用可能

デフォルトでは、拡張可能問合せオブティマイザは使用可能です。拡張可能問合せオブティマイザを使用禁止にするには、次の文を発行します。

```
DISASSOCIATE STATISTICS FROM INDEXTYPES ConText;  
DISASSOCIATE STATISTICS FROM PACKAGES ctx_contains;
```

拡張可能問合せオブティマイザを使用禁止にした後も、再度、使用可能にできます。使用可能にするには、次の SQL 文を CTXSYS として発行します。

```
ASSOCIATE STATISTICS WITH INDEXTYPES ConText USING textoptstats;  
ASSOCIATE STATISTICS WITH PACKAGES ctx_contains USING textoptstats;
```

応答時間短縮のための問合せの最適化

デフォルトでは、問合せはスループット向上のために最適化されます。これによって、問合せは最短時間ですべての行を戻します。

ただし、大きなヒットリストからできるだけ短い時間で、最初の **n** 個のヒットのみを取得するときは、多くの場合（特に Web アプリケーションの場合）、問合せを応答時間が短縮されるように最適化する必要があります。

次の項では、テキスト問合せを応答時間の短縮のために最適化する方法について説明します。これには、2 つの方法があります。

- FIRST_ROWS ヒントの使用
- CHOOSE ヒントおよび DOMAIN_INDEX_SORT ヒントの使用

注意： いずれの方法も応答時間が短縮されるように最適化しますが、EXPLAIN PLAN で取得した 2 つの方法の実行計画は、実行した問合せとは異なる場合があります。

FIRST_ROWS による応答時間の短縮

デフォルトの問合せオプティマイザ・モードを、FIRST_ROWS ヒントを使用して、応答時間の短縮のために最適化するように変更できます。問合せが、応答時間が短縮されるように最適化された場合は、最初の **n** 行が最短時間で戻されます。

たとえば、次の PL/SQL ブロックでは、カーソルを使用して問合せの最初の 20 ヒットを取り出し、FIRST_ROWS ヒントを使用して応答時間が短縮されるように最適化しています。

```
declare
cursor c is
select /*+ FIRST_ROWS */ pk, score(1), col from ctx_tab
      where contains(txt_col, 'test', 1) > 0 order by score(1) desc;
begin
for i in c
loop
insert into t_s values(i.pk, i.col);
exit when c%rowcount > 21;
end loop;
end;
/
```

このカーソル *c* は SELECT 文で、ワード *test* を含む ROWID をソートして戻します。コードがカーソルをループして、最初の 20 行を取り出します。取り出された行は、一時表 *t_s* に格納されます。

FIRST_ROWS

ヒントを指定した場合には、Oracle はテキスト索引に ROWID をスコア順で戻すように指示します。

ヒントがない場合は、テキスト索引が CONTAINS 述語を満たすすべての行を未ソート状態で戻した後で、Oracle が ROWID をソートします。このように結果セット全体を取り出すと時間がかかります。

この問合せでは、最初の 20 ヒットのみが必要なため、ヒントを使用することでパフォーマンスが向上します。

注意： 問合せで最初の数ヒットのみが必要な場合は、FIRST_ROWS ヒントを使用してください。結果セット全体が必要な場合は、パフォーマンスの低下につながるため、このヒントは使用しないでください。

また、FIRST_ROWS ヒントは拡張可能問合せオプティマイザを使用可能にしてもなくても使用できます。

その他の FIRST_ROWS 動作

ヒットをスコア順で戻すようにテキスト索引に指示する他に、FIRST_ROWS ヒントは、応答時間が短縮されるように問合せを最適化するときに、ブロック操作を回避します。ブロック操作は、マージ結合、ハッシュ結合およびビットマップ操作を含みます。

その結果、応答時間の短縮のための最適化に FIRST_ROWS ヒントを使用すると、同じように応答時間の短縮のために最適化する CHOOSE を DOMAIN_INDEX_SORT で使用した場合とは実行計画が異なる場合があります。

SQL の EXPLAIN PLAN コマンドを使用して、問合せ実行計画を調べることができます。

参照： 問合せオプティマイザの詳細、および FIRST_ROWS、CHOOSE などのヒントの使用方法は、『Oracle8i 概要』および『Oracle8i パフォーマンスのための設計およびチューニング』を参照してください。

EXPLAIN PLAN コマンドの詳細は、『Oracle8i SQL リファレンス』を参照してください。

CHOOSE による応答時間の短縮

CHOOSE または ALL_ROWS オプティマイザ・ヒントを使用する場合は、問合せはスループットの向上のために最適化されます。これはデフォルトのオプティマイザ・モードです。このモードでは、Oracle はテキスト・ドメイン・インデックスに行をスコア順にソートして戻すように指示せずに、Oracle 側でテキスト索引からフェッチされたすべての行をソートします。

CHOOSE モードまたは ALL_ROWS モードで高速な応答を得るために最適化するには、次のように DOMAIN_INDEX_SORT ヒントを使用できます。

```
declare
cursor c is
select /*+ CHOOSE DOMAIN_INDEX_SORT */ pk, score(1), col from ctx_tab
      where contains(txt_col, 'test', 1) > 0 order by score(1) desc;
begin
for i in c
loop
insert into t_s values(i.pk, i.col);
exit when c%rowcount > 21;
end loop;
end;
/
```

注意： FIRST_ROWS のみを使用した場合と同様に、この方法でも応答時間が短縮されるように最適化できますが、EXPLAIN PLAN で取得した 2 つの方法の実際の実行計画は、実行した問合せとは異なる場合があります。

参照： 問合せオプティマイザの詳細、および FIRST_ROWS、CHOOSE などのヒントの使用方法は、『Oracle8i 概要』および『Oracle8i パフォーマンスのための設計およびチューニング』を参照してください。

EXPLAIN PLAN コマンドの詳細は、『Oracle8i SQL リファレンス』を参照してください。

スループット向上のための問合せの最適化

CHOOSE モードおよび ALL ROWS モード

デフォルトでは、問合せは CHOOSE モードおよび ALL ROWS モードでスループットが向上するように最適化されます。問合せがスループット向上のために最適化された場合は、すべての行が最短時間で戻されます。

FIRST_ROWS モード

FIRST_ROWS モードでは、可能な場合、拡張可能問合せオプティマイザは、テキスト・ドメイン・インデックスにスコア順にソートされた行を戻させて高速な応答を得るように最適化します。これは、FIRST_ROWS ヒントを使用する場合のデフォルトの動作です。

FIRST_ROWS でスループットをさらに向上するために最適化するには、DOMAIN_INDEX_NO_SORT ヒントを使用します。スループットの向上とは、すべての行を最短時間で問合せに取り込むことです。

次の例では、スコア順にソートした行を戻すためにテキスト・ドメイン・インデックスを使用せずにスループットを向上させています。かわりに、CONTAINS 述語を満たすすべての行が索引から取り出された後で、Oracle によって行がソートされます。

```
select /*+ FIRST_ROWS DOMAIN_INDEX_NO_SORT */ pk, score(1), col from ctx_tab
       where contains(txt_col, 'test', 1) > 0 order by score(1) desc;
```

参照： 問合せオプティマイザの詳細、および FIRST_ROWS、CHOOSE などのヒントの使用方法は、『Oracle8i 概要』および『Oracle8i パフォーマンスのための設計およびチューニング』を参照してください。

ブロック操作による問合せのチューニング

複数の述語を持つ問合せを発行すると、実行計画でブロック操作が行われる場合があります。たとえば、次の複合問合せについて考えます。

```
select docid from myindex where contains(text, 'oracle', 1) > 0
AND colA > 5
AND colB > 1
AND colC > 3;
```

すべての述語が非選択的で、colA、colB および colC がビットマップ索引を持つと想定します。Oracle のコストベースのオプティマイザは、次の実行計画を選択します。

```
TABLE ACCESS BY ROWIDS
  BITMAP CONVERSION TO ROWIDS
    BITMAP AND
      BITMAP INDEX COLA_BMX
      BITMAP INDEX COLB_BMX
      BITMAP INDEX COLC_BMX
    BITMAP CONVERSION FROM ROWIDS
      SORT ORDER BY
        DOMAIN INDEX MYINDEX
```

BITMAP AND はブロック操作のため、Oracle は、BITMAP AND 操作を実行する前に、*interMedia Text* のドメイン・インデックスから戻された ROWID とスコアのペアを一時的に保存する必要があります。

Oracle は、これらの ROWID とスコアのペアをメモリーに保存しようとします。ただし、これらの ROWID とスコアのペアを含む結果セットのサイズが SORT_AREA_SIZE 初期化パラメータを超える場合、Oracle は、これらの結果をディスク上の一時セグメントに排出します。

ディスクに結果を保存すると余分なオーバーヘッドが発生するため、次のように ALTER SESSION を使用して SORT_AREA_SIZE パラメータを増やすことによって、パフォーマンスを向上できます。

```
alter session set SORT_AREA_SIZE = <new memory size in bytes>;
```

たとえば、バッファを約 8MB に設定するには、次の文を発行します。

```
alter session set SORT_AREA_SIZE = 8300000;
```

参照： SORT_AREA_SIZE の詳細は、『Oracle8i リファレンス・マニュアル』を参照してください。

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する必要があります。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必要です。

この項では、次のタイプの結果表およびそれ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 B-1](#) に、CTX_QUERY.EXPLAIN がその結果を書き込む表の構造を示します。

表 B-1

列名	データ型	説明
EXPLAIN_ID	VARCHAR2(30)	FEEDBACK コールで指定した <i>explain_id</i> 引数の値
ID	NUMBER	問合せ実行ツリーでのそれぞれのノードに割り当てられた数値 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、左から右の順に順番が付けられます。
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称 有効な値は、 表 B-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 B-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

表 B-1

列名	データ型	説明
POSITION	NUMBER	すべて同じ PARENT_ID を持つノードの処理の順序 1 から始めて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 B-2 に、実行計画表および HFEEDBACK 表の OPERATION 列の有効な値を示します。

表 B-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

次の表に、実行計画表の OPTIONS 列の有効な値を示します。

表 B-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 B-4 に、CTX_QUERY.HFEEDBACK がその結果を書き込む表の構造を示します。

表 B-4

列名	データ型	説明
FEEDBACK_ID	VARCHAR2(30)	HFEEDBACK コールで指定した <i>feedback_id</i> 引数の値
ID	NUMBER	問合せ実行ツリーでのそれぞれのノードに割り当てられた数値 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、左から右の順に順番が付けられます。
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 B-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 B-5 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 から始めて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 B-6 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 B-6 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 B-6 を参照してください。

OPTIONS 列値

次の表に、FEEDBACK 表の OPTIONS 列の値を示します。

表 B-5

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 B-6

CTX_FEEDBACK_ITEM_TYPE メンバーおよびメソッド	型	説明
テキスト	メンバー	フィードバック語
カーディナリティ	メンバー	(将来の拡張用)
スコア	メンバー	(将来の拡張用)
ランク	メソッド	(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;
```

参照： HFEEDBACK 表およびその NESTED TABLE からの選択方法の例は、[第 10 章](#)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、表を作成する必要があります。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必要です。

この項では、次のタイプの結果表およびそれに必要な列について説明します。

- [フィルタ表](#)
- [要旨表](#)
- [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必要です。

表 B-7

列名	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

要旨表

要旨表には、CTX_DOC.GIST が生成する要旨 / テーマ・サマリーごとに 1 行が格納されます。

要旨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必要です。

表 B-8

列名	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知識ベース 内での表現方法によって異なります。 POV は、ドキュメントの要旨に対しては、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、要旨または 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、CTX_DOC.HIGHLIGHT が生成した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。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。

ドキュメントが書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して、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させることができます。

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必要です。

表 B-9

列名	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	1 の位置を持つドキュメントの始めと比較した、ドキュメント内のハイライト表示の位置
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必要です。

表 B-10

列名	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必要です。

表 B-11

列名	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

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 型の表に拡張を戻すことができます。オプションで、*restab* 引数で表の名前を指定します。

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ます。

EXP_REC フィールド	説明
<i>xrel</i>	<i>xrel</i> 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、 <i>xrel</i> 値は PHRASE です。翻訳に対しては、 <i>xrel</i> 値は言語です。
<i>xlevel</i>	<i>xlevel</i> フィールドは、リレーションのレベルです。これは、主に <i>xrel</i> が階層リレーション (BT*/NT*) の場合に使用されます。 <i>xlevel</i> フィールドは、 <i>xrel</i> が PHRASE のときは 0 (ゼロ) です。 <i>xlevel</i> フィールドは、TRSYN でのシノニムの翻訳の場合は 2 です。 <i>xlevel</i> 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。
<i>xphrase</i>	<i>xphrase</i> は関連語句です。関連語句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。

サポートされているフィルタ形式

この付録では、Inso フィルタ処理テクノロジー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。

- Inso フィルタ処理テクノロジー
- 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

Inso フィルタ処理テクノロジー

Oracle8i *interMedia* Text は、Inso Corporation からライセンス許可を受けたドキュメント・フィルタ処理テクノロジーを使用しています。このフィルタ処理テクノロジーによって、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、このテクノロジーによって、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。

参照： 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 パッケージを使用してドキュメントを HTML に変換するために Inso フィルタ処理テクノロジーを使用する場合、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、ご使用の環境は、この付録で説明するとおり、Inso フィルタ処理テクノロジー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ーは、Inso Corporation および Adobe Corporation 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

Inso フィルタ・テクノロジーは次のプラットフォームでサポートされています。

- Sun Solaris Sparc (2.4 ~ 2.6)
- IBM AIX RS 6000 (4.1.4 ~ 4.3)
- HP/UX HP9000 (9.0 ~ 11.0)
- Compaq Tru64 UNIX (4.0 以降)
- SGI IRIX (6.3 "New 32-bit")
- Intel 搭載の NT (x86 NT 3.51 以降)
- DEC Alpha NT (4.0 以降)

環境変数の場所

Inso フィルタ処理に関連するすべての環境変数は、*interMedia* Text で参照できる必要があります。これらの変数を次の場所に設定します。

- listener.ora ファイル
これによって、PL/SQL 処理 extproc が環境変数を参照できるようになります。
- ctxsrv サーバーが起動されるオペレーティング・システムのシェル
これによって、バックグラウンド DML を行う ctxsrv 処理が環境変数を参照できるようになります。

UNIX プラットフォームに対する考慮点

次の考慮点は、Solaris、IBM AIX、HP/UX、Compaq Tru64 UNIX および SGI 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 に *.flt ファイルへのパスを追加してください。具体的には、ファイル isunx2.flt の場所に設定します。
- Inso テクノロジーがファイルを \$HOME ディレクトリのサブディレクトリ (.inso) に書き込めるように、環境変数 \$HOME を設定してください。
- 実行している X-Window サーバーにアクセスするには、グラフィック変換を実行する必要があります。

Solaris

環境変数 \$PATH およびシステムの共有ライブラリのパス (\$LD_LIBRARY_PATH) 環境変数は、\$ORACLE_HOME/ctx/lib (Inso フィルタ処理の共有ライブラリの場所) を含むように設定してください。

IBM AIX

環境変数 \$PATH およびシステムの共有ライブラリのパス (\$LIBPATH) 環境変数は、\$ORACLE_HOME/ctx/lib (Inso フィルタ処理の共有ライブラリの場所) を含むように設定してください。

/usr/lib:/lib を \$LIBPATH に含める必要があります。これは、環境変数 \$LIBPATH が一度設定されると、これらのディレクトリはデフォルトでは検索されないためです。

HP/UX

環境変数 \$PATH およびシステムの共有ライブラリのパス (\$SHLIB_PATH) 環境変数は、\$ORACLE_HOME/ctx/lib (Inso フィルタ処理の共有ライブラリの場所) を含むように設定してください。

SGI

環境変数 `$PATH` およびシステムの共有ライブラリのパス (`$LD_LIBRARY_PATH`) 環境変数は、`$ORACLE_HOME/ctx/lib` (Inso フィルタ処理の共有ライブラリの場所) を含むように設定してください。

Compaq Tru64 UNIX

環境変数 `$PATH` およびシステムの共有ライブラリのパス (`$LD_LIBRARY_PATH`) 環境変数は、`$ORACLE_HOME/ctx/lib` (Inso フィルタ処理の共有ライブラリの場所) を含むように設定してください。

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。

- X サーバーを実行して、(ビットマップ・ファイル形式でなく) ベクトル・グラフィック形式をフィルタ処理します。X サーバーが存在しない (システムが X ライブラリを検出しない) 場合は、ベクトル・グラフィック変換は実行されません。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、ビットマップ形式の他に、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・テクノロジーが OLE2 オブジェクトを使用して行う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 テクノロジーはこれを使用して 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は、Inso テクノロジーは 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 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

次の表に、フィルタ処理に対して *interMedia Text* がサポートするすべてのドキュメント形式を示します。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) のために使用されます。このフィルタ処理テクノロジーは、Inso Corporation の HTML エクスポート・テクノロジーに基づいており、Inso Corporation からライセンス許可を受けています。

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提供されている場合、外部フィルタ・フレームワークを使用すると、Oracle はすべてのドキュメント形式が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	すべてのバージョン
IBM Revisable Form Text	すべてのバージョン
IBM FFT	バージョン 3.0 以前
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	バージョン 4.1 以前
DisplayWrite 2 & 3 (TXT)	すべてのバージョン
DisplayWrite 4 & 5	リリース 2.0 以前のバージョン
Enable	バージョン 3.0、4.0 および 4.5
First Choice	バージョン 3.0 以前

形式	バージョン
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 以前
MASS11	バージョン 8.0 以前
Microsoft Word	バージョン 6.0 以前
Microsoft Works	バージョン 2.0 以前
MultiMate	バージョン 4.0 以前
Navy DIF	すべてのバージョン
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 以前
PC-File+ Letter	バージョン 3.0 以前
PFS:Write	バージョン A、B および C
Professional Write	バージョン 2.1 以前
Q&A	バージョン 2.0
Samna Word	Samna Word IV+ 以前のバージョン
SmartWare II	バージョン 1.02
Sprint	バージョン 1.0 以前
Total Word	バージョン 1.2
Volkswriter 3 & 4	バージョン 1.0 以前
Wang PC (IWP)	バージョン 2.6 以前
WordMARC	Composer Plus 以前のバージョン
WordPerfect	バージョン 7.0 以前
WordStar	バージョン 7.0 以前
WordStar 2000	バージョン 3.0 以前
XyWrite	III Plus 以前のバージョン

ワード処理 - インターナショナル

形式	バージョン
一太郎	バージョン 8

ワード処理 - Windows

形式	バージョン
AMI/AMI Professional	バージョン 3.1 以前
Corel WordPerfect for Windows	バージョン 8.0 以前
JustWrite	バージョン 3.0 以前
Legacy	バージョン 1.1 以前
Lotus WordPro (Intel 搭載の NT のみ)	WordPro 96、97 および SmartSuite for the Millennium
Microsoft Windows Works	バージョン 4.0 以前
Microsoft Windows Write	バージョン 3.0 以前
Microsoft Word 97	Word 97
Microsoft Word 2000	Beta 2
Microsoft Word for Windows	バージョン 7.0 以前
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 以前
Professional Write Plus	バージョン 1.0
Q&A Write for Windows	バージョン 3.0
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 (Mac)	バージョン 2.0 以前
MacWrite II	バージョン 1.1

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 以前
Framework	バージョン 3.0
Lotus 1-2-3 (DOS & Windows)	バージョン 5.0 以前
Lotus 1-2-3 for SmartSuite 97	SmartSuite 97
Lotus 1-2-3 for SmartSuite for the Millennium	SmartSuite for the Millennium
Lotus 1-2-3 Charts (DOS & Windows)	バージョン 5.0 以前
Lotus 1-2-3 (OS/2)	バージョン 2.0 以前
Lotus 1-2-3 Charts (OS/2)	バージョン 2.0 以前
Lotus Symphony	バージョン 1.0、1.1 および 2.0
Microsoft Excel 97	Excel 97
Microsoft Excel 2000	Beta 2
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

形式	バージョン
Microsoft Windows Works	バージョン 4.0 以前
Microsoft Works (DOS)	バージョン 2.0 以前
Microsoft Works (Mac)	バージョン 2.0 以前
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 以前
QuattroPro for Windows	バージョン 8.0 以前
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 以前
dBASE	バージョン 5.0 以前
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 以前
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 以前
Microsoft Works (DOS)	バージョン 2.0 以前
Microsoft Works (Mac)	バージョン 2.0 以前
Paradox (DOS)	バージョン 4.0 以前
Paradox (Windows)	バージョン 1.0 以前

形式	バージョン
Personal R:BASE	バージョン 1.0
R:BASE 5000	バージョン 3.1 以前
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 以前
SmartWare II	バージョン 1.02

表示形式

形式	バージョン
PDF - Portable Document Format	バージョン 1.0、1.1 および 1.2 DEC Alpha NT ではサポートされて いません。 日本語の PDF はサポートされてい ません。

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
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
Freelance 96 for Windows 95	Freelance 96
Freelance for Windows 95	SmartSuite 97 および Millennium
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 以前
Microsoft PowerPoint for Windows	バージョン 7.0 以前
Microsoft PowerPoint 97	PowerPoint 97
Microsoft PowerPoint 2000	Beta 2
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

次の表に、Inso フィルタが認識するグラフィック形式を示します。これは、これらの形式を含むテキスト列を索引付けしてもエラーが発生しないことを意味します。このように、列がこれらの形式のどれを含んでいても問題ありません。

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
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 ～ 5.0
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 13 以前
EMF (Intel 搭載の NT のみ)	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
JPEG	すべてのバージョン
PCX	PC Paintbrush
PBM - Portable Bitmap	固有のバージョンなし
PCD - Kodak Photo CD	バージョン 1
Perfect Works (Draw)	Novell バージョン 2.0

形式	バージョン
PGM - Portable Graymap	固有のバージョンなし
PIC	Lotus
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
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 以前
TIFF CCITT Group 3 & 4	Fax Systems
WMF	Windows Metafile
WordPerfect Graphics [WPG および WPG2]	バージョン 2.0 以前
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
vCard Electronic Business Card	固有のバージョンなし

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があるドキュメントは、Inso フィルタではサポートされていません。

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](#)
- [ctxload テキスト・ロード・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 列を主キーとしています。これは、テキスト表に必要な制約です。text 列は VARCHAR2 です。

```
create table docs (id number primary key, text varchar2(80));
```

text 列を移入するには、次の 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

参照： SQL*Loader の使用方法の詳細は、『Oracle8i ユーティリティ・ガイド』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。これは、テキスト表に必要な制約です。ドキュメントは、BLOB 型である text 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 format 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 text 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句（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 はインポートのとき、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）は予約語のため、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（共通、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（共通、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（共通、部分またはインスタンス）語がなく、1 つ以上の下位（共通、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（*interMedia Text* のシソーラスでは、BT/NT、BTG/NTG、BTP/NTP および BTI/NIT 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、*interMedia Text* シソーラスでは明示的に定義されません。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 の関連語句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* という関連語句があります。

注意： 関連語句には推移性はありません。句に 2 つ以上の関連語句がある場合、その語はそれぞれ親句と関連しているだけであり、関連語句同士が互いに関連しているではありません。

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形式設定済の階層 (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句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句のエントリは、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句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 マーカーを必要とします。

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
```

- 複数ワードを持つ語句を、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なので、*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。

- 関連語句エントリは、句または別の関連語句エントリの後に続けて指定する必要があります。
- 関連語句エントリは 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句を指定します。
- 複数の関連語句には複数の 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句を指定できます。

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

ctxload テキスト・ロード・ファイルの構造

ctxload テキストのロード・ファイルは、各ドキュメントおよびドキュメントの構造化データに、次の形式を使用する必要があります。

```
<TEXTSTART: col_name1=doc_data, col_name2=doc_data,...col_nameN=doc_data>
text. . .
<TEXTEND>
```

各パラメータの説明を次に示します。

<TEXTSTART: ... >

ドキュメントの始まりを示すヘッダー・マーカーです。ドキュメントの構造化データを指定するために、次のフィールドを1つ以上含みます。

col_name

ドキュメントの構造化データを格納する列の名前です。

doc_data

col_name に格納される構造化データです。

text

ロード対象のドキュメントのテキスト、またはロード対象のテキストが格納されているオペレーティング・システム・ファイルの名前（および必要な場合は位置）です。

注意： *text* のデータ（テキストの文字列およびファイル名ポインタ）は、リテラル・データ（英数字以外のすべての文字および空白を含む）として ctxload によって処理されます。このため、*text* には、ctxload で処理するデータを正確に入力する必要があります。

たとえば、ctxload を使用して複数の個別ファイルからテキストをロードする場合、ロード・ファイル内のファイル名には、テキストが格納されているオペレーティング・システム・ファイルの名前を正確に入力する必要があります。ファイル名に1つでも空白があると、ctxload はそのファイルを見つけることができないため、テキストはロードされません。

<TEXTEND>

ドキュメントの終わりを示します。

ロード・ファイルの構造

ロード・ファイルの構造には、次の条件があります。

- ドキュメントをロードする場合、ドキュメントのテキスト、または別ファイルのポインタのいずれかがロード・ファイルに格納されている必要があります。
- 1つのロード・ファイルで埋込みテキストと別ファイルのポインタの両方は使用できません。
- ドキュメントのテキストがロード・ファイルに埋め込まれている場合、そのテキストは ASCII 形式である必要があります。
- 別ファイルのポインタが使用されている場合、そのファイルのテキストは、ASCII 形式またはアプリケーション固有の形式（MS Word など）にします。
- 別ファイルにあるテキストがアプリケーション固有の形式の場合、その形式は ConText でサポートされ、LONG RAW 列にロード可能な形式である必要があります。
- 別ファイルには1つずつドキュメントが格納されている必要があります（別ファイルの内容は、表に行単位で格納されます）。

ロード・ファイル構文

テキストのロード・ファイルで使用される構文には次の条件が適用されます。

- <TEXTSTART: ... > および <TEXTEND> は、それぞれ改行して開始する必要があります。
- <TEXTSTART: ... > 文字列にある構造化データのパラメータは、どのような順序でも指定できます。
- col_name とそれに対応する doc_data の始まりとの間に、改行（ハード / ソフトのいずれも）を入れることはできません。

注意： doc_data の値のすべてを col_name と同じ行に指定する必要はありませんが、その値の始まりは col_name と同じ行に指定してください。

- 最初の col_name は 'TEXTSTART:' と同じ行に指定する必要があります。
- <TEXTSTART: ... > 文字列の最後を示す「>」文字は、ドキュメントの最後の doc_data フィールドと同じ行に指定する必要があります。
- 構造化データおよび LONG データは、2 行にまたがって指定できます。

- 一重引用符 (') は、doc_data 中ではエスケープ・シークエンスを使用して表現する必要があります (たとえば、*don't* は *don''t* と入力する必要があります)。
- 各 <TEXTSTART: ... > 文字列の直後には、ドキュメントのテキストまたは別ファイルのポインタを続ける必要があります。
- テキストまたはファイル・ポインタは、完成した <TEXTSTART: ... > 文字列の次に、改行して続ける必要があります。
- ロード・ファイルの最後の文字は改行文字にする必要があります。

ロード・ファイルの埋込みテキスト例

次の例は、構造化された従業員情報、つまり従業員番号 (1010、1024) や従業員名 (Joe Smith、Mary Jones)、および各ドキュメントのテキストなどを含む、形式設定済テキストのロード・ファイルを示しています。

```
<TEXTSTART: EMPNO=1000, ELNAME='Smith', EFNAME='Joe'>
Joe has an interesting resume, includes...cliff-diving.
<TEXTEND>
<TEXTSTART: EMPNO=1024, EFNAME='Mary', ELNAME='Jones'>
Mary has many excellent skills, including...technical,
marketing, and organizational. Team player.
<TEXTEND>
```

ロード・ファイルのファイル名ポインタ例

次の例は、構造化された従業員情報、つまり従業員番号 (1010、1024) や従業員名 (Joe Smith、Mary Jones)、および各ドキュメントのファイル・ポインタなどを含む、形式設定済テキストのロード・ファイルを示しています。

```
<TEXTSTART: EMPNO=1024, EFNAME='Mary', ELNAME='Jones'>
mjones.doc
<TEXTEND>
<TEXTSTART: EMPNO=1000, EFNAME='Joe', EFNAME='Smith'>
jsmith.doc
<TEXTEND>
```

注意： この例でロード・ファイルを使用するには、ctxload を実行するとき、コマンドラインに `-separate` 引数を指定する必要があります。

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
- デンマーク語 (DA)
- オランダ語 (NL)
- フィンランド語 (FI)
- フランス語 (FR)
- ドイツ語 (DE)
- イタリア語 (IT)
- ポルトガル語 (PR)
- スペイン語 (ES)
- スウェーデン語 (SE)

英語

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	

デンマーク語 (DA)

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）

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	eerdad	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
althoewel	bovenvermeld	enigszins	het	later	of	sindsdien	voor	wier
alias	buiten	enkel	hierbeneden	liever	ofschoon	slechts	vooraf	wij
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 (FI)

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	milloinkin	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 (FR)

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	certain	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 (DE)

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	zwischen
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 (IT)

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（PR）

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 (ES)

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	misimos	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 (SE)

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	mycket	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	

代替スペルの規則

この付録では、*interMedia 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。この章では、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](#)
- [デンマーク語](#)
- [スウェーデン語](#)

概要

この章では、*interMedia 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わ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

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。次の表に、*interMedia 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

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。次の表に、*interMedia 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

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。次の表に、*interMedia 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ます。

文字	代替スペル
ä	ae
å	aa
ö	oe
Ä	AE
Å	AA
Ö	OE

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必要があります。

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。これは、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 管理者が、ALTER INDEX の使用またはバックグラウンドでの ctxsrv の実行によって索引を同期化する必要があります。

参照： ALTER INDEX の詳細は、第 2 章の「ALTER INDEX」を参照してください。

ctxsrv の詳細は、第 12 章の「ctxsrv」を参照してください。

この付録では、*interMedia Text* が提供するすべてのビューについて説明します。

概要

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_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_SERVERS](#)
- [CTX_SQES](#)
- [CTX_STOPLISTS](#)
- [CTX_STOPWORDS](#)
- [CTX_SUB_LEXERS](#)
- [CTX_THESAURI](#)
- [CTX_USER_INDEXES](#)
- [CTX_USER_INDEX_ERRORS](#)
- [CTX_USER_INDEX_OBJECTS](#)
- [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_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わ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わ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 (INDEXED または INDEXING)
IDX_LANGUAGE_COLUMN	VARCHAR2(256)	ベース表の言語列名
IDX_FORMAT_COLUMN	VARCHAR2(256)	ベース表のフォーマット列名
IDX_CHARSET_COLUMN	VARCHAR2(256)	ベース表のキャラクタ・セット列名

CTX_INDEX_ERRORS

このビューは、DML エラーを表示し、CTXSYS で問い合わ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わ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_VALUES

このビューは、索引で使用す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わ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。CTXSYS で、このビューを問い合わ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わ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

CTX_OBJECT_ATTRIBUTE_LOV

このビューは、*interMedia Text* によって提供される特定のオブジェクト属性の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わ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わせることができます。

列名	型	説明
PAR_NAME	VARCHAR2(30)	パラメータ名 <ul style="list-style-type: none">max_index_memorydefault_index_memorydefault_datastoredefault_filter_binarydefault_filter_textdefault_filter_filedefault_section_htmldefault_section_textdefault_lexerdefault_wordlistdefault_stoplistdefault_storagelog_directory
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わ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_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、*interMedia Text* ユーザーによって作成されるプリファレンスおよび *interMedia 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わ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わ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わ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 または SPECIAL 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・セクションの参照できる標識)

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わ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_SERVERS

このビューは、現在アクティブな ctxsrv サーバーを表示します。CTXSYS でのみ、このビューを問い合わせることができます。

列名	型	説明
SER_NAME	VARCHAR2(60)	サーバー識別子
SER_STATUS	VARCHAR2(8)	サーバーのステータス (IDLE、RUN、EXIT)
SER_ADMMBX	VARCHAR2(60)	サーバーの管理パイプ・メールボックスの名前
SER_OOBBMX	VARCHAR2(60)	サーバーのバンド外メールボックスの名前
SER_SESSION	NUMBER	使用されていません
SER_AUDSID	NUMBER	サーバーの監査セッション ID
SER_DBID	NUMBER	サーバーのデータベース ID
SER_PROCID	VARCHAR2(10)	使用されていません
SER_PERSON_MASK	VARCHAR2(30)	サーバーのパーソナリティ・マスク
SER_STARTED_AT	DATE	サーバー起動日
SER_IDLE_TIME	NUMBER	サーバーのアイドル時間 (秒)
SER_DB_INSTANCE	VARCHAR2(10)	データベースのインスタンス ID
SER_MACHINE	VARCHAR2(64)	サーバー実行中のホスト・マシンの名前

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わ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わ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

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わ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
SPL_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

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わ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 *interMedia Text* ユーザーが、このビューを問い合わ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_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わ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_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)	ベース表の言語列名
IDX_FORMAT_COLUMN	VARCHAR2(256)	ベース表のフォーマット列名
IDX_CHARSET_COLUMN	VARCHAR2(256)	ベース表のキャラクタ・セット列名

CTX_USER_INDEX_ERRORS

このビューは、DML エラーを表示し、すべてのユーザーが問い合わ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わ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_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。すべてのユーザーが、このビューを問い合わ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わせることができます。

列名	型	説明
PND_INDEX_NAME	VARCHAR2(30)	索引名
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わ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わ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わ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 または SPECIAL 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・セクションの参照できる標識)

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わ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わ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、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わ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

CTX_USER_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わ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

CTX_USER_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わ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わせることができます。

列名	型	説明
THS_NAME	VARCHAR2(30)	シソーラス名

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、それから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。すべての表で、「ストップワード式」の列は、問合せ式または問合せ式のコンポーネントを説明します。また、右側の列は Oracle による問合せの書換えを表します。

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および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、元の実行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そのもの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
<i>non_stopword AND stopwords</i>	non_stopword
<i>non_stopword AND no_token</i>	non_stopword
<i>stopword AND non_stopword</i>	non_stopword
<i>no_token AND non_stopword</i>	non_stopword
<i>stopword AND stopwords</i>	no_token
<i>no_token AND stopwords</i>	no_token
<i>stopword AND no_token</i>	no_token
<i>no_token AND no_token</i>	no_token

OR 変換

ストップワード式	変換後の式
<i>non_stopword OR stopwords</i>	non_stopword
<i>non_stopword OR no_token</i>	non_stopword
<i>stopword OR non_stopword</i>	non_stopword
<i>no_token OR non_stopword</i>	non_stopword

ストップワード式	変換後の式
<i>stopword</i> OR <i>stopword</i>	<i>no_token</i>
<i>no_token</i> OR <i>stopword</i>	<i>no_token</i>
<i>stopword</i> OR <i>no_token</i>	<i>no_token</i>
<i>no_token</i> OR <i>no_token</i>	<i>no_token</i>

ACCUMulate 変換

ストップワード式	変換後の式
<i>non_stopword</i> ACCUM <i>stopword</i>	<i>non_stopword</i>
<i>non_stopword</i> ACCUM <i>no_token</i>	<i>non_stopword</i>
<i>stopword</i> ACCUM <i>non_stopword</i>	<i>non_stopword</i>
<i>no_token</i> ACCUM <i>non_stopword</i>	<i>non_stopword</i>
<i>stopword</i> ACCUM <i>stopword</i>	<i>no_token</i>
<i>no_token</i> ACCUM <i>stopword</i>	<i>no_token</i>
<i>stopword</i> ACCUM <i>no_token</i>	<i>no_token</i>
<i>no_token</i> ACCUM <i>no_token</i>	<i>no_token</i>

MINUS 変換

ストップワード式	変換後の式
<i>non_stopword</i> MINUS <i>stopword</i>	<i>non_stopword</i>
<i>non_stopword</i> MINUS <i>no_token</i>	<i>non_stopword</i>
<i>stopword</i> MINUS <i>non_stopword</i>	<i>no_token</i>
<i>no_token</i> MINUS <i>non_stopword</i>	<i>no_token</i>
<i>stopword</i> MINUS <i>stopword</i>	<i>no_token</i>
<i>no_token</i> MINUS <i>stopword</i>	<i>no_token</i>
<i>stopword</i> MINUS <i>no_token</i>	<i>no_token</i>
<i>no_token</i> MINUS <i>no_token</i>	<i>no_token</i>

NOT 変換

ストップワード式	変換後の式
<i>non_stopword</i> NOT <i>stopword</i>	<i>non_stopword</i>
<i>non_stopword</i> NOT <i>no_token</i>	<i>non_stopword</i>
<i>stopword</i> NOT <i>non_stopword</i>	<i>no_token</i>
<i>no_token</i> NOT <i>non_stopword</i>	<i>no_token</i>
<i>stopword</i> NOT <i>stopword</i>	<i>no_token</i>
<i>no_token</i> NOT <i>stopword</i>	<i>no_token</i>
<i>stopword</i> NOT <i>no_token</i>	<i>no_token</i>
<i>no_token</i> NOT <i>no_token</i>	<i>no_token</i>

EQUIValence 変換

ストップワード式	変換後の式
<i>non_stopword</i> EQUIV <i>stopword</i>	<i>non_stopword</i>
<i>non_stopword</i> EQUIV <i>no_token</i>	<i>non_stopword</i>
<i>stopword</i> EQUIV <i>non_stopword</i>	<i>non_stopword</i>
<i>no_token</i> EQUIV <i>non_stopword</i>	<i>non_stopword</i>
<i>stopword</i> EQUIV <i>stopword</i>	<i>no_token</i>
<i>no_token</i> EQUIV <i>stopword</i>	<i>no_token</i>
<i>stopword</i> EQUIV <i>no_token</i>	<i>no_token</i>
<i>no_token</i> EQUIV <i>no_token</i>	<i>no_token</i>

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
<i>non_stopword</i> NEAR <i>stopword</i>	<i>non_stopword</i>
<i>non_stopword</i> NEAR <i>no_token</i>	<i>non_stopword</i>
<i>stopword</i> NEAR <i>non_stopword</i>	<i>non_stopword</i>
<i>no_token</i> NEAR <i>non_stopword</i>	<i>non_stopword</i>
<i>stopword</i> NEAR <i>stopword</i>	<i>no_token</i>
<i>no_token</i> NEAR <i>stopword</i>	<i>no_token</i>
<i>stopword</i> NEAR <i>no_token</i>	<i>no_token</i>
<i>no_token</i> NEAR <i>no_token</i>	<i>no_token</i>

WEIGHT 変換

ストップワード式	変換後の式
<i>stopword</i> * n	no_token
<i>no_token</i> * n	no_token

THRESHOLD 変換

ストップワード式	変換後の式
<i>stopword</i> > n	no_token
<i>no_token</i> > n	no_token

WITHIN 変換

ストップワード式	変換後の式
<i>stopword</i> WITHIN <i>section</i>	no_token
<i>no_token</i> WITHIN <i>section</i>	no_token

知識ベース - カテゴリの階層

この付録では、カテゴリとなる知識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、それぞれ知識ベースの6つの主なブランチに対応しています。

- [Branch 1: science and technology](#)
- [Branch 2: business and economics](#)
- [Branch 3: government and military](#)
- [Branch 4: social environment](#)
- [Branch 5: geography](#)
- [Branch 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、知識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。この付録に含まれるのは、カテゴリとなる概念（階層の親ノード）のみです。

Branch 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

steroids	[6] hormones	[6] immune systems
	[7] androgens and anabolic	[7] antigens and antibodies
	[7] blood sugar hormones	[6] lymphatic systems
	[7] corticosteroids	[6] muscular systems
	[7] estrogens and progestins	[6] nervous systems
	[7] gonadotropins	[6] reproductive systems
	[7] pituitary hormones	[6] respiratory systems
	[7] thyroid hormones	[6] skeletal systems
	[6] lipids and fatty acids	[6] tissue systems
	[6] nucleic acids	[6] torso
	[6] sugars and carbohydrates	[6] urinary systems
	[6] toxins	[5] reproduction and development
	[6] vitamins	[4] populations and vivisystems
	[5] cell reproduction	[5] biological evolution
	[5] cell structure and function	[5] ecology
	[5] molecular genetics	[6] ecological conservation
[4] botany		[6] environmental pollution
	[5] algae	[5] genetics and heredity
[5] fungi		[4] zoology
	[5] plant diseases	[5] invertebrates
[5] plant kingdom		[6] aquatic sponges
	[6] ferns	[6] arthropods
	[6] flowering plants	[7] arachnids
	[7] cacti	[8] mites and ticks
	[7] grasses	[8] scorpions
	[6] mosses	[8] spiders
	[6] trees and shrubs	[7] crustaceans
	[7] conifers	[7] insects
	[7] deciduous trees	[6] coral and sea anemones
	[7] palm trees	[6] jellyfish
[5] plant physiology		[6] mollusks
	[6] plant development	[7] clams, oysters, and
[6] plant parts		
		[7] octopi and squids
[4] lower life forms		[7] snails and slugs
	[5] bacteria	[6] starfish and sea urchins
[5] viruses		[6] worms
	[4] paleontology	[5] vertebrates
[5] dinosaurs		[6] amphibians
	[4] physiology	[6] birds
[5] anatomy		[7] birds of prey
	[6] cardiovascular systems	[8] owls
	[6] digestive systems	[7] game birds
	[6] extremities and appendages	[7] hummingbirds
	[6] glandular systems	[7] jays, crows, and magpies
	[6] head and neck	[7] parrots and parakeets
	[7] ear anatomy	[7] penguins
	[7] eye anatomy	[7] pigeons and doves
	[7] mouth and teeth	[7] warblers and sparrows

swans	[7] water birds	[4] biometrics
	[8] ducks, geese, and	[5] voice recognition technology
	[8] gulls and terns	[4] genetic engineering
	[8] pelicans	[4] pharmaceutical industry
	[7] woodpeckers	[5] anesthetics
	[7] wrens	[6] general anesthetics
[6] fish		[6] local anesthetics
[7] boneless fish		[5] antagonists and antidotes
[8] rays and skates		[5] antibiotics, antimicrobials, and antiparasitics
[8] sharks		[6] anthelmintics
[7] bony fish		[6] antibacterials
[8] deep sea fish		[7] antimalarials
[8] eels		[7] antituberculars and
[8] tropical fish	antileprotics	
[7] jawless fish		[6] antifungals
[6] mammals		[6] antivirals
[7] anteaters and sloths		[6] local anti-infectives
[8] aardvarks		[5] antigout agents
[7] carnivores		[5] autonomic nervous system drugs
[8] canines		[6] neuromuscular blockers
[8] felines		[6] skeletal muscle relaxants
[7] chiropterans		[5] blood drugs
[7] elephants		[5] cardiovascular drugs
[7] hoofed mammals		[6] antihypertensives
[8] cattle		[5] central nervous system drugs
[8] goats		[6] analgesics and antipyretics
[8] horses		[6] antianxiety agents
[8] pigs		[6] antidepressants
[8] sheep		[6] antipsychotics
[7] hyraxes		[6] narcotic and opioid analgesics
[7] marine mammals	drugs	[6] nonsteroidal anti-inflammatory
[8] seals and walruses		[6] sedative-hypnotics
[9] manatees		[5] chemotherapeutics, antineoplastic
[8] whales and porpoises		
[7] marsupials	agents	
[7] monotremes		[5] dermatomucosal agents
[7] primates		[6] topical corticosteroids
[8] lemurs		[5] digestive system drugs
[7] rabbits		[6] antacids, adsorbents, and antiflatulents
[7] rodents		[6] antidiarrheals
[6] reptiles		[6] antiemetics
[7] crocodilians		[6] antiulcer agents
[7] lizards		[6] digestants
[7] snakes		[6] laxatives
[7] turtles		[5] eye, ear, nose, and throat drugs
[3] biotechnology		[6] nasal agents
[4] antibody technology		[6] ophthalmics
[5] immunoassays		

	[7] ophthalmic		[6] communicable diseases
vasoconstrictors			[7] sexually transmitted
	[6] otics, ear care drugs	diseases	
	[5] fluid and electrolyte balance		[5] injuries
drugs			[5] medical disabilities
	[6] diuretics		[5] neurological disorders
	[5] hormonal agents		[5] respiratory disorders
	[5] immune system drugs		[5] skin conditions
	[6] antitoxins and antivenins		[4] nutrition
	[6] biological response modifiers		[4] practice of medicine
	[6] immune serums		[5] alternative medicine
	[6] immunosuppressants		[5] medical diagnosis
	[6] vaccines and toxoids		[6] medical imaging
	[5] oxytocics		[5] medical personnel
	[5] respiratory drugs		[5] medical procedures
	[6] antihistamines		[6] physical therapy
	[6] bronchodilators		[6] surgical procedures
	[6] expectorants and antitussives		[7] cosmetic surgery
	[5] spasmolytics		[4] veterinary medicine
	[5] topical agents	[2] machinery	
[3] health and medicine		[3] machine components	
[4] healthcare industry		[2] mathematics	
[5] healthcare providers and practices		[3] algebra	
[5] medical disciplines and		[4] linear algebra	
specialties		[4] modern algebra	
	[6] cardiology	[3] arithmetic	
	[6] dentistry	[4] elementary algebra	
	[6] dermatology	[3] calculus	
	[6] geriatrics	[3] geometry	
	[6] neurology	[4] mathematical topology	
	[6] obstetrics and gynecology	[4] plane geometry	
	[6] oncology	[4] trigonometry	
	[6] ophthalmology	[3] math tools	
	[6] pediatrics	[3] mathematical analysis	
[5] medical equipment		[3] mathematical foundations	
[6] artificial limbs and organs		[4] number theory	
[6] dressings and supports		[4] set theory	
[5] medical equipment manufacturers		[4] symbolic logic	
[5] medical facilities		[3] statistics	
[4] medical problems		[2] mechanical engineering	
[5] blood disorders		[2] physics	
[5] cancers and tumors		[3] acoustics	
[6] carcinogens		[3] cosmology	
[5] cardiovascular disorders		[4] astronomy	
[5] developmental disorders		[5] celestial bodies	
[5] environment-related afflictions		[6] celestial stars	
[5] gastrointestinal disorders		[6] comets	
[5] genetic and hereditary disorders		[6] constellations	
[5] infectious diseases		[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

Branch 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**

Branch 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**

Branch 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] deshabelle
 - [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**

Branch 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] Tajikistan
 - [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] Moldova
 - [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

Branch 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

\ エスケープ文字, 5-3
{ } エスケープ文字, 5-3
! 演算子, 4-28
\$ 演算子, 4-29
- 演算子, 4-17
% ワイルド・カード, 4-41
* 演算子, 4-39
, 演算子, 4-9
= 演算子, 4-15
> 演算子, 4-33
? 演算子, 4-16
_ ワイルド・カード, 4-41

A

ABOUT 問合せ, 4-6
英語のデフォルト, 1-8
ハイライト・オフセット, 8-10
ハイライト・マークアップ, 8-13
表示の拡張, 10-6
例, 1-10, 4-7
ACCUMULATE 演算子, 4-9
スコア付け, 4-9
ストップワード変換, I-4
ADD_ATTR_SECTION プロシージャ, 7-3
ADD_FIELD_SECTION プロシージャ, 7-5
ADD_SPECIAL_SECTION プロシージャ, 7-9
ADD_STOP_SECTION プロシージャ, 7-12
ADD_STOPCLASS プロシージャ, 7-11
ADD_STOPTHEME プロシージャ, 7-14
ADD_STOPWORD プロシージャ, 7-15
ADD_SUB_LEXER プロシージャ, 7-17
例, 3-35

ADD_ZONE_SECTION プロシージャ, 7-20
adjective 属性, 3-37
adverb 属性, 3-37
ALL_ROWS ヒント
応答時間の短縮, A-7
ALTER INDEX, 2-2
REBUILD 構文, 2-4
RENAME 構文, 2-2
例, 2-7
ALTER_PHRASE プロシージャ, 11-3
ALTER_THESAURUS プロシージャ, 11-5
alternate_spelling 属性, 3-33
AND 演算子, 4-11
ストップワード変換, I-3
AUTO_SECTION_GROUP オブジェクト, 3-46, 7-28
AUTO ステミング, 3-39

B

base_letter 属性, 3-32
BASIC_LEXER オブジェクト, 3-28
キャラクタ・セットのサポート, 3-28
BASIC_LEXER システム定義プリファレンス, 3-52
BASIC_SECTION_GROUP オブジェクト, 3-46, 7-27
BASIC_STORAGE オブジェクト
属性, 3-43
デフォルト, 3-44
例, 3-44
BASIC_WORDLIST オブジェクト
属性, 3-39
例, 3-42
BFILE 列, 1-5
索引付け, 1-7, 2-14
binary 属性, 3-4, 3-15
BLOB 列, 1-5

- 索引付け, 1-7, 2-14
- ロード例, D-3
- BROADER TERM 演算子
 - 例, 4-12
- BROWSE_WORDS プロシージャ, 10-2
- BTG 演算子, 4-12
- BTG ファンクション, 11-10
- BTI 演算子, 4-12
- BTI ファンクション, 11-12
- BTP 演算子, 4-12
- BTP ファンクション, 11-14
- BT 演算子, 4-12
- BT ファンクション, 11-7

C

- CHARSET_FILTER
 - 属性, 3-19
 - 例, 3-20
- charset 属性, 3-19
- CHAR 列, 1-5
 - 索引付け, 2-14
- CHINESE_VGRAM_LEXER オブジェクト, 3-36
- CHOOSE ヒント
 - 応答時間の短縮, A-7
- CLOB 列, 1-5
 - 索引付け, 1-7, 2-14
- command 属性, 3-25
- composite 属性 (韓国語レクサー), 3-38
- composite 属性 (基本レクサー), 3-33
- CONTAINS 演算子
 - 構文, 2-12
 - 例, 2-13
- ConText の索引型, 2-14
- continuation 属性, 3-29
- COUNT_HITS プロシージャ, 10-5
- CREATE INDEX, 2-14
 - デフォルトのパラメータ, 3-55
 - 例, 1-7, 2-18
- CREATE_PHRASE プロシージャ, 11-16
- CREATE_PREFERENCE プロシージャ, 7-24
- CREATE_RELATION プロシージャ, 11-18
- CREATE_SECTION_GROUP プロシージャ, 7-27
- CREATE_STOPLIST プロシージャ, 7-30
- CREATE_THESAURUS ファンクション, 11-20
- CTX_ADM パッケージ
 - RECOVER, 6-2
 - SET_PARAMETER, 6-3
 - SHUTDOWN, 6-5
- CTX_CLASSES ビュー, H-4
- CTX_DDL パッケージ
 - ADD_ATTR_SECTION, 7-3
 - ADD_FIELD_SECTION, 7-5
 - ADD_SPECIAL_SECTION, 7-9
 - ADD_STOP_SECTION, 7-12
 - ADD_STOPCLASS, 7-11
 - ADD_STOPTHEME, 7-14
 - ADD_STOPWORD, 7-15
 - ADD_SUB_LEXER, 7-17
 - ADD_ZONE_SECTION, 7-20
 - CREATE_PREFERENCE, 7-24
 - CREATE_SECTION_GROUP, 7-27
 - CREATE_STOPLIST, 7-30
 - DROP_PREFERENCE, 7-31
 - DROP_SECTION GROUP, 7-32
 - DROP_STOPLIST, 7-33
 - REMOVE_SECTION, 7-36
 - REMOVE_STOPCLASS, 7-38
 - REMOVE_STOPTHEME, 7-39
 - REMOVE_STOPWORD, 7-40
 - SET_ATTRIBUTE, 7-41
 - UNSET_ATTRIBUTE, 7-43
- CTX_DOC_KEY_TYPE システム・パラメータ, 3-54
- CTX_DOC パッケージ, 8-1
 - FILTER, 8-2
 - GIST, 8-5
 - HIGHLIGHT, 8-10
 - MARKUP, 8-13
 - PKENCODE, 8-19
 - SET_KEY_TYPE, 8-20
 - THEMES, 8-21
 - 結果表, B-8
- CTX_FEEDBACK_ITEM_TYPE 型, B-6
- CTX_FEEDBACK_TYPE 型, 10-10, B-6
- CTX_INDEX_ERRORS ビュー, H-5
 - 例, 2-20
- CTX_INDEX_OBJECTS ビュー, H-5
- CTX_INDEX_VALUES ビュー, H-6
- CTX_INDEXES ビュー, H-4
- CTX_OBJECT_ATTRIBUTE_LOV ビュー, H-7
- CTX_OBJECT_ATTRIBUTES ビュー, H-7
- CTX_OBJECTS ビュー, H-6
- CTX_OUTPUT パッケージ, 9-1
 - END_LOG, 9-2

- LOGFILENAME, 9-3
- START_LOG, 9-4
- CTX_PARAMETERS ビュー, 3-54, H-8
- CTX_PENDING ビュー, H-9
- CTX_PREFERENCE_VALUES ビュー, H-9
- CTX_PREFERENCES ビュー, H-9
- CTX_QUERY パッケージ
 - BROWSE_WORDS, 10-2
 - COUNT_HITS, 10-5
 - EXPLAIN, 10-6
 - HFEEDBACK, 10-9
 - REMOVE_SQE, 10-13
 - STORE_SQE, 10-14
 - 結果表, B-2
- CTX_SECTION_GROUPS ビュー, H-10
- CTX_SECTIONS ビュー, H-10
- CTX_SERVERS ビュー, H-11
- CTX_SQES ビュー, H-11
- CTX_STOPLISTS ビュー, H-12
- CTX_STOPWORDS ビュー, H-12
- CTX_SUB_LEXERS ビュー, H-13
- CTX_THESAURI ビュー, H-13
- CTX_THES パッケージ, 11-1
 - ALTER_PHRASE, 11-3
 - ALTER_THESAURUS, 11-5
 - BT, 11-7
 - BTG, 11-10
 - BTI, 11-12
 - BTP, 11-14
 - CREATE_PHRASE, 11-16
 - CREATE_RELATION, 11-18
 - CREATE_THESAURUS, 11-20
 - DROP_PHRASE, 11-21
 - DROP_RELATION, 11-23
 - DROP_THESAURUS, 11-26
 - NT, 11-27
 - NTG, 11-30
 - NTI, 11-32
 - NTP, 11-34
 - OUTPUT_STYLE, 11-36
 - PT, 11-37
 - RT, 11-39
 - SN, 11-41
 - SYN, 11-40, 11-42
 - THES_TT, 11-45
 - TR, 11-46
 - TRSYN, 11-48

- TT, 11-50
- 結果表, B-12
- CTX_USER_INDEX_ERRORS ビュー, H-15
 - 例, 2-20
- CTX_USER_INDEX_OBJECTS ビュー, H-15
- CTX_USER_INDEX_VALUES ビュー, H-15
- CTX_USER_INDEXES ビュー, H-14
- CTX_USER_PENDING ビュー, H-16
 - 例, 2-9
- CTX_USER_PREFERENCE_VALUES ビュー, H-16
- CTX_USER_PREFERENCES ビュー, H-16
- CTX_USER_SECTION_GROUPS ビュー, H-17
- CTX_USER_SECTIONS ビュー, H-17
- CTX_USER_SQES ビュー, H-17
- CTX_USER_STOPLISTS ビュー, H-18
- CTX_USER_STOPWORDS ビュー, H-18
- CTX_USER_SUB_LEXERS ビュー, H-19
- CTX_USER_THESAURI ビュー, H-19
- CTXAPP ロール, 1-4
- ctxkbtc コンパイラ, 12-12
- ctxload, 12-6
 - インポート・ファイルの構造, D-6
 - 例, 12-11
 - ロード・ファイル形式, D-13
 - ロード・ファイル例, D-15
- ctxsrv
 - アクティブ・サーバーの表示, H-11
 - 概要, 1-9, 12-2
 - 起動シェルの Inso 変数, C-3
 - 停止, 6-5, 12-4
- ctxsrv のパーソナリティ・マスク, 12-2
- CTXSYS ユーザー, 1-4

D

- DATE 列, 1-5, 1-7, 2-14
- DEFAULT_DATASTORE システム定義索引付けプリファレンス, 3-50
- DEFAULT_DATASTORE システム・パラメータ, 3-55
- DEFAULT_FILTER_BINARY システム・パラメータ, 3-55
- DEFAULT_FILTER_FILE システム・パラメータ, 3-55
- DEFAULT_FILTER_TEXT システム・パラメータ, 3-55
- DEFAULT_INDEX_MEMORY システム・パラメータ, 3-54
- DEFAULT_LEXER システム定義索引付けプリファレンス, 3-51

DEFAULT_LEXER システム・パラメータ, 3-56
DEFAULT_SECTION_HTML システム・パラメータ,
3-55
DEFAULT_SECTION_TEXT システム・パラメータ,
3-55
DEFAULT_STOPLIST システム定義プリファレンス,
3-53
DEFAULT_STOPLIST システム・パラメータ, 3-56
DEFAULT_STORAGE システム定義プリファレンス,
3-53
DEFAULT_STORAGE システム・パラメータ, 3-55
DEFAULT_WORDLIST システム定義プリファレンス,
3-53
DEFAULT_WORDLIST システム・パラメータ, 3-56
DEFAULT シソーラス, 4-13, 4-19
DETAIL_DATASTORE オブジェクト, 3-4
例, 3-4
detail_key 属性, 3-4
detail_lineno 属性, 3-4
detail_table 属性, 3-4
detail_text 属性, 3-4
DIRECT_DATASTORE オブジェクト, 3-3
例, 3-3
DML
スコア付けへの影響, G-3
表示の保留, 2-9
DML エラー
表示, 12-4, H-5
DML キュー
表示, 12-4, H-9
DML 処理, 1-9
バックグラウンド, 12-2
バッチ, 2-4
例, 2-8
DOMAIN_INDEX_NO_SORT ヒント
スループットの向上例, A-8
DOMAIN_INDEX_SORT ヒント
応答時間の短縮例, A-7
DROP INDEX, 2-11
DROP_PHRASE プロシージャ, 11-21
DROP_PREFERENCE プロシージャ, 7-31
DROP_RELATION プロシージャ, 11-23
DROP_SECTION_GROUP プロシージャ, 7-32
DROP_STOPLIST プロシージャ, 7-33
DROP_THESAURUS プロシージャ, 11-26

E

END_LOG プロシージャ, 9-2
endjoin 属性, 3-31
EQUIVALENCE 演算子, 4-15
NEAR, 4-21
ストップワード変換, I-6
EXP_TAB 表タイプ, B-12
EXPLAIN プロシージャ, 10-6
結果表, B-2
例, 10-8
extproc 処理, C-3

F

FILE_DATASTORE オブジェクト, 3-7
例, 3-7
FILE_DATASTORE システム定義プリファレンス,
3-50
FILTER プロシージャ, 8-2
結果表, B-8
メモリー内の例, 8-3
例, 8-4
FIRST_ROWS ヒント
応答時間の短縮例, A-5
スループットの向上例, A-8
例, 1-11
ftp_proxy 属性, 3-10
fuzzy_match 属性, 3-41
fuzzy_numresults 属性, 3-41
fuzzy_score 属性, 3-41
FUZZY 演算子, 4-16

G

GIST プロシージャ, 8-5
結果表, B-8
例, 8-8

H

HFEEDBACK プロシージャ, 10-9
結果表, B-5
例, 10-10
HIGHLIGHT プロシージャ, 8-10
結果表, B-10
例, 8-12

HOME 環境変数

Inso の設定, C-3

HTML

索引付け, 3-24, 3-46

ゾーン・セクションの例, 7-21

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, 8-13

フィルタ処理, 1-13, 8-2

フィルタ処理のバイパス, 3-21

例のハイライト, 8-17

HTML_SECTION_GROUP オブジェクト, 3-46, 7-21, 7-27

NULL_FILTER, 3-24

HTML_SECTION_GROUP システム定義プリファレンス, 3-52

http_proxy 属性, 3-10

I

i_index_clause 属性, 3-44

i_table_clause 属性, 3-43

index_text 属性, 3-33

index_themes 属性, 3-33

INSERT 文

テキスト例のロード, 1-5

ロード例, D-2

INSO_FILTER オブジェクト, 3-21

キャラクタ・セット変換, 3-22

INSO_FILTER システム定義プリファレンス, 3-51

Inso フィルタ

索引プリファレンス・オブジェクト, 3-21

サポートされていない形式, C-13

サポートされている形式, C-5

サポートされているプラットフォーム, C-2
設定, C-2

interMedia Text

関連マニュアル, xiv

interMedia Text Manager ツール, 12-2, 12-5

interMedia Text の機能

概要, 1-3

コード・サンプル・ディレクトリ, 1-3

新機能

「interMedia Text の移行」を参照

問合せおよび索引, 1-11

J

JAPANESE_VGRAM_LEXER オブジェクト, 3-37

JOB_QUEUE_INTERVAL 初期化パラメータ, 2-15

JOB_QUEUE_PROCESSES 初期化パラメータ, 2-15

K

k_table_clause 属性, 3-43

KOREAN_LEXER オブジェクト, 3-37

L

listener.ora

環境変数の設定, C-3

LOB 列

索引付け, 1-7

ロード, D-3

LOG_DIRECTORY システム・パラメータ, 3-54, 9-3

LOGFILENAME プロシージャ, 9-3

LONG RAW 列

テキストのロード, 12-6

LONG 列, 1-5

索引付け, 2-14

テキストのロード, 12-6

M

MARKUP プロシージャ, 8-13

HTML ハイライト例, 8-17

結果表, B-10

例, 8-17

MAX_INDEX_MEMORY システム・パラメータ, 3-54

maxdocsize 属性, 3-10

maxthreads 属性, 3-9

maxurls 属性, 3-10

META タグ

ゾーン・セクションの作成, 7-21

フィールド・セクションの作成, 7-6

MINUS 演算子, 4-17

ストップワード変換, I-5

mixed_case 属性, 3-32

morpheme 属性, 3-38

MULTI_LEXER オブジェクト, 3-35

CREATE INDEX の例, 2-19

例, 3-35

N

n_table_clause 属性, 3-44
NARROWER TERM 演算子
 例, 4-18
NCLOB 列, 1-5, 1-7
NEAR 演算子, 4-20
 WITHIN, 4-44
 下位互換性, 4-22
 スコア付け, 4-21
 ストップワード変換, I-6
 その他の演算子, 4-21
 ハイライト表示, 4-22
NEAR 演算子の max_span パラメータ, 4-20
NEAR 演算子のクランプ・サイズ, 4-20
nested_column 属性, 3-15
NESTED_DATASTORE オブジェクト, 3-15
NESTED_DATASTORE 属性, 3-16
nested_lineno 属性, 3-15
nested_text 属性, 3-15
nested_type 属性, 3-15
newline 属性, 3-32
NEWS_SECTION_GROUP オブジェクト, 3-46, 7-28
no_proxy 属性, 3-10
NOT 演算子, 4-24
 ストップワード変換, I-5
NTG 演算子, 4-18
NTG ファンクション, 11-30
NTI 演算子, 4-18
NTI ファンクション, 11-32
NTP 演算子, 4-18
NTP ファンクション, 11-34
NT 演算子, 4-18
NT ファンクション, 11-27
NULL_FILTER オブジェクト, 3-24
NULL_FILTER システム定義プリファレンス, 3-51
NULL_SECTION_GROUP オブジェクト, 3-46, 7-27
NULL_SECTION_GROUP システム定義プリファレンス, 3-52
number 属性, 3-37
NUMBER 列, 1-5, 1-7, 2-14
numgroup 属性, 3-29
numjoin 属性, 3-30

O

onechar 属性, 3-37

OPTIONS 列

 HFEEDBACK 表, B-6
 実行計画表, B-4

Oracle Enterprise Manager, 12-5

OR 演算子, 4-25

 ストップワード変換, I-3

OUTPUT_STYLE プロシージャ, 11-36

output_type 属性, 3-13

P

p_table_clause, 3-44

PARAGRAPH キーワード, 4-46

PATH 環境変数

 Inso の設定, C-3

path 属性, 3-7

PKENCODE ファンクション, 8-19

PREFERRED TERM 演算子

 例, 4-26

printjoin 属性, 3-30

procedure 属性, 3-12

PT 演算子, 4-26

PT ファンクション, 11-37

punctuation 属性, 3-30

R

r_table_clause 属性, 3-44

RECOVER プロシージャ, 6-2

RELATED TERM 演算子, 4-27

REMOVE_SECTION プロシージャ, 7-36

REMOVE_SQE プロシージャ, 10-13

REMOVE_STOPCLASS プロシージャ, 7-38

REMOVE_STOPTHEME プロシージャ, 7-39

REMOVE_STOPWORD プロシージャ, 7-40

RFC 1738 URL 仕様, 3-8

RT 演算子, 4-27

RT ファンクション, 11-39

S

SCORE 演算子, 2-22

SENTENCE キーワード, 4-46

SET_ATTRIBUTE プロシージャ, 7-41

SET_KEY_TYPE プロシージャ, 8-20

SET_PARAMETER プロシージャ, 3-54, 6-3

SHUTDOWN プロシージャ, 6-5

skipjoin 属性, 3-31
SN プロシージャ, 11-41
SORT_AREA_SIZE パラメータ, A-9
SOUNDEX 演算子, 4-28
SQE 演算子, 4-30
SQL*Loader
 制御ファイル例, D-4
 データ・ファイル例, D-5
 例, D-3
sqlldr 例, D-3
SQL 演算子
 CONTAINS, 2-12
 SCORE, 2-22
SQL コマンド
 ALTER INDEX, 2-2
 CREATE INDEX, 2-14
 DROP INDEX, 2-11
START_LOG プロシージャ, 9-4
startjoin 属性, 3-31
stemmer 属性, 3-40
STEM 演算子, 4-29
STORE_SQE プロシージャ
 構文, 10-14
 例, 4-30
substring_index 属性, 3-41
SYNONYM 演算子, 4-31
SYN 演算子, 4-31
SYN ファンクション, 11-40, 11-42

T

theme_language 属性, 3-33
THEMES プロシージャ, 8-21
 結果表, B-11
THES_TT プロシージャ, 11-45
THRESHOLD 演算子, 4-33
 ストップワード変換, I-7
timeout 属性, 3-9
tohangeul 属性, 3-38
TOP TERM 演算子, 4-38
toupper 属性, 3-38
TRANSLATION TERM SYNONYM 演算子, 4-36
TRANSLATION TERM 演算子, 4-34
TRSYN 演算子, 4-36
TRSYN ファンクション, 11-48
TR 演算子, 4-34
TR ファンクション, 11-46

TT 演算子, 4-38
TT ファンクション, 11-50

U

udic 属性, 3-37
UNIX プラットフォーム
 Inso の変数の設定, C-3
UNSET_ATTRIBUTE プロシージャ, 7-43
URL_DATASTORE オブジェクト
 属性, 3-8
 例, 3-11
URL_DATASTORE システム定義プリファレンス,
 3-50
urlsize 属性, 3-10
URL の構文, 3-8
USER_DATASTORE オブジェクト, 3-12
 例, 3-13
USER_FILTER オブジェクト, 3-25
 例, 3-25
UTF8 キャラクタ・セット, 3-28, 3-37

V

VARCHAR2 列, 1-5
 索引付け, 2-14
verb 属性, 3-37

W

WEIGHT 演算子, 4-39
 ストップワード変換, I-7
whitespace 属性, 3-32
WITHIN 演算子, 4-43
 ストップワード変換, I-7
 制限事項, 4-49
 属性セクション, 4-46
 ネスト, 4-45
 優先順位, 4-4

X

xdic 属性, 3-38
XML_SECTION_GROUP オブジェクト, 3-46, 7-27
XML ドキュメント
 索引付け, 3-46, 7-28
 属性セクション, 7-3

問合せ, 4-46
ドキュメント・タイプを区別するセクション, 7-21

あ

アメリカ英語
索引デフォルト, 3-51
アメリカ英語のデフォルト, 1-8

い

イギリス英語
索引デフォルト, 3-51
イタリア語
ステミング, 3-39
提供されるストップリスト, E-8
ファジー・マッチング, 3-40
意味を限定する語
シソーラス・インポート・ファイル内, D-9
シソーラスの問合せ, 4-12
インポート・ファイル
構造, D-6
例, D-11

え

英語
索引デフォルト, 1-8
提供されるストップリスト, E-2
エスケープ文字, 5-3
エラー
索引付け, 2-9, 2-20
演算子, 4-1
ABOUT, 4-6
ACCUMULATE, 4-9
BROADER TERM, 4-12
EQUIVALENCE, 4-15
FUZZY, 4-16
MINUS, 4-17
NARROWER TERM, 4-18
NEAR, 4-20
NOT, 4-24
OR, 4-25
PREFERRED TERM, 4-26
RELATED TERM, 4-27
SOUNDEX, 4-28
SQE, 4-30

STEM, 4-29
SYNONYM, 4-31
THRESHOLD, 4-33
TOP TERM, 4-38
TRANSLATION TERM, 4-34
TRANSLATION TERM SYNONYM, 4-36
WEIGHT, 4-39
WITHIN, 4-43

演算子の拡張
表示, 10-6
演算子の優先順位, 4-3
EQUIVALENCE 演算子, 4-15
表示, 10-6
変更, 4-5, 5-2
例, 4-4

お

応答時間
最適化の例, 1-11
短縮, A-5
大 / 小文字が区別される索引
作成, 3-32
ドイツ語デフォルト, 1-8
オブジェクト
索引の表示, H-6
オブジェクト値
表示, H-6
オランダ語
コンボジット・ワードの索引付け, 3-33
索引デフォルト, 1-8, 3-51
ステミング, 3-39
提供されるストップリスト, E-4
ファジー・マッチング, 3-40

か

下位語問合せフィードバック, 10-9
階層関係
シソーラス・インポート・ファイル内, D-8
階層問合せフィードバック情報
生成, 10-9
概念問合せ
「ABOUT」を参照
外部フィルタ
指定, 3-25
拡張演算子

- FUZZY, 4-16
- SOUNDEX, 4-28
- STEM, 4-29
- 表示, 10-6
- 拡張可能問合せオペティマイザ, A-2
 - 使用可能 / 使用禁止, A-4
- カッコ
 - 文字のグループ化, 5-2
 - 優先順位の変更, 4-5, 5-2
- ガベージ・コレクション, 2-7
- 空の索引
 - 作成, 2-18
- 簡易中国語
 - 索引デフォルト, 3-52
- 環境変数
 - Inso フィルタの設定, C-2
- 韓国語
 - 索引デフォルト, 3-52
- 韓国語キャラクタ・セットのサポート, 3-37
- 韓国語テキスト
 - 索引付け, 3-37
- 完全最適化
 - 例, 2-7
- 管理ツール, 12-5
- 関連語問合せフィールドバック, 10-9
- 関連ランク付け
 - ワード問合せ, G-2

き

- 記憶域オブジェクト, 3-43
- 記憶域索引プリファレンス
 - 例, 7-25
- 記憶域デフォルト, 3-44
- 逆頻度スコア付け, G-2
- キャラクタ・セット
 - 韓国語, 3-37
 - 中国語のサポート, 3-36
 - 日本語, 3-37
 - 複合列の索引付け, 3-19
- キャラクタ・セット変換
 - INSO_FILTER, 3-22
- 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, 4-46

け

- 形式設定されたドキュメント
 - フィルタ処理, 3-21
- 結果表, B-1
 - CTX_DOC, B-8
 - CTX_QUERY, B-2
 - CTX_THES, B-12
- 結果バッファ・サイズ
 - 増加, A-9
- 言語
 - 索引付けのためのデフォルト設定, 1-7
 - 設定, 3-27
- 言語固有のデフォルト, 1-8

こ

- 高速最適化
 - 例, 2-7
- コード・サンプル・ディレクトリ, 1-3
- コードのサンプル, 1-3
- 語形変化のステミング
 - 使用可能, 3-40
- 混合列フォーマット
 - サポートされている形式, C-5
- コンボジット索引
 - オランダ語デフォルト, 1-8
 - ドイツ語デフォルト, 1-8
- コンボジット・テキストキー
 - コード化, 8-19
- コンボジット・ワード
 - 表示, 10-6
- コンボジット・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, 3-33

さ

- サーバー
 - DML, 12-2
 - アクティブの表示, H-11
 - 停止, 6-5, 12-4
- 索引

- 改名, 2-2
- 作成, 2-14
- 登録の表示, H-4
- 索引エラー, 2-9
 - 表示, 2-20
- 索引オブジェクト, 3-1
 - 表示, H-5, H-6
- 索引型の ConText, 2-14
- 索引最適化, 2-5
- 索引作成パラメータ
 - 例, 3-44
- 索引付け
 - テーマ, 3-33
 - パラレル, 2-15
 - マスター / ディテールの例, 3-5
 - 例, 1-7
- 索引付けのデフォルト
 - 表示, H-8
- 索引デフォルト
 - 言語固有, 1-8
 - 汎用, 1-7
- 索引内のワードのブラウズ, 10-2
- 索引の圧縮, 2-7
- 索引の改名, 2-2
- 索引の再構築
 - 構文, 2-4
 - 例, 2-7
- 索引の最適化, 2-5
 - 完全 (例), 2-7
 - 高速 (例), 2-7
- 索引の作成
 - カスタム・プリファレンスの例, 2-18
 - デフォルトの例, 2-18
- 索引の断片化, 2-18, 12-4
 - バックグラウンド DML, 12-4
- 索引の同期化, 1-9
 - ctxsrv, 12-2
 - バックグラウンド, 12-2
 - バッチ例, 2-8
 - 例, 2-8
- 索引のメンテナンス, 2-2
- 索引表領域パラメータ
 - 指定, 3-43
- 索引プリファレンス
 - 概要, 3-2
 - 作成, 3-2, 7-24
- 索引メンテナンス, 1-9, 2-2

- 索引要求
 - ログイン, 9-4
- 索引要求のログイン, 9-4
- サブ文字列索引
 - 作成の例, 3-42
- サブレクサー
 - 表示, H-13

し

- システム定義プリファレンス, 3-50
- システムのリカバリ
 - 手動, 6-2
- システム・パラメータ, 3-54
 - 索引付けのデフォルト, 3-55
- シソーラス
 - DEFAULT, 4-13
 - インポート / エクスポート, 12-6
 - インポート / エクスポート例, 12-11
 - 改名および切捨て, 11-5
 - 関係の作成, 11-16
 - コンパイル, 12-12
 - 削除, 11-26
 - 作成, 11-20
 - 参照用プロシージャ, 11-1
 - 情報の表示, H-13
- シソーラス・インポート・ファイル
 - 構造, D-6
 - 例, D-11
- シソーラス句
 - 削除, 11-21
 - 変更, 11-3
- シソーラスの最上位語
 - すべて検索, 11-45
- シソーラスのスコープ・ノート
 - 検索, 11-41
- シソーラスのリレーション
 - 削除, 11-23
 - 作成, 11-18
- シソーラスのロード, 12-6
- 実行計画表
 - 構造, B-2
 - 作成, 10-7
 - データ例の取出し, 10-8
- 実行計画表の OPERATION 列
 - 値, B-3
- 失敗した索引操作

- 再開, 2-5
- 失敗した索引の再開, 2-5
 - 例, 2-7
- 上位語問合せフィードバック, 10-9
- 書式
 - サポートされている, 1-5
- 新機能
 - 「interMedia Text の移行」を参照
- シングルバイト言語
 - 索引付け, 3-28

す

- スウェーデン語
 - 索引デフォルト, 1-8, 3-52
 - 代替スペル, F-5
 - 提供されるストップリスト, E-11
- スコア付け
 - ACCUMULATE, 4-9
 - DML の影響, G-3
 - NEAR 演算子, 4-21
- スコア付けの Salton の式, G-2
- スコア付けのアルゴリズム
 - ワード問合せ, G-2
- スコープ・ノート
 - 検索, 11-41
- ステミング, 3-40
 - 自動, 3-39
 - 使用可能な例, 3-42
 - デフォルト, 1-7
- ストアド・クエリー, 4-30
- ストアド・クエリー式
 - 削除, 10-13
 - 作成, 10-14
 - 定義の表示, H-11
 - 表示, H-17
- ストップクラス
 - 削除, 7-38
 - 定義, 7-11
- ストップテーマ
 - 削除, 7-39
 - 定義, 7-14
- ストップリスト
 - イタリア語, E-8
 - 英語, E-2
 - オランダ語, E-4
 - 概要, 3-48

- 削除, 7-33
- 作成, 3-48, 7-30
- スウェーデン語, E-11
- スペイン語, E-10
- デフォルト, 1-7
- デンマーク語, E-3
- ドイツ語, E-7
- 表示, H-12
- フィンランド語, E-5
- フランス語, E-6
- 変更, 3-48
- ポルトガル語, E-9
- ストップワード
 - 削除, 3-49, 7-40
 - ストップリストへのすべての表示, H-12
 - 定義, 7-15
 - 動的な追加, 2-4, 2-5, 3-48
- ストップワード変換, I-2
 - 表示, 10-6
- スペイン語
 - ステミング, 3-39
 - 提供されるストップリスト, E-10
 - ファジー・マッチング, 3-40
- スループット
 - 問合せの向上, A-8

せ

- 制御ファイル例
 - SQL*Loader, D-4
- セクション
 - オーバーラップ, 7-22
 - 繰返しゾーン, 7-22
 - 繰返しフィールド, 7-7
 - 削除, 7-36
 - 情報の表示, H-10
 - ゾーンの作成, 7-20
 - 属性の作成, 7-3
 - 動的な追加, 2-4
 - 動的な追加の制約, 2-9
 - ネスト, 7-22
 - フィールド作成, 7-5
- セクション・グループ
 - 削除, 7-32
 - 作成, 7-27
 - 情報の表示, H-10
- セクション・グループのタイプ, 3-46, 7-27

- セクション・グループの例, 3-47
- セクション検索, 4-43
 - ネスト, 4-45
- 全テーマ
 - 例, 8-23

そ

- ゾーン・セクション
 - 繰返し, 7-22
 - 作成, 7-20
 - 問合せ, 4-43
 - 動的な追加, 2-5
 - 動的な追加の例, 2-8
- ゾーン・セクションのオーバーラップ, 7-22
- 属性
 - alternate_spelling, 3-33
 - base_letter, 3-32
 - binary, 3-4
 - charset, 3-19
 - command, 3-25
 - composite, 3-33
 - continuation, 3-29
 - detail_key, 3-4
 - detail_linenos, 3-4
 - detail_table, 3-4
 - detail_text, 3-4
 - endjoin, 3-31
 - ftp_proxy, 3-10
 - fuzzy_match, 3-41
 - fuzzy_numresults, 3-41
 - fuzzy_score, 3-41
 - http_proxy, 3-10
 - i_index_clause, 3-44
 - i_table_clause, 3-43
 - index_text, 3-33
 - index_themes, 3-33
 - k_table_clause, 3-43
 - maxdocsize, 3-10
 - maxthreads, 3-9
 - maxurls, 3-10
 - mixed_case, 3-32
 - n_table_clause, 3-44
 - newline, 3-32
 - no_proxy, 3-10
 - numgroup, 3-29
 - numjoin, 3-30

- output_type, 3-13
- p_table_clause, 3-44
- path, 3-7
- printjoin, 3-30
- procedure, 3-12
- punctuation, 3-30
- r_table_clause, 3-44
- skipjoin, 3-31
- startjoin, 3-31
- stemmer, 3-40
- timeout, 3-9
- urlsize, 3-10
- whitespace, 3-32
- 可能な値の表示, H-7
- 使用禁止, 7-43
- 設定, 7-41
- 表示, H-7
- 属性セクション
 - WITHIN 例, 4-46
 - 定義, 7-3
 - 問合せ, 4-43
 - 動的な追加, 2-6, 2-8

た

- 大カッコ
 - 文字のグループ化, 5-2
 - 優先順位の変更, 4-5, 5-2
- 代替スベル
 - オランダ語デフォルト, 1-8
 - 概要, F-2
 - 使用可能な例, F-2
 - 使用禁止の例, 7-43, F-2
 - スウェーデン語, F-5
 - デンマーク語, F-4
 - デンマーク語デフォルト, 1-8
 - ドイツ語, F-3
 - ドイツ語デフォルト, 1-8
 - フィンランド語、ノルウェー語およびスウェーデン語のデフォルト, 1-8
- タグ付きテキスト
 - 検索, 4-43
- ダブル・トランケーテッド検索
 - パフォーマンスの向上, 3-41
- ダブル・トランケーテッド問合せ, 4-41
- 単一テーマ
 - 取得 (例), 8-23

- 断片化解消, 2-7
- 段落セクション
 - 定義, 7-9
 - 問合せ, 4-43
- 段落レベルの要旨およびテーマ・サマリー生成, 8-5

ち

- 知識カタログ
 - カテゴリの階層, J-1
- 知識カタログでの概念, J-1
- 知識カタログ内のカテゴリ, J-1
- 知識ベース拡張コンパイラ, 12-12
- 知識ベースの拡張, 12-12
- 中カッコエスケープ文字, 5-3
- 中国語キャラクタ・セットのサポート, 3-36
- 中国語テキスト
 - 索引付け, 3-36

て

- 提供されるストップリスト, E-1
- 停止セクション
 - 追加, 7-12
 - 動的な追加, 2-6
 - 動的な追加の例, 2-8
- データ記憶域
 - URL, 3-8
 - 外部, 3-7
 - 索引デフォルト, 1-7
 - ダイレクト, 3-3
 - プロシージャ的に定義, 3-12
 - マスター / ディテール, 3-4
 - 例, 7-24
- データストア・オブジェクト, 3-3
- テーマ
 - 索引付け, 3-33
 - ドキュメントに対しての作成, 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, 8-13
- テーマ索引
 - 英語のデフォルト, 1-8, 3-51
 - 作成, 3-33
- テーマ・サマリー
 - 生成, 8-5
 - 例, 8-9

- テーマ問合せ
 - 「ABOUT」を参照
- テーマのリスト
 - 例, 8-23
- 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, 8-17
 - オフセットの生成, 8-10
 - マークアップの生成, 8-13
- テーマ表
 - 構造, B-11
 - 例, 8-23
- テーマ・ベース, J-1
- テキスト・データ・ディクショナリ
 - クリーン・アップ, 6-2
- テキストのみの索引
 - 使用可能, 3-33
 - 例, 7-24
- テキストのロード
 - ctxload, 12-6, D-13
 - SQL*Loader 例, D-3
 - SQL の INSERT 例, 1-5, D-2
 - 概要, 1-5
- テキスト列
 - サポートするタイプ, 1-5, 2-14
 - ロード
 - 「テキストのロード」を参照
- デフォルト
 - 索引, 1-7
- デフォルトの索引
 - 例, 2-18
- デフォルトのパラメータ
 - 表示, 3-56
 - 変更, 3-56
- デンマーク語
 - 索引デフォルト, 1-8, 3-51
 - 代替スペル, F-4
 - 提供されるストップリスト, E-3

と

- fG スケープ文字, 5-3
- 問合せ
 - ACCUMULATE, 4-9
 - AND, 4-11
 - BROADER TERM, 4-12
 - EQUIVALENCE, 4-15

MINUS, 4-17
NARROWER TERM, 4-18
NOT, 4-24
OR, 4-25
PREFERRED TERM, 4-26
RELATED TERM, 4-27
SYNONYM, 4-31
THRESHOLD, 4-33
TOP TERM, 4-38
TRANSLATION TERM, 4-34
TRANSLATION TERM SYNONYM, 4-36
WEIGHT, 4-39
階層フィードバック, 10-9
概要, 1-10
ストアド, 4-30
スループットの向上のための最適化, A-8
ブロック操作, A-9
例, 2-13
問合せ機能, 1-11
問合せのチューニング
 応答時間, 1-11
 応答時間の短縮, A-5
問合せの最適化, A-2
 応答時間, 1-11, A-5
 スループット, A-8
 統計, A-2
 ブロック操作, A-9
問合せのチューニング
 結果バッファ・サイズの増加, A-9
 スループットの向上, A-8
 統計の利用, A-2
問合せ例, 1-10
ドイツ語
 alternate spelling 属性, 3-33
 コンボジット・ワードの索引付け, 3-33
 索引デフォルト, 1-8, 3-51
 ステミング, 3-39
 代替スペルの規則, F-3
 提供されるストップリスト, E-7
 ファジー・マッチング, 3-40
統計
 最適化, A-2
同形異義語
 BROADER TERM 問合せ, 4-13
 シソーラス・インポート・ファイル内, D-9
 問合せ内, 4-12
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
 テーマの作成, 8-21
 要旨およびテーマ・サマリー, 8-5
ドキュメント形式
 サポートされていない, C-13
 サポートされている, 1-5, C-5
ドキュメント・サービス
 概要, 1-13
 要求のロギング, 9-4
ドキュメントの表示方法
 概要, 1-13
 プロシージャ, 8-1
ドキュメントのロード
 ctxload, 12-6
 SQL*Loader, D-3
 方法, 1-5
ドキュメント・フィルタ処理
 Inso, C-2
特殊セクション
 定義, 7-9
 問合せ, 4-43
特殊文字のエスケープ, 5-3

に

日本語
 索引付け, 3-37
 索引デフォルト, 3-52
 ファジー・マッチング, 3-40
日本語キャラクタ・セットのサポート, 3-37

ね

ネストされたセクション検索, 4-45
ネストされたゾーン・セクション, 7-22

の

ノルウェー語
 索引デフォルト, 1-8, 3-52

は

ハイライト表
 構造, B-10
 例, 8-12

- ハイライト表示
 - NEAR 演算子, 4-22
 - オフセットの生成, 8-10
 - 概要, 1-13
 - マークアップの生成, 8-13
- 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, 3-40
- バックグラウンド DML, 12-2
- バックスラッシュ・エスケープ文字, 5-3
- バッチ処理
 - 例, 2-8
- パフォーマンス
 - ワイルド・カード検索, 4-42
- パラメータ
 - システム定義の表示, H-8
 - 設定, 6-3
- パラレル索引付け, 2-15
 - 例, 2-19

ひ

- ヒット数のカウント, 10-5
- ビュー, H-1
- 表記規則, xv
- 表構造
 - HFEEDBACK, B-5
 - 実行計画, B-2
 - テーマ, B-11
 - ハイライト, B-10
 - フィルタ, B-8
 - マークアップ, B-10
 - 要旨, B-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, 7-24
- ファジー・マッチング
 - 言語の指定, 3-41
 - 自動言語検出, 3-40
 - 使用可能な例, 3-42
 - デフォルト, 1-7
- フィールド・セクション

- WITHIN 例, 4-45
- 繰返し, 4-46
- 制限事項, 7-7
- 定義, 7-5
- 問合せ, 4-43
- 動的な追加, 2-6
- 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, 4-45
- フィルタ
 - Inso, 3-21, C-2
 - キャラクタ・セット, 3-19
 - ユーザー, 3-25
- フィルタ・オブジェクト, 3-18
- フィルタ形式
 - サポート, C-5
- フィルタ処理
 - 索引デフォルト, 1-7
 - プレーン・テキストおよび HTML, 1-13, 8-2
- フィルタ表
 - 構造, B-8
- フィンランド語
 - 索引デフォルト, 1-8, 3-52
 - 提供されるストップリスト, E-5
- 複合キャラクタ・セット列
 - 索引付け, 3-19
- 複合書式列
 - フィルタ処理, 3-21
 - 索引付け, 3-21
- フランス語
 - 提供されるストップリスト, E-6
 - ファジー・マッチング, 3-40
- フランス語ステミング, 3-39
- プリファレンス
 - 置換え, 2-4
 - 概要, 3-2
 - 索引付けの指定, 2-15
 - 削除, 7-31
 - 作成, 7-24
 - システム定義, 3-50
 - 表示, H-9
- プリファレンス・クラス
 - 表示, H-4
- プリファレンス値
 - 表示, H-9
- プリファレンスの置換, 2-4
- プレーン・テキスト
 - NULL_FILTER による索引付け, 3-24

- 索引付け, 3-22
- 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, 8-13
- フィルタ処理, 8-2
- フィルタ処理のバイパス, 3-21
- プレーン・テキストのフィルタ処理, 1-13
- ブロック操作
 - 問合せのチューニング, A-9
- 文セクション
 - 定義, 7-9
 - 問合せ, 4-43
- 文レベルの要旨およびテーマ・サマリー生成, 8-5

へ

- 変換
 - ストップワード, I-2

ほ

- 保留中の DML
 - 表示, 2-9, 12-4, H-9
- ボルトガル語
 - 提供されるストップリスト, E-9

ま

- マークアップ表
 - 構造, B-10
 - 例, 8-17
- マスター表 / ディテール表
 - 索引付けの例, 3-5
- マスター表 / ディテール表のデータ記憶域, 3-4
 - 例, 3-4, 7-25
- マルチ言語索引, 7-17
- マルチ言語表
 - 問合せ, 2-13, 3-36

め

- メモリー
 - 索引付け, 2-8, 2-17
 - 索引の同期化用, 2-5

も

- 文字
 - continuation, 3-29
 - newline の指定, 3-32
 - numgroup, 3-29
 - numjoin, 3-30
 - printjoin, 3-30
 - punctuation, 3-30
 - skipjoin, 3-31
 - startjoin および endjoin, 3-31
 - whitespace の指定, 3-32

や

- 予約語
 - エスケープ, 5-3

ゆ

- ユーザー
 - システム定義, 1-4
- ユーティリティ
 - ctxload, 12-6

よ

- 要旨
 - 生成, 8-5
 - 例, 8-8
- 要旨表
 - 構造, B-8
 - 例, 8-8
- 予約語, 5-4

れ

- レクサー・オブジェクト, 3-27
- 列型
 - 索引付けにサポートされている, 1-5, 2-14
- レフト・トランケーテッド検索
 - パフォーマンスの向上, 3-41

ろ

- ロード・ファイル
 - 形式, D-13

- 例, D-15
- ロール
 - システム定義, 1-4
- 論理演算子
 - NEAR, 4-21

わ

- ワード問合せ
 - 例, 1-10
- ワイルド・カード検索, 4-41
 - パフォーマンスの改善, 4-42

