

Oracle OLAP

リファレンス

リリース 2 (9.2.0.4)

2003 年 11 月

部品番号 : J08245-01

Oracle OLAP リファレンス, リリース 2 (9.2.0.4)

部品番号: J08245-01

原本名: Oracle OLAP Reference, Release 2 (9.2.0.4)

原本部品番号: B10334-01

Copyright © 2002, 2003, 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載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.....	xvii
対象読者	xvii
このマニュアルの構成	xviii
関連文書	xx
表記規則	xx
1 DBMS_AWM を使用したアナリティック・ワークスペースの作成	
概要	1-2
ソース・キューブの OLAP カタログ・メタデータの作成	1-3
ワークスペース・ディメンションの作成および移入	1-4
ワークスペース・キューブの作成および移入	1-4
アナリティック・ワークスペースにおけるキューブのデータの集計	1-5
ワークスペース・キューブへのリレーショナル・アクセスの有効化	1-5
DBMS_AWM によって作成したメタデータの表示	1-5
アクティブ・カタログ・ビュー	1-5
アナリティック・ワークスペース・メンテナンス・ビュー	1-6
DBMS_AWM プロシージャの理解	1-6
ディメンションに関するメソッド	1-7
キューブに関するメソッド	1-7
ディメンション・ロード仕様に関するメソッド	1-8
キューブ・ロード仕様に関するメソッド	1-9
集計仕様に関するメソッド	1-9
コンポジット仕様に関するメソッド	1-10
ワークスペース・ディメンションの作成およびリフレッシュ	1-11
ディメンションのメタデータのリフレッシュ	1-12

ディメンションをリフレッシュするタイミング	1-12
ディメンションのリフレッシュ後に必要な処理	1-13
ワークスペース・キューブの作成およびリフレッシュ	1-13
データ型の変換	1-14
キューブのメタデータのリフレッシュ	1-15
キューブをリフレッシュするタイミング	1-15
キューブのリフレッシュ後に必要な処理	1-16
スパースなデータの管理およびワークスペース・キューブの最適化	1-16
ディメンションの順序	1-17
コンポジット仕様の作成および変更	1-17
アナリティック・ワークスペースにおけるデータの集計	1-18
集計仕様の作成	1-19
集計メソッドの選択	1-20
ワークスペース・キューブへのリレーショナル・アクセスの作成	1-22
手順:有効化スクリプトの生成および実行	1-23
手順:有効化スクリプトの自動実行	1-23
OLAP API イネーブラ・プロシージャ	1-24
アナリティック・ワークスペースの有効化メタデータ	1-25
リレーショナル・アクセスの無効化	1-25
ディメンション・ビューのデフォルト名	1-25
ファクト・ビューのデフォルト名	1-26
ディメンション有効化ビューの列構造	1-27
ディメンション・ビューの例	1-28
グループID列	1-28
有効化ファクト・ビューの列構造	1-28
例:OLAP APIによるワークスペース・キューブへのアクセスを可能にする	1-30

2 CWM2 による OLAP カタログ・メタデータの作成

OLAP メタデータのエンティティ	2-1
ディメンションの作成	2-2
手順:OLAP ディメンションの作成	2-2
例:製品ディメンションの作成	2-3
手順:時間ディメンションの作成	2-6
例:時間ディメンションの作成	2-6
キューブの作成	2-9

手順: キューブの作成.....	2-9
例: Costs キューブの作成.....	2-9
OLAP メタデータのマッピング.....	2-10
列へのマッピング.....	2-11
ディメンションのマッピング.....	2-11
メジャーのマッピング.....	2-11
ファクト表とディメンション表の結合.....	2-11
OLAP メタデータの検証およびコミット.....	2-12
OLAP メタデータの検証.....	2-12
妥当性ステータスの表示.....	2-15
OLAP API のメタデータ表のリフレッシュ.....	2-15
プロシージャの起動.....	2-15
セキュリティ・チェックおよびエラー条件.....	2-15
パラメータのサイズ要件.....	2-16
パラメータの大 / 小文字識別要件.....	2-16
出力の送信.....	2-17
OLAP メタデータの表示.....	2-17

3 アクティブ・カタログ・ビュー

スタンダード・フォームのアクティブ・カタログ.....	3-1
スタンダード・フォームのクラス.....	3-2
アクティブ・カタログおよびスタンダード・フォームのクラス.....	3-2
例: アナリティック・ワークスペース・キューブの問合せ.....	3-3
アクティブ・カタログ・ビューの概要.....	3-4
ALL_OLAP2_AWS.....	3-5
ALL_OLAP2_AW_ATTRIBUTES.....	3-5
ALL_OLAP2_AW_CUBES.....	3-6
ALL_OLAP2_AW_CUBE_AGG_LVL.....	3-6
ALL_OLAP2_AW_CUBE_AGG_MEAS.....	3-7
ALL_OLAP2_AW_CUBE_AGG_OP.....	3-7
ALL_OLAP2_AW_CUBE_AGG_SPECS.....	3-8
ALL_OLAP2_AW_CUBE_DIM_USES.....	3-8
ALL_OLAP2_AW_CUBE_MEASURES.....	3-9
ALL_OLAP2_AW_DIMENSIONS.....	3-10
ALL_OLAP2_AW_DIM_HIER_LVL_ORD.....	3-10
ALL_OLAP2_AW_DIM_LEVELS.....	3-11

ALL_OLAP2_AW_PHYS_OBJ	3-11
ALL_OLAP2_AW_PHYS_OBJ_PROP	3-12

4 アナリティック・ワークスペース・メンテナンス・ビュー

アナリティック・ワークスペースの構築およびメンテナンス	4-1
例：ワークスペース・ディメンションのロード・パラメータおよび有効化パラメータの間合せ.....	4-2
アナリティック・ワークスペース・メンテナンス・ビューの概要	4-3
ALL_AW_CUBE_AGG_LEVELS	4-4
ALL_AW_CUBE_AGG_MEASURES	4-4
ALL_AW_CUBE_AGG_PLANS	4-5
ALL_AW_CUBE_ENABLED_HIERCOMBO	4-5
ALL_AW_CUBE_ENABLED_VIEWS	4-5
ALL_AW_DIM_ENABLED_VIEWS	4-6
ALL_AW_LOAD_CUBES	4-7
ALL_AW_LOAD_CUBE_DIMS	4-7
ALL_AW_LOAD_CUBE_FILTERS	4-8
ALL_AW_LOAD_CUBE_MEASURES	4-9
ALL_AW_LOAD_CUBE_PARMS	4-10
ALL_AW_LOAD_DIMENSIONS	4-10
ALL_AW_LOAD_DIM_FILTERS	4-11
ALL_AW_LOAD_DIM_PARMS	4-11

5 OLAP カタログ・メタデータのビュー

OLAP カタログのビューへのアクセス	5-1
多次元モデルのビュー	5-2
マッピング情報のビュー	5-3
ALL_OLAP2_AGGREGATION_USES	5-3
ALL_OLAP2_CATALOGS	5-4
ALL_OLAP2_CATALOG_ENTITY_USES	5-4
ALL_OLAP2_CUBES	5-5
ALL_OLAP2_CUBE_DIM_USES	5-5
ALL_OLAP2_CUBE_MEASURES	5-6
ALL_OLAP2_CUBE_MEASURE_MAPS	5-6
ALL_OLAP2_CUBE_MEAS_DIM_USES	5-7
ALL_OLAP2_DIMENSIONS	5-7
ALL_OLAP2_DIM_ATTRIBUTES	5-8
ALL_OLAP2_DIM_ATTR_USES	5-8

ALL_OLAP2_DIM_HIERARCHIES	5-9
ALL_OLAP2_DIM_HIER_LEVEL_USES	5-9
ALL_OLAP2_DIM_LEVELS	5-10
ALL_OLAP2_DIM_LEVEL_ATTRIBUTES	5-10
ALL_OLAP2_DIM_LEVEL_ATTR_MAPS	5-11
ALL_OLAP2_ENTITY_DESC_USES	5-11
ALL_OLAP2_FACT_LEVEL_USES	5-12
ALL_OLAP2_FACT_TABLE_GID	5-13
ALL_OLAP2_HIER_CUSTOM_SORT	5-13
ALL_OLAP2_JOIN_KEY_COLUMN_USES	5-14
ALL_OLAP2_LEVEL_KEY_COL_USES	5-15

6 OLAP 動的パフォーマンス・ビュー

OLAP パフォーマンス・ビューが参照するシステム表	6-1
OLAP パフォーマンス・ビューの概要	6-2
V\$AW_CALC	6-2
V\$AW_OLAP	6-4
V\$AW_SESSION_INFO	6-5

7 CWM2_OLAP_CATALOG

メジャー・フォルダの理解	7-1
例: メジャー・フォルダの作成	7-2
CWM2_OLAP_CATALOG サブプログラムの概要	7-3
ADD_CATALOG_ENTITY プロシージャ	7-3
CREATE_CATALOG プロシージャ	7-4
DROP_CATALOG プロシージャ	7-4
LOCK_CATALOG プロシージャ	7-5
REMOVE_CATALOG_ENTITY プロシージャ	7-5
SET_CATALOG_NAME プロシージャ	7-6
SET_DESCRIPTION プロシージャ	7-6
SET_PARENT_CATALOG プロシージャ	7-7

8 CWM2_OLAP_CUBE

キューブの理解	8-1
例: キューブの作成	8-2
CWM2_OLAP_CUBE サブプログラムの概要	8-3

ADD_DIMENSION_TO_CUBE プロシージャ	8-3
CREATE_CUBE プロシージャ	8-4
DROP_CUBE プロシージャ	8-5
LOCK_CUBE プロシージャ	8-5
REMOVE_DIMENSION_FROM_CUBE プロシージャ	8-6
SET_AGGREGATION_OPERATOR プロシージャ	8-6
SET_CUBE_NAME プロシージャ	8-8
SET_DEFAULT_CUBE_DIM_CALC_HIER プロシージャ	8-9
SET_DESCRIPTION プロシージャ	8-9
SET_DISPLAY_NAME プロシージャ	8-10
SET_MV_SUMMARY_CODE プロシージャ	8-10
SET_SHORT_DESCRIPTION プロシージャ	8-11

9 CWM2_OLAP_DIMENSION

ディメンションの理解	9-1
例: CWM2 ディメンションの作成	9-2
CWM2_OLAP_DIMENSION サブプログラムの概要	9-3
CREATE_DIMENSION プロシージャ	9-3
DROP_DIMENSION プロシージャ	9-4
LOCK_DIMENSION プロシージャ	9-5
SET_DEFAULT_DISPLAY_HIERARCHY プロシージャ	9-5
SET_DESCRIPTION プロシージャ	9-6
SET_DIMENSION_NAME プロシージャ	9-6
SET_DISPLAY_NAME プロシージャ	9-7
SET_PLURAL_NAME プロシージャ	9-7
SET_SHORT_DESCRIPTION プロシージャ	9-8

10 CWM2_OLAP_DIMENSION_ATTRIBUTE

ディメンション属性の理解	10-1
例: ディメンション属性の作成	10-2
CWM2_OLAP_DIMENSION_ATTRIBUTE サブプログラムの概要	10-4
CREATE_DIMENSION_ATTRIBUTE プロシージャ	10-4
DROP_DIMENSION_ATTRIBUTE プロシージャ	10-6
LOCK_DIMENSION_ATTRIBUTE プロシージャ	10-6
SET_DESCRIPTION プロシージャ	10-7
SET_DIMENSION_ATTRIBUTE_NAME プロシージャ	10-7
SET_DISPLAY_NAME プロシージャ	10-8

SET_SHORT_DESCRIPTION プロシージャ	10-9
11 CWM2_OLAP_HIERARCHY	
階層の理解	11-1
例: 階層の作成	11-2
CWM2_OLAP_HIERARCHY サブプログラムの概要	11-3
CREATE_HIERARCHY プロシージャ	11-3
DROP_HIERARCHY プロシージャ	11-4
LOCK_HIERARCHY プロシージャ	11-5
SET_DESCRIPTION プロシージャ	11-5
SET_DISPLAY_NAME プロシージャ	11-6
SET_HIERARCHY_NAME プロシージャ	11-7
SET_SHORT_DESCRIPTION プロシージャ	11-7
SET_SOLVED_CODE プロシージャ	11-8
12 CWM2_OLAP_LEVEL	
レベルの理解	12-1
例: レベルの作成	12-2
CWM2_OLAP_LEVEL サブプログラムの概要	12-3
ADD_LEVEL_TO_HIERARCHY プロシージャ	12-3
CREATE_LEVEL プロシージャ	12-4
DROP_LEVEL プロシージャ	12-5
LOCK_LEVEL プロシージャ	12-5
REMOVE_LEVEL_FROM_HIERARCHY プロシージャ	12-6
SET_DESCRIPTION プロシージャ	12-7
SET_DISPLAY_NAME プロシージャ	12-7
SET_LEVEL_NAME プロシージャ	12-8
SET_PLURAL_NAME プロシージャ	12-8
SET_SHORT_DESCRIPTION プロシージャ	12-9
13 CWM2_OLAP_LEVEL_ATTRIBUTE	
レベル属性の理解	13-1
例: レベル属性の作成	13-2
CWM2_OLAP_LEVEL_ATTRIBUTE サブプログラムの概要	13-4
CREATE_LEVEL_ATTRIBUTE プロシージャ	13-4
DROP_LEVEL_ATTRIBUTE プロシージャ	13-6

LOCK_LEVEL_ATTRIBUTE プロシージャ	13-6
SET_DESCRIPTION プロシージャ.....	13-7
SET_DISPLAY_NAME プロシージャ	13-8
SET_LEVEL_ATTRIBUTE_NAME プロシージャ	13-8
SET_SHORT_DESCRIPTION プロシージャ.....	13-10

14 CWM2_OLAP_MEASURE

メジャーの理解	14-1
例:メジャーの作成.....	14-2
CWM2_OLAP_MEASURE サブプログラムの概要	14-3
CREATE_MEASURE プロシージャ.....	14-3
DROP_MEASURE プロシージャ	14-4
LOCK_MEASURE プロシージャ	14-4
SET_DESCRIPTION プロシージャ.....	14-5
SET_DISPLAY_NAME プロシージャ	14-6
SET_MEASURE_NAME プロシージャ	14-6
SET_SHORT_DESCRIPTION プロシージャ.....	14-7

15 CWM2_OLAP_METADATA_REFRESH

キャッシュされた OLAP カタログ・メタデータのビュー.....	15-1
キャッシュされたアクティブ・カタログ・メタデータのビュー	15-2
CWM2_OLAP_METADATA_REFRESH サブプログラムの概要	15-3
MR_REFRESH プロシージャ	15-3
MR_AC_REFRESH プロシージャ	15-3

16 CWM2_OLAP_PC_TRANSFORM

前提条件	16-1
親子ディメンション	16-2
解決済のレベルベース・ディメンション	16-2
例:解決済のレベルベース・ディメンション表の作成.....	16-3
グルーピング ID 列.....	16-4
埋込み合計キー列	16-4
CWM2_OLAP_PC_TRANSFORM サブプログラムの概要	16-6
CREATE_SCRIPT プロシージャ.....	16-6

17 CWM2_OLAP_TABLE_MAP

OLAP メタデータのマッピングの理解	17-1
例: デイメンションのマッピング	17-2
例: キューブのマッピング	17-2
CWM2_OLAP_TABLE_MAP サブプログラムの概要	17-4
MAP_DIMTBL_HIERLEVELATTR プロシージャ	17-5
MAP_DIMTBL_HIERLEVEL プロシージャ	17-5
MAP_DIMTBL_HIERSORTKEY プロシージャ	17-6
MAP_DIMTBL_LEVELATTR プロシージャ	17-7
MAP_DIMTBL_LEVEL プロシージャ	17-8
MAP_FACTTBL_LEVELKEY プロシージャ	17-9
MAP_FACTTBL_MEASURE プロシージャ	17-11
REMOVEMAP_DIMTBL_HIERLEVELATTR プロシージャ	17-12
REMOVEMAP_DIMTBL_HIERLEVEL プロシージャ	17-13
REMOVEMAP_DIMTBL_HIERSORTKEY プロシージャ	17-14
REMOVEMAP_DIMTBL_LEVELATTR プロシージャ	17-14
REMOVEMAP_DIMTBL_LEVEL プロシージャ	17-15
REMOVEMAP_FACTTBL_LEVELKEY プロシージャ	17-15
REMOVEMAP_FACTTBL_MEASURE プロシージャ	17-16

18 CWM2_OLAP_VALIDATE

OLAP カタログ・メタデータの検証について	18-1
構造の検証	18-1
キューブ	18-2
デイメンション	18-2
マッピングの検証	18-2
キューブ	18-2
デイメンション	18-2
検証タイプ	18-3
CWM2_OLAP_VALIDATE サブプログラムの概要	18-4
VALIDATE_ALL_CUBES プロシージャ	18-4
VALIDATE_ALL_DIMENSIONS プロシージャ	18-5
VALIDATE_CUBE プロシージャ	18-5
VALIDATE_DIMENSION プロシージャ	18-6
VALIDATE_OLAP_CATALOG プロシージャ	18-7

19 CWM2_OLAP_VERIFY_ACCESS

OLAP キューブのアクセス可能性の検証	19-1
CWM2_OLAP_VERIFY_ACCESS サブプログラムの概要	19-3
VERIFY_CUBE_ACCESS プロシージャ	19-3

20 DBMS_AW

OLAP DML の SQL 文への埋込み	20-1
OLAP DML コマンドの実行方法	20-2
OLAP DML コマンドで引用符を使用する場合のガイドライン	20-2
カスタム・メジャーの SELECT 文への埋込み	20-2
集計アドバイザの使用	20-5
ワークスペース内での集計機能	20-6
例: ADVISE_REL プロシージャの使用	20-6
DBMS_AW サブプログラムの概要	20-10
ADVISE_CUBE プロシージャ	20-11
ADVISE_REL プロシージャ	20-12
EXECUTE プロシージャ	20-13
GETLOG ファンクション	20-14
INTERP ファンクション	20-15
INTERPCLOB ファンクション	20-16
INTERP_SILENT プロシージャ	20-17
OLAP_EXPRESSION ファンクション	20-18
OLAP_EXPRESSION_BOOL ファンクション	20-19
OLAP_EXPRESSION_DATE ファンクション	20-20
OLAP_EXPRESSION_TEXT ファンクション	20-21
PRINTLOG プロシージャ	20-22

21 DBMS_AW_UTILITIES

カスタム・メジャーについて	21-1
カスタム・メジャーに対する問合せ	21-2
CWM2\$_AW_PERM_CUST_MEAS_MAP	21-2
CWM2\$_AW_TEMP_CUST_MEAS_MAP	21-3
例: カスタム・メジャーの作成	21-3
DBMS_AW_UTILITIES サブプログラムの概要	21-6
CREATE_CUSTOM_MEASURE プロシージャ	21-6
DELETE_CUSTOM_MEASURE プロシージャ	21-8

UPDATE_CUSTOM_MEASURE プロシージャ.....	21-8
-----------------------------------	------

22 DBMS_AWM

DBMS_AWM サブプログラムのパラメータ	22-1
DBMS_AWM サブプログラムの概要	22-3
ADD_AWCOMP_SPEC_COMP_MEMBER プロシージャ.....	22-6
ADD_AWCOMP_SPEC_MEMBER プロシージャ.....	22-8
ADD_AWCUBEAGG_SPEC_LEVEL プロシージャ	22-9
ADD_AWCUBEAGG_SPEC_MEASURE プロシージャ.....	22-10
ADD_AWCUBELOAD_SPEC_COMP プロシージャ	22-11
ADD_AWCUBELOAD_SPEC_FILTER プロシージャ	22-12
ADD_AWCUBELOAD_SPEC_MEASURE プロシージャ	22-13
ADD_AWDIMLOAD_SPEC_FILTER プロシージャ.....	22-15
AGGREGATE_AWCUBE プロシージャ	22-16
CREATE_AWCOMP_SPEC プロシージャ	22-17
CREATE_AWCUBE プロシージャ.....	22-19
CREATE_AWCUBE_ACCESS プロシージャ	22-21
CREATE_AWCUBE_ACCESS_FULL プロシージャ.....	22-23
CREATE_AWCUBEAGG_SPEC プロシージャ	22-24
CREATE_AWCUBELOAD_SPEC プロシージャ	22-26
CREATE_AWDIMENSION プロシージャ	22-28
CREATE_AWDIMENSION_ACCESS プロシージャ	22-30
CREATE_AWDIMENSION_ACCESS_FULL プロシージャ	22-32
CREATE_AWDIMLOAD_SPEC プロシージャ	22-33
DELETE_AWCOMP_SPEC プロシージャ	22-35
DELETE_AWCOMP_SPEC_MEMBER プロシージャ.....	22-35
DELETE_AWCUBE_ACCESS プロシージャ	22-36
DELETE_AWCUBEAGG_SPEC プロシージャ	22-37
DELETE_AWCUBEAGG_SPEC_LEVEL プロシージャ	22-38
DELETE_AWCUBEAGG_SPEC_MEASURE プロシージャ.....	22-39
DELETE_AWCUBELOAD_SPEC プロシージャ	22-40
DELETE_AWCUBELOAD_SPEC_COMP プロシージャ	22-40
DELETE_AWCUBELOAD_SPEC_FILTER プロシージャ	22-41
DELETE_AWCUBELOAD_SPEC_MEASURE プロシージャ	22-42
DELETE_AWDIMENSION_ACCESS プロシージャ.....	22-42
DELETE_AWDIMLOAD_SPEC プロシージャ	22-44
DELETE_AWDIMLOAD_SPEC_FILTER プロシージャ.....	22-44
REFRESH_AWCUBE プロシージャ.....	22-45

REFRESH_AWDIMENSION プロシージャ	22-47
SET_AWCOMP_SPEC_CUBE プロシージャ	22-49
SET_AWCOMP_SPEC_MEMBER_NAME プロシージャ	22-50
SET_AWCOMP_SPEC_MEMBER_POS プロシージャ	22-51
SET_AWCOMP_SPEC_MEMBER_SEG プロシージャ	22-52
SET_AWCOMP_SPEC_NAME プロシージャ	22-54
SET_AWCUBE_VIEW_NAME プロシージャ	22-55
SET_AWCUBEAGG_SPEC_AGGOP プロシージャ	22-56
SET_AWCUBELOAD_SPEC_CUBE プロシージャ	22-57
SET_AWCUBELOAD_SPEC_LOADTYPE プロシージャ	22-58
SET_AWCUBELOAD_SPEC_NAME プロシージャ	22-58
SET_AWCUBELOAD_SPEC_PARAMETER プロシージャ	22-59
SET_AWDIMENSION_VIEW_NAME プロシージャ	22-60
SET_AWDIMLOAD_SPEC_DIMENSION プロシージャ	22-61
SET_AWDIMLOAD_SPEC_LOADTYPE プロシージャ	22-62
SET_AWDIMLOAD_SPEC_NAME プロシージャ	22-63
SET_AWDIMLOAD_SPEC_PARAMETER プロシージャ	22-64

23 DBMS_ODM

マテリアライズド・ビューでのサマリー管理	23-1
グルーピング・セット	23-2
ファクト表の集計	23-2
手順: マテリアライズド・ビューの自動生成	23-3
手順: マテリアライズド・ビューの手動生成	23-4
例: Sales キューブのマテリアライズド・ビューの作成	23-4
DBMS_ODM サブプログラムの概要	23-8
CREATECUBELEVELTUPLE プロシージャ	23-8
CREATEDIMLEVTUPLE プロシージャ	23-9
CREATEDIMMV_GS プロシージャ	23-10
CREATEFACTMV_GS プロシージャ	23-11
CREATESTDFACTMV プロシージャ	23-12

24 OLAP_API_SESSION_INIT

OLAP API の初期化パラメータ	24-1
構成表の表示	24-2
ALL_OLAP_ALTER_SESSION ビュー	24-2
OLAP_API_SESSION_INIT サブプログラムの概要	24-3

ADD_ALTER_SESSION プロシージャ	24-3
CLEAN_ALTER_SESSION プロシージャ	24-4
DELETE_ALTER_SESSION プロシージャ	24-4

25 OLAP_TABLE

OLAP_TABLE の使用	25-1
例: ビューの作成	25-2
例: 埋込み合計ディメンションのビューの作成	25-2
例: 埋込み合計メジャーのビューの作成	25-4
例: ロールアップ形式のビューの作成	25-5
例: OLAP_TABLE での FETCH コマンドの使用	25-7
OLAP_TABLE の構文	25-9
構文	25-9
パラメータ	25-9
注意: OLAP_TABLE における処理の順序	25-18

索引

例目次

1-1	アナリティック・ワークスペースにおける CHANNEL ディメンションの作成	1-11
1-2	アナリティック・ワークスペースにおける ANALYTIC_CUBE キューブの作成	1-13
1-3	アナリティック・ワークスペースにおけるキューブのディメンションの定義	1-17
1-4	アナリティック・ワークスペースにおける Costs および Quota の事前集計	1-19
1-5	キューブの作成、リフレッシュおよび有効化	1-30
2-1	PRODUCTS 表に OLAP ディメンションを作成	2-4
2-2	OLAP 時間ディメンションの作成	2-7
2-3	COSTS ファクト表に OLAP キューブを作成	2-9
2-4	PRODUCT ディメンションの検証レポートの生成	2-13
3-1	ワークスペース・キューブの情報に関するアクティブ・カタログの問合せ	3-3
4-1	CHANNEL と TIME のロード・パラメータおよび有効化ビュー名の問合せ	4-2
20-1	OLAP_EXPRESSION: WHERE 句が定義された時系列ファンクション	20-4
20-2	OLAP_EXPRESSION: ORDER BY 句が定義された数値計算	20-4
20-3	ADVISE_REL: Customer ディメンションについて提示された事前集計	20-8
25-1	OLAP_TABLE を使用したビュー作成用テンプレート	25-2
25-2	OLAP_TABLE を使用したディメンション・ビュー作成用スクリプト	25-3
25-3	OLAP_TABLE を使用したメジャー・ビュー作成用スクリプト	25-4
25-4	OLAP_TABLE を使用した製品のロールアップ・ビュー作成用スクリプト	25-6
25-5	OLAP_TABLE の FAMILYREL 句で QDR を使用するスクリプト	25-6
25-6	OLAP_TABLE で FETCH を使用するスクリプト	25-8
25-7	OLAP_TABLE の LIMIT_MAP パラメータの構文	25-12

はじめに

このマニュアルでは、Oracle データベース・サーバーの OLAP オプションに同梱されている Oracle PL/SQL パッケージについて説明します。

対象読者

このマニュアルは、次の作業を行うデータベース管理者およびアプリケーション開発者を対象としています。

- データベースの管理
- アナリティック・ワークスペースの管理
- データ・ウェアハウスおよびデータ・マートの構築およびメンテナンス
- メタデータの定義
- 分析アプリケーションの開発

このマニュアルを読む場合、Oracle OLAP に関する前提知識は必要ありません。

このマニュアルの構成

このマニュアルは、次の章で構成されています。

第1章「DBMS_AWM を使用したアナリティック・ワークスペースの作成」

この章では、DBMS_AWM パッケージの使用方法について説明します。

第2章「CWM2 による OLAP カタログ・メタデータの作成」

この章では、CWM2 パッケージの使用方法について説明します。

第3章「アクティブ・カタログ・ビュー」

この章では、アクティブ・カタログのビューについて説明します。

第4章「アナリティック・ワークスペース・メンテナンス・ビュー」

この章では、アナリティック・ワークスペースのメンテナンス情報のビューについて説明します。

第5章「OLAP カタログ・メタデータのビュー」

この章では、OLAP カタログ・メタデータのビューについて説明します。

第6章「OLAP 動的パフォーマンス・ビュー」

この章では、Oracle OLAP の動的パフォーマンス・ビューについて説明します。

第7章「CWM2_OLAP_CATALOG」

この章では、CWM2_OLAP_CATALOG パッケージのプロシージャの構文について説明します。

第8章「CWM2_OLAP_CUBE」

この章では、CWM2_OLAP_CUBE パッケージのプロシージャの構文について説明します。

第9章「CWM2_OLAP_DIMENSION」

この章では、CWM2_OLAP_DIMENSION パッケージのプロシージャの構文について説明します。

第10章「CWM2_OLAP_DIMENSION_ATTRIBUTE」

この章では、CWM2_OLAP_DIMENSION_ATTRIBUTE パッケージのプロシージャの構文について説明します。

第11章「CWM2_OLAP_HIERARCHY」

この章では、CWM2_OLAP_HIERARCHY パッケージのプロシージャの構文について説明します。

第12章 「CWM2_OLAP_LEVEL」

この章では、CWM2_OLAP_LEVEL パッケージのプロシージャの構文について説明します。

第13章 「CWM2_OLAP_LEVEL_ATTRIBUTE」

この章では、CWM2_OLAP_LEVEL_ATTRIBUTE パッケージのプロシージャの構文について説明します。

第14章 「CWM2_OLAP_MEASURE」

この章では、CWM2_OLAP_MEASURE パッケージのプロシージャの構文について説明します。

第15章 「CWM2_OLAP_METADATA_REFRESH」

この章では、CWM2_OLAP_METADATA_REFRESH パッケージのプロシージャの構文について説明します。

第16章 「CWM2_OLAP_PC_TRANSFORM」

この章では、CWM2_OLAP_PC_TRANSFORM パッケージのプロシージャの構文について説明します。

第17章 「CWM2_OLAP_TABLE_MAP」

この章では、CWM2_OLAP_TABLE_MAP パッケージのプロシージャの構文について説明します。

第18章 「CWM2_OLAP_VALIDATE」

この章では、CWM2_OLAP_VALIDATE パッケージのプロシージャの構文について説明します。

第19章 「CWM2_OLAP_VERIFY_ACCESS」

この章では、CWM2_OLAP_VERIFY_ACCESS パッケージのプロシージャの構文について説明します。

第20章 「DBMS_AW」

この章では、DBMS_AW パッケージのプロシージャの構文について説明します。

第21章 「DBMS_AW_UTILITIES」

この章では、DBMS_AW_UTILITIES パッケージのプロシージャの構文について説明します。

第22章 「DBMS_AWM」

この章では、DBMS_AWM パッケージのプロシージャの構文について説明します。

第 23 章 「DBMS_ODM」

この章では、DBMS_ODM パッケージのプロシージャの構文について説明します。

第 24 章 「OLAP_API_SESSION_INIT」

この章では、OLAP_API_SESSION_INIT パッケージのプロシージャの構文について説明します。

第 25 章 「OLAP_TABLE」

この章では、OLAP_TABLE ファンクションの構文について説明します。

関連文書

詳細は、次の Oracle リソースを参照してください。

- 『Oracle OLAP アプリケーション開発者ガイド』
- 『Oracle9i OLAP 開発者ガイド - Oracle OLAP API』
- 『Oracle9i データ・ウェアハウス・ガイド』
- 『PL/SQL ユーザーズ・ガイドおよびリファレンス』

表記規則

このマニュアルで使用される表記規則は、次のとおりです。

規則	意味
.	例における垂直の省略記号は、例に直接関連しない情報が省略されていることを示します。
...	文またはコマンドにおける水平の省略記号は、例に直接関連しない文またはコマンドの一部が省略されていることを示します。
太字	本文中の太字は、本文中または用語集（あるいはその両方）で定義されている用語を示します。
<>	山カッコは、ユーザーが指定する名前を囲んでいます。
[]	大カッコは、任意に選択できるオプション句を囲んでいます。
\$	ドル記号は、Windows ではコマンドプロンプトを、Digital UNIX では Bourne シェル・プロンプトを示します。

DBMS_AWM を使用したアナリティック・ワークスペースの作成

DBMS_AWM パッケージでは、スター・スキーマからアナリティック・ワークスペース・キューブを作成し、OLAP API によってアクセスできるようにするためのストアド・プロシージャが提供されます。DBMS_AWM パッケージは、Analytic Workspace Manager によって使用されます。この章では、DBMS_AWM プロシージャを直接使用して処理する方法を説明します。

参照：

- [第 22 章「DBMS_AWM」](#) を参照してください。
- [第 3 章「アクティブ・カタログ・ビュー」](#) を参照してください。
- [第 4 章「アナリティック・ワークスペース・メンテナンス・ビュー」](#) を参照してください。

この章では、次の項目について説明します。

- [概要](#)
- [DBMS_AWM プロシージャの理解](#)
- [ワークスペース・ディメンションの作成およびリフレッシュ](#)
- [ワークスペース・キューブの作成およびリフレッシュ](#)
- [スパースなデータの管理およびワークスペース・キューブの最適化](#)
- [アナリティック・ワークスペースにおけるデータの集計](#)
- [ワークスペース・キューブへのリレーショナル・アクセスの作成](#)

概要

スター・スキーマまたはスノーフレイク・スキーマにデータが格納されている場合、DBMS_AWM パッケージを使用してアナリティック・ワークスペースへのデータのロード・プロセスを簡略化できます。

最初の手順は、スキーマの機能を多次元用語で（つまり、ディメンション、属性およびメジャーを持つキューブとして）記述する OLAP カタログ・メタデータを作成することです。次に、DBMS_AWM パッケージを使用して、アナリティック・ワークスペース内のこれらのオブジェクトのインスタンス化、ワークスペース・オブジェクトのリレーショナル・ビューの作成、およびワークスペース・ビューにマッピングされる OLAP カタログ・メタデータの 2 番目のセットの生成（オプション）を行うことができます。

注意： DBMS_AWM プロシージャによって作成されるアナリティック・ワークスペースは**データベース・スタンダード・フォーム**であるため、関連する Oracle OLAP のツールおよびユーティリティとの互換性は保証されます。スタンダード・フォームの詳細は、『**Oracle OLAP アプリケーション開発者ガイド**』を参照してください。

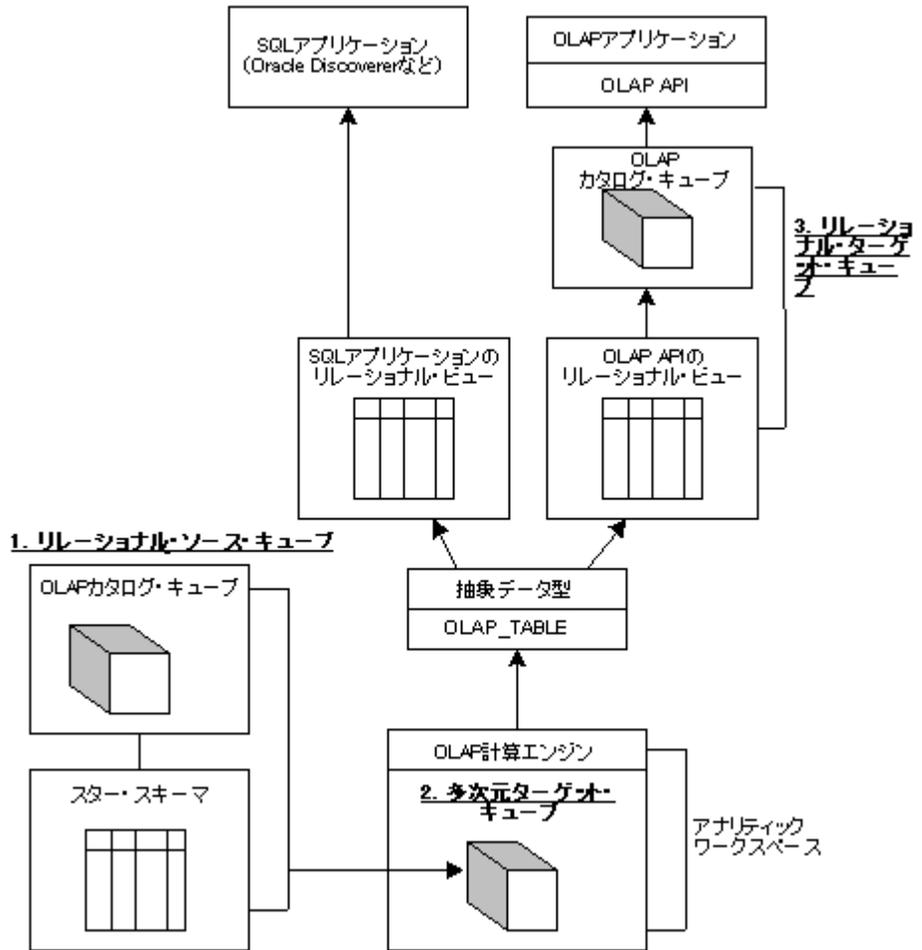
DBMS_AWM パッケージでは、アナリティック・ワークスペースの管理に使用できる豊富な機能の API セットが提供されます。これらの API を有効に使用するには、リレーショナル・ソースから多次元ターゲットへデータを移動するために各 API がどのように連動するのか、および API がそのターゲットへのリレーショナル・アクセスをどのように確立するのかを理解する必要があります。

イベントの基本的なフローでは、次の 3 つの異なる論理キューブを作成します。

1. **リレーショナル・ソース・キューブ。** このキューブは、任意の DBMS_AWM プロシージャをコールする前に存在する必要があります。キューブのメタデータは OLAP カタログ内で定義されます。データは未解決（最低レベルのデータのみ）で、スター・スキーマに格納されます。
2. **多次元ターゲット・キューブ。** DBMS_AWM プロシージャは、リレーショナル・ソース・キューブからこのキューブを定義および移入します。キューブのスタンダード・フォーム・メタデータは、アナリティック・ワークスペース内で定義されます。そのデータはワークスペースに格納され、通常完全にまたは部分的に集計されています。
3. **リレーショナル・ターゲット・キューブ。** DBMS_AWM プロシージャは、多次元ターゲット・キューブからこのキューブを定義します。キューブのメタデータは OLAP カタログ内で定義されます。データはアナリティック・ワークスペースに格納され、リレーショナル・ビューを介してアクセスされます。ビューは完全に解決した状態のデータを表示します（すべてのレベルの組合せに対する埋込み合計を含む）。

DBMS_AWM パッケージを使用したアナリティック・ワークスペースの作成および有効化の基本プロセスを [図 1-1](#) に示します。

図 1-1 DBMS_AWM を使用したアナリティック・ワークスペースの作成および有効化



ソース・キューブの OLAP カタログ・メタデータの作成

DBMS_AWM プロシージャを使用する前に、OLAP カタログ内にキューブを作成し、そのキューブをソースのファクト表およびディメンション表にマップする必要があります。ソース表は基本的なスター・スキーマまたはスノーフレーク・スキーマ内で構成される必要があります。

Enterprise Manager を使用するか、または PL/SQL パッケージ CWM2 を使用してスクリプトを作成します (第 2 章を参照)。また、Oracle Warehouse Builder を使用すると、OLAP カタログ・メタデータを作成できます。

このキューブは図 1-1 に示されている **リレーショナル・ソース・キューブ** です。

ワークスペース・ディメンションの作成および移入

OLAP カタログに定義されたキューブの各ディメンションに対して、次の一般的な作業を行うには、DBMS_AWM パッケージ内のプロシージャのセットを実行する必要があります。

1. **ディメンション・ロード仕様** (アナリティック・ワークスペースにおけるディメンションの移入手順を含む) の作成。ロード仕様には、ソース・ディメンション表からデータを選択するための基準を識別するフィルタが含まれる場合があります。
2. アナリティック・ワークスペースにおけるディメンションのコンテナの作成。
3. ディメンション・ロード仕様を使用した、ソース・ディメンション表からのアナリティック・ワークスペースにおけるディメンションの移入。

参照: 1-11 ページの「[ワークスペース・ディメンションの作成およびリフレッシュ](#)」を参照してください。

ワークスペース・キューブの作成および移入

キューブのディメンションを作成した後は、別のプロシージャのセットを実行し、キューブ自身を作成および移入します。

1. **キューブ・ロード仕様** (アナリティック・ワークスペースにおけるキューブのメジャーの移入手順を含む) の作成。ロード仕様には、ソース・ファクト表からデータを選択するための基準を識別するフィルタが含まれる場合があります。
2. **コンポジット仕様** (アナリティック・ワークスペースにおけるキューブのディメンションの順序付けおよびスパースなデータの格納手順を含む) の作成。
3. キューブ・ロード仕様へのコンポジット仕様の追加。
4. アナリティック・ワークスペースにおけるキューブのコンテナの作成。
5. キューブ・ロード仕様を使用した、ソース・ファクト表からのアナリティック・ワークスペースにおけるキューブのメジャーの移入。

このキューブは図 1-1 に示されている **多次元ターゲット・キューブ** です。

参照: 1-13 ページの「[ワークスペース・キューブの作成およびリフレッシュ](#)」、および 1-16 ページの「[スパースなデータの管理およびワークスペース・キューブの最適化](#)」を参照してください。

アナリティック・ワークスペースにおけるキューブのデータの集計

ワークスペース・キューブに対しては、プロシージャのセットを実行して次の作業を行います。

1. **集計仕様**（アナリティック・ワークスペースにおけるサマリー・データの格納手順を含む）の作成。
2. 集計仕様を使用したワークスペース・キューブの集計。

参照： 1-18 ページの「[アナリティック・ワークスペースにおけるデータの集計](#)」を参照してください。

ワークスペース・キューブへのリレーショナル・アクセスの有効化

アナリティック・ワークスペースにおいてキューブを作成、移入および集計した後は、別のプロシージャのセットを実行して、リレーショナル・アクセスを有効化します。**有効化プロセス**は、有効化スクリプト・セットの生成および実行が主体となります。これらのスクリプトは、OLAP_TABLE ファンクションを使用してワークスペース・キューブへアクセスするリレーショナル・ビューを作成します。スクリプトで、ビューにマップされる OLAP カタログ・キューブを作成する場合があります。

有効化スクリプトで作成したキューブは、[図 1-1](#) に示されているリレーショナル・ターゲット・キューブです。

ワークスペース・キューブを有効化するには、スクリプトを生成し実行するか、スクリプトを自動的に作成および実行する 1 段階のプロシージャを使用できます。

参照： 1-22 ページの「[ワークスペース・キューブへのリレーショナル・アクセスの作成](#)」を参照してください。

DBMS_AWM によって作成したメタデータの表示

2 つのビューのセットは、アナリティック・ワークスペースに関連付けられたメタデータを表示します。**アクティブ・カタログ・ビュー**は、アナリティック・ワークスペース内に格納されたメタデータを表示します。**アナリティック・ワークスペース・メンテナンス・ビュー**は、OLAP カタログ内に格納されたメタデータを表示します。

アクティブ・カタログ・ビュー

これらのビューは OLAP_TABLE ファンクションを使用して、アナリティック・ワークスペース内の論理スタンダード・フォーム・オブジェクトに関する情報を戻します。たとえば、アクティブ・カタログ・ビューを問い合わせ、ワークスペース・キューブのディメンションに関する情報を取得できます。アクティブ・カタログ・ビューの名前には、接頭辞 ALL_OLAP2_AW が付きます。詳細は、[第 3 章](#)を参照してください。

アナリティック・ワークスペース・メンテナンス・ビュー

これらのビューは、アナリティック・ワークスペース・キューブの構築およびメンテナンスに関する情報を戻します。たとえば、アナリティック・ワークスペース・メンテナンス・ビューを問い合わせ、アナリティック・ワークスペースのディメンションまたはキューブに関連付けられたロード仕様に関する情報を取得できます。アナリティック・ワークスペース・メンテナンス・ビューの名前には、接頭辞 ALL_AW が付きます。詳細は、[第4章](#)を参照してください。

DBMS_AWM プロシージャの理解

DBMS_AWM パッケージ内のプロシージャは、様々な種類の論理エンティティに関するメソッドをサポートしています。これらのエンティティについては、[表 1-1](#)を参照してください。

参照： [第22章「DBMS_AWM」](#)を参照してください。

表 1-1 DBMS_AWM パッケージ内の論理エンティティ

エンティティ	説明
ディメンション	OLAP カタログ内のディメンションおよびアナリティック・ワークスペース内の対応ディメンション。
キューブ	OLAP カタログ内のキューブおよびアナリティック・ワークスペース内の対応キューブ。
ディメンション・ロード仕様	OLAP カタログ・ディメンションのディメンション表からのアナリティック・ワークスペース・ディメンションの移入手順。
キューブ・ロード仕様	OLAP カタログ・キューブのファクト表からのアナリティック・ワークスペース・キューブの移入手順。
キューブ集計仕様	アナリティック・ワークスペースにおけるサマリー・データの作成手順。
キューブ・コンポジット仕様	アナリティック・ワークスペースにおけるディメンションの順序付けおよびスパースなデータの格納手順。

ディメンションに関するメソッド

ディメンションに関して実行可能なメソッドを表 1-2 に示します。

表 1-2 DBMS_AWM のディメンションに関するメソッド

メソッド	説明	プロシージャ
作成	OLAP カタログ内に定義されたディメンションのコンテナをアナリティック・ワークスペースに作成します。	CREATE_AWDIMENSION プロシージャ
リフレッシュ	ディメンション・ロード仕様を使用して、OLAP カタログ・ディメンションのディメンション表からアナリティック・ワークスペース・ディメンションを移入します。	REFRESH_AWDIMENSION プロシージャ
アクセスの作成	アナリティック・ワークスペース内のディメンションへのリレーショナル・アクセスを有効化するスクリプトを作成します。	CREATE_AWCUBELOAD_SPEC プロシージャ
アクセスの削除	アナリティック・ワークスペース内のディメンションへのリレーショナル・アクセスを無効化するスクリプトを作成します。	DELETE_AWDIMENSION_ACCESS プロシージャ
ビュー名の設定	アナリティック・ワークスペース内のディメンションのリレーショナル・ビューに新しい名前を指定します。	SET_AWDIMENSION_VIEW_NAME プロシージャ

キューブに関するメソッド

キューブに関して実行可能なメソッドを表 1-3 に示します。

表 1-3 DBMS_AWM のキューブに関するメソッド

メソッド	説明	プロシージャ
作成	OLAP カタログで定義されたキューブのコンテナをアナリティック・ワークスペースに作成します。	CREATE_AWCUBE プロシージャ
リフレッシュ	キューブ・ロード仕様を使用して、OLAP カタログ・キューブのファクト表からアナリティック・ワークスペース・キューブのメジャーを移入します。	REFRESH_AWCUBE プロシージャ
集計	集計仕様を使用して、アナリティック・ワークスペース内のキューブを集計します。	AGGREGATE_AWCUBE プロシージャ

表 1-3 DBMS_AWM のキューブに関するメソッド (続き)

メソッド	説明	プロシージャ
アクセスの作成	アナリティック・ワークスペース内のキューブへのリレーショナル・アクセスを有効化するスクリプトを作成します。	CREATE_AWCUBE_ACCESS プロシージャ
アクセスの削除	アナリティック・ワークスペース内のキューブへのリレーショナル・アクセスを無効化するスクリプトを作成します。	DELETE_AWCUBE_ACCESS プロシージャ
ビュー名の設定	アナリティック・ワークスペース内のキューブ・データのリレーショナル・ビューに新しい名前を指定します。	SET_AWCUBE_VIEW_NAME プロシージャ

ディメンション・ロード仕様に関するメソッド

ディメンション・ロード仕様に関して実行可能なメソッドを表 1-4 に示します。

表 1-4 DBMS_AWM のディメンション・ロード仕様に関するメソッド

メソッド	説明	プロシージャ
作成 / 削除	ディメンション・ロード仕様を作成または削除します。	CREATE_AWDIMLOAD_SPEC プロシージャ DELETE_AWDIMLOAD_SPEC プロシージャ
情報の再設定	ディメンション・ロード仕様の様々なコンポーネントを変更します。	SET_AWDIMLOAD_SPEC_DIMENSION プロシージャ SET_AWDIMLOAD_SPEC_LOADTYPE プロシージャ SET_AWDIMLOAD_SPEC_NAME プロシージャ SET_AWDIMLOAD_SPEC_PARAMETER プロシージャ
フィルタの追加 / 削除	ディメンション・ロード仕様のフィルタを追加または削除します。	ADD_AWDIMLOAD_SPEC_FILTER プロシージャ DELETE_AWDIMLOAD_SPEC_FILTER プロシージャ

キューブ・ロード仕様に関するメソッド

キューブ・ロード仕様に関して実行可能なメソッドを表 1-5 に示します。

表 1-5 DBMS_AWM のキューブ・ロード仕様に関するメソッド

メソッド	説明	プロシージャ
作成 / 削除	キューブ・ロード仕様を作成または削除します。	CREATE_AWCUBELOAD_SPEC プロシージャ DELETE_AWCUBELOAD_SPEC プロシージャ
情報の再設定	キューブ・ロード仕様の様々なコンポーネントを変更します。	SET_AWCUBELOAD_SPEC_CUBE プロシージャ SET_AWCUBELOAD_SPEC_LOADTYPE プロシージャ SET_AWCUBELOAD_SPEC_NAME プロシージャ SET_AWCUBELOAD_SPEC_PARAMETER プロシージャ
フィルタの追加 / 削除	キューブ・ロード仕様のフィルタを追加または削除します。	ADD_AWCUBELOAD_SPEC_FILTER プロシージャ DELETE_AWCUBELOAD_SPEC_FILTER プロシージャ
コンポジット使用の追加 / 削除	キューブ・ロード仕様のコンポジット仕様を追加または削除します。	ADD_AWCUBELOAD_SPEC_COMP プロシージャ DELETE_AWCUBELOAD_SPEC_COMP プロシージャ

集計仕様に関するメソッド

集計仕様に関して実行可能なメソッドを表 1-6 に示します。

表 1-6 DBMS_AWM の集計仕様に関するメソッド

メソッド	説明	プロシージャ
作成 / 削除	集計仕様を作成または削除します。	CREATE_AWCUBEAGG_SPEC プロシージャ DELETE_AWCUBEAGG_SPEC_MEASURE プロシージャ
演算子の設定	ディメンションに集計演算子を設定します。	SET_AWCUBEAGG_SPEC_AGGOP プロシージャ

表 1-6 DBMS_AWM の集計仕様に関するメソッド (続き)

メソッド	説明	プロシージャ
レベルの追加 / 削除	集計仕様のレベルを追加または削除します。	ADD_AWCUBEAGG_SPEC_LEVEL プロシージャ DELETE_AWCUBEAGG_SPEC_LEVEL プロシージャ
メジャーの追加 / 削除	集計仕様のメジャーを追加または削除します。	ADD_AWCUBEAGG_SPEC_MEASURE プロシージャ DELETE_AWCUBEAGG_SPEC_MEASURE プロシージャ

コンポジット仕様に関するメソッド

コンポジット仕様に関して実行可能なメソッドを表 1-7 に示します。

表 1-7 DBMS_AWM のコンポジット仕様に関するメソッド

メソッド	説明	プロシージャ
作成 / 削除	コンポジット仕様を作成または削除します。	CREATE_AWCOMP_SPEC プロシージャ DELETE_AWCOMP_SPEC プロシージャ
情報の再設定	コンポジット仕様の名前を変更するか、別のキューブに関連付けます。	SET_AWCOMP_SPEC_CUBE プロシージャ SET_AWCOMP_SPEC_NAME プロシージャ
メンバーの追加 / 削除	仕様のメンバーを追加または削除します。メンバーは、ディメンションまたはコンポジットです。	ADD_AWCOMP_SPEC_MEMBER プロシージャ DELETE_AWCOMP_SPEC_MEMBER プロシージャ
メンバー情報の再設定	仕様のメンバーに関する情報を変更します。	SET_AWCOMP_SPEC_MEMBER_NAME プロシージャ SET_AWCOMP_SPEC_MEMBER_POS プロシージャ SET_AWCOMP_SPEC_MEMBER_SEG プロシージャ
コンポジット・メンバーの追加	仕様のコンポジットにメンバーを追加します。	ADD_AWCOMP_SPEC_COMP_MEMBER プロシージャ

ワークスペース・ディメンションの作成およびリフレッシュ

ソース・ディメンション表に対して OLAP カタログ内でディメンションを定義したら、アナリティック・ワークスペースにディメンションを作成できます。

OLAP カタログ内の指定したディメンションから作成できるワークスペース・ディメンションは1つのみです。たとえば、OLAP カタログの PRODUCT ディメンションを、アナリティック・ワークスペースの PROD_AW ディメンションのソースとして使用した場合、同じワークスペース内の同じソース・ディメンションから別のディメンション PROD_AW2 を作成することはできません。

注意： CREATE_AWDIMENSION は、読み込み / 書き込みアクセスでアナリティック・ワークスペースを開きます。これによってワークスペースは更新されますが、SQL の COMMIT は実行されません。

CREATE_AWDIMENSION や DBMS_AWM パッケージのその他すべてのプロシージャをコールする時点で、アナリティック・ワークスペースが存在している必要があります。

例 1-1 に、XADEMO.CHANNEL ディメンションのワークスペース・オブジェクトを定義および移入するためのプロシージャ・コールを示します。ロード仕様には、'DIRECT' の行のみがロードされるようにするフィルタ条件が含まれます。

例 1-1 アナリティック・ワークスペースにおける CHANNEL ディメンションの作成

```

--- SET UP
set serveroutput on
execute cwm2_olap_manager.set_echo_on;
execute cwm2_olap_manager.begin_log
    ('/users/myxademo/myscripts' , 'channel.log');

--- CREATE THE ANALYTIC WORKSPACE
execute dbms_aw.execute ('aw create 'myaw');

--- CREATE AND POPULATE THE DIMENSION
execute dbms_awm.create_awdimension
    ('XADEMO', 'CHANNEL', 'MYSHEMA', 'MYAW', 'AW_CHAN');
execute dbms_awm.create_awdimload_spec
    ('CHAN_LOAD', 'XADEMO', 'CHANNEL', 'FULL_LOAD');
execute dbms_awm.add_awdimload_spec_filter
    ('CHAN_LOAD', 'XADEMO', 'CHANNEL', 'XADEMO',
    'XADEMO_CHANNEL', ''CHAN_STD_CHANNEL' = 'DIRECT'');
execute dbms_awm.refresh_awdimension
    ('MYSHEMA', 'MYAW', 'AW_CHAN', 'CHAN_LOAD');

```

```
--- COMMIT AND WRAP UP
commit;
execute cwm2_olap_manager.set_echo_off;
execute cwm2_olap_manager.end_log
```

アクティブ・カタログ・ビュー ALL_OLAP2_AW_DIMENSIONS を問い合わせると、次の行が表示されます。

AW_OWNER	AW_NAME	AW_LOGICAL_NAME	SOURCE_OWNER	SOURCE_NAME
MYSCHEMA	MYAW	AW_CHAN	XADEMO	CHANNEL

ディメンションのメタデータのリフレッシュ

CREATE_AWDIMENSION は、ディメンションをサポートする一般的なスタンダード・フォーム・オブジェクトがワークスペースに存在するようにし、ワークスペースの指定ディメンションを登録します。ただし、この特定のディメンションの論理構造を定義するメタデータは、REFRESH_AWDIMENSION をコールするまでワークスペース内でインスタンス化されません。

たとえば、ソース・ディメンション XADEMO.PRODUCT から XADEMO 内のワークスペース MYAW にディメンション AW_PROD を作成済であれば、アクティブ・カタログを問い合わせるワークスペースをチェックできます。

```
SQL>select * from ALL_OLAP2_AW_DIMENSIONS WHERE AW_LOGICAL_NAME in 'AW_PROD';
```

AW_OWNER	AW_NAME	AW_LOGICAL_NAME	SOURCE_OWNER	SOURCE_NAME
XADEMO	MYAW	AW_PROD	XADEMO	PRODUCT

次の問合せでは、ディメンションに関連付けられているレベルがないことが示されています。レベル、階層、属性および説明は、ディメンションのリフレッシュ時にインスタンス化されます。

```
SQL>select * from ALL_OLAP2_AW_DIM_LEVELS where AW_LOGICAL_NAME in 'AW_PROD';
```

```
no rows selected
```

ディメンションをリフレッシュするタイミング

ソース・ディメンション表に変更が生じるたびに、ディメンションをリフレッシュする必要があります。これらの変更は、製品ディメンションからのある製品の削除など、ディメンション・メンバーの追加や削除の場合や、時間ディメンションへの日レベルの追加など、ディメンションのメタデータへの変更の場合があります。

ディメンションをリフレッシュする場合は、ディメンションが含まれる各キューブもリフレッシュする必要があります。

ディメンションのリフレッシュ後に必要な処理

メタデータの変更のためにディメンションをリフレッシュする場合は、ディメンションおよびその関連キューブを再有効化する必要があります。データの変更が理由でディメンションをリフレッシュする場合は、再有効化する必要はありません。

キューブがアナリティック・ワークスペース内で（集計仕様の結果）ストアド・サマリーと関連しているディメンションをリフレッシュする場合、そのキューブも再集計する必要があります。

ワークスペース・キューブの作成およびリフレッシュ

スター・スキーマに対して OLAP カタログ内でキューブを定義したら、アナリティック・ワークスペースにキューブを作成できます。

CREATE_AWCUBE をコールしてキューブを作成する前に、CREATE_AWDIMENSION をコールしてキューブの各ディメンションを作成する必要があります。キューブを移入するには、REFRESH_AWDIMENSION をコールしてキューブの各ディメンションを移入してから、REFRESH_AWCUBE をコールしてキューブのメジャーをリフレッシュする必要があります。この後のリフレッシュでリフレッシュする必要があるのは変更されたディメンションのみです。

アナリティック・ワークスペース内では、ディメンションは複数のキューブによって共有できます。新しいワークスペース・キューブを作成する場合は、ワークスペースにすでに存在しているキューブのディメンションのソースとして使用されていない OLAP カタログ・ディメンションに対して CREATE_AWDIMENSION をコールするのみです。

注意： CREATE_AWCUBE は、読み込み / 書き込みアクセスでアナリティック・ワークスペースを開きます。これによってワークスペースは更新されますが、SQL の COMMIT は実行されません。

CREATE_AWCUBE や DBMS_AWM パッケージのその他すべてのプロシージャをコールする時点で、アナリティック・ワークスペースが存在している必要があります。

例 1-2 に、アナリティック・ワークスペースの XADEMO.ANALYTIC_CUBE キューブを作成および移入するプロシージャ・コールを示します。

例 1-2 アナリティック・ワークスペースにおける ANALYTIC_CUBE キューブの作成

```
--- SET UP
set serveroutput on
execute cwm2_olap_manager.set_echo_on;
execute cwm2_olap_manager.begin_log
        ('/users/myxademo/myscripts' , 'anacube.log');
```

```

--- CREATE THE ANALYTIC WORKSPACE
execute dbms_aw.execute ('aw create 'myaw');

--- CREATE AND REFRESH THE DIMENSIONS
execute dbms_awm.create_awdimension
('XADEMO','CHANNEL','MYSHEMA', 'MYAW', 'AW_CHAN');
execute dbms_awm.create_awdimension
('XADEMO','GEOGRAPHY','MYSHEMA','MYAW', 'AW_GEOG');
execute dbms_awm.create_awdimension
('XADEMO','PRODUCT','MYSHEMA', 'MYAW', 'AW_PROD');
execute dbms_awm.create_awdimension
('XADEMO','TIME','MYSHEMA', 'MYAW', 'AW_TIME');
execute dbms_awm.refresh_awdimension
('MYSHEMA', 'MYAW', 'AW_CHAN');
execute dbms_awm.refresh_awdimension
('MYSHEMA', 'MYAW', 'AW_PROD');
execute dbms_awm.refresh_awdimension
('MYSHEMA', 'MYAW', 'AW_GEOG');
execute dbms_awm.refresh_awdimension
('MYSHEMA', 'MYAW', 'AW_TIME');

--- CREATE AND REFRESH THE CUBE
execute dbms_awm.create_awcube
('XADEMO', 'ANALYTIC_CUBE','MYSHEMA', 'MYAW', 'AW_ANACUBE');
execute dbms_awm.create_awcubeload_spec
('AC_CUBELOADSPEC', 'XADEMO', 'ANALYTIC_CUBE', 'LOAD_DATA');
execute dbms_awm.refresh_awcube
('MYSHEMA', 'MYAW', 'AW_ANACUBE', 'AC_CUBELOADSPEC');

--- COMMIT AND WRAP UP
commit;
execute cwm2_olap_manager.set_echo_off;
execute cwm2_olap_manager.end_log

```

アクティブ・カタログ・ビュー ALL_OLAP2_AW_CUBES を問い合わせると、次の行が表示されます。

AW_OWNER	AW_NAME	AW_LOGICAL_NAME	SOURCE_OWNER	SOURCE_NAME
MYSHEMA	MYAW	AW_ANACUBE	XADEMO	ANALYTIC_CUBE

データ型の変換

ソース・ファクト表のメジャーには、数値、テキストまたは日付のデータ型を格納できません。REFRESH_AWCUBE はデータをワークスペース・キューブ内にロードする際、RDBMS

のデータ型をアナリティック・ワークスペース固有の型に変換します。データ型の変換を表 1-8 に示します。

ソース・メジャーを表 1-8 で示されていないデータ型がある場合、そのメジャーは REFRESH_AWCUBE によって無視され、そのデータやメタデータはアナリティック・ワークスペースにロードされません。

表 1-8 RDBMS データ型のワークスペース・データ型への変換

RDBMS のデータ型	アナリティック・ワークスペースのデータ型
NUMBER	DECIMAL
CHAR, LONG, VARCHAR, VARCHAR2	TEXT
NCHAR, NVARCHAR2	NTEXT
DATE	DATE

キューブのメタデータのリフレッシュ

CREATE_AWCUBE は、キューブをサポートする一般的なスタンダード・フォーム・オブジェクトがワークスペースに存在するようにし、ワークスペースの指定キューブを登録します。ただし、この特定のキューブの論理構造を定義するメタデータは、REFRESH_AWCUBE をコールするまでワークスペース内でインスタンス化されません。

たとえば、ソース・キューブ XADEMO.ANALYTIC_CUBE から MYSCHEMA 内のワークスペース MYAW にキューブ AW_ANACUBE を作成済であれば、アクティブ・カタログを問い合わせてワークスペースをチェックできます。

```
SQL>select * from ALL_OLAP2_AW_CUBES where AW_LOGICAL_NAME in 'AW_ANACUBE';
```

AW_OWNER	AW_NAME	AW_LOGICAL_NAME	SOURCE_OWNER	SOURCE_NAME
MYSCHEMA	MYAW	AW_ANACUBE	XADEMO	ANALYTIC_CUBE

次の問合せでは、キューブに関連付けられているメジャーがないことが示されています。メジャー、ディメンションおよび説明は、キューブのリフレッシュ時にインスタンス化されません。

```
SQL>select * from ALL_OLAP2_AW_CUBE_MEASURES where AW_CUBE_NAME in 'AW_ANACUBE';
```

```
no rows selected
```

キューブをリフレッシュするタイミング

ソース・ファクト表に変更が生じるたびに、キューブをリフレッシュする必要があります。これらの変更は、売上高の更新などのデータの追加や削除の場合や、メジャーの追加や説明名の変更など、キューブのメタデータへの変更の場合があります。

キューブをリフレッシュする場合、最初に、変更されたキューブのディメンションをすべてリフレッシュする必要があります。

キューブのリフレッシュ後に必要な処理

メタデータの変更のためにキューブをリフレッシュする場合は、キューブおよびその関連ディメンションを再有効化する必要があります。データの変更が理由でキューブをリフレッシュする場合は、再有効化する必要はありません。

関連集計仕様を持つキューブをリフレッシュするたびに、そのキューブを再集計する必要があります。

キューブに関連付けられたコンポジット仕様を変更する場合は、そのキューブを削除してアナリティック・ワークスペースに再作成する必要があります。変更されたコンポジット仕様を持つキューブをリフレッシュすることはできません。

スパースなデータの管理およびワークスペース・キューブの最適化

コンポジットは、アナリティック・ワークスペースの変数にスパースなデータをコンパクトに格納するために使用されるオブジェクトです。1つのコンポジットはディメンション値の組合せのリスト（各値はコンポジットがベースとする各ディメンションから取得される）から構成されます。データが存在する組合せのみがコンポジットに含まれます。

コンポジットは、OLAP エンジンによって自動的にメンテナンスされます。コンポジットによって、アナリティック・ワークスペースのサイズを最小限に維持し、パフォーマンスを向上することができます。アナリティック・ワークスペースにおけるスパース性の管理およびパフォーマンス最適化については、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

たとえば、すべての地域で売れていない製品がアナリティック・キューブにいくつかあるとします。これらの **PRODUCT** と **GEOGRAPHY** の組合せのデータ・セルは空のはずです。このような場合に、**PRODUCT** と **GEOGRAPHY** をコンポジットとして定義します。このキューブの **Costs** メジャーのディメンションを定義する OLAP DML 構文は次のようになります。

```
DEFINE prod_geog COMPOSITE <product geography>  
DEFINE costs VARIABLE INTEGER <time channel prod_geog<product geography>>
```

キューブのディメンションのこの定義を使用してアナリティック・ワークスペースにキューブのデータをロードするように指定するには、そのキューブに**コンポジット仕様**を定義します。コンポジット仕様は次の式を定義します。

```
<time channel prod_geog<product geography>>
```

コンポジット仕様の各メンバーは、名前、型および位置を持ちます。前述の例におけるこの情報を表 1-9 に示します。

表 1-9 XADEMO.ANALYTIC_CUBE のコンポジット仕様のメンバー

メンバー	型	位置
TIME	ディメンション	1
CHANNEL	ディメンション	2
PROD_GEOG	コンポジット	3
PRODUCT	ディメンション	4
GEOGRAPHY	ディメンション	5

ディメンションの順序

ディメンションの順序によって、アナリティック・ワークスペースにおけるキューブのデータの格納方法およびアクセス方法が決まります。ディメンションの定義の最初のディメンションが最初に変化し、最後のディメンションが最後に変化します。

デフォルトでは、REFRESH_AWCUBE によって、ワークスペース・キューブの **Time** のディメンションが最初に変化するディメンションとして定義され、その後その他のすべてのディメンションのコンポジットが続きます。コンポジットのディメンションはそれぞれのサイズに応じて順序付けられます。最も多くのメンバーを持つディメンションが最初で、最小メンバーのディメンションが最後です。たとえば、アナリティック・ワークスペースの ANALYTIC_CUBE のデフォルトのディメンションは次のようになります。

```
<time comp_name<geography, product, channel>>
```

コンポジット仕様を指定し、キューブ・ロード仕様に含めることによって、デフォルトのディメンションを上書きできます。

ディメンションの順序付けおよびディメンション・ストレージのセグメント・サイズの指定については、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

コンポジット仕様の作成および変更

例 1-3 の文は、ANALYTIC_CUBE の comp1 というコンポジット仕様を作成します。

例 1-3 アナリティック・ワークスペースにおけるキューブのディメンションの定義

```
exec dbms_awm.create_awcomp_spec
    ('comp1', 'xademo', 'analytic_cube');
exec dbms_awm.add_awcomp_spec_member
    ('comp1', 'xademo', 'analytic_cube', 'comp1_time', 'dimension',
    'xademo', 'time');
exec dbms_awm.add_awcomp_spec_member
    ('comp1', 'xademo', 'analytic_cube', 'comp1_channel', 'dimension',
    'xademo', 'channel');
```

```
exec dbms_awm.add_awcomp_spec_member
    ('comp1', 'xademo', 'analytic_cube', 'comp1_prod_geog', 'composite');
exec dbms_awm.add_awcomp_spec_comp_member
    ('comp1', 'xademo', 'analytic_cube', 'comp1_prod_geog',
    'comp1_product', 'dimension', 'xademo', 'product');
exec dbms_awm.add_awcomp_spec_comp_member
    ('comp1', 'xademo', 'analytic_cube', 'comp1_prod_geog',
    'comp1_geography', 'dimension', 'xademo', 'geography');
exec dbms_awm.add_awcubeload_spec_comp
    ('my_cube_load', 'xademo', 'analytic_cube', 'comp1');
```

コンポジット仕様は、別のキューブに適用するか、別の名前を指定することによって変更できます。コンポジット仕様のメンバーについて、名前の変更、移動およびセグメント・サイズの変更が可能です。ただし、コンポジットのメンバーについては、名前の変更、移動またはセグメント・サイズの変更ができません。コンポジット自身を編集するには、削除してから新たなコンポジットを定義する必要があります。

Time ではなく **Channel** をアナリティック・ワークスペースにおけるキューブの最初に変化するディメンションにするとします。この場合、コンポジット仕様の **Channel** を次のように再配置できます。

```
exec dbms_awm.set_awcomp_spec_member_pos
    ('comp1', 'xademo', 'analytic_cube', 'comp1_channel', 1);
```

アナリティック・ワークスペースにおけるデータの集計

DBMS_AWM パッケージを使用すると、ワークスペース・キューブにおけるメジャーのレベルの組合せの集計データを格納できます。

アナリティック・ワークスペース内のストアド・サマリーは、リレーショナル・データのマテリアライズド・ビューに似ています。ただし、ワークスペース・キューブは、アプリケーションによる SQL アクセスが有効な場合、埋込み合計で完全に解決された状態で常に表示されます。どのワークスペース・データも事前集計しない場合、すべての集計データが使用可能ですが、その場で計算される必要があります。

ワークスペース・データの一部またはすべてを事前集計すると、ほとんどの環境において問合せパフォーマンスが向上します。集計方針の選択については、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

注意： 集計プロセス (AGGREGATE_AWCUBE) は、読み込み / 書き込みアクセスでアナリティック・ワークスペースを開きます。これによってワークスペースは更新されますが、SQL の COMMIT は実行されません。

キューブのリフレッシュ・プロセスでは、ワークスペースに詳細データが格納され、動的集計をサポートするように構造が設定されます。データの一部またはすべてを事前集計する場

合、集計仕様を作成し、ワークスペース・キューブに対して別の集計プロシージャを実行する必要があります。

集計仕様の作成

例 1-4 に、XADEMO.ANALYTIC_CUBE から作成したアナリティック・ワークスペース・キューブ AC2 の Costs メジャーおよび Quota メジャーを事前集計するプロシージャ・コールの例を示します。

製品グループ (PRODUCT のレベル 'L3') の四半期合計 (TIME のレベル 'L2')、製品部門 (PRODUCT のレベル 'L2')、およびすべてのチャネル (CHANNEL のレベル 'STANDARD-2') は、アナリティック・ワークスペースにおいて計算および格納されます。

例 1-4 アナリティック・ワークスペースにおける Costs および Quota の事前集計

```
execute dbms_awm.create_awcubeagg_spec
    ('AGG1', 'MYSHEMA', 'MYAW', 'AC2');
execute dbms_awm.add_awcubeagg_spec_level
    ('AGG1', 'MYSHEMA', 'MYAW', 'AC2', 'PRODUCT', 'L3');
execute dbms_awm.add_awcubeagg_spec_level
    ('AGG1', 'MYSHEMA', 'MYAW', 'AC2', 'PRODUCT', 'L2');
execute dbms_awm.add_awcubeagg_spec_level
    ('AGG1', 'MYSHEMA', 'MYAW', 'AC2', 'CHANNEL', 'STANDARD_2');
execute dbms_awm.add_awcubeagg_spec_level
    ('AGG1', 'MYSHEMA', 'MYAW', 'AC2', 'TIME', 'L2');
execute dbms_awm.add_awcubeagg_spec_measure
    ('AGG1', 'XADEMOAW', 'UK', 'AC2', 'XXF_COSTS');
execute dbms_awm.add_awcubeagg_spec_measure
    ('AGG1', 'XADEMOAW', 'UK', 'AC2', 'XXF_QUOTA');
execute dbms_awm.aggregate_awcube('MYSHEMA', 'MYAW', 'AC2', 'AGG1');
```

次の文は、アナリティック・ワークスペースにおける集計計画のメジャーおよび PRODUCT のレベルを示します。

```
execute dbms_aw.execute ('aw attach MYSCHEMA.MYAW ro');
execute dbms_aw.execute ('fulldsc agg1');
```

```
DEFINE AGG1 DIMENSION TEXT
LD List of Measures which use this AggPlan
PROPERTY 'AW$CLASS' -
'IMPLEMENTATION'
PROPERTY 'AW$CREATEDBY' -
'AW$CREATE'
PROPERTY 'AW$LASTMODIFIED' -
'.'
PROPERTY 'AW$LOGICAL_NAME' -
'AGG1'
```

```
PROPERTY 'AW$PARENT_NAME' -
'AC2'
PROPERTY 'AW$ROLE' -
'AGGDEF'
PROPERTY 'AW$STATE' -
'ACTIVE'

execute dbms_aw.execute('rpr agg1')

AGG1
-----
XXF.COSTS
XXF.QUOTA

execute dbms_aw.execute('fulldsc agg1_product');

DEFINE AGG1_PRODUCT VALUESET PRODUCT_LEVELLIST
LD List of Levels for this AggPlan
PROPERTY 'AW$AGGOPERATOR' -
'SUM'
PROPERTY 'AW$CLASS' -
'IMPLEMENTATION'
PROPERTY 'AW$CREATEDBY' -
'AW$CREATE'
PROPERTY 'AW$LASTMODIFIED' -
'.'
PROPERTY 'AW$PARENT_CUBE' -
'AC2'
PROPERTY 'AW$PARENT_DIM' -
'PRODUCT'
PROPERTY 'AW$PARENT_NAME' -
'AGG1'
PROPERTY 'AW$ROLE' -
'AGGDEF_LEVELS'
PROPERTY 'AW$STATE' -
'ACTIVE'

execute dbms_aw.execute('shw values (agg1_product)');

L3
L2
```

集計メソッドの選択

集計メソッドはレベルごとのデータの集計に使用される操作を指定します。デフォルトの集計メソッドは加算です。たとえば、売上げデータは通常、期間ごとに値が追加されることによって時間の経過とともに集計されます。

OLAP カタログは集計メソッドのセットをサポートしており、これは、キューブの定義に含まれる場合があります。これらの集計メソッドについては、表 1-10 を参照してください。

ワークスペース・キューブをリフレッシュする際、OLAP カタログに指定された集計演算子は、OLAP DML の RELATION コマンドによってサポートされている対応する演算子に変換されます。これらの演算子は、キューブの動的集計を制御する集計マップに組み込まれています。

ストアド・サマリーに異なる演算子を指定するには、SET_AWCUBEAGG_SPEC_AGGOP プロシージャを使用します。このプロシージャを使用すると、OLAP DML の RELATION コマンドでサポートされている任意の演算子を指定してデータを事前集計できます。

注意： DBMS_AWM パッケージは現在、加重集計演算子をサポートしていません。たとえば、OLAP カタログでキューブのディメンションの 1 つに対する集計に加重合計または加重平均を指定した場合、これらは、キューブがアナリティック・ワークスペースにおいてリフレッシュされる際に同等のスカラー演算（合計または平均）に変換されます。SET_AWCUBEAGG_SPEC_AGGOP によって指定される加重演算子も同様に変換されます。

OLAP カタログおよび対応する OLAP DML の集計演算子を表 1-10 に示します。

表 1-10 集計演算子

OLAP カタログ	OLAP DML	DML 略称	説明
SUM	SUM	SU	合計。データ値を追加します（デフォルト）。
SCALED SUM	SSUM	SS	SUM に変換されます。
WEIGHTED SUM	WSUM	WS	SUM に変換されます。
AVERAGE	AVERAGE	AV	平均。データ値を追加し、一緒に追加されたデータ値の数で合計を除算します。
HIERARCHICAL AVERAGE	HAVERAGE	HA	階層の平均。データ値を追加し、ディメンション階層内の子の数で合計を除算します。
WEIGHTED AVERAGE	WAVERAGE	WA	AVERAGE に変換されます。
	HWVERAGE	HW	HIERARCHICAL AVERAGE に変換されます。
MAX	MAX	MA	最大。すべての親データ値の子の中で最大のデータ値。
MIN	MIN	MI	最小。すべての親データ値の子の中で最小のデータ値。
FIRST	FIRST	FI	最初。最初の非 NA データ値。

表 1-10 集計演算子 (続き)

OLAP カタログ	OLAP DML	DML 略称	説明
	HFIRST	HF	階層の最初。階層によって指定された最初のデータ値 (NA 値を含む)。
LAST	LAST	LA	最後。最後の非 NA データ値。
	HLAST	HL	階層の最後。階層によって指定された最後のデータ値 (NA 値を含む)。
AND	AND	AN	(ブール変数のみ) 任意の子のデータ値が FALSE の場合、その親のデータ値は FALSE です。親が TRUE となるのは、その子がすべて TRUE の場合のみです。
OR	OR	OR	(ブール変数のデフォルト) 任意の子のデータ値が TRUE の場合、その親のデータ値は TRUE です。親が FALSE となるのは、その子がすべて FALSE の場合のみです。
COUNT		NO	NOAGG に変換されます。
	NOAGG	NO	このディメンションのすべてのデータを集計しません。

ワークスペース・キューブへのリレーショナル・アクセスの作成

アナリティック・ワークスペース・キューブを作成してそのデータをリフレッシュおよび集計したら、アプリケーションで標準 SQL を使用してそのデータへアクセスできるようにするビューを生成できます。ビューを生成する DBMS_AWM プロシージャは、**OLAP API イネーブラ**・プロシージャとして知られています。このプロシージャは、次のような OLAP API および BI Beans で必要とされる形式でビューおよび OLAP カタログ・メタデータを生成します。

- 各ディメンション階層の埋込み合計ディメンション・ビュー
- ディメンション階層の各組合せの埋込み合計ファクト・ビュー

アナリティック・ワークスペースが異なるアプリケーションをサポートする場合、そのアプリケーションの要件を満たすビューを生成する必要があります。OLAP_TABLE ファンクション (第 25 章を参照) を使用して、様々な形式のビューを生成できます。

ワークスペース・キューブを有効化するには、スクリプトを生成し実行するか、スクリプトを自動的に作成および実行する 1 段階のプロシージャを使用できます。

手順 : 有効化スクリプトの生成および実行

次の手順を使用して、OLAP API および BI Beans によるワークスペース・キューブへのアクセスを可能にします。

1. システムを構成してファイルへ書き込む方法を決めます。イネーブラ・プロシージャでは、ディレクトリ・オブジェクトまたはディレクトリ・パスを指定できます。ディレクトリ・オブジェクトを指定する場合、ユーザー ID にそのオブジェクトに対する適切なアクセス権が付与されていることを確認してください。パスを指定する場合、そのパスがインスタンスの UTL_FILE_DIR 初期化パラメータの値であることを確認してください。
2. REFRESH_AWCUBE および REFRESH_AWDIMENSION プロシージャをコールして、アナリティック・ワークスペースのキューブのデータおよびメタデータをリフレッシュします。リフレッシュ・プロセスでは、有効化プロセスで使用されるメタデータがアナリティック・ワークスペースに作成されます。このメタデータには、有効化スクリプトによって生成されるビューのデフォルト名が含まれます。
3. システムが生成したビュー名を独自のビュー名に置き換える場合、SET_AWDIMENSION_VIEW_NAME および SET_AWCUBE_VIEW_NAME プロシージャをコールします。
4. キューブの各ディメンションに対して、CREATE_AWDIMENSION_ACCESS プロシージャをコールします。access_type パラメータを OLAP に設定します。各プロシージャ・コールによって、指定したディレクトリに有効化スクリプトが作成されます。このスクリプトには、ディメンション・ビューおよびそのビューにマップされる OLAP カタログ・ディメンションを作成する文が含まれます。
5. CREATE_AWCUBE_ACCESS プロシージャをコールします。access_type パラメータを OLAP に設定します。このプロシージャ・コールによって、指定したディレクトリに有効化スクリプトが作成されます。このスクリプトには、ファクト・ビューおよびそのビューにマップされる OLAP カタログ・キューブを作成する文が含まれます。
6. 有効化スクリプトを実行します。このスクリプトは、ビューおよびメタデータを新たに作成する前に、古い世代のビューおよびメタデータをすべて削除します。

手順 : 有効化スクリプトの自動実行

有効化スクリプトを自動的に作成および実行するには、次の手順を使用します。

1. 1-23 ページの「[手順 : 有効化スクリプトの生成および実行](#)」に示すように、アナリティック・ワークスペースのキューブおよびそのディメンションをリフレッシュします。
2. キューブの各ディメンションに対して、CREATE_AWDIMENSION_ACCESS_FULL をコールします。このプロシージャによって、有効化スクリプトが一時メモリーに作成および実行され、ディメンション・ビューおよび OLAP カタログ・メタデータが作成されます。このスクリプトは、ビューおよびメタデータを新たに作成する前に、前のビューおよび OLAP カタログ・メタデータをすべて削除します。

3. プロシージャ `CREATE_AWCUBE_ACCESS_FULL` をコールして、キューブのファクト・ビューを作成します。このプロシージャは、ディメンション用の対応プロシージャと同じ基本手順を実行します。

OLAP API イネーブラ・プロシージャ

OLAP API イネーブラ・プロシージャを表 1-11 にリストします。

表 1-11 OLAP API イネーブラ・プロシージャ

プロシージャ	説明
<code>CREATE_AWCUBE_ACCESS</code> プロシージャ	アナリティック・ワークスペース内のキューブへのアクセスを有効化するスクリプトを作成します。
<code>CREATE_AWCUBE_ACCESS_FULL</code> プロシージャ	アナリティック・ワークスペース内のキューブへのアクセスを有効化します。
<code>CREATE_AWDIMENSION_ACCESS</code> プロシージャ	アナリティック・ワークスペース内のディメンションへのアクセスを有効化するスクリプトを作成します。
<code>CREATE_AWDIMENSION_ACCESS_FULL</code> プロシージャ	アナリティック・ワークスペース内のディメンションへのアクセスを有効化します。
<code>DELETE_AWCUBE_ACCESS</code> プロシージャ	アナリティック・ワークスペース内のキューブへのアクセスを無効化するスクリプトを作成します。
<code>DELETE_AWDIMENSION_ACCESS</code> プロシージャ	アナリティック・ワークスペース内のディメンションへのアクセスを無効化するスクリプトを作成します。
<code>SET_AWCUBE_VIEW_NAME</code> プロシージャ	アナリティック・ワークスペース・キューブのビューのシステムが生成した名前を置き換えます。
<code>SET_AWDIMENSION_VIEW_NAME</code> プロシージャ	アナリティック・ワークスペース・ディメンションのビューのシステムが生成した名前を置き換えます。

注意： Analytic Workspace Manager によって生成された SQL を取得し、それを独自のスクリプトの作成に使用する場合、有効化プロシージャ・コールを編集する必要があります。Analytic Workspace Manager では、有効化プロシージャの異なるバージョンが使用されるからです。スクリプトでは、このマニュアルで説明している構文を使用する必要があります。

アナリティック・ワークスペースの有効化メタデータ

リフレッシュ・プロセスは、データのロードおよびディメンションやキューブの論理構造を定義するメタデータのリフレッシュに加え、有効化に関連するメタデータをアナリティック・ワークスペースに作成します。このメタデータには、有効化スクリプトによって作成されるビューのデフォルト名のセットが含まれます。

リフレッシュするたびに新しいビュー名が生成されます。SET_AWDIMENSION_VIEW_NAME および SET_AWCUBE_VIEW_NAME を使用して以前に独自の名前を作成している場合、リフレッシュ・プロセスでその名前が新しい名前の基準として使用されます。

リフレッシュしてもソース・キューブのメタデータが変更されない場合、有効化スクリプトを再作成する必要はありません。ただし、ワークスペースに格納されている内部ビュー名と実際のビュー名との一貫性が失われます。

リレーショナル・アクセスの無効化

有効化プロセスは、古い世代のビューおよびOLAPカタログ・メタデータをすべて自動的に削除します。ただし、特定の環境においては、ビューおよびメタデータを再作成せずに削除する場合があります。特に、ワークスペース・キューブまたはワークスペース自体を削除する場合、孤立したビューおよびメタデータをクリーン・アップする必要があります。

この場合、DELETE_AWDIMENSION_ACCESS および DELETE_AWCUBE_ACCESS プロセスを実行して、キューブへのリレーショナル・アクセスを有効化するビューおよびメタデータを削除するスクリプトを生成できます。これらのスクリプトによって、アナリティック・ワークスペースに格納されている有効化メタデータが削除されることはありません。

ディメンション・ビューのデフォルト名

REFRESH_AWDIMENSION はビューのデフォルト名を作成します。

SET_AWDIMENSION_VIEW_NAME をコールすることによって、デフォルト名を上書きできます。

デフォルトのビュー名は `aaaa_bbbbb_ccccc_dddd#view` となり、名前の構成要素は次のとおりです。

`aaaa`: アナリティック・ワークスペース所有者の最初の 4 文字

`bbbb`: アナリティック・ワークスペース名の最初の 5 文字

`cccc`: アナリティック・ワークスペース・ディメンション名の最初の 5 文字

`ddd`: アナリティック・ワークスペース階層名の最初の 5 文字

`#`: 一意性を保証するために自動的に生成される 1～9,999 の順序番号

OLAP_TABLE によって移入される抽象オブジェクト (ADT) にもデフォルト名が生成されます。たとえば、XADEMO スキーマの AWTEST というワークスペース内のワークスペース・

ディメンション AWGEOG では、STANDARD 階層に次のシステム生成名を持つことになりません。

デフォルト名	説明
XADE_AWTES_AWGE0_STAND34VIEW	リレーショナル・ビューの名前。
XADE_AWTES_AWGEOG34OBJ	OLAP_TABLE によって移入されたオブジェクトの抽象表の行を定義する抽象オブジェクトの名前。
XADE_AWTES_AWGEOG34TBL	OLAP_TABLE によって移入された抽象表型の名前。

ファクト・ビューのデフォルト名

REFRESH_AWCUBE プロシージャはビューのデフォルト名を作成します。
SET_AWCUBE_VIEW_NAME をコールすることによって、デフォルト名を上書きできます。

デフォルトのビュー名は `aaaa_bbbbb_ccccccc#view` となり、名前の構成要素は次のとおりです。

`aaaa`: アナリティック・ワークスペース所有者の最初の 4 文字

`bbbbbb`: アナリティック・ワークスペース名の最初の 5 文字

`ccccccc`: アナリティック・ワークスペース・キューブ名の最初の 8 文字

`#`: 一意性を保証するために自動的に生成される 1～9,999 の順序番号

OLAP_TABLE によって移入される抽象オブジェクト (ADT) にもデフォルト名が生成されます。たとえば、XADEMO スキーマの AWTEST というワークスペース内のワークスペース・キューブ AWCUBE は、次のシステム生成名を持つことになります。

デフォルト名	説明
XADE_AWTES_AWCUBE8VIEW	最初の階層組合せのリレーショナル・ファクト・ビューの名前。
XADE_AWTES_AWCUBE9VIEW	2 番目の階層組合せのリレーショナル・ファクト・ビューの名前。
XADE_AWTES_AWCUBE10VIEW	3 番目の階層組合せのリレーショナル・ファクト・ビューの名前。
XADE_AWTES_AWCUBE11VIEW	4 番目の階層組合せのリレーショナル・ファクト・ビューの名前。
XADE_AWTES_AWCUBE7OBJ	OLAP_TABLE によって移入されたオブジェクトの抽象表の行を定義する抽象オブジェクトの名前。
XADE_AWTES_AWCUBE7TBL	OLAP_TABLE によって移入された抽象表型の名前。

ディメンション有効化ビューの列構造

有効化プロセスによって、各ディメンション階層の個別のビューが生成されます。たとえば、表 1-12 に示す 4 つのディメンションを持つワークスペース・キューブの場合、2 つのディメンションにそれぞれ 2 つの階層があるため、合計 6 つのディメンション・ビューがあります。

表 1-12 ディメンション階層の例

ディメンション	階層	ビューの数
geography	standard	2
	consolidated	
product	standard	1
channel	standard	1
time	standard	2
	ytd	

ディメンション・ビューはレベルベースであり、すべての行のすべてのレベル値の関係が含まれます。このディメンションに関連するファクト表には、すべてのレベルの組合せに対する埋込み合計が含まれているため、このタイプのディメンション表は**解決済**です。

各ディメンション・ビューには、表 1-13 に示す列が含まれます。

表 1-13 ディメンション・ビューの列

列	説明
ET キー	埋込み合計キー列には、行に移入済の最低レベルの値が格納されます。
親 ET キー	親埋込み合計キー列には、各 ET キー値の親が格納されます。
GID	グルーピング ID 列は、各行に関連付けられた階層レベルを識別します (1-28 ページの「グルーピング ID 列」を参照)。
親 GID	親グルーピング ID 列には、各 GID 値の親が格納されます。
レベル列	ディメンション階層の各レベルの 1 列。これらの列は、単一行内の各ディメンション・メンバーの完全な祖先を提供します。
レベル属性列	各レベル属性の 1 列。

ディメンション・ビューの例

TOTAL_US、REGION および STATE のレベルを持つ標準的な地理階層の場合、ディメンション・ビューには次のような列が含まれます。レベル属性列も含まれます。

GID	PARENT_GID	ET	KEY	PARENT_ET_KEY	TOTAL_US	REGION	STATE
0	1		MA	Northeast	USA	Northeast	MA
0	1		NY	Northeast	USA	Northeast	NY
0	1		GA	Southeast	USA	Southeast	GA
0	1		CA	Southwest	USA	Southwest	CA
0	1		AZ	Southwest	USA	Southwest	AZ
1	3		Northeast	USA	USA	Northeast	
1	3		Southeast	USA	USA	Southeast	
1	3		Southwest	USA	USA	Southwest	
3	NA		USA	NA	USA		

グループング ID 列

GID は、レベル列の NULL 以外の値に 0、NULL 値に 1 を割り当てて、各行に関連付けられた階層レベルを識別します。その結果生成された 2 進数が GID 値です。

たとえば、1 という GID が、次の 3 つのレベルを持つ行に割り当てられています。

TOTAL_US	REGION	STATE
USA	Southwest	
0	0	1

3 という GID が、次の 5 つのレベルを持つ行に割り当てられています。

TOTAL_GEOG	COUNTRY	REGION	STATE	CITY
World	USA	Northeast		
0	0	0	1	1

有効化ファクト・ビューの列構造

CREATE_AWCUBE_ACCESS プロシージャは、ディメンション / 階層の各組合せに対して個別のビューを生成します。たとえば、表 1-12 の 4 つのディメンションを持つアナリティック・ワークスペース・キューブには、表 1-14 に示すように、階層の組合せごとに 1 つずつ、4 つの個別のファクト・ビューがあります。

表 1-14 ディメンション / 階層の組合せの例

地理ディメンション	製品ディメンション	チャネル・ディメンション	時間ディメンション
geography/ standard	product/standard	channel/standard	time/standard
geography/ standard	product/standard	channel/standard	time/ytd
geography/ consolidated	product/standard	channel/standard	time/standard
geography/ consolidated	product/standard	channel/standard	time/ytd

ファクト・ビューは完全に**解決済**です。ファクト・ビューには、すべてのレベルの組合せに対する埋込み合計が含まれます。各ビューには、キューブのメジャーに対する列、およびファクト・ビューとそれに関連付けられたディメンション・ビューをリンクするキー列があります。

各ファクト・ビューには、表 1-15 に示す列が含まれます。

表 1-15 ファクト・ビューの列

列	説明
各ディメンション / 階層の ET キー	ET キー列は、関連付けられたディメンション表の主キーにマップされる外部キーで、メジャー表とディメンション表の結合に使用されます。
各ディメンション / 階層の GID	GID 列は、応答時間の最適化のために OLAP API で必要とされるグルーピング ID を提供します。GID 列は、関連付けられたディメンション表の GID 列と同一です。
メジャー列	キューブの各メジャーの列。
R2C	カスタム・メジャーの動的計算に必要な情報。表 25-2 「OLAP_TABLE の LIMIT_MAP パラメータの構成要素」で説明している ROWTOCELL キーワードを参照してください。
CUST_MEAS_TEXT <i>n</i>	VARCHAR2 (255) データ型を持つ 100 個の空の連番列。 これらの列は、テキスト・データ型を持つ事前定義のカスタム・メジャーを戻します。これらのカスタム・メジャーは、アナリティック・ワークスペース内の式を実行した結果生成されたもので、DBMS_AW_UTILITIES パッケージのプロシージャによって管理されます。詳細は、第 21 章を参照してください。

表 1-15 ファクト・ビューの列 (続き)

列	説明
CUST_MEAS_NUM#	NUMBER(38,6) データ型を持つ 100 個の空の連番列。 これらの列は、数値データ型を持つ事前定義のカスタム・メジャーを戻します。これらのカスタム・メジャーは、アナリティック・ワークスペース内の式を実行した結果生成されたもので、DBMS_AW_UTILITIES パッケージのプロシージャによって管理されます。詳細は、 第 21 章 を参照してください。

例 : OLAP API によるワークスペース・キューブへのアクセスを可能にする

次の例は、ソース・キューブ XADEMO.ANALYTIC_CUBE に基づいて、キューブ AWUSR.AWTEST を作成、リフレッシュおよび有効化します。

例 1-5 キューブの作成、リフレッシュおよび有効化

```
-- SET UP
set serveroutput on size 1000000
execute cwm2_olap_manager.set_echo_on;
execute cwm2_olap_manager.begin_log ('/users/awuser/scripts' , 'awtest.log');

--- CREATE AW
execute dbms_aw.execute ('aw create 'AWTEST'');

-- CREATE DIMENSIONS
execute dbms_awm.create_awdimension
    ('XADEMO','CHANNEL', 'AWUSR', 'AWTEST', 'AWCHAN');
execute dbms_awm.create_awdimension
    ('XADEMO','GEOGRAPHY', 'AWUSR', 'AWTEST', 'AWGEOG');
execute dbms_awm.create_awdimension
    ('XADEMO','PRODUCT', 'AWUSR', 'AWTEST', 'AWPROD');
execute dbms_awm.create_awdimension
    ('XADEMO','TIME', 'AWUSR', 'AWTEST', 'AWTIME');

-- CREATE CUBE
execute dbms_awm.create_awcube
    ('XADEMO', 'ANALYTIC_CUBE', 'AWUSR', 'AWTEST', 'AWCUBE');

-- REFRESH DIMENSIONS
execute dbms_awm.refresh_awdimension ('AWUSR', 'AWTEST', 'AWCHAN');
execute dbms_awm.refresh_awdimension ('AWUSR', 'AWTEST', 'AWGEOG');
execute dbms_awm.refresh_awdimension ('AWUSR', 'AWTEST', 'AWPROD');
execute dbms_awm.refresh_awdimension ('AWUSR', 'AWTEST', 'AWTIME');

-- REFRESH CUBE
```

```
execute dbms_awm.refresh_awcube ('AWUSR', 'AWTEST', 'AWCUBE');

-- SET DIMENSION VIEW NAMES
exec dbms_awm.set_awdimension_view_name
    ('AWUSR', 'AWTEST', 'awprod', 'standard', 'prod_std_view');
exec dbms_awm.set_awdimension_view_name
    ('AWUSR', 'AWTEST', 'awchan', 'standard', 'chan_std_view');
exec dbms_awm.set_awdimension_view_name
    ('AWUSR', 'AWTEST', 'awgeog', 'consolidated', 'geog_csd_view');
exec dbms_awm.set_awdimension_view_name
    ('AWUSR', 'AWTEST', 'awgeog', 'standard', 'geog_std_view');
exec dbms_awm.set_awdimension_view_name
    ('AWUSR', 'AWTEST', 'awtime', 'standard', 'time_std_view');
exec dbms_awm.set_awdimension_view_name
    ('AWUSR', 'AWTEST', 'awtime', 'ytd', 'time_ytd_view');

-- SET CUBE VIEW NAMES
exec dbms_awm.set_awcube_view_name
    ('AWUSR', 'AWTEST', 'awcube', 1, 'AWCUBE_view1');
exec dbms_awm.set_awcube_view_name
    ('AWUSR', 'AWTEST', 'awcube', 2, 'AWCUBE_view2');
exec dbms_awm.set_awcube_view_name
    ('AWUSR', 'AWTEST', 'awcube', 3, 'AWCUBE_view3');
exec dbms_awm.set_awcube_view_name
    ('AWUSR', 'AWTEST', 'awcube', 4, 'AWCUBE_view4');

-- ENABLE DIMENSIONS
exec dbms_awm.create_awdimension_access
    ('AWUSR', 'AWTEST', 'awprod', 'olap',
     '/users/awuser/scripts', 'awprod_views.sql', 'w');
exec dbms_awm.create_awdimension_access
    ('AWUSR', 'AWTEST', 'awchan', 'olap',
     '/users/awuser/scripts', 'awchan_views.sql', 'w');
exec dbms_awm.create_awdimension_access
    ('AWUSR', 'AWTEST', 'awgeog', 'olap',
     '/users/awuser/scripts', 'awgeog_views.sql', 'w');
exec dbms_awm.create_awdimension_access
    ('AWUSR', 'AWTEST', 'awtime', 'olap',
     '/users/awuser/scripts', 'awtime_views.sql', 'w');

-- ENABLE CUBE
exec dbms_awm.create_awcube_access
    ('AWUSR', 'AWTEST', 'awcube', 'olap',
     '/users/awuser/scripts', 'awcube_views.sql', 'w');

-- COMMIT and WRAPUP
commit;
```

```
execute cwm2_olap_manager.end_log;
```

次の問合せは、結果のワークスペース・キューブおよびディメンションを OLAP カタログ内のソース・キューブおよびディメンションとともに表示します。

```
select * from all_olap2_aw_dimensions where AW_OWNER = 'AWUSER';
```

AW_OWNER	AW_NAME	AW_LOGICAL_NAME	AW_PHYSICAL_OBJECT	SOURCE_OWNER	SOURCE_NAME
AWUSER	AWTEST	AWCHAN	AWCHAN	XADEMO	CHANNEL
AWUSER	AWTEST	AWGEOG	AWGEOG	XADEMO	GEOGRAPHY
AWUSER	AWTEST	AWPROD	AWPROD	XADEMO	PRODUCT
AWUSER	AWTEST	AWTIME	AWTIME	XADEMO	TIME

```
select * from all_olap2_aw_CUBE$ where AW_OWNER = 'AWUSER';
```

AW_OWNER	AW_NAME	AW_LOGICAL_NAME	AW_PHYSICAL_OBJECT	SOURCE_OWNER	SOURCE_NAME
AWUSER	AWTEST	AWCUBE	AWCUBE	XADEMO	ANALYTIC_CUBE

次の問合せは、ディメンション有効化ビューのシステム名およびユーザー名を表示します。

```
select * from all_aw_dim_ENABLED_VIEWS where AW_OWNER = 'AWUSER';
```

AW_OWNER	AW_NAME	DIMENSION	HIERARCHY	SYSTEM_VIEWNAME	USER_VIEWNAME
AWUSER	AWTEST	AWCHAN	STANDARD	AWUS_AWTES_AWCHA_STAND144VIEW	CHAN_STD_VIEW
AWUSER	AWTEST	AWGEOG	CONSOLIDATED	AWUS_AWTES_AWGEO_CONSO145VIEW	GEOG_CSD_VIEW
AWUSER	AWTEST	AWGEOG	STANDARD	AWUS_AWTES_AWGEO_STAND146VIEW	GEOG_STD_VIEW
AWUSER	AWTEST	AWPROD	STANDARD	AWUS_AWTES_AWPRO_STAND147VIEW	PROD_STD_VIEW
AWUSER	AWTEST	AWTIME	STANDARD	AWUS_AWTES_AWTIM_STAND148VIEW	TIME_STD_VIEW
AWUSER	AWTEST	AWTIME	YTD	AWUS_AWTES_AWTIM_YTD149VIEW	TIME_YTD_VIEW

次の問合せは、キューブ有効化ビューのシステム名およびユーザー名を表示します。階層の組合せ番号（この場合は、1～4）、およびこのキューブのディメンション階層の一意の各組合せからなる階層文字列が含まれます。

```
select * from all_aw_CUBE_ENABLED_VIEWS where AW_OWNER = 'AWUSER';
```

AW_OWN	AW_NA	CUBE_NAM	HIER	HIERCOMBO_STR	SYSTEM_VIEWNAME	USER_VIEWNAME
AWUSER	AWTEST	AWCUBE	1	DIM:AWCHAN/HIER:STANDARD;DIM:AWGEOG/HIER:CONSOLIDATED;DIM:AWPROD/HIER:STANDARD;DIM:AWTIME/HIER:STANDARD	AWUS_AWTES_AWCUBE151VIEW	AWCUBE_VIEW1
AWUSER	AWTEST	AWCUBE	2	DIM:AWCHAN/HIER:STANDARD;DIM:AWGEOG/HIER:CONSOLIDATED;DIM:AWPROD/HIER:STANDARD;DIM:AWTIME/HIER:YTD	AWUS_AWTES_AWCUBE152VIEW	AWCUBE_VIEW2
AWUSER	AWTEST	AWCUBE	3	DIM:AWCHAN/HIER:STANDARD;DIM:AWGEOG/HIER:STANDARD;DIM:AWPROD/HIER:STANDARD;DIM:AWTIME/HIER:STANDARD	AWUS_AWTES_AWCUBE153VIEW	AWCUBE_VIEW3

```
AWUSER  AWTEST  AWCUBE  4      DIM:AWCHAN/HIER:STANDARD;DIM:AWGEOG  AWUS_AWTES_AWCUBE154VIEW  AWCUBE_VIEW4
/HIER:STANDARD;DIM:AWPROD/HIER:STAN
DARD;DIM:AWTIME/HIER:YTD
```

最後の手順は、有効化スクリプトを実行し、アナリティック・ワークスペース・キューブのビューおよび OLAP カタログ・メタデータを生成することです。この例で作成されたスクリプトを次に示します。

ディレクトリ	スクリプト	説明
/users/awuser/scripts	awprod_views.sql	PRODUCT デイメンションの抽象オブジェクト、オブジェクトの表、およびビューを作成します。ビューにマップされる OLAP カタログ・デイメンション AWUSER.AWPROD の作成および検証も行います。
/users/awuser/scripts	awchan_views.sql	CHANNEL デイメンションの抽象オブジェクト、オブジェクトの表、およびビューを作成します。ビューにマップされる OLAP カタログ・デイメンション AWUSER.AWCHAN の作成および検証も行います。
/users/awuser/scripts	awgeog_views.sql	GEOGRAPHY デイメンションの各階層の抽象オブジェクト、オブジェクトの表、およびビューを作成します。ビューにマップされる OLAP カタログ・デイメンション AWUSER.AWGEOG の作成および検証も行います。
/users/awuser/scripts	awtime_views.sql	TIME デイメンションの各階層の抽象オブジェクト、オブジェクトの表、およびビューを作成します。ビューにマップされる OLAP カタログ・デイメンション AWUSER.AWTIME の作成および検証も行います。
/users/awuser/scripts	awcube_views.sql	AWCUBE キューブの各階層の組合せの抽象オブジェクト、オブジェクトの表、および個別のビューを作成します。ビューにマップされる OLAP カタログ・キューブ AWUSER.AWCUBE の作成および検証も行います。

CWM2 による OLAP カタログ・メタデータの作成

OLAP カタログ CWM2 PL/SQL パッケージでは、OLAP メタデータを作成、削除および更新するためのストアド・プロシージャが提供されます。この章では、CWM2 プロシージャを使用した処理方法を説明します。構文の詳細は、各パッケージについての章を参照してください。

この章では、次の項目について説明します。

- OLAP メタデータのエンティティ
- ディメンションの作成
- キューブの作成
- OLAP メタデータのマッピング
- OLAP メタデータの検証およびコミット
- プロシージャの起動
- 出力の送信
- OLAP メタデータの表示

OLAP メタデータのエンティティ

OLAP メタデータのエンティティは、**ディメンション**、**階層**、**レベル**、**レベル属性**、**ディメンション属性**、**メジャー**、**キューブ**および**メジャー・フォルダ**です。各タイプのエンティティには、個別の PL/SQL パッケージが存在します。このパッケージでは、そのタイプのエンティティの説明を作成、削除、ロックおよび指定するためのプロシージャが提供されます。たとえば、ディメンションを作成するには `CWM2_OLAP_DIMENSION.CREATE_DIMENSION` をコールし、レベルを作成するには `CWM2_OLAP_LEVEL.CREATE_LEVEL` をコールします。

メタデータの各エンティティは、その所有者および名前で一意に識別されます。

OLAP メタデータのエンティティを作成する場合は、そのタイプのすべてのエンティティを識別する OLAP カタログに行を追加します。エンティティを作成しても、ディメンションまたはキューブは完全には定義されません。また、ウェアハウスのディメンション表またはファクト表へのマッピングも実行されません。

注意： すべての OLAP カタログ・メタデータのエンティティは VARCHAR (30) として定義されます。

ディメンションまたはキューブを完全に構成するには、コンポーネント・メタデータのエンティティ間の階層関係について理解する必要があります。

ディメンションの作成

ディメンションのエンティティの作成は、ディメンションの OLAP メタデータを構成する場合の最初の手順です。各ディメンションには、1 つ以上のレベルが必要です。通常、複数のレベル、階層および属性が含まれます。表 2-1 に、ディメンションのメタデータ・コンポーネント間の親子関係を示します。

表 2-1 ディメンションのコンポーネント間の階層関係

親エンティティ	子エンティティ
ディメンション	ディメンション属性、階層、レベル
ディメンション属性	レベル属性
階層	レベル
レベル	レベル属性

注意： CWM2 プロシージャで作成した OLAP カタログ・ディメンションは純粋な論理エンティティです。これらのディメンションは、データベースのディメンション・オブジェクトとは関係がありません。ただし、Enterprise Manager で作成された OLAP カタログ・ディメンションは、データベースのディメンション・オブジェクトに関連付けられています。

手順 : OLAP ディメンションの作成

通常、ディメンションの作成後、ディメンション・レベルおよびレベル属性を作成する前に、階層およびディメンション属性を作成します。レベルおよびレベル属性は、定義後、1 つ以上のウェアハウス・ディメンション表にマップできます。手順は次のとおりです。

1. CWM2_OLAP_DIMENSION のプロシージャをコールして、ディメンションを作成します。

2. CWM2_OLAP_DIMENSION_ATTRIBUTE のプロシージャをコールして、ディメンション属性を作成します。通常は、'long description' および 'short description' のディメンション属性を定義する必要があります。

OLAP API では、アナリティック・ワークスペースのビューなど、埋込み合計ディメンション表に次のディメンション属性が必要となります。'ET Key'、'Parent ET Key'、'Grouping ID' および 'Parent Grouping ID' です。詳細は、表 10-1「予約済のディメンション属性」を参照してください。

3. CWM2_OLAP_HIERARCHY のプロシージャをコールして、ディメンションのレベルに階層関係を定義します。
4. CWM2_OLAP_LEVEL のプロシージャをコールして、レベルを作成し、階層に割り当てます。
5. CWM2_OLAP_LEVEL_ATTRIBUTE のプロシージャをコールして、レベル属性を作成し、ディメンション属性に割り当てます。'long description'、'short description' およびその他の予約済ディメンション属性に対して、各レベルで同じ名前のレベル属性を作成します。

OLAP API では、アナリティック・ワークスペースのビューなど、埋込み合計ディメンション表に次のレベル属性が必要となります。'ET Key'、'Parent ET Key'、'Grouping ID' および 'Parent Grouping ID' です。詳細は、表 13-1「予約済のレベル属性」を参照してください。

6. CWM2_OLAP_TABLE_MAP のプロシージャをコールして、ディメンションのレベルおよびレベル属性をディメンション表の列にマップします。

例：製品ディメンションの作成

例 2-1 の PL/SQL 文は、論理 CWM2 ディメンションの PRODUCT_DIM を、SH スキーマの PRODUCTS ディメンション表に作成します。

次の表に、PRODUCTS 表の列を示します。

列名	データ型
PROD_ID	NUMBER
PROD_NAME	VARCHAR2
PROD_DESC	VARCHAR2
PROD_SUBCATEGORY	VARCHAR2
PROD_SUBCAT_DESC	VARCHAR2
PROD_CATEGORY	VARCHAR2
PROD_CAT_DESC	VARCHAR2

列名	データ型
PROD_WEIGHT_CLASS	NUMBER
PROD_UNIT_OF_MEASURE	VARCHAR2
PROD_PACK_SIZE	VARCHAR2
SUPPLIER_ID	NUMBER
PROD_STATUS	VARCHAR2
PROD_LIST_PRICE	NUMBER
PROD_MIN_PRICE	NUMBER
PROD_TOTAL	VARCHAR2

例 2-1 PRODUCTS 表に OLAP ディメンションを作成

```

--- CREATE THE PRODUCT DIMENSION ---
exec cwm2_olap_dimension.create_dimension
    ('SH', 'PRODUCT_DIM', 'Product','Products', 'Product Dimension',
     'Product Dimension Values');

--- CREATE DIMENSION ATTRIBUTES ---
exec cwm2_olap_dimension_attribute.create_dimension_attribute
    ('SH', 'PRODUCT_DIM', 'Long Description', 'Long Descriptions',
     'Long Desc', 'Long Product Descriptions', true);
exec cwm2_olap_dimension_attribute.create_dimension_attribute
    ('SH', 'PRODUCT_DIM', 'PROD_NAME_DIM', 'Product Name',
     'Prod Name', 'Product Name');

--- CREATE STANDARD HIERARCHY ---
exec cwm2_olap_hierarchy.create_hierarchy
    ('SH', 'PRODUCT_DIM', 'STANDARD', 'Standard', 'Std Product',
     'Standard Product Hierarchy', 'Unsolved Level-Based');
exec cwm2_olap_dimension.set_default_display_hierarchy
    ('SH', 'PRODUCT_DIM', 'standard');

--- CREATE LEVELS ---
exec cwm2_olap_level.create_level
    ('SH', 'PRODUCT_DIM', 'L4', 'Product ID', 'Product Identifiers',
     'Prod Key','Product Key');
exec cwm2_olap_level.create_level
    ('SH', 'PRODUCT_DIM', 'L3', 'Product Sub-Category',
     'Product Sub-Categories', 'Prod Sub-Category',
     'Sub-Categories of Products');
exec cwm2_olap_level.create_level

```

```

('SH', 'PRODUCT_DIM', 'L2', 'Product Category',
 'Product Categories', 'Product Category', 'Categories of Products');
exec cwm2_olap_level.create_level
('SH', 'PRODUCT_DIM', 'L1', 'Total Product', 'Total Products',
 'Total Prod', 'Total Product');

--- CREATE LEVEL ATTRIBUTES ---
exec cwm2_olap_level_attribute.create_level_attribute
('SH', 'PRODUCT_DIM', 'Long Description', 'L4', 'Long Description',
 'PRODUCT_LABEL', 'L4 Long Desc',
 'Long Labels for PRODUCT Identifiers', TRUE);
exec cwm2_olap_level_attribute.create_level_attribute
('SH', 'PRODUCT_DIM', 'Long Description', 'L3', 'Long Description',
 'SUBCATEGORY_LABEL', 'L3 Long Desc',
 'Long Labels for PRODUCT Sub-Categories', TRUE);
exec cwm2_olap_level_attribute.create_level_attribute
('SH', 'PRODUCT_DIM', 'Long Description', 'L2', 'Long Description',
 'CATEGORY_LABEL', 'L2 Long Desc',
 'Long Labels for PRODUCT Categories', TRUE);
exec cwm2_olap_level_attribute.create_level_attribute
('SH', 'PRODUCT_DIM', 'PROD_NAME_DIM', 'L4', 'PROD_NAME_LEV',
 'Product Name', 'Product Name', 'Product Name');

--- ADD LEVELS TO HIERARCHIES ---
exec cwm2_olap_level.add_level_to_hierarchy
('SH', 'PRODUCT_DIM', 'STANDARD', 'L4', 'L3');
exec cwm2_olap_level.add_level_to_hierarchy
('SH', 'PRODUCT_DIM', 'STANDARD', 'L3', 'L2');
exec cwm2_olap_level.add_level_to_hierarchy
('SH', 'PRODUCT_DIM', 'STANDARD', 'L2', 'L1');
exec cwm2_olap_level.add_level_to_hierarchy
('SH', 'PRODUCT_DIM', 'STANDARD', 'L1');

--- CREATE MAPPINGS ---
exec cwm2_olap_table_map.Map_DimTbl_HierLevel
('SH', 'PRODUCT_DIM', 'STANDARD', 'L4',
 'SH', 'PRODUCTS', 'PROD_ID');
exec cwm2_olap_table_map.Map_DimTbl_HierLevelAttr
('SH', 'PRODUCT_DIM', 'Long Description', 'STANDARD',
 'L4', 'Long Description', 'SH', 'PRODUCTS', 'PROD_DESC');
exec cwm2_olap_table_map.Map_DimTbl_HierLevelAttr
('SH', 'PRODUCT_DIM', 'PROD_NAME_DIM', 'STANDARD', 'L4',
 'PROD_NAME_LEV', 'SH', 'PRODUCTS', 'PROD_NAME');
exec cwm2_olap_table_map.Map_DimTbl_HierLevel
('SH', 'PRODUCT_DIM', 'STANDARD', 'L3', 'SH', 'PRODUCTS',
 'PROD_SUBCATEGORY');
exec cwm2_olap_table_map.Map_DimTbl_HierLevelAttr

```

```
( 'SH', 'PRODUCT_DIM', 'Long Description', 'STANDARD', 'L3',  
  'Long Description', 'SH', 'PRODUCTS', 'PROD_SUBCATEGORY_DESC' );  
exec cwm2_olap_table_map.Map_DimTbl_HierLevel  
  ( 'SH', 'PRODUCT_DIM', 'STANDARD', 'L2', 'SH', 'PRODUCTS',  
    'PROD_CATEGORY' );  
exec cwm2_olap_table_map.Map_DimTbl_HierLevelAttr  
  ( 'SH', 'PRODUCT_DIM', 'Long Description', 'STANDARD', 'L2',  
    'Long Description', 'SH', 'PRODUCTS', 'PROD_CATEGORY_DESC' );  
exec cwm2_olap_table_map.Map_DimTbl_HierLevel  
  ( 'SH', 'PRODUCT_DIM', 'STANDARD', 'L1', 'SH', 'PRODUCTS',  
    'PROD_TOTAL' );
```

手順 : 時間ディメンションの作成

時間ディメンション表のメタデータを構成する際、他の OLAP ディメンションに対する手順と同じ一般的な手順に従います。ただし、いくつかの追加要件が適用されます。時間ディメンションを作成する場合の一般的な手順は次のとおりです。

1. CWM2_OLAP_DIMENSION のプロシージャをコールして、ディメンションを作成します。ディメンション・タイプ・パラメータに 'TIME' を指定します。
2. CWM2_OLAP_DIMENSION_ATTRIBUTE のプロシージャをコールして、ディメンション属性を作成します。通常のディメンションに必要なディメンション属性に加え、'End Date' 属性および 'Time Span' 属性を定義します。
3. CWM2_OLAP_HIERARCHY のプロシージャをコールして、ディメンションのレベルに階層関係を定義します。通常の階層は、Calendar および Fiscal です。
4. CWM2_OLAP_LEVEL のプロシージャをコールして、レベルを作成し、階層に割り当てます。通常のレベルは、Month、Quarter および Year です。
5. CWM2_OLAP_LEVEL_ATTRIBUTE のプロシージャをコールして、レベル属性を作成し、ディメンション属性に割り当てます。通常のディメンション属性に必要なレベル属性に加え、各レベルに 'End Date' および 'Time Span' 属性を作成し、'End Date' および 'Time Span' ディメンション属性に関連付けます。
6. CWM2_OLAP_TABLE_MAP のプロシージャをコールして、ディメンションのレベルおよびレベル属性をディメンション表の列にマップします。'End Date' レベル属性を日付データ型の列にマップします。'Time Span' レベル属性を数値データ型の列にマップします。

例 : 時間ディメンションの作成

例 2-1 の PL/SQL 文は、論理 CWM2 時間ディメンションの TIME_DIM を、SH スキーマの TIMES ディメンション表に作成します。

TIMES 表には次の列が含まれます。

列名	データ型
TIME_ID	DATE
TIME_ID_KEY	NUMBER
DAY_NAME	VARCHAR2(9)
CALENDAR_MONTH_NUMBER	NUMBER(2)
CALENDAR_MONTH_DESC	VARCHAR2(8)
CALENDAR_MONTH_DESC_KEY	NUMBER
END_OF_CAL_MONTH	DATE
CALENDAR_MONTH_NAME	VARCHAR2(9)
CALENDAR_QUARTER_DESC	CHAR(7)
CALENDAR_QUARTER_DESC_KEY	NUMBER
END_OF_CAL_QUARTER	DATE
CALENDAR_QUARTER_NUMBER	NUMBER(1)
CALENDAR_YEAR	NUMBER(4)
CALENDAR_YEAR_KEY	NUMBER
END_OF_CAL_YEAR	DATE

例 2-2 OLAP 時間ディメンションの作成

```

--- CREATE THE TIME DIMENSION
exec cwm2_olap_dimension.create_dimension
    ('SH', 'TIME_DIM', 'Time','Time', 'Time Dimension',
     'Time Dimension Values', 'TIME');

--- CREATE DIMENSION ATTRIBUTE END DATE
exec cwm2_olap_dimension_attribute.create_dimension_attribute
    ('SH', 'TIME_DIM', 'END DATE', 'End Date',
     'End Date', 'Last date of time period', true);

--- CREATE CALENDAR HIERARCHY
exec cwm2_olap_hierarchy.create_hierarchy
    ('SH', 'TIME_DIM', 'CALENDAR', 'Calendar', 'Calendar Hierarchy',
     'Calendar Hierarchy', 'Unsolved Level-Based');
exec cwm2_olap_dimension.set_default_display_hierarchy
    ('SH', 'TIME_DIM', 'CALENDAR');

```

```

--- CREATE LEVELS
exec cwm2_olap_level.create_level
    ('SH', 'TIME_DIM', 'MONTH', 'Month', 'Months', 'Month', 'Month');
exec cwm2_olap_level.create_level
    ('SH', 'TIME_DIM', 'QUARTER', 'Quarter', 'Quarters', 'Quarter', 'Quarter');
exec cwm2_olap_level.create_level
    ('SH', 'TIME_DIM', 'YEAR', 'Year', 'Years', 'Year', 'Year');

--- CREATE LEVEL ATTRIBUTES ---
exec cwm2_olap_level_attribute.create_level_attribute
    ('SH', 'TIME_DIM', 'END DATE', 'Month', 'END DATE',
     'End Date', 'End Date',
     'Last date of time period', TRUE);
exec cwm2_olap_level_attribute.create_level_attribute
    ('SH', 'TIME_DIM', 'END DATE', 'Quarter', 'END DATE',
     'End Date', 'End Date',
     'Last date of time period', TRUE);
exec cwm2_olap_level_attribute.create_level_attribute
    ('SH', 'TIME_DIM', 'END DATE', 'Year', 'END DATE',
     'End Date', 'End Date',
     'Last date of time period', TRUE);

--- ADD LEVELS TO HIERARCHIES
exec cwm2_olap_level.add_level_to_hierarchy
    ('SH', 'TIME_DIM', 'CALENDAR', 'Month', 'Quarter');
exec cwm2_olap_level.add_level_to_hierarchy
    ('SH', 'TIME_DIM', 'CALENDAR', 'Quarter', 'Year');
exec cwm2_olap_level.add_level_to_hierarchy
    ('SH', 'TIME_DIM', 'CALENDAR', 'Year');

--- CREATE MAPPINGS
exec cwm2_olap_table_map.Map_DimTbl_HierLevel
    ('SH', 'TIME_DIM', 'CALENDAR', 'Year',
     'SH', 'TIMES', 'CALENDAR_YEAR_ID');
exec cwm2_olap_table_map.Map_DimTbl_HierLevelAttr
    ('SH', 'TIME_DIM', 'END DATE', 'CALENDAR',
     'Year', 'END DATE', 'SH', 'TIMES', 'END_OF_CAL_YEAR');
exec cwm2_olap_table_map.Map_DimTbl_HierLevel
    ('SH', 'TIME_DIM', 'CALENDAR', 'Quarter', 'SH', 'TIMES',
     'CALENDAR_QUARTER_NUMBER');
exec cwm2_olap_table_map.Map_DimTbl_HierLevelAttr
    ('SH', 'TIME_DIM', 'END DATE', 'CALENDAR',
     'Quarter', 'END DATE', 'SH', 'TIMES', 'END_OF_CAL_QUARTER');
exec cwm2_olap_table_map.Map_DimTbl_HierLevel
    ('SH', 'TIME_DIM', 'CALENDAR', 'Month', 'SH', 'TIMES',
     'CALENDAR_MONTH_NUMBER');
exec cwm2_olap_table_map.Map_DimTbl_HierLevelAttr

```

```
('SH', 'TIME_DIM', 'END DATE', 'CALENDAR',
'Month', 'END DATE', 'SH', 'TIMES', 'END_OF_CAL_MONTH');
```

キューブの作成

キューブのエントリティの作成は、キューブの OLAP メタデータを構成する場合の最初の手順です。各キューブには、1つ以上のディメンションおよび1つ以上のメジャーが必要です。通常、複数のディメンションおよび複数のメジャーが含まれます。

手順：キューブの作成

キューブを構成する場合の手順は次のとおりです。

1. 「**手順：OLAP ディメンションの作成**」の手順に従って、キューブの各ディメンションを作成します。
2. CWM2_OLAP_CUBE のプロシージャをコールして、キューブを作成し、そのディメンションを識別します。
3. CWM2_OLAP_MEASURE のプロシージャをコールして、キューブのメジャーを作成します。
4. CWM2_OLAP_TABLE_MAP のプロシージャをコールして、キューブのメジャーをファクト表の列に、ファクト表の外部キー列をディメンション表のキー列にマップします。

例：Costs キューブの作成

例 2-3 の PL/SQL 文は、論理 CWM2 キューブ・オブジェクトの ANALYTIC_CUBE を、SH スキーマの COSTS ファクト表に作成します。キューブのディメンションは、PRODUCT_DIM (例 2-1 を参照) および TIME_DIM (例 2-2 を参照) です。

COSTS ファクト表には次の列が含まれます。

列名	データ型
PROD_ID	NUMBER
TIME_ID	DATE
UNIT_COST	NUMBER
UNIT_PRICE	NUMBER

例 2-3 COSTS ファクト表に OLAP キューブを作成

```
--- CREATE THE ANALYTIC_CUBE CUBE ---
cwm2_olap_cube.create_cube('SH', 'ANALYTIC_CUBE', 'Analytics');
```

```
'Analytic Cube','Unit Cost and Price Analysis');

--- ADD THE DIMENSIONS TO THE CUBE ---
cwm2_olap_cube.add_dimension_to_cube('SH', 'ANALYTIC_CUBE',
    'SH', 'TIME_DIM');
cwm2_olap_cube.add_dimension_to_cube('SH', 'ANALYTIC_CUBE',
    'SH', 'PRODUCT_DIM');

--- CREATE THE MEASURES ---
cwm2_olap_measure.create_measure('SH', 'ANALYTIC_CUBE', 'UNIT_COST',
    'Unit Cost','Unit Cost', 'Unit Cost');
cwm2_olap_measure.create_measure('SH', 'ANALYTIC_CUBE', 'UNIT_PRICE',
    'Unit Price','Unit Price', 'Unit Price');

--- CREATE THE MAPPINGS ---
cwm2_olap_table_map.Map_FactTbl_LevelKey
('SH', 'ANALYTIC_CUBE','SH', 'COSTS', 'LOWESTLEVEL',
    'DIM:SH.PRODUCTS/HIER:STANDARD/LVL:L4/COL:PROD_ID;
    DIM:SH.TIME/HIER:CALENDAR/LVL:L3/COL:MONTH;');
cwm2_olap_table_map.Map_FactTbl_Measure
('SH', 'ANALYTIC_CUBE','UNIT_COST', 'SH', 'COSTS', 'UNIT_COST',
    'DIM:SH.PRODUCTS/HIER:STANDARD/LVL:L4/COL:PROD_ID;
    DIM:SH.TIME/HIER:CALENDAR/LVL:L3/COL:MONTH;');
cwm2_olap_table_map.Map_FactTbl_Measure
('SH', 'ANALYTIC_CUBE','UNIT_PRICE', 'SH', 'COSTS', 'UNIT_PRICE',
    'DIM:SH.PRODUCTS/HIER:STANDARD/LVL:L4/COL:PROD_ID;
    DIM:SH.TIME/HIER:CALENDAR/LVL:L3/COL:MONTH;');
```

OLAP メタデータのマッピング

OLAP メタデータのマッピングは、論理メタデータのエンティティと、データが格納されている実際の場所の間にリンクを確立する処理です。ディメンション・レベルおよびレベル属性は、ディメンション表の列にマップされます。メジャーは、ファクト表の列にマップされます。マッピング処理は、ファクト表とファクト表に関連付けられたディメンション表の結合関係も指定できます。

注意： ディメンション表およびファクト表は、ビューとして実装される場合があります。たとえば、DBMS_AWM パッケージを使用して生成可能なビューが、OLAP メタデータのデータ・ソースである場合があります。これらのビューには、データが実際に存在するアナリティック・ワークスペース上のリレーショナル・ファクト表およびディメンション表のイメージが表示されます。詳細は、22-21 ページの「[CREATE_AWCUBE_ACCESS プロシージャ](#)」を参照してください。

列へのマッピング

CWM2_OLAP_TABLE_MAP パッケージには、CWM2 メタデータのマッピング・プロシージャが含まれます。ディメンション・レベル、レベル属性およびメジャーは、階層のコンテキストの内外でマップできます。

ディメンションのマッピング

各レベルは、ディメンション表の1つ以上の列にマップされます。複数列からなるレベルのすべての列は、同じ表内にマップされる必要があります。ディメンションのすべてのレベルは、同じ表（従来のスター・スキーマ）の列、または個別の表（スノーフレイク・スキーマ）の列にマップできます。

各レベル属性は、関連付けられたレベルと同じ表の単一列にマップされます。

メジャーのマッピング

各メジャーは、ファクト表の単一列にマップされます。同じファクト表内にマップされたすべてのメジャーは、同じディメンションを共有する必要があります。

キューブ内で1つ以上の階層コンテキストが可能な場合（少なくともキューブのディメンションの1つに複数の階層がある）、階層の各組合せは個別のファクト表にマップできます。この場合、各表はキューブの各メジャーに対して列を持つ必要があり、メジャー列は各表において同じ順序である必要があります。

ファクト表とディメンション表の結合

レベル、レベル属性およびメジャーをマップした後、ディメンション表のレベル・キー列へのファクト表の論理外部キー列のマッピングを指定できます。

MAP_FACTTBL_LEVELKEY プロシージャは、キューブとそのディメンションの結合関係を定義します。このプロシージャには、キューブ名、ファクト表名、マッピング文字列および格納タイプ・インジケータ（データをファクト表に格納する方法を指定）を入力します。

格納タイプ・インジケータには、次のいずれかの値を指定できます。

- **'LOWESTLEVEL'**
単一ファクト表に、キューブのすべてのメジャーに対する未解決データが格納されます（スター・スキーマ）。キューブのいずれかのディメンションに複数の階層が存在する場合、それらのすべての階層には同じ最低レベルが含まれる必要があります。ファクト表の各外部キー列は、ディメンション表のレベル・キー列にマップされます。
- **'ET'**
ファクト表には、キューブのディメンションの特定の階層に対する解決済データ（埋込み合計を含む）が格納されます。通常、階層の各組合せに対するデータは、個別のファクト表に格納されます。各ファクト表には、同じ列が必要です。ディメンションに存在する複数の階層では、同じ最低レベルを共有する必要はありません。

ファクト表の埋込み合計キーおよびグルーピング ID キー (GID) は、対応する列 (解決済ディメンション表のディメンション階層を識別する列) にマップされます。ET キーは、行に存在する最低レベルの値を識別します。GID は、各行に関連付けられた階層レベルを識別します。詳細は、1-28 ページの「[グルーピング ID 列](#)」を参照してください。ファクト表およびディメンション表間のキーの関係のマッピングについては、17-9 ページの「[MAP_FACTTBL_LEVELKEY プロシージャ](#)」を参照してください。

OLAP API は、ET ディメンションに特定の属性を必要とします。表 10-1「[予約済のディメンション属性](#)」を参照してください。

ファクト表とディメンション表が格納タイプ LOWESTLEVEL で結合される際、キューブの階層は `solved_code` が、'UNSOLVED LEVEL-BASED' となります。

ファクト表とディメンション表が格納タイプ ET で結合される際、キューブの階層は `solved_code` が、'SOLVED LEVEL-BASED' となります。

11-8 ページの「[SET_SOLVED_CODE プロシージャ](#)」を参照してください。

OLAP メタデータの検証およびコミット

OLAP メタデータを作成、マップまたは検証する CWM2 プロシージャに COMMIT は含まれません。

OLAP API のメタデータを準備するには、新しいメタデータを作成、マップおよび検証し、次に OLAP API Metadata Reader の表をリフレッシュする文をすべて最初にスクリプトで実行する必要があります。リフレッシュ・プロセスには COMMIT が含まれます。2-15 ページの「[OLAP API のメタデータ表のリフレッシュ](#)」を参照してください。

別の種類のアプリケーションに OLAP メタデータを準備する場合、スクリプトにはメタデータの作成、マップおよび検証後の COMMIT を含める必要があります。

OLAP メタデータの検証

OLAP メタデータの妥当性を検証するには、`CWM2_OLAP_VALIDATE` および `CWM2_OLAP_VERIFY_ACCESS` パッケージを使用します。検証プロシージャでは、メタデータの構造的な整合性がチェックされ、メタデータがディメンション表およびファクト表の列に適切にマップされているかどうかを確認されます。必要に応じて、OLAP API に固有の追加の検証プロセスを実行することもできます。

`CWM2_OLAP_VERIFY_ACCESS` パッケージは、キューブの検証後に 2 つの追加チェックを行います。1 つは、キューブの CWM2 メタデータが、OLAP API Metadata Reader で問い合わせたキャッシュされたメタデータ表と整合性があることのチェックです。もう 1 つは、コール元ユーザーがソース表および列へのアクセス権を所有していることのチェックです。

参照:

- 2-15 ページの「[OLAP API のメタデータ表のリフレッシュ](#)」を参照してください。
- 第 18 章「[CWM2_OLAP_VALIDATE](#)」を参照してください。
- 第 19 章「[CWM2_OLAP_VERIFY_ACCESS](#)」を参照してください。

注意: CWM2 メタデータ同様、Enterprise Manager で作成または更新したメタデータも検証してください。

検証プロセスを実行する際、検証プロセスのサマリーや詳細レポートを生成することができます。画面上での出力表示またはファイルへの出力書込みについては、2-17 ページの「[出力の送信](#)」を参照してください。

例 2-4 に、XADEMO の PRODUCT ディメンションを検証し、詳細な検証レポートを生成する文を示します。このレポートは画面に表示され、ログ・ファイルに書き込まれます。

例 2-4 PRODUCT ディメンションの検証レポートの生成

```
set echo on
set linesize 135
set pagesize 50
set serveroutput on size 1000000

execute cwm2_olap_manager.set_echo_on;
execute cwm2_olap_manager.begin_log('/users/myxademo/myscripts' , 'x.log');

execute cwm2_olap_validate.validate_dimension
        ('xademo','product','default','yes');

execute cwm2_olap_manager.end_log;
execute cwm2_olap_manager.set_echo_off;
```

検証レポートは、次のように表示されます。

```
Validate Dimension: XADEMO.PRODUCT      Type of Validation: DEFAULT      Verbose Report: YES
Validating Dimension in OLAP Catalog 1
```

ENTITY TYPE	ENTITY NAME	STATUS	COMMENT
Dimension	.	VALID	
Dimension	XADEMO.PRODUCT	VALID	
LevelAttribute	PROD_STD_TOP_LLABEL	VALID	DimensionAttribute "Long Description"
LevelAttributeMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_TOP_LLABEL"
LevelAttribute	PROD_STD_TOP_SLABEL	VALID	DimensionAttribute "Short Description"

LevelAttributeMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_TOP_SLABEL"
Hierarchy	STANDARD	VALID	
Level	L4	VALID	Hierarchy depth 1 (Lowest Level)
LevelMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_PRODUCT"
LevelAttribute	PROD_COLOR	VALID	DimensionAttribute "Color"
LevelAttributeMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_COLOR"
LevelAttribute	PROD_SIZE	VALID	DimensionAttribute "Size"
LevelAttributeMap		VALID	Mapped to Column "XADEMO.XADEMO_ PRODUCT.PROD_SIZE"
LevelAttribute	PROD_STD_PRODUCT_LLABEL	VALID	DimensionAttribute "Long Description"
LevelAttributeMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_PRODUCT_LLABEL"
LevelAttribute	PROD_STD_PRODUCT_SLABEL	VALID	DimensionAttribute "Short Description"
LevelAttributeMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_PRODUCT_SLABEL"
Level	L3	VALID	Hierarchy depth 2
LevelMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_GROUP"
LevelAttribute	PROD_STD_GROUP_LLABEL	VALID	DimensionAttribute "Long Description"
LevelAttributeMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_GROUP_LLABEL"
LevelAttribute	PROD_STD_GROUP_SLABEL	VALID	DimensionAttribute "Short Description"
LevelAttributeMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_GROUP_SLABEL"
Level	L2	VALID	Hierarchy depth 3
LevelMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_DIVISION"
LevelAttribute	PROD_STD_DIVISION_LLABEL	VALID	DimensionAttribute "Long Description"
LevelAttributeMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_DIVISION_LLABEL"
LevelAttribute	PROD_STD_DIVISION_SLABEL	VALID	DimensionAttribute "Short Description"
LevelAttributeMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_DIVISION_SLABEL"
Level	L1	VALID	Hierarchy depth 4 (Top Level)
LevelMap		VALID	Mapped to Column "XADEMO.XADEMO_PRODUCT .PROD_STD_TOP"

注意： メタデータのエンティティが無効の場合、検証レポートの COMMENT 列に、問題がこのエンティティに起因するのか、そのエンティティが依存する別のエンティティにあるのかを示します。たとえば、レベルが無効の場合、その従属レベル属性も無効となります。

妥当性ステータスの表示

ALL_OLAP2_CUBES および ALL_OLAP2_DIMENSIONS ビューの INVALID 列を選択して、キューブおよびディメンションの妥当性ステータスを確認できます。次のいずれかの値が表示されます。

- Y -- キューブまたはディメンションが無効です。
- N -- キューブまたはディメンションが基本的な検証基準を満たしています。
- O -- キューブが基本的な検証基準および OLAP API 固有の追加基準を満たしています。

詳細は、5-5 ページの「[ALL_OLAP2_CUBES](#)」および 5-7 ページの「[ALL_OLAP2_DIMENSIONS](#)」を参照してください。

OLAP API のメタデータ表のリフレッシュ

OLAP API からメタデータへアクセスできるようにするには、CWM2_OLAP_METADATA_REFRESH パッケージを使用して、OLAP API Metadata Reader の表をリフレッシュします。

これらの表に構築されたビューは、OLAP API Metadata Reader による問合せ用に最適化された OLAP カタログへの読み込み API を表示します。Metadata Reader のビューには、接頭辞 MRV_OLAP2 が付くパブリック・シノニムがあります。詳細は、[第 15 章](#)を参照してください。

注意： OLAP API でアクセスできるようにするには、Metadata Reader の表をリフレッシュする必要があります。

CWM2 API をコールして OLAP メタデータを作成するスクリプトがある場合、メタデータの検証および Metadata Reader 表のリフレッシュを行うコールを含めます。

OLAP メタデータの作成に Enterprise Manager を使用する場合、メタデータの作成後に、検証とリフレッシュのプロシージャを別々に実行する必要があります。

プロシージャの起動

OLAP カタログの書き込み API を使用する場合は、すべての CWM2 プロシージャに共通の論理および規則を理解しておく必要があります。

セキュリティ・チェックおよびエラー条件

各 CWM2 プロシージャは、まず、コール元ユーザーのセキュリティ権限を確認します。コール元ユーザーは OLAP_DBA ロールを持っている必要があります。通常、コール元ユーザーは

エンティティの所有者である必要があります。コール元ユーザーがセキュリティ要件を満たしていない場合、プロシージャは正常に実行されず、例外が戻されます。たとえば、ユーザー ID が `jsmith` の場合、`jjones` が所有する階層に対して、`CWM2_OLAP_HIERARCHY.DROP_HIERARCHY` は正常に実行できません。

各プロシージャは、セキュリティ要件を確認した後、エンティティおよびその親エンティティの存在を確認します。エンティティが存在しない場合、`CREATE` プロシージャを除くすべてのプロシージャがエラーを戻します。たとえば、レベルが存在しない場合に `CWM2_OLAP_LEVEL.SET_DESCRIPTION` をコールすると、プロシージャは正常に実行されません。

パラメータのサイズ要件

CWM2 メタデータ・エンティティは、説明および表示名を使用して作成されます。たとえば、`CWM2_OLAP_CUBE` パッケージの `CREATE_CUBE` プロシージャは次のパラメータを必要とします。

```
CREATE_CUBE (  
    cube_owner          IN    VARCHAR2,  
    cube_name           IN    VARCHAR2,  
    display_name        IN    VARCHAR2,  
    short_description   IN    VARCHAR2,  
    description         IN    VARCHAR2);
```

エンティティ名および説明には、OLAP カタログ・モデル表の格納される列幅に基づくサイズの制限があります。表 2-2 に、サイズの制限を示します。

表 2-2 CWM2 メタデータ・エンティティのサイズ制限

メタデータ・エンティティ	最大サイズ
エンティティの所有者	30 文字
エンティティ名	30 文字
表示名	30 文字
簡単な説明	240 文字
説明	2000 文字

パラメータの大 / 小文字識別要件

CWM2 プロシージャの引数は、小文字または大文字（あるいは大 / 小文字の組合せ）で指定できます。

引数は、メタデータのエンティティ名 (`dimension_name` など) または他のプロシージャでの追加処理に使用される値 (階層の `solved_code` など) の場合、プロシージャによって大文字に変換されます。他の引数の場合は、指定した文字が保持されます。

出力の送信

CWM2 スクリプトの開発およびデバッグに役立つ使用可能なツールや設定がいくつかあります。

CWM2 プロシージャからの出力およびメッセージを SQL バッファにエコーできます。次の文を使用します。

```
SQL>exec cwm2_olap_manager.set_echo_on;
```

デフォルトでは、エコーはオフです。エコーをオンに設定している場合、次の文でオフにできます。

```
SQL>exec cwm2_olap_manager.set_echo_off;
```

次の文を使用して、SQL バッファの内容を画面に表示するように SQL*Plus で設定できます。

```
SQL>set serveroutput on
```

SQL バッファのデフォルトの最小サイズは 2KB です。サイズは、次の文を使用して最大 1MB まで拡張できます。

```
SQL>set serveroutput on size 1000000
```

バッファ・オーバーフロー状態を回避するには `serveroutput` を最大サイズに設定する必要があります。

参照： `serveroutput` の設定の詳細は、『SQL*Plus ユーザーズ・ガイド およびリファレンス』を参照してください。

より多くの量の出力を格納するには、ファイルに出力を送信する必要があります。次の文を使用します。

```
SQL>exec cwm2_olap_manager.begin_log('directory_path','filename');
```

`directory_path` には、自身のユーザー ID に適切なアクセス権が付与されているディレクトリ・オブジェクトか、インスタンスの `UTL_FILE_DIR` 初期化パラメータで設定されているディレクトリ・パスを指定できます。

バッファの内容をフラッシュし、ロギングをオフにするには、次の文を使用します。

```
SQL>exec cwm2_olap_manager.end_log;
```

OLAP メタデータの表示

接頭辞 `ALL_OLAP2` によって識別されるビューのセットに、OLAP カタログのメタデータが表示されます。このメタデータは、CWM2 PL/SQL パッケージまたは Oracle Enterprise

Manager で作成されています。ALL_OLAP2 ビューは、メタデータが変更されると、自動的に移入されます。

接頭辞 MRV_OLAP によって識別される 2 番目のビューのセットにも、OLAP カタログ・メタデータが表示されます。ただし、これらのビューは、OLAP API Metadata Reader による高速問合せをサポートするために特別に構成されています。これらのビューは、メタデータが変更されると、明示的にリフレッシュする必要があります。

参照：

- ALL_OLAP2 ビューの詳細は、[第 5 章「OLAP カタログ・メタデータのビュー」](#)を参照してください。
- OLAP API のメタデータ表のリフレッシュの詳細は、[第 15 章「CWM2_OLAP_METADATA_REFRESH」](#)を参照してください。

アクティブ・カタログ・ビュー

この章では、アナリティック・ワークスペースのスタンダード・フォーム・オブジェクトのリレーショナル・ビューについて説明します。ワークスペース内で、スタンダード・フォーム・オブジェクトは、DBMS_AWM パッケージのプロシージャによって自動的に作成および移入されます。

参照：

- [第 1 章「DBMS_AWM を使用したアナリティック・ワークスペースの作成」](#) を参照してください。
- [第 22 章「DBMS_AWM」](#) を参照してください。
- [15-2 ページの「キャッシュされたアクティブ・カタログ・メタデータのビュー」](#) を参照してください。

この章では、次の項目について説明します。

- [スタンダード・フォームのアクティブ・カタログ](#)
- [例：アナリティック・ワークスペース・キューブの間合せ](#)
- [アクティブ・カタログ・ビューの概要](#)

スタンダード・フォームのアクティブ・カタログ

OLAP の処理は、キューブ、メジャー、ディメンション、階層、レベルおよび属性から構成されるデータ・モデルに基づいています。OLAP カタログ・メタデータはリレーショナル・ソースにこの論理モデルを定義します。スタンダード・フォーム・メタデータは、アナリティック・ワークスペース内の論理モデルを定義します。

DBMS_AWM パッケージのプロシージャは、アナリティック・ワークスペースのディメンションとキューブを作成およびリフレッシュする際に、スタンダード・フォーム・メタデータを作成およびメンテナンスします。OLAP カタログ・メタデータが DBA によって明示的に作成される必要があるのに対して、スタンダード・フォーム・メタデータは、ワークスペース

管理の一部としてアクティブに生成されます。このメタデータのビューは、アナリティック・ワークスペース内で自動的に生成される情報が移入されるため、一般に**アクティブ・カタログ**と呼ばれます。

アクティブ・カタログのビューは `OLAP_TABLE` ファンクションを使用して、リレーショナル・フォーマットでワークスペース・データを戻します。`OLAP_TABLE` の詳細は、[第 25 章](#) を参照してください。

注意： アクティブ・カタログに対する問合せのパフォーマンスを向上させるには、`MRV_OLAP2_AW` ビューの基礎となるキャッシュされたメタデータ表をリフレッシュします。詳細は、[15-2 ページの「キャッシュされたアクティブ・カタログ・メタデータのビュー」](#) を参照してください。

スタンダード・フォームのクラス

スタンダード・フォームの各ワークスペース・オブジェクトは、次の 4 つのクラスのいずれかに属します。

- **実装クラス。** このクラスのオブジェクトは論理モデルを実装します。
- **カタログ・クラス。** このクラスのオブジェクトは、論理モデルに関する情報を保持します。
- **機能クラス。** このクラスのオブジェクトは、論理モデルの特定オブジェクトに関する情報を保持します。
- **拡張クラス。** このクラスのオブジェクトは独自に使用します。

アクティブ・カタログおよびスタンダード・フォームのクラス

アクティブ・カタログ・ビューの情報の主要なソースは、**カタログ・クラス**のオブジェクトです。これには、アナリティック・ワークスペースのキューブ、メジャー、ディメンション、レベルおよび属性のすべてのリストが含まれます。

アクティブ・カタログ・ビューは、**カタログ・クラス**の論理オブジェクトを **OLAP** カタログのソース・オブジェクトおよび実装クラスのコンテナに関連付ける情報も提供します。

最後に、2 つのアクティブ・カタログ・ビューが、スタンダード・フォームのすべてのオブジェクトおよびこれらのオブジェクトのすべてのプロパティを提供します。

注意： アクティブ・カタログ・ビューは、現行のユーザーがアクセスできるすべてのアナリティック・ワークスペースのスタンダード・フォーム・オブジェクトに関する情報を提供します。

参照：

- スタンダード・フォームのアナリティック・ワークスペースについては、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

例：アナリティック・ワークスペース・キューブの問合せ

例 3-1 では、XADEMO キューブ ANALYTIC_CUBE を使用して 2 つのアクティブ・カタログ・ビューを示します。

例 3-1 ワークスペース・キューブの情報に関するアクティブ・カタログの問合せ

次の文は、アナリティック・ワークスペース XADEMO.MY_AW のディメンションを作成します。

```
execute dbms_awm.create_awdimension
    ('XADEMO','CHANNEL','XADEMO', 'MY_AW', 'AW_CHAN');
execute dbms_awm.create_awdimension
    ('XADEMO','PRODUCT','XADEMO', 'MY_AW', 'AW_PROD');
execute dbms_awm.create_awdimension
    ('XADEMO','GEOGRAPHY','XADEMO', 'MY_AW', 'AW_GEOG');
execute dbms_awm.create_awdimension
    ('XADEMO','TIME','XADEMO', 'MY_AW', 'AW_TIME');
```

次の問合せを使用して、アナリティック・ワークスペースの論理ディメンションを表示できます。

```
SQL>select * from ALL_OLAP2_AW_DIMENSIONS;
```

AW_OWNER	AW_NAME	AW_LOGICAL_NAME	AW_PHYSICAL_OBJECT	SOURCE_OWNER	SOURCE_NAME
XADEMO	MY_AW	AW_CHAN	AW_CHAN	XADEMO	CHANNEL
XADEMO	MY_AW	AW_PROD	AW_PROD	XADEMO	PRODUCT
XADEMO	MY_AW	AW_GEOG	AW_GEOG	XADEMO	GEOGRAPHY
XADEMO	MY_AW	AW_TIME	AW_TIME	XADEMO	TIME

次の文はキューブを作成します。

```
execute dbms_awm.create_awcube
    ('XADEMO','ANALYTIC_CUBE','XADEMO', 'MY_AW', 'MY_ANALYTIC_CUBE');
```

次の問合せを使用して、アナリティック・ワークスペースの論理キューブを表示できます。

```
SQL>select * from ALL_OLAP2_AW_CUBES;
```

AW_OWNER	AW_NAME	AW_LOGICAL_NAME	AW_PHYSICAL_OBJECT	SOURCE_OWNER	SOURCE_NAME
----------	---------	-----------------	--------------------	--------------	-------------

```
XADEMO MY_AW MY_ANALYTIC_CUBE MY_ANALYTIC_CUBE XADEMO ANALYTIC_CUBE
```

次の問合せは、アナリティック・ワークスペース・キューブを関連するディメンションとともに戻します。

```
SQL>select * from ALL_OLAP2_AW_CUBE_DIM_USES;
```

AW_OWNER	AW_NAME	AW_LOGICAL_NAME	DIMENSION_ AW_OWNER	DIMENSION_ AW_NAME	DIMENSION_ SOURCE_OWNER	DIMENSION_ SOURCE_NAME
XADEMO	MY_AW	MY_ANALYTIC_CUBE	XADEMO	AW_CHAN	XADEMO	CHANNEL
XADEMO	MY_AW	MY_ANALYTIC_CUBE	XADEMO	AW_GEOG	XADEMO	GEOGRAPHY
XADEMO	MY_AW	MY_ANALYTIC_CUBE	XADEMO	AW_PROD	XADEMO	PRODUCT
XADEMO	MY_AW	MY_ANALYTIC_CUBE	XADEMO	AW_TIME	XADEMO	TIME

アクティブ・カタログ・ビューの概要

アナリティック・ワークスペースのアクティブ・カタログ・ビューの要約を次の表に示します。

表 3-1 アクティブ・カタログ・ビュー

パブリック・シノニム	説明
ALL_OLAP2_AWS	アナリティック・ワークスペースのリスト。
ALL_OLAP2_AW_ATTRIBUTES	アナリティック・ワークスペースのディメンション属性のリスト。
ALL_OLAP2_AW_CUBES	アナリティック・ワークスペースのキューブのリスト。
ALL_OLAP2_AW_CUBE_AGG_LVL	アナリティック・ワークスペースの集計計画のレベルのリスト。
ALL_OLAP2_AW_CUBE_AGG_MEAS	アナリティック・ワークスペースの集計計画のメジャーのリスト。
ALL_OLAP2_AW_CUBE_AGG_OP	アナリティック・ワークスペースの集計計画の集計演算子のリスト。
ALL_OLAP2_AW_CUBE_AGG_SPECS	アナリティック・ワークスペースの集計計画のリスト。
ALL_OLAP2_AW_CUBE_DIM_USES	アナリティック・ワークスペースのキューブと関連ディメンションのリスト。
ALL_OLAP2_AW_CUBE_MEASURES	アナリティック・ワークスペースのキューブと関連メジャーのリスト。
ALL_OLAP2_AW_DIMENSIONS	アナリティック・ワークスペースのディメンションのリスト。
ALL_OLAP2_AW_DIM_HIER_LVL_ORD	アナリティック・ワークスペースの階層レベルのリスト。

表 3-1 アクティブ・カタログ・ビュー (続き)

パブリック・シノニム	説明
ALL_OLAP2_AW_DIM_LEVELS	アナリティック・ワークスペースのレベルのリスト。
ALL_OLAP2_AW_PHYS_OBJ	アナリティック・ワークスペースのスタンダード・フォーム・オブジェクトのリスト。
ALL_OLAP2_AW_PHYS_OBJ_PROP	アナリティック・ワークスペースのスタンダード・フォーム・オブジェクトに関連付けられたプロパティのリスト。

ALL_OLAP2_AWS

ALL_OLAP2_AWS は、現行のユーザーがアクセスできるすべてのアナリティック・ワークスペースのリストを提供します。これには、スタンダード・フォームおよびスタンダード・フォームでないアナリティック・ワークスペースの両方が含まれます。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	アナリティック・ワークスペースの所有者。
AW	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_NUMBER	NUMBER	NOT NULL	アナリティック・ワークスペースに対する一意の識別子。

ALL_OLAP2_AW_ATTRIBUTES

ALL_OLAP2_AW_ATTRIBUTES は、スタンダード・フォームのアナリティック・ワークスペースの属性を表示します。

ディメンションに関連付けられた属性は、DBMS_AWM.REFRESH_AWDIMENSION プロシージャによってアナリティック・ワークスペースに作成されます。1-12 ページの「[ディメンションのメタデータのリフレッシュ](#)」を参照してください。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2 (30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_DIMENSION_NAME	VARCHAR2 (1000)		アナリティック・ワークスペースのディメンションの名前。
AW_LOGICAL_NAME	VARCHAR2 (90)		アナリティック・ワークスペースの属性の論理名。
AW_PHYSICAL_OBJECT	VARCHAR2 (1000)		アナリティック・ワークスペースの属性のスタンダード・フォーム名。

列	データ型	NULL	説明
DISPLAY_NAME	VARCHAR2(1000)		属性の表示名。
DESCRIPTION	VARCHAR2(1000)		属性の説明。
ATTRIBUTE_TYPE	VARCHAR2(1000)		属性のタイプ。表 10-1 「予約済のディメンション属性」を参照。
SOURCE_OWNER	VARCHAR2(1000)		OLAP カタログのソース属性の所有者。
SOURCE_DIMENSION_NAME	VARCHAR2(1000)		OLAP カタログのソース・ディメンションの名前。
SOURCE_NAME	VARCHAR2(1000)		OLAP カタログのソース属性の名前。

ALL_OLAP2_AW_CUBES

ALL_OLAP2_AW_CUBES は、スタンダード・フォームのアナリティック・ワークスペースのキューブを表示します。

スタンダード・フォームのキューブは、DBMS_AWM.CREATE_AWCUBE プロシージャによってアナリティック・ワークスペースに作成されます。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2(30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2(30)		アナリティック・ワークスペースの名前。
AW_LOGICAL_NAME	VARCHAR2(90)		アナリティック・ワークスペースのキューブの論理名。
AW_PHYSICAL_OBJECT	VARCHAR2(1000)		アナリティック・ワークスペースのキューブのスタンダード・フォーム名。
SOURCE_OWNER	VARCHAR2(1000)		OLAP カタログのソース・キューブの所有者。
SOURCE_NAME	VARCHAR2(1000)		OLAP カタログのソース・キューブの名前。

ALL_OLAP2_AW_CUBE_AGG_LVL

ALL_OLAP2_AW_CUBE_AGG_LVL は、スタンダード・フォームのアナリティック・ワークスペースの集計仕様のレベルを表示します。

集計仕様は、サマリー・データの計算方法およびアナリティック・ワークスペースへの格納方法を決定します。レベルは、DBMS_AWM.ADD_AWCUBEAGG_LEVEL プロシージャによって集計仕様に追加されます。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2 (30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_CUBE_NAME	VARCHAR2 (90)		アナリティック・ワークスペースのキューブの名前。
AW_AGGSPEC_NAME	VARCHAR2 (1000)		キューブの集計仕様の名前。
AW_DIMENSION_NAME	VARCHAR2 (1000)		キューブのワークスペース・ディメンションの名前。
AW_LEVEL_NAME	VARCHAR2 (1000)		ディメンションのワークスペース・レベルの名前。このレベルは集計仕様内にあります。

ALL_OLAP2_AW_CUBE_AGG_MEAS

ALL_OLAP2_AW_CUBE_AGG_MEAS は、スタンダード・フォームのアナリティック・ワークスペースの集計仕様のメジャーを表示します。

集計仕様は、サマリー・データの計算方法およびアナリティック・ワークスペースへの格納方法を決定します。メジャーは、DBMS_AWM.ADD_AWCUBEAGG_SPEC_MEASURE プロシージャによって集計仕様追加されます。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2 (30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_CUBE_NAME	VARCHAR2 (90)		アナリティック・ワークスペースのキューブの名前。
AW_AGGSPEC_NAME	VARCHAR2 (1000)		キューブの集計仕様の名前。
AW_MEASURE_NAME	VARCHAR2 (1000)		キューブのワークスペース・メジャーの名前。このメジャーは集計仕様内にあります。

ALL_OLAP2_AW_CUBE_AGG_OP

ALL_OLAP2_AW_CUBE_AGG_OP は、スタンダード・フォームのアナリティック・ワークスペースの集計仕様の集計演算子を表示します。

集計仕様は、サマリー・データの計算方法およびアナリティック・ワークスペースへの格納方法を決定します。集計演算子は、DBMS_AWM.SET_AWCUBEAGG_SPEC_AGGOP プロシージャによって集計仕様追加されます。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2(30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2(30)		アナリティック・ワークスペースの名前。
AW_CUBE_NAME	VARCHAR2(90)		アナリティック・ワークスペースのキューブの名前。
AW_MEASURE_NAME	VARCHAR2		集計するワークスペース・メジャーの名前。
AW_AGGSPEC_NAME	VARCHAR2(1000)		キューブの集計仕様の名前。
AW_DIMENSION_NAME	VARCHAR2(1000)		キューブのワークスペース・ディメンションの名前。
OPERATOR	VARCHAR2(1000)		このディメンションに対する集計用演算子。有効な演算子については、表 1-10「集計演算子」を参照してください。

ALL_OLAP2_AW_CUBE_AGG_SPECS

ALL_OLAP2_AW_CUBE_AGG_SPECS は、スタンダード・フォームのアナリティック・ワークスペースの集計仕様を表示します。

集計仕様は、サマリー・データの計算方法およびアナリティック・ワークスペースへの格納方法を決定します。集計仕様は、DBMS_AWM.CREATE_AWCUBEAGG_SPEC プロシージャによって作成されます。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2(30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2(30)		アナリティック・ワークスペースの名前。
AW_CUBE_NAME	VARCHAR2(90)		アナリティック・ワークスペースのキューブの名前。
AW_AGGSPEC_NAME	VARCHAR2(1000)		キューブの集計計画の名前。

ALL_OLAP2_AW_CUBE_DIM_USES

ALL_OLAP2_AW_CUBE_DIM_USES は、スタンダード・フォームのアナリティック・ワークスペースのキューブのディメンションを表示します。

ディメンションは、DBMS_AWM.CREATE_AWCUBE プロシージャによってワークスペース・キューブに関連付けられます。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2(30)		アナリティック・ワークスペースの所有者。

列	データ型	NULL	説明
AW_NAME	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_LOGICAL_NAME	VARCHAR2 (90)		アナリティック・ワークスペースのキューブの名前。
DIMENSION_AW_OWNER	VARCHAR2 (1000)		キューブのワークスペース・ディメンションの所有者。
DIMENSION_AW_NAME	VARCHAR2 (1000)		キューブのワークスペース・ディメンションの名前。
DIMENSION_SOURCE_OWNER	VARCHAR2 (1000)		OLAP カタログのソース・ディメンションの所有者。
DIMENSION_SOURCE_NAME	VARCHAR2 (1000)		OLAP カタログのソース・ディメンションの名前。

ALL_OLAP2_AW_CUBE_MEASURES

ALL_OLAP2_AW_CUBE_MEASURES は、スタンダード・フォームのアナリティック・ワークスペースのキューブのメジャーを表示します。

メジャーは、DBMS_AWM.REFRESH_AWCUBE プロシージャによってキューブに関連付けられます。個別のメジャーが、DBMS_AWM.ADD_AWCUBELOAD_SPEC_MEASURE へのコールによって指定されていない場合、キューブのすべてのメジャーはキューブのリフレッシュ時にロードされます。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2 (30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_CUBE_NAME	VARCHAR2 (90)		アナリティック・ワークスペースのキューブの名前。
AW_MEASURE_NAME	VARCHAR2 (1000)		キューブのメジャーの論理名。
AW_PHYSICAL_OBJECT	VARCHAR2 (1000)		メジャーのスタンダード・フォーム名。
MEASURE_SOURCE_NAME	VARCHAR2 (1000)		OLAP カタログのソース・メジャーの名前。
DISPLAY_NAME	VARCHAR2 (1000)		アナリティック・ワークスペースのメジャーの表示名。
DESCRIPTION	VARCHAR2 (1000)		アナリティック・ワークスペースのメジャーの説明。
IS_AGGREGATEABLE	VARCHAR2 (1000)		OLAP DML の AGGREGATE コマンドでこのメジャーを集計するかどうか。メジャーが OLAP 変数として実装されるか、基になる格納形式が変数の場合、値は YES です。たとえば、値が変数に格納される式としてメジャーを実装できます。

ALL_OLAP2_AW_DIMENSIONS

ALL_OLAP2_AW_DIMENSIONS は、スタンダード・フォームのアナリティック・ワークスペースのディメンションを表示します。

ワークスペース・ディメンションは、DBMS_AWM.CREATE_AWDIMENSION プロシージャによって作成されます。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2 (30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_LOGICAL_NAME	VARCHAR2 (90)		アナリティック・ワークスペースのディメンションの論理名。
AW_PHYSICAL_NAME	VARCHAR2 (1000)		アナリティック・ワークスペースのディメンションのスタンダード・フォーム名。
SOURCE_OWNER	VARCHAR2 (1000)		OLAP カタログのソース・ディメンションの所有者。
SOURCE_NAME	VARCHAR2 (1000)		OLAP カタログのソース・ディメンションの名前。

ALL_OLAP2_AW_DIM_HIER_LVL_ORD

ALL_OLAP2_AW_DIM_HIER_LVL_ORD は、スタンダード・フォームのアナリティック・ワークスペースの階層のレベルを表示します。これには、階層内の各レベルの位置が含まれます。

ワークスペース・ディメンションは、DBMS_AWM.CREATE_AWDIMENSION プロシージャによって作成されます。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2 (30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_DIMENSION_NAME	VARCHAR2 (90)		アナリティック・ワークスペースのディメンションの名前。
AW_HIERARCHY_NAME	VARCHAR2 (1000)		ワークスペース・ディメンションの階層の名前。
IS_DEFAULT_HIER	VARCHAR2 (1000)		この階層がデフォルトの階層かどうか。
AW_LEVEL_NAME	VARCHAR2 (1000)		ワークスペース階層のレベルの名前。
POSITION	NUMBER		階層のレベルの位置。

ALL_OLAP2_AW_DIM_LEVELS

ALL_OLAP2_AW_DIM_LEVELS は、スタンダード・フォームのアナリティック・ワークスペースのディメンションのレベルを表示します。

ワークスペース・レベルは、DBMS_AWM.CREATE_AWDIMENSION プロシージャによって作成されます。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2 (30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_LOGICAL_NAME	VARCHAR2 (90)		アナリティック・ワークスペースのディメンションの名前。
LEVEL_NAME	VARCHAR2 (1000)		ディメンションのワークスペース・レベルの名前。
DISPLAY_NAME	VARCHAR2 (1000)		レベルの表示名。
DESCRIPTION	VARCHAR2 (1000)		レベルの説明。

ALL_OLAP2_AW_PHYS_OBJ

ALL_OLAP2_AW_PHYS_OBJ は、アナリティック・ワークスペースのスタンダード・フォーム・オブジェクトを表示します。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2 (30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_OBJECT_NAME	VARCHAR2 (90)		アナリティック・ワークスペースのスタンダード・フォーム・オブジェクトの名前。
AW_OBJECT_TYPE	VARCHAR2 (1000)		スタンダード・フォーム・オブジェクトのタイプ。タイプは、OLAP DML で定義可能な任意のネイティブ・オブジェクト・タイプです (ディメンション、リレーション、変数、式、コンポジットおよび値セットを含む)。
AW_OBJECT_DATATYPE	VARCHAR2 (1000)		スタンダード・フォーム・オブジェクトのデータ型。データ型は、OLAP DML でサポートされている任意のネイティブ型 (テキスト、ブールまたは整数を含む) か、スタンダード・フォームに固有の定義型です。

ALL_OLAP2_AW_PHYS_OBJ_PROP

ALL_OLAP2_AW_PHYS_OBJ_PROP は、スタンダード・フォーム・オブジェクトのプロパティとともに表示します。

列	データ型	NULL	説明
AW_OWNER	VARCHAR2 (30)		アナリティック・ワークスペースの所有者。
AW_NAME	VARCHAR2 (30)		アナリティック・ワークスペースの名前。
AW_OBJECT_NAME	VARCHAR2 (90)		アナリティック・ワークスペースのスタンダード・フォーム・オブジェクトの名前。
AW_PROP_NAME	VARCHAR2 (1000)		スタンダード・フォーム・オブジェクトのプロパティ名。
AW_PROP_VALUE	VARCHAR2 (1000)		プロパティの値。

アナリティック・ワークスペース・メンテナンス・ビュー

この章では、DBMS_AWM パッケージで作成したアナリティック・ワークスペースのメンテナンス情報を取得するために問合せ可能なビューについて説明します。

参照：

- [第 1 章「DBMS_AWM を使用したアナリティック・ワークスペースの作成」](#) を参照してください。
- [第 22 章「DBMS_AWM」](#) を参照してください。

この章では、次の項目について説明します。

- [アナリティック・ワークスペースの構築およびメンテナンス](#)
- [例：ワークスペース・ディメンションのロード・パラメータおよび有効化パラメータの問合せ](#)
- [アナリティック・ワークスペース・メンテナンス・ビューの概要](#)

アナリティック・ワークスペースの構築およびメンテナンス

DBMS_AWM パッケージは、スタンダード・フォームのアナリティック・ワークスペースのライフ・サイクルを管理します。これには、リレーショナル・ソースからのワークスペース・キューブの作成、データのロードおよびリレーショナル・アクセスのためのワークスペース・キューブの有効化が含まれます。

DBMS_AWM パッケージは、ワークスペース構築に関する情報を OLAP カタログに格納します。この情報は、アナリティック・ワークスペース・メンテナンス・ビューを問い合わせることで取得できます。たとえば、ワークスペース・キューブとそのリレーショナル・ソースのリスト、ロード仕様のリスト、またはコンポジット仕様のリストを取得できます。

DBMS_AWM パッケージは、ワークスペース有効化に関する情報をアナリティック・ワークスペース自身に格納します。アナリティック・ワークスペース・メンテナンス・ビューは OLAP_TABLE ファンクションを使用して、ワークスペース・キューブの有効化に関する情報を戻します。これらのビューを問い合せて、有効化ビューおよび階層の組合せの名前を取得できます。

例：ワークスペース・ディメンションのロード・パラメータおよび有効化パラメータの問合せ

次の例では、XADEMO のディメンション CHANNEL および TIME を使用して、アナリティック・ワークスペース・メンテナンス・ビューをいくつか示します。

例 4-1 CHANNEL と TIME のロード・パラメータおよび有効化ビュー名の問合せ

次の文は、ディメンション AW_CHAN および AW_TIME をアナリティック・ワークスペース MY_SCHEMA.MY_AW に作成します。

```
execute dbms_awm.create_awdimension
      ('XADEMO','CHANNEL','MY_SCHEMA', 'MY_AW', 'AW_CHAN');
execute dbms_awm.create_awdimension
      ('XADEMO','TIME','MY_SCHEMA', 'MY_AW', 'AW_TIME');
```

次の文は、ディメンションのロード仕様を作成します。

```
execute dbms_awm.create_awdimload_spec
      ('CHAN_DIMLOADSPEC', 'XADEMO', 'CHANNEL', 'FULL_LOAD');
execute dbms_awm.add_awdimload_spec_filter
      ('CHAN_DIMLOADSPEC', 'XADEMO', 'CHANNEL', 'XADEMO', 'XADEMO_CHANNEL',
      ' ''CHAN_STD_CHANNEL'' = ''DIRECT'' ');
execute dbms_awm.create_awdimload_spec
      ('TIME_DIMLOADSPEC', 'XADEMO', 'TIME', 'FULL_LOAD');
execute dbms_awm.add_awdimload_spec_filter
      ('TIME_DIMLOADSPEC', 'XADEMO', 'TIME', 'XADEMO', 'XADEMO_TIME',
      ' ''TIME_STD_YEAR'' = ''1997'' ');
```

次の問合せは、ディメンションのロード仕様に関連付けられたフィルタ条件を戻します。

```
SQL>select * from all_aw_load_dim_filters;
```

OWNER	DIMENSION_NAME	LOAD_NAME	TABLE_OWNER	TABLE_NAME	FILTER_CONDITION
XADEMO	TIME	TIME_DIMLOADSPEC	XADEMO	XADEMO_TIME	'TIME_STD_YEAR' = '1997'
XADEMO	CHANNEL	CHAN_DIMLOADSPEC	XADEMO	XADEMO_CHANNEL	'CHAN_STD_CHANNEL' = 'DIRECT'

次の文は、アナリティック・ワークスペースのディメンションをロードします。ロード・プロセスの一部として、有効化ビューで使用されるシステム生成名がワークスペースに作成されます。

```
execute dbms_awm.refresh_awdimension
    ('MY_SCHEMA', 'MY_AW', 'AWCHAN', 'CHAN_DIMLOADSPEC');
execute dbms_awm.refresh_awdimension
    ('MY_SCHEMA', 'MY_AW', 'AWTIME', 'TIME_DIMLOADSPEC');
```

次の問合せは、ディメンションの有効化ビューのシステム生成名を戻します。

```
SQL>select * from all_aw_dim_enabled_views;
```

AW_OWNER	AW_NAME	DIMENSION_NAME	HIERARCHY_NAME	SYSTEM_VIEWNAME	USER_VIEWNAME
MY_SCHEMA	MY_AW	AWCHAN	STANDARD	MY_S_MY_AW_AWCHA_STAND35VIEW	
MY_SCHEMA	MY_AW	AWTIME	STANDARD	MY_S_MY_AW_AWTIM_STAND36VIEW	
MY_SCHEMA	MY_AW	AWTIME	YTD	MY_S_MY_AW_AWTIM_YTD37VIEW	

アナリティック・ワークスペース・メンテナンス・ビューの概要

アナリティック・ワークスペース・メンテナンス・ビューの要約を次の表に示します。

表 4-1 アナリティック・ワークスペース・メンテナンス・ビュー

パブリック・シノニム	説明
ALL_AW_CUBE_AGG_LEVELS	キューブの集計仕様のレベルを表示します。
ALL_AW_CUBE_AGG_MEASURES	キューブの集計仕様のメジャーを表示します。
ALL_AW_CUBE_AGG_PLANS	キューブの集計仕様を表示します。
ALL_AW_CUBE_ENABLED_HIERCOMBO	キューブに関連付けられた階層の組合せを表示します。
ALL_AW_CUBE_ENABLED_VIEWS	ワークスペース・キューブの生成可能なファクト・ビューを表示します。
ALL_AW_DIM_ENABLED_VIEWS	ワークスペース・ディメンションの生成可能なディメンション・ビューを表示します。
ALL_AW_LOAD_CUBES	キューブのロード仕様を表示します。
ALL_AW_LOAD_CUBE_DIMS	キューブのコンポジット仕様を表示します。
ALL_AW_LOAD_CUBE_FILTERS	キューブのロード仕様に関連付けられたフィルタ条件を表示します。
ALL_AW_LOAD_CUBE_MEASURES	キューブのロード仕様のメジャーを表示します。
ALL_AW_LOAD_CUBE_PARMs	キューブのロード仕様のパラメータを表示します。

表 4-1 アナリティック・ワークスペース・メンテナンス・ビュー (続き)

パブリック・シノニム	説明
ALL_AW_LOAD_DIMENSIONS	ディメンションのロード仕様を表示します。
ALL_AW_LOAD_DIM_FILTERS	ディメンションのロード仕様に関連付けられたフィルタ条件を表示します。
ALL_AW_LOAD_DIM_PARAMS	ディメンションのロード仕様のパラメータを表示します。

ALL_AW_CUBE_AGG_LEVELS

ALL_AW_CUBE_AGG_LEVELS は、キューブの集計仕様のレベルを表示します。

集計仕様は、アナリティック・ワークスペースにおけるキューブのディメンションに対するデータの集計方法を決定します。集計仕様は、DBMS_AWM.CREATE_AWCUBEAGG_SPEC プロシージャによって作成されます。

列	データ型	NULL	説明
owner	varchar2(240)		キューブの所有者。
cube_name	varchar2(240)		キューブの名前。
aggregation_name	varchar2(60)		集計仕様の名前。
dimension_owner	varchar2(30)		集計するディメンションの所有者。
dimension_name	varchar2(240)		集計するディメンションの名前。
level_name	varchar2(240)		このディメンションの集計のレベル名。

ALL_AW_CUBE_AGG_MEASURES

ALL_AW_CUBE_AGG_MEASURES は、キューブの集計仕様のメジャーを表示します。

集計仕様は、アナリティック・ワークスペースにおけるキューブのディメンションに対するメジャーの集計方法を決定します。集計仕様は、DBMS_AWM.CREATE_AWCUBEAGG_SPEC プロシージャによって作成されます。

列	データ型	NULL	説明
cube_owner	varchar2(240)		キューブの所有者。
cube_name	varchar2(240)		キューブの名前。
aggregation_name	varchar2(60)		集計仕様の名前。
measure_name	varchar2(240)		集計するメジャーの名前。

ALL_AW_CUBE_AGG_PLANS

ALL_AW_CUBE_AGG_PLANS は、キューブの集計仕様を表示します。

集計仕様は、アナリティック・ワークスペースにおけるキューブのディメンションに対するデータの集計方法を決定します。集計仕様は、DBMS_AWM.CREATE_AWCUBEAGG_SPEC プロシージャによって作成されます。

列	データ型	NULL	説明
owner	varchar2(240)		キューブの所有者。
cube_name	varchar2(240)		キューブの名前。
aggregation_name	varchar2(60)		集計仕様の名前。

ALL_AW_CUBE_ENABLED_HIERCOMBO

ALL_AW_CUBE_ENABLED_HIERCOMBO は、アナリティック・ワークスペースのキューブに関連付けられた階層の組合せを表示します。

階層の各組合せは一意的の数によって識別されます。OLAP API イネーブラは、階層の各組合せに対して別個のファクト・ビューを作成します。

このビューの情報は、リフレッシュ済のすべてのスタンダード・フォーム・キューブで使用可能です。

DBMS_AWM.REFRESH_AWCUBE プロシージャおよび DBMS_AWM.CREATE_AWCUBE_ACCESS プロシージャを参照してください。

列	データ型	NULL	説明
aw_owner	varchar2(30)		アナリティック・ワークスペースの所有者。
aw_name	varchar2(30)		アナリティック・ワークスペースの名前。
cube_name	varchar2(1000)		アナリティック・ワークスペースのキューブの名前。
hiercombo_num	number		階層の組合せを識別する一意の数。
hiercombo_str	varchar2(1000)		有効化キューブのファクト・ビューのディメンションを定義する階層のリスト。

ALL_AW_CUBE_ENABLED_VIEWS

ALL_AW_CUBE_ENABLED_VIEWS は、アナリティック・ワークスペースのキューブの生成可能なファクト・ビューを表示します。

ビューの説明は、キューブのリフレッシュ時に作成されます。ビューは、`DBMS_AWM.CREATE_AWCUBE_ACCESS` を実行して結果スクリプトが実行されるまでインスタンス化されません。

`ALL_AW_CUBE_ENABLED_VIEWS` は、ビューの説明を表示します。必ずしもビュー自体が存在しているわけではありません。

ファクト・ビューに関するメタデータは、`DBMS_AWM.REFRESH_AWCUBE` プロシージャによって生成されます。ワークスペース・キューブのビューを作成するスクリプトは、`DBMS_AWM.CREATE_AWCUBE_ACCESS` プロシージャによって作成されます。

列	データ型	NULL	説明
<code>aw_owner</code>	<code>varchar2(30)</code>		アナリティック・ワークスペースの所有者。
<code>aw_name</code>	<code>varchar2(30)</code>		アナリティック・ワークスペースの名前。
<code>cube_name</code>	<code>varchar2(1000)</code>		アナリティック・ワークスペースのキューブの名前。
<code>hiercombo_num</code>	<code>number</code>		階層の組合せを識別する一意の数。
<code>hiercombo_str</code>	<code>varchar2(1000)</code>		有効化キューブのファクト・ビューのディメンションを定義する階層のリスト。
<code>system_viewname</code>	<code>varchar2(1000)</code>		<code>DBMS_AWM.REFRESH_AWCUBE</code> プロシージャによって作成されるデフォルトのビュー名。
<code>user_viewname</code>	<code>varchar2(1000)</code>		<code>DBMS_AWM.SET_AWCUBE_VIEWNAME</code> プロシージャによって指定されるユーザー定義のビュー名。

ALL_AW_DIM_ENABLED_VIEWS

`ALL_AW_DIM_ENABLED_VIEWS` は、アナリティック・ワークスペースのディメンションの生成可能なディメンション・ビューを表示します。

ビューの説明は、ディメンションのリフレッシュ時に作成されます。ビューは、`DBMS_AWM.CREATE_AWDIMENSION_ACCESS` を実行して結果スクリプトが実行されるまでインスタンス化されません。

`ALL_AW_DIM_ENABLED_VIEWS` は、ビューの説明を表示します。必ずしもビュー自体が存在しているわけではありません。

ディメンション・ビューに関するメタデータは、`DBMS_AWM.REFRESH_AWDIMENSION` プロシージャによって生成されます。ワークスペース・ディメンションのビューを作成するスクリプトは、`DBMS_AWM.CREATE_AWDIMENSION_ACCESS` プロシージャによって作成されません。

列	データ型	NULL	説明
aw_owner	varchar2(30)		アナリティック・ワークスペースの所有者。
aw_name	varchar2(30)		アナリティック・ワークスペースの名前。
dimension_name	varchar2(1000)		アナリティック・ワークスペースのディメンションの名前。
hierarchy_name	varchar2(1000)		アナリティック・ワークスペースの階層の名前。
system_viewname	varchar2(1000)		DBMS_AWM.REFRESH_AWCUBE プロシージャによって作成されるデフォルトのビュー名。
user_viewname	varchar2(1000)		DBMS_AWM.SET_AWDIMENSION_VIEWNAME プロシージャによって指定されるユーザー定義のビュー名。

ALL_AW_LOAD_CUBES

ALL_AW_LOAD_CUBES は、キューブのロード仕様を表示します。

ロード仕様は、ソース・ファクト表からアナリティック・ワークスペースへデータをロードする方法を決定します。キューブのロード仕様は、DBMS_AWM.CREATE_AWCUBELOAD_SPEC プロシージャによって作成されます。

列	データ型	NULL	説明
cube_owner	varchar2(240)		OLAP カタログのソース・キューブの所有者。
cube_name	varchar2(240)		OLAP カタログのソース・キューブの名前。
load_name	varchar2(60)		キューブのロード仕様の名前。
load_type	varchar2(60)		'LOAD_DATA' -- ファクト表からアナリティック・ワークスペースのターゲット・キューブにデータをロードします。 'LOAD_PROGRAM' -- アナリティック・ワークスペースのロード・プログラムを作成します (実行はしません)。データをロードするには、プログラムを手動で実行します。キューブのロード・プログラム名は、アナリティック・ワークスペースのスタンダード・フォーム・キューブの AW\$LOADPGRGS プロパティに格納されます。

ALL_AW_LOAD_CUBE_DIMS

ALL_AW_LOAD_CUBE_DIMS は、キューブのコンポジット仕様を表示します。

コンポジット仕様は、アナリティック・ワークスペースにおけるキューブのディメンションの最適化方法を決定します。コンポジット仕様は、DBMS_AWM.CREATE_AWCOMP_SPEC プロシージャによって作成されます。

ALL_AW_LOAD_CUBE_FILTERS

列	データ型	NULL	説明
cube_owner	varchar2(240)		OLAP カタログのソース・キューブの所有者。
cube_name	varchar2(240)		OLAP カタログのソース・キューブの名前。
cubeload_name	varchar2(60)		キューブのロード仕様の名前。
compspec_name	varchar2(30)		このロード仕様に関連付けられたコンポジット仕様の名前。
composite_name	varchar2(30)		仕様のメンバーの1つであるコンポジットの名前。コンポジットにはキューブのスペース・ディメンションが含まれます。
segwidth	number		仕様のこのメンバーによってディメンション化されたデータの格納用セグメント幅。
compspec_position	number		仕様内のメンバーの位置。
dimension_owner	varchar2(30)		仕様のメンバーである OLAP カタログのソース・ディメンションの所有者。
dimension_name	varchar2(240)		仕様のメンバーである OLAP カタログのソース・ディメンションの名前。
composite_position	number		コンポジット・メンバー内のメンバーの位置。
nested_level	number		仕様のメンバーのネストのレベル。たとえば、稠密ディメンションは、1のネスト・レベルを持ちます。コンポジット内のスペース・ディメンションは2、ネストされたコンポジットは3のネスト・レベルを持ちます。
nested_type	varchar2(10)		仕様のメンバーのタイプ。DIMENSION または COMPOSITE のいずれかです。
nested_name	varchar2(30)		仕様のメンバーの名前。これは、ディメンションの名前かコンポジットの名前です。

ALL_AW_LOAD_CUBE_FILTERS

ALL_AW_LOAD_CUBE_FILTERS は、キューブのロード仕様に関連付けられたフィルタ条件を表示します。

フィルタ条件は SQL の WHERE 句で、ファクト表からアナリティック・ワークスペースへロードするデータのサブセットを識別します。

フィルタ条件は、DBMS_AWM.ADD_AWCUBELOAD_SPEC_FILTER プロシージャによって作成されます。

列	データ型	NULL	説明
owner	varchar2(240)		OLAP カタログのソース・キューブの所有者。
cube_name	varchar2(240)		OLAP カタログのソース・キューブの名前。
load_name	varchar2(60)		キューブのロード仕様の名前。
table_owner	varchar2(30)		ファクト表の所有者。
table_name	varchar2(30)		ファクト表の名前。
filter_condition	varchar2(4000)		SQL の WHERE 句。

ALL_AW_LOAD_CUBE_MEASURES

ALL_AW_LOAD_CUBE_MEASURES は、キューブ・ロード仕様のメジャーとスタンダード・フォームのアナリティック・ワークスペースの対応するターゲット・メジャーを表示します。

メジャーは、DBMS_AWM.ADD_AWCUBELOAD_SPEC_MEASURE プロシージャによってキューブのロード仕様に追加されます。このプロシージャを使用して、アナリティック・ワークスペースのメジャーのターゲット名および表示名を指定できます。このプロシージャをコールしないか、ターゲット名を指定しない場合、OLAP カタログの名前が使用されます。

列	データ型	NULL	説明
owner	varchar2(240)		OLAP カタログのソース・キューブの所有者。
cube_name	varchar2(240)		OLAP カタログのソース・キューブの名前。
load_name	varchar2(60)		ソース・キューブのロード仕様の名前。
measure_name	varchar2(240)		ソース・キューブのメジャーの名前。
measure_target_name	varchar2(60)		アナリティック・ワークスペースのメジャーの名前。
measure_target_display_name	varchar2(60)		アナリティック・ワークスペースのメジャーの表示名。これは OLAP カタログからの表示名か、ユーザー定義の名前です。
measure_target_description	varchar2(4000)		アナリティック・ワークスペースのメジャーの説明。これは OLAP カタログからの説明か、ユーザー定義の説明です。

ALL_AW_LOAD_CUBE_PARMS

ALL_AW_LOAD_CUBE_PARMS は、キューブのロード仕様のパラメータを表示します。

キューブのロード仕様は、ファクト表からアナリティック・ワークスペースへキューブのデータをロードする方法を決定します。

キューブのロード仕様のパラメータは、DBMS_AWM.SET_AWCUBELOAD_SPEC_PARAMETER プロシージャによって設定されます。

列	データ型	NULL	説明
owner	varchar2(240)		OLAP カタログのソース・キューブの所有者。
cube_name	varchar2(240)		OLAP カタログのソース・キューブの名前。
load_name	varchar2(60)		ソース・キューブのロード仕様の名前。
parm_name	varchar2(16)		パラメータの名前。'DISPLAY NAME'のみが現在使用可能です。このパラメータを設定しない場合、OLAP カタログのキューブの表示名がアナリティック・ワークスペースで使用されます。
parm_value	varchar2(30)		アナリティック・ワークスペースのターゲット・キューブで使用される表示名。

ALL_AW_LOAD_DIMENSIONS

ALL_AW_LOAD_DIMENSIONS は、ディメンションのロード仕様を表示します。

ディメンションのロード仕様は、DBMS_AWM.CREATE_AWDIMLOAD_SPEC プロシージャによって作成されます。

列	データ型	NULL	説明
owner	varchar2(30)		OLAP カタログのソース・ディメンションの所有者。
dimension_name	varchar2(30)		OLAP カタログのソース・ディメンションの名前。
load_name	varchar2(60)		ロード仕様の名前。
load_type	varchar2(60)		'FULL_LOAD_ADDITIONS_ONLY' -- 新しいディメンションのメンバーのみが、ディメンションのリフレッシュ時にロードされます。(デフォルト) 'FULL_LOAD' -- ディメンションのリフレッシュ時にワークスペースのすべてのディメンションのメンバーが削除され、ソース・ディメンションのすべてのメンバーがロードされます。

ALL_AW_LOAD_DIM_FILTERS

ALL_AW_LOAD_DIM_FILTERS は、ディメンションのロード仕様に関連付けられたフィルタ条件を表示します。

フィルタ条件は SQL の WHERE 句で、ディメンション表からアナリティック・ワークスペースへロードするデータのサブセットを識別します。

フィルタ条件は、DBMS_AWM.ADD_AWDIMLOAD_SPEC_FILTER プロシージャによって作成されます。

列	データ型	NULL	説明
owner	varchar2(30)		OLAP カタログのソース・ディメンションの所有者。
dimension_name	varchar2(30)		OLAP カタログのソース・ディメンションの名前。
load_name	varchar2(60)		ディメンションのロード仕様の名前。
table_owner	varchar2(30)		ディメンション表の所有者。
table_name	varchar2(30)		ディメンション表の名前。
filter_condition	varchar2(4000)		SQL の WHERE 句。

ALL_AW_LOAD_DIM_PARMS

ALL_AW_LOAD_DIM_PARMS は、ディメンションのロード仕様のパラメータを表示します。

ディメンションのロード仕様は、ディメンション表からアナリティック・ワークスペースへディメンションのメンバーをロードする方法を決定します。

ディメンションのロード仕様のパラメータは、DBMS_AWM.SET_AWDIMLOAD_SPEC_PARAMETER プロシージャによって設定されます。

列	データ型	NULL	説明
owner	varchar2(30)		OLAP カタログのソース・ディメンションの所有者。
dimension_name	varchar2(30)		OLAP カタログのソース・ディメンションの名前。
load_name	varchar2(60)		ディメンションのロード仕様の名前。

列	データ型	NULL	説明
parm_name	varchar2(16)		<p>'UNIQUE_RDBMS_KEY' -- このディメンションのメンバーが、ソース表のすべてのレベルで一意かどうか。</p> <p>'DISPLAY_NAME' -- アナリティック・ワークスペースのターゲット・ディメンションの表示名。</p> <p>'PLURAL_DISPLAY_NAME' -- アナリティック・ワークスペースのターゲット・ディメンションの複数表示名。</p>
parm_value	varchar2(4000)		<p>UNIQUE_RDBMS_KEY の値: NO -- ディメンションのメンバー名が RDBMS 表のレベルで一意でない。アナリティック・ワークスペースの対応するディメンションのメンバー名に、接頭辞としてレベル名を含みません。(デフォルト) YES -- ディメンションのメンバー名が RDBMS 表のレベルで一意。アナリティック・ワークスペースの対応するディメンションのメンバー名に、ソース・リレーショナル・ディメンション内のものと同じ名前を含みます。</p> <p>DISPLAY_NAME の値は、アナリティック・ワークスペースのターゲット・ディメンションの表示名です。</p> <p>PLURAL_DISPLAY_NAME の値は、アナリティック・ワークスペースのターゲット・ディメンションの複数表示名です。</p>

OLAP カタログ・メタデータのビュー

この章では、OLAP カタログ・メタデータのビューについて説明します。すべての OLAP メタデータは、CWM2 PL/SQL パッケージまたは Oracle Enterprise Manager のいずれを使用して作成した場合でも、これらのビューに表示されます。

参照： 第 2 章「CWM2 による OLAP カタログ・メタデータの作成」を参照してください。

注意： 2 番目のビューのセット (OLAP API Metadata Reader のビュー) には、OLAP カタログのビューと同様の情報が表示されます。Metadata Reader ビューは、OLAP API による高速問合せを簡単に実行できるように構成されています。詳細は、第 15 章を参照してください。

この章では、次の項目について説明します。

- OLAP カタログのビューへのアクセス
- 多次元モデルのビュー
- マッピング情報のビュー

OLAP カタログのビューへのアクセス

OLAP カタログの読み込み API は、次の 2 つのセットの対応するビューで構成されます。

- 現行のユーザーがアクセスできるすべての有効な OLAP メタデータを表示する ALL_ ビュー。
- データベース全体のすべての OLAP メタデータ (有効および無効の両方) を表示する DBA_ ビュー。DBA_ ビューは、管理者のみを対象としています。

注意： OLAP カタログの表は OLAPSYS が所有します。これらの表に OLAP メタデータを作成するには、OLAP_DBA ロールが必要です。

ALL_ ビューと DBA_ ビューの列は同一です。この章では、ALL_ ビューのみを示します。

多次元モデルのビュー

次のビューには、OLAP メタデータの基本的な多次元モデルが表示されます。

論理モデルの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

表 5-1 OLAP カタログの多次元モデルのビュー

ビュー名のシノニム	説明
ALL_OLAP2_CATALOGS	Oracle インスタンス内のすべてのメジャー・フォルダ（カタログ）を表示します。
ALL_OLAP2_CATALOG_ENTITY_USES	各メジャー・フォルダ内のメジャーを表示します。
ALL_OLAP2_CUBES	Oracle インスタンスのすべてのキューブを表示します。
ALL_OLAP2_CUBE_DIM_USES	各キューブ内のディメンションを表示します。
ALL_OLAP2_CUBE_MEASURES	各キューブ内のメジャーを表示します。
ALL_OLAP2_CUBE_MEAS_DIM_USES	各メジャーをディメンションごとに集計する方法を表示します。
ALL_OLAP2_DIMENSIONS	Oracle インスタンスのすべての OLAP ディメンションを表示します。
ALL_OLAP2_DIM_ATTRIBUTES	各ディメンション内のディメンション属性を表示します。
ALL_OLAP2_DIM_ATTR_USES	レベル属性と各ディメンション属性の関係を表示します。
ALL_OLAP2_DIM_HIERARCHIES	各ディメンション内の階層を表示します。
ALL_OLAP2_DIM_HIER_LEVEL_USES	各階層内のレベルの順序付けを表示します。
ALL_OLAP2_DIM_LEVELS	各ディメンション内のレベルを表示します。
ALL_OLAP2_DIM_LEVEL_ATTRIBUTES	各レベル内のレベル属性を表示します。
ALL_OLAP2_ENTITY_DESC_USES	アプリケーション固有の意味を持つ予約済の属性を表示します。たとえば、詳細および簡単な説明と時系列計算（最終日、期間、以前の時間間隔など）に使用するディメンション属性などです。

マッピング情報のビュー

次のビューには、基本的な多次元モデルをリレーショナル表またはビューにマップする方法が表示されます。

表 5-2 OLAP カタログのマッピングのビュー

ビュー名のシノニム	説明
ALL_OLAP2_CUBE_MEASURE_MAPS	列への各メジャーのマッピングを表示します。
ALL_OLAP2_DIM_LEVEL_ATTR_MAPS	列への各レベル属性のマッピングを表示します。
ALL_OLAP2_FACT_LEVEL_USES	スター・スキーマまたはスノーフレーク・スキーマ内のディメンション表とファクト表の結合を表示します。
ALL_OLAP2_FACT_TABLE_GID	各ファクト表内の各階層のグルーピング ID 列を表示します。
ALL_OLAP2_HIER_CUSTOM_SORT	階層内のレベル列のデフォルトのソート順序を表示します。
ALL_OLAP2_JOIN_KEY_COLUMN_USES	階層内の 2 つのレベルの結合を表示します。
ALL_OLAP2_LEVEL_KEY_COL_USES	一意のキー列への各レベルのマッピングを表示します。

ALL_OLAP2_AGGREGATION_USES

ALL_OLAP2_AGGREGATION_USES は、スター・スキーマまたはスノーフレーク・スキーマとして構成されるリレーショナル表にマップされるキューブに関連付けられた集計演算子を表示します。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	キューブの所有者。
CUBE_NAME	VARCHAR2 (30)	NOT NULL	キューブの名前。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	キューブのディメンションの名前。
HIERARCHY_NAME	VARCHAR2 (30)		キューブのディメンションの階層の名前。
DIM_HIER_COMBO_ID	NUMBER	NOT NULL	キューブ内の階層組合せの識別子。
AGGREGATION_NAME	VARCHAR2 (240)		このディメンションの集計演算子の名前。1-21 ページの表 1-10 「集計演算子」を参照。
AGGREGATION_ORDER	NUMBER		集計演算子の優先順位。
TABLE_OWNER	VARCHAR2 (30)		加重演算子の WEIGHTBY 要素を含む表の所有者。演算子が加重されていない場合、この列は NULL です。

ALL_OLAP2_CATALOGS

列	データ型	NULL	説明
TABLE_NAME	VARCHAR2 (30)		加重演算子の WEIGHTBY 要素を含む表の名前。演算子が加重されていない場合、この列は NULL です。
COLUMN_NAME	VARCHAR2 (30)		加重演算子の WEIGHTBY 要素を含む列の名前。演算子が加重されていない場合、この列は NULL です。

ALL_OLAP2_CATALOGS

ALL_OLAP2_CATALOGS は、Oracle インスタンス内のすべてのメジャー・フォルダ（カタログ）を表示します。

列	データ型	NULL	説明
CATALOG_ID	NUMBER	NOT NULL	メジャー・フォルダの ID。
CATALOG_NAME	VARCHAR2 (30)	NOT NULL	メジャー・フォルダの名前。
PARENT_CATALOG_ID	NUMBER		親メジャー・フォルダの ID。この列は、メジャー・フォルダ・ツリーのルートでは NULL です。
DESCRIPTION	VARCHAR2 (2000)		メジャー・フォルダの説明。

ALL_OLAP2_CATALOG_ENTITY_USES

ALL_OLAP2_CATALOG_ENTITY_USES は、各メジャー・フォルダ内のメジャーを表示します。

列	データ型	NULL	説明
CATALOG_ID	NUMBER	NOT NULL	メジャー・フォルダの ID。
ENTITY_OWNER	VARCHAR2 (30)	NOT NULL	メジャーのキューブの所有者。
ENTITY_NAME	VARCHAR2 (30)	NOT NULL	メジャーのキューブの名前。
CHILD_ENTITY_NAME	VARCHAR2 (30)	NOT NULL	メジャー・フォルダのメジャーの名前。

ALL_OLAP2_CUBES

ALL_OLAP2_CUBES は、Oracle インスタンスのすべてのキューブを表示します。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	メジャーを含むキューブの所有者。
CUBE_NAME	VARCHAR2 (30)	NOT NULL	メジャーを含むキューブの名前。
INVALID	VARCHAR2 (2)	NOT NULL	このキューブが有効か無効か。2-12 ページの「 OLAP メタデータの検証およびコミット 」を参照。
DISPLAY_NAME	VARCHAR2 (30)		キューブの表示名。
DESCRIPTION	VARCHAR2 (2000)		キューブの説明。
MV_SUMMARYCODE	VARCHAR2 (2)		このキューブに、関連付けられたマテリアライズド・ビューが存在する場合、キューブがグルーピング・セット (groupingset) 形式か、ロールアップ (rollup) 形式かは、マテリアライズド・ビューのサマリー・コードによって指定されます。 第 23 章「 DBMS_ODM 」を参照。

ALL_OLAP2_CUBE_DIM_USES

ALL_OLAP2_CUBE_DIM_USES は、各キューブ内のディメンションを表示します。

ディメンションは、同じキューブに複数回関連付けることができますが、関連付けごとに、一意のディメンションの別名を使用して個別の行に指定します。

列	データ型	NULL	説明
CUBE_DIMENSION_USE_ID	NUMBER	NOT NULL	キューブとディメンション間の関連付けの ID。
OWNER	VARCHAR2 (30)	NOT NULL	キューブの所有者。
CUBE_NAME	VARCHAR2 (30)	NOT NULL	キューブの名前。
DIMENSION_OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
DIMENSION_ALIAS	VARCHAR2 (30)		ディメンションの別名 (キューブ内でディメンションを使用する場合に一意の ID を指定するために使用)。

列	データ型	NULL	説明
DEFAULT_CALC_HIERARCHY_NAME	VARCHAR2 (30)		デフォルトの階層 (ディメンション内をドリルダウンまたはドリルアップするために使用)。
DEPENDENT_ON_DIM_USE_ID	NUMBER		このキューブ / ディメンションの関連付けの基になるキューブ / ディメンションの ID。

ALL_OLAP2_CUBE_MEASURES

ALL_OLAP2_CUBE_MEASURES は、各キューブ内のメジャーを表示します。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	メジャーを含むキューブの所有者。
CUBE_NAME	VARCHAR2 (30)	NOT NULL	メジャーを含むキューブの名前。
MEASURE_NAME	VARCHAR2 (30)	NOT NULL	メジャーの名前。
DISPLAY_NAME	VARCHAR2 (30)		メジャーの表示名。
DESCRIPTION	VARCHAR2 (2000)		メジャーの説明。

ALL_OLAP2_CUBE_MEASURE_MAPS

ALL_OLAP2_CUBE_MEASURE_MAPS は、列への各メジャーのマッピングを表示します。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	キューブの所有者。
CUBE_NAME	VARCHAR2 (30)	NOT NULL	キューブの名前。
MEASURE_NAME	VARCHAR2 (30)	NOT NULL	このキューブに含まれているメジャーの名前。
DIM_HIER_COMBO_ID	NUMBER	NOT NULL	このメジャーとメジャーのディメンション階層の組合せとの間の関係付けの ID。
FACT_TABLE_OWNER	VARCHAR2 (30)	NOT NULL	ファクト表の所有者。
FACT_TABLE_NAME	VARCHAR2 (30)	NOT NULL	ファクト表の名前。
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	このメジャーのデータが格納されるファクト表の列の名前。

ALL_OLAP2_CUBE_MEAS_DIM_USES

ALL_OLAP2_CUBE_MEAS_DIM_USES は、各メジャーをディメンションごとに集計する方法を表示します。デフォルトの集計メソッドは加算です。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	このメジャーを含むキューブの所有者。
CUBE_NAME	VARCHAR2 (30)	NOT NULL	このメジャーを含むキューブの名前。
MEASURE_NAME	VARCHAR2 (30)	NOT NULL	メジャーの名前。
DIMENSION_OWNER	VARCHAR2 (30)	NOT NULL	このメジャーに関連付けられたディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
DIMENSION_ALIAS	VARCHAR2 (30)		ディメンションの別名。
DEFAULT_AGGR_FUNCTION_USE_ID	NUMBER		ディメンションに対してこのメジャーのデータを集計する場合に使用するデフォルトの集計メソッド。この列が NULL の場合、集計メソッドは加算です。

ALL_OLAP2_DIMENSIONS

ALL_OLAP2_DIMENSIONS は、Oracle インスタンスのすべての OLAP ディメンションを表示します。

CWM2 API で作成された OLAP ディメンションは、データベースのディメンション・オブジェクトとは関連付けられていません。Oracle Enterprise Manager で作成された OLAP ディメンションは、データベースのディメンション・オブジェクトに基づいています。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
PLURAL_NAME	VARCHAR2 (30)		ディメンションの複数名。表示に使用されます。
DISPLAY_NAME	VARCHAR2 (30)		ディメンションの表示名。
DESCRIPTION	VARCHAR2 (2000)		ディメンションの説明。
DEFAULT_DISPLAY_HIERARCHY	VARCHAR2 (30)	NOT NULL	ディメンションのデフォルトの表示階層。
INVALID	VARCHAR2 (1)	NOT NULL	ディメンションが有効か無効か。2-12 ページの「 OLAP メタデータの検証およびコミット 」を参照。

列	データ型	NULL	説明
DIMENSION_TYPE	VARCHAR2 (10)		使用せず。

ALL_OLAP2_DIM_ATTRIBUTES

ALL_OLAP2_DIM_ATTRIBUTES は、各ディメンション内のディメンション属性を表示します。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
ATTRIBUTE_NAME	VARCHAR2 (30)	NOT NULL	ディメンション属性の名前。
DISPLAY_NAME	VARCHAR2 (30)		ディメンション属性の表示名。
DESCRIPTION	VARCHAR2 (2000)		ディメンション属性の説明。
DESC_ID	NUMBER		属性が予約済の場合、その型がこの列に表示されます。予約済のディメンション属性の例は、詳細および簡単な説明と時間に関連する属性（最終日、期間、以前の時間間隔など）です。

ALL_OLAP2_DIM_ATTR_USES

ALL_OLAP2_DIM_ATTR_USES は、レベル属性を各ディメンション属性に関連付ける方法を表示します。

同じレベル属性を複数のディメンション属性に含めることができます。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
DIM_ATTRIBUTE_NAME	VARCHAR2 (30)	NOT NULL	ディメンション属性の名前。
LEVEL_NAME	VARCHAR2 (30)	NOT NULL	ディメンション内のレベルの名前。
LVL_ATTRIBUTE_NAME	VARCHAR2 (30)	NOT NULL	このレベル属性の名前。このレベル属性は、ディメンション属性に含まれます。

ALL_OLAP2_DIM_HIERARCHIES

ALL_OLAP2_DIM_HIERARCHIES は、各ディメンション内の階層を表示します。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
HIERARCHY_NAME	VARCHAR2 (30)	NOT NULL	階層の名前。
DISPLAY_NAME	VARCHAR2 (30)		階層の表示名。
DESCRIPTION	VARCHAR2 (2000)		階層の説明。
SOLVED_CODE	VARCHAR2 (2)	NOT NULL	解決済コードは、次のいずれかになります。 UNSOLVED LEVEL-BASED は、埋込み合計を含まず、レベルベース・ディメンション表に格納される階層に使用します。 SOLVED LEVEL-BASED は、埋込み合計を含み、レベルベース・ディメンション表に格納される、グルーピングIDを持つ階層に使用します。 SOLVED VALUE-BASED は、埋込み合計を含み、親子ディメンション表に格納される階層に使用します。 異なる解決済コードでの階層のマッピングについては、2-11 ページの「 ファクト表とディメンション表の結合 」を参照してください。

ALL_OLAP2_DIM_HIER_LEVEL_USES

ALL_OLAP2_DIM_HIER_LEVEL_USES は、レベルを各階層内で順序付ける方法を表示します。

個別の階層内では、同じ親レベルが異なる子レベルに階層的に関連する場合があります。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
HIERARCHY_NAME	VARCHAR2 (30)	NOT NULL	階層の名前。
PARENT_LEVEL_NAME	VARCHAR2 (30)	NOT NULL	親レベルの名前。
CHILD_LEVEL_NAME	VARCHAR2 (30)	NOT NULL	子レベルの名前。

列	データ型	NULL	説明
POSITION	NUMBER	NOT NULL	階層内のこの親子関係の位置 (1 が最も詳細)。

ALL_OLAP2_DIM_LEVELS

ALL_OLAP2_DIM_LEVELS は、各ディメンション内のレベルを表示します。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	このレベルを含むディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	このレベルを含むディメンションの名前。
LEVEL_NAME	VARCHAR2 (30)	NOT NULL	レベルの名前。
DISPLAY_NAME	VARCHAR2 (30)		レベルの表示名。
DESCRIPTION	VARCHAR2 (2000)		レベルの説明。
LEVEL_TABLE_OWNER	VARCHAR2 (30)	NOT NULL	このレベルの列を含むディメンション表の所有者。
LEVEL_TABLE_NAME	VARCHAR2 (30)	NOT NULL	このレベルの列を含むディメンション表の名前。

ALL_OLAP2_DIM_LEVEL_ATTRIBUTES

ALL_OLAP2_DIM_LEVEL_ATTRIBUTES は、各レベル内のレベル属性を表示します。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	レベルを含むディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	レベルを含むディメンションの名前。
ATTRIBUTE_NAME	VARCHAR2 (30)		レベル属性の名前。属性名が指定されていない場合、列名が使用されます。
DISPLAY_NAME	VARCHAR2 (30)		レベル属性の表示名。
DESCRIPTION	VARCHAR2 (2000)		レベル属性の説明。
DETERMINED_BY_LEVEL_NAME	VARCHAR2 (30)	NOT NULL	レベルの名前。

ALL_OLAP2_DIM_LEVEL_ATTR_MAPS

ALL_OLAP2_DIM_LEVEL_ATTR_MAPS は、列への各レベル属性のマッピングを表示します。レベルへのレベル属性のマッピングは階層によって異なります。同じレベルを異なる階層で使用すると、同じレベルに異なる属性が含まれる場合があります。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
HIERARCHY_NAME	VARCHAR2 (30)		このレベルを含む階層の名前。
ATTRIBUTE_NAME	VARCHAR2 (30)		このレベル属性を含むディメンション属性のグルーピングの名前。
LVL_ATTRIBUTE_NAME	VARCHAR2 (30)	NOT NULL	レベル属性の名前（レベル属性の名前が指定されていない場合は列名）。
LEVEL_NAME	VARCHAR2 (30)	NOT NULL	レベルの名前。
TABLE_OWNER	VARCHAR2 (30)	NOT NULL	レベルおよびレベル属性を含むディメンション表の所有者。
TABLE_NAME	VARCHAR2 (30)	NOT NULL	レベル列およびレベル属性列を含むディメンション表の名前。
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	レベル属性を含む列の名前。
DTYPE	VARCHAR2 (10)	NOT NULL	レベル属性を含む列のデータ型。

ALL_OLAP2_ENTITY_DESC_USES

ALL_OLAP2_ENTITY_DESC_USES は、予約済の属性を表示し、ディメンションが時間ディメンションかどうかを示します。

列	データ型	NULL	説明
DESCRIPTOR_ID	NUMBER	NOT NULL	予約済の属性またはディメンション・タイプの名前。 10-2 ページの表 10-1 「予約済のディメンション属性」に、予約済のディメンション属性を示します。 13-2 ページの表 13-1 「予約済のレベル属性」に、予約済のレベル属性を示します。
ENTITY_OWNER	VARCHAR2 (30)	NOT NULL	メタデータのエンティティの所有者。
ENTITY_NAME	VARCHAR2 (30)	NOT NULL	メタデータのエンティティの名前。

ALL_OLAP2_FACT_LEVEL_USES

列	データ型	NULL	説明
CHILD_ENTITY_NAME	VARCHAR2 (30)		子エンティティの名前 (存在する場合)。ディメンション属性は、ディメンションの子エンティティです。レベル属性は、ディメンション属性の子エンティティです。
SECONDARY_CHILD_ENTITY_NAME	VARCHAR2 (30)		2番目の子エンティティの名前 (存在する場合)。ディメンション属性は、ディメンションの子エンティティです。レベル属性は、ディメンション属性の子エンティティです。レベル属性は、ディメンションの2番目の子エンティティである場合があります。

ALL_OLAP2_FACT_LEVEL_USES

ALL_OLAP2_FACT_LEVEL_USES は、スター・スキーマまたはスノーflake・スキーマ内のディメンション表とファクト表の結合を表示します。詳細は、2-11 ページの「[ファクト表とディメンション表の結合](#)」を参照してください。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	キューブの所有者。
CUBE_NAME	VARCHAR2 (30)	NOT NULL	キューブの名前。
DIMENSION_OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	NUMBER	NOT NULL	ディメンションの名前。
DIMENSION_ALIAS	VARCHAR2 (30)		ディメンションの別名 (存在する場合)。
HIERARCHY_NAME		NOT NULL	階層の名前。
DIM_HIER_COMBO_ID	NUMBER	NOT NULL	ファクト表に関連付けられたディメンション階層の組合せの ID。
LEVEL_NAME	VARCHAR2 (30)		マッピングが行われる階層内のレベルの名前。
FACT_TABLE_OWNER	VARCHAR2 (30)	NOT NULL	ファクト表の所有者。
FACT_TABLE_NAME	VARCHAR2 (30)	NOT NULL	ファクト表の名前。
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	ファクト表の外部キー列の名前。
POSITION	NUMBER		複数列からなるキー内のこの列の位置。

列	データ型	NULL	説明
DIMENSION_ KEYMAP_TYPE	VARCHAR2 (30)	NOT NULL	ファクト表のキー・マッピングのタイプ。値は次のとおりです。 LL (最低レベル) は、最低レベルのディメンション・メンバーがキー列に格納される場合に指定します。ファクト表は、未解決です。 ET (埋込み合計) は、すべてのレベルの組合せに対するディメンション・メンバーがキー列に格納される場合に指定します。ファクト表は解決済です (すべてのレベルの組合せに対する埋込み合計を含みます)。
FOREIGN_KEY_NAME	VARCHAR2 (30)		外部キー列に適用される外部キー制約の名前。制約は、CWM2 API では使用されません。

ALL_OLAP2_FACT_TABLE_GID

ALL_OLAP2_FACT_TABLE_GID は、各ファクト表の各階層のグルーピング ID 列を表示します。詳細は、1-28 ページの「グルーピング ID 列」を参照してください。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	キューブの所有者。
CUBE_NAME	VARCHAR2 (30)	NOT NULL	キューブの名前。
DIMENSION_OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
HIERARCHY_NAME	VARCHAR2 (30)	NOT NULL	階層の名前。
DIM_HIER_COMBO_ID	NUMBER	NOT NULL	ディメンションと階層の関連付けの ID。
FACT_TABLE_OWNER	VARCHAR2 (30)	NOT NULL	ファクト表の所有者。
FACT_TABLE_NAME	VARCHAR2 (30)	NOT NULL	ファクト表の名前。
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	GID 列の名前。

ALL_OLAP2_HIER_CUSTOM_SORT

ALL_OLAP2_HIER_CUSTOM_SORT は、階層内のレベル列のソート順序を表示します。カスタム・ソート情報はオプションです。

カスタム・ソート情報は、ディメンション表の列に基づいて階層のメンバーをソートする方法を指定します。ディメンション表の特定の列は、キー列と同じであるか、関連する属性列です。

ALL_OLAP2_JOIN_KEY_COLUMN_USES

カスタム・ソートでは、列を昇順または降順のどちらでソートするか、また、その場合に NULL を先頭に置くか最後に置くかを指定できます。カスタム・ソートは、ディメンションの複数のレベルで適用できます。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
HIERARCHY_NAME	VARCHAR2 (30)	NOT NULL	階層の名前。
TABLE_OWNER	VARCHAR2 (30)	NOT NULL	ディメンション表の所有者。
TABLE_NAME	VARCHAR2 (30)	NOT NULL	ディメンション表の名前。
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	格納される列の名前。
POSITION	NUMBER	NOT NULL	複数列からなる SORT_POSITION 内の位置を表示します。ほとんどの場合、単一系列には、SORT_POSITION が表示されます。POSITION の値は 1 です。
SORT_POSITION	NUMBER	NOT NULL	ソートするレベルのソート順序内の位置。
SORT_ORDER	VARCHAR2 (4)	NOT NULL	ソート順序。Ascending または Descending のいずれかです。
NULL_ORDER	VARCHAR2 (5)	NOT NULL	ソート順序に NULL 値を挿入する位置。Nulls First または Nulls Last のいずれかです。

ALL_OLAP2_JOIN_KEY_COLUMN_USES

ALL_OLAP2_JOIN_KEY_COLUMN_USES は、階層内の 2 つのレベルの結合を表示します。これらの結合は、スノーフレイク・スキーマ内のディメンション表間およびスター・スキーマ内のレベル列間に存在します。

レベルが複数の列にマップされる場合、各列のマッピングはビューの個別の行に表示されません。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
HIERARCHY_NAME	VARCHAR2 (30)	NOT NULL	階層の名前。
CHILD_LEVEL_NAME	VARCHAR2 (30)	NOT NULL	階層の子レベル。
TABLE_OWNER	VARCHAR2 (30)	NOT NULL	ディメンション表の所有者。

列	データ型	NULL	説明
TABLE_NAME	VARCHAR2 (30)	NOT NULL	ディメンション表の名前。
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	ディメンション表の子レベル列の名前。スター・スキーマでは、CHILD_LEVEL_NAMEに関連付けられた列です。スノーフレーク・スキーマでは、同じディメンション表のCHILD_LEVEL_NAMEの親列です。
POSITION	NUMBER		キー内の列の位置。複数列からなるキー（レベルが複数の列にマップされている）のみに適用されます。
JOIN_KEY_TYPE	VARCHAR2 (30)	NOT NULL	結合キーが論理外部キーの場合、このキーはスノーフレーク型です。結合キーが同じ表内の列を参照する場合、このキーはスター型です。

ALL_OLAP2_LEVEL_KEY_COL_USES

ALL_OLAP2_LEVEL_KEY_COL_USES は、一意キー列への各レベルのマッピングを表示します。

レベルが複数の列にマップされる場合、各列のマッピングはビューの個別の行に表示されません。

列	データ型	NULL	説明
OWNER	VARCHAR2 (30)	NOT NULL	ディメンションの所有者。
DIMENSION_NAME	VARCHAR2 (30)	NOT NULL	ディメンションの名前。
HIERARCHY_NAME	VARCHAR2 (30)		このレベルを含む階層の名前。
CHILD_LEVEL_NAME	VARCHAR2 (30)	NOT NULL	レベルの名前。
TABLE_OWNER	VARCHAR2 (30)	NOT NULL	ディメンション表の所有者。
TABLE_NAME	VARCHAR2 (30)	NOT NULL	ディメンション表の名前。
COLUMN_NAME	VARCHAR2 (30)	NOT NULL	CHILD_LEVEL_NAME が格納される列の名前。
POSITION	NUMBER		キー内の列の位置。複数列からなるキー（レベルが複数の列にマップされている）のみに適用されます。

OLAP 動的パフォーマンス・ビュー

Oracle では、パフォーマンスの統計を固定表に収集し、これらの表から、ユーザーがアクセスできるビューを作成します。この章では、Oracle OLAP に関するパフォーマンス・データを含むビューについて説明します。

参照： 動的パフォーマンス表およびビューの詳細は、次のマニュアルを参照してください。

- 『Oracle10i Database Reference』
- 『Oracle10i Database Performance Tuning Guide』

この章では、次の項目について説明します。

- [OLAP パフォーマンス・ビューが参照するシステム表](#)
- [OLAP パフォーマンス・ビューの概要](#)
- [V\\$AW_CALC](#)
- [V\\$AW_OLAP](#)
- [V\\$AW_SESSION_INFO](#)

OLAP パフォーマンス・ビューが参照するシステム表

各 Oracle データベース・インスタンスは、現行のデータベース・アクティビティを記録する一連の仮想的な表を保持します。これらの表を**動的パフォーマンス表**と呼びます。

動的パフォーマンス表は、内部ディスク構造およびメモリー構造のデータを収集します。動的パフォーマンス表は、データベースの使用中、連続的に更新されます。その中には、Oracle OLAP のデータを収集する表もあります。

OLAP 動的パフォーマンス表の名前は、V\$AW で始まります。動的パフォーマンス表は SYS ユーザーが所有します。また、SELECT CATALOG ロールを持つユーザーであれば、この表にアクセスできます。

システムによって、これらの表からビューが作成され、これらのビューに対してパブリック・シノニムが作成されます。データベース管理者がビューを変更または削除することはできないため、これらのビューは**固定ビュー**と呼ばれる場合もあります。また、シノニム名はV\$AWで始まります。ビューもSYSが所有しますが、DBAがこれらのビューへのアクセス権をさらに広範囲なユーザーに付与することもできます。

次のSQL*Plusセッションの例では、OLAPシステム表のリストを示します。

```
% sqlplus '/ as sysdba'
.
.
.
SQL> SELECT name FROM v$fixed_table WHERE name LIKE 'V$AW%';

NAME
-----
V$AW_OLAP
V$AW_CALC
V$AW_SESSION_INFO
```

OLAP パフォーマンス・ビューの概要

表 6-1 に、各 OLAP パフォーマンス・ビューの概要を示します。

表 6-1 OLAP パフォーマンス・ビュー

固定ビュー	説明
V\$AW_CALC	キャッシュ領域の使用状況に関する情報を収集します。
V\$AW_OLAP	アクティブなアナリティック・ワークスペースの状態に関する情報を収集します。
V\$AW_SESSION_INFO	各アクティブ・セッションに関する情報を収集します。

V\$AW_CALC

V\$AW_CALC は、Oracle OLAP が使用する様々なキャッシュの効果をレポートします。OLAP 問合せは繰り返し実行される傾向があるため、通常、1つのセッション中に同じデータが繰り返し問い合わせされます。データが問合せごとに再計算される場合に比べ、キャッシュを使用することで、セッションですでに計算されたデータへのアクセスが大幅に高速になります。

キャッシュによる効果が向上すると、ユーザーへの応答時間も短縮されます。効果の少ないキャッシュ（つまり、ほとんどヒットせず、ミスが多いキャッシュ）の場合、データは、表示される場合とは異なる形式で保存されている場合があります。実行時のパフォーマンスを改善するには、変数のディメンションを並べ替える必要があります（つまり、最初に変化するディメンションと、最後に変化するディメンションを入れ替える必要があります）。

Oracle OLAP は、次のキャッシュを使用します。

- **集計キャッシュ。** OLAP DML の AGGREGATE ファンクションが使用するオプションのキャッシュ。AGGREGATE ファンクションは、問合せへの応答で実行時に集計データを計算します。キャッシュが保持されると、AGGREGATE は、データが問い合わせられるたびに再計算するのではなく、セッション中に事前に計算されたデータを取得することができます。
- **セッション・キャッシュ。** Oracle OLAP はセッションごとにキャッシュを保持し、計算結果を格納します。セッションが終了すると、キャッシュの内容は廃棄されます。
- **ページ・プール。** データベースのプログラム・グローバル領域 (PGA) から割り当てられたキャッシュ。Oracle OLAP がセッションに対して保持します。ページ・プールは特定のセッションに関連付けられ、アタッチされたすべてのアナリティック・ワークスペースによって共有されます。ページ・プールが一杯になると、Oracle OLAP はそのページの一部をデータベース・キャッシュに書き込みます。OLAP DML で UPDATE コマンドが発行されると、そのアナリティック・ワークスペースに関連付けられた変更済ページが永続 LOB に書き込まれます。その場合、ディスクにデータをストリームするためのステージ領域として、一時セグメントが使用されます。ページ・プールのサイズは、OLAP_PAGE_POOL 初期化パラメータによって制御されます。
- **データベース・キャッシュ。** データベース・インスタンス用に Oracle RDBMS が保持する大容量のキャッシュ。

列	データ型	説明
AGGREGATE_CACHE_HITS	NUMBER	集計キャッシュでディメンション・メンバーが検索された回数 (ヒット)。 実行時集計でのヒット数は、稠密ディメンション間のデータ・フェッチによって増加する場合があります。
AGGREGATE_CACHE_MISSES	NUMBER	集計キャッシュでディメンション・メンバーが検索されず、ディスクからの読み込みが必要になった回数 (ミス)。
SESSION_CACHE_HITS	NUMBER	セッション・キャッシュでデータが検索された回数 (ヒット)。
SESSION_CACHE_MISSES	NUMBER	セッション・キャッシュでデータが検索されなかった回数 (ミス)。
POOL_HITS	NUMBER	OLAP ページ・プールのページでデータが検索された回数 (ヒット)。
POOL_MISSES	NUMBER	OLAP ページ・プールでデータが検索されなかった回数 (ミス)。
POOL_NEW_PAGES	NUMBER	アナリティック・ワークスペース LOB にはまだ書き込まれていない、OLAP ページ・プールで新しく作成されたページ数。
POOL_RECLAIMED_PAGES	NUMBER	新しいデータで再利用された未使用のページ数。

列	データ型	説明
CACHE_WRITES	NUMBER	OLAP ページ・プールのデータがデータベース・キャッシュに書き込まれた回数。
POOL_SIZE	NUMBER	OLAP ページ・プールのページ数。

V\$AW_OLAP

V\$AW_OLAP は、アクティブ・セッションのレコードおよびアナリティック・ワークスペースでの使用状況を表示します。アナリティック・ワークスペースが作成またはアタッチされるたびに、行が生成されます。セッションの最初の行は、最初の DML コマンドの発行時に作成されます。セッションの最初の行は、各セッションに自動的にアタッチされる SYS.EXPRESS アナリティック・ワークスペースを識別します。特定のアナリティック・ワークスペースに関連する行は、アナリティック・ワークスペースがセッションからデタッチされたとき、またはセッションの終了時に削除されます。

列	データ型	説明
SESSION_ID	NUMBER	セッションの一意の数値識別子。
AW_NUMBER	NUMBER	アナリティック・ワークスペースの一意の数値識別子。
ATTACH_MODE	VARCHAR2 (10)	READ ONLY または READ WRITE。
GENERATION	NUMBER	アナリティック・ワークスペースの世代。UPDATE を実行するたびに新しい世代が作成されます。同一 UPDATE コマンド内で同じアナリティック・ワークスペースをアタッチするセッションは、同じ世代を共有します。
TEMP_SPACE_PAGES	NUMBER	アナリティック・ワークスペースの一時セグメントに格納されるページ数。
TEMP_SPACE_READS	NUMBER	データが一時セグメントから読み込まれた回数、およびページ・プールから読み込まれなかった回数。
LOB_READS	NUMBER	アナリティック・ワークスペースが格納されている表 (永続 LOB) からデータが読み込まれた回数。
POOL_CHANGED_PAGES	NUMBER	このアナリティック・ワークスペースで変更されたページ・プールのページ数。
POOL_UNCHANGED_PAGES	NUMBER	このアナリティック・ワークスペースで変更されていないページ・プールのページ数。

V\$AW_SESSION_INFO

V\$AW_SESSION_INFO は、各アクティブ・セッションに関する情報を表示します。

トランザクションは、クライアント・セッションと Oracle OLAP 間の 1 回のやりとりです。DBMS_AW.EXECUTE プロシージャへのコールなど、1 回のトランザクションで複数の OLAP DML コマンドを実行できます。

列	データ型	説明
CLIENT_TYPE	VARCHAR2 (64)	OLAP
SESSION_STATE	VARCHAR2 (64)	TRANSACTING、NOT_TRANSACTING、EXCEPTION_HANDLING、CONSTRUCTING、CONSTRUCTED、DECONSTRUCTING または DECONSTRUCTED。
SESSION_HANDLE	NUMBER	セッション識別子。
USERID	VARCHAR2 (64)	セッションを開くデータベース・ユーザー名。
CURR_DML_COMMAND	VARCHAR2 (64)	実行中の DML コマンド。
PREV_DML_COMMAND	VARCHAR2 (64)	最後に実行した DML コマンド。
TOTAL_TRANSACTION	NUMBER	セッションで実行したトランザクションの合計数。通常、この数が、セッションでのアクティビティのレベルを示します。
TOTAL_TRANSACTION_TIME	NUMBER	トランザクションが実行されてからの合計経過時間 (ミリ秒)。
AVERAGE_TRANSACTION_TIME	NUMBER	トランザクションが完了するまでの平均経過時間 (ミリ秒)。
TRANSACTION_CPU_TIME	NUMBER	最後のトランザクションが完了するまでの合計 CPU 時間 (ミリ秒)。
TOTAL_TRANSACTION_CPU_TIME	NUMBER	このセッションですべてのトランザクションを実行するための合計 CPU 時間。実行中のトランザクションは含まれません。
AVERAGE_TRANSACTION_CPU_TIME	NUMBER	トランザクションを完了するまでの平均 CPU 時間。実行中のトランザクションは含まれません。

CWM2_OLAP_CATALOG

CWM2_OLAP_CATALOG パッケージでは、メジャー・フォルダの管理のためのプロシージャが提供されます。

注意： CWM2_OLAP_CATALOG パッケージで使用する**カタログ**という用語は、メジャー・フォルダを表します。

参照：

- [第 14 章「CWM2_OLAP_MEASURE」](#) を参照してください。
- [第 2 章「CWM2 による OLAP カタログ・メタデータの作成」](#) を参照してください。

この章では、次の項目について説明します。

- [メジャー・フォルダの理解](#)
- [例：メジャー・フォルダの作成](#)
- [CWM2_OLAP_CATALOG サブプログラムの概要](#)

メジャー・フォルダの理解

メジャー・フォルダは、OLAP メタデータのエンティティです。これは、メジャー・フォルダが名前と所有者で識別される OLAP カタログ内の論理オブジェクトであることを意味します。

CWM2_OLAP_CATALOG パッケージのプロシージャを使用して、メジャー・フォルダの作成、移入、削除およびロック、表示に使用する記述情報の指定を行います。

メジャー・フォルダは、関連するメジャーのグルーピング・メカニズムを提供します。メジャー・フォルダには、メジャーおよびネストしたメジャー・フォルダを格納できます。メ

ジャー・フォルダへのアクセスは、スキーマに依存しません。メジャー・フォルダはすべて、任意のクライアントに表示されます。ただし、メジャー自身へのアクセスは、基になる表へのクライアントのアクセス権に依存します。

参照： メジャー・フォルダおよび OLAP メタデータ・モデルの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

例：メジャー・フォルダの作成

次の文は、PHARMACEUTICALS メジャー・フォルダを作成し、キューブ SH.COST_CUBE から UNIT_COST メジャーを追加します。メジャー・フォルダは、ルート・レベルにあります。

```
execute cwm2_olap_catalog.create_catalog
    ('PHARMACEUTICALS', 'Pharmaceutical Sales and Planning');
execute cwm2_olap_catalog.add_catalog_entity
    ('PHARMACEUTICALS', 'SH', 'COST_CUBE', UNIT_COST');
```

CWM2_OLAP_CATALOG サブプログラムの概要

表 7-1 CWM2_OLAP_CATALOG サブプログラム

サブプログラム	説明
ADD_CATALOG_ENTITY プロシージャ (7-3 ページ)	メジャー・フォルダにメジャーを追加します。
CREATE_CATALOG プロシージャ (7-4 ページ)	メジャー・フォルダを作成します。
DROP_CATALOG プロシージャ (7-4 ペー ジ)	メジャー・フォルダを削除します。
LOCK_CATALOG プロシージャ (7-5 ペー ジ)	メジャー・フォルダをロックします。
REMOVE_CATALOG_ENTITY プロシ ージャ (7-5 ページ)	メジャー・フォルダからメジャーを削除します。
SET_CATALOG_NAME プロシージャ (7-6 ページ)	メジャー・フォルダの名前を設定します。
SET_DESCRIPTION プロシージャ (7-6 ページ)	メジャー・フォルダの説明を設定します。
SET_PARENT_CATALOG プロシージャ (7-7 ページ)	メジャー・フォルダの親フォルダを設定します。

ADD_CATALOG_ENTITY プロシージャ

このプロシージャは、メジャー・フォルダにメジャーを追加します。

構文

```
ADD_CATALOG_ENTITY (
    catalog_name    IN    VARCHAR2,
    cube_owner      IN    VARCHAR2,
    cube_name       IN    VARCHAR2,
    measure_name    IN    VARCHAR2);
```

パラメータ

表 7-2 ADD_CATALOG_ENTITY プロシージャのパラメータ

パラメータ	説明
catalog_name	メジャー・フォルダの名前。
cube_owner	キューブの所有者。
cube_name	キューブの名前。
measure_name	メジャー・フォルダに追加するメジャーの名前。

CREATE_CATALOG プロシージャ

このプロシージャは、新しいメジャー・フォルダを作成します。

メジャー・フォルダ作成の一部として、説明および表示プロパティも指定する必要があります。メジャー・フォルダの作成後、このパッケージの他のプロシージャをコールして、これらのプロパティを上書きできます。

構文

```
CREATE_CATALOG (
    catalog_name      IN  VARCHAR2,
    description       IN  VARCHAR2,
    parent_catalog    IN  VARCHAR2 DEFAULT NULL);
```

パラメータ

表 7-3 CREATE_CATALOG プロシージャのパラメータ

パラメータ	説明
catalog_name	メジャー・フォルダの名前。
description	メジャー・フォルダの説明。
parent_catalog	オプションの親メジャー・フォルダ。

DROP_CATALOG プロシージャ

このプロシージャは、メジャー・フォルダを削除します。メジャー・フォルダに他のメジャー・フォルダが格納されている場合、それらも削除されます。

構文

```
DROP_CATALOG (
```

```
catalog_name    IN    VARCHAR2);
```

パラメータ

表 7-4 DROP_CATALOG プロシージャのパラメータ

パラメータ	説明
catalog_name	メジャー・フォルダの名前。

LOCK_CATALOG プロシージャ

このプロシージャは、CWM2 モデル表のメジャー・フォルダを識別する行でデータベース・ロックを取得して、更新のためにメジャー・フォルダのメタデータをロックします。

構文

```
LOCK_CATALOG (
    catalog_name    IN    VARCHAR2,
    wait_for_lock   IN    BOOLEAN DEFAULT FALSE);
```

パラメータ

表 7-5 LOCK_CATALOG プロシージャのパラメータ

パラメータ	説明
catalog_name	メジャー・フォルダの名前。
wait_for_lock	(オプション) メジャー・フォルダが他のユーザーによってすでにロックされている場合、メジャー・フォルダが使用可能になるまで待機するかどうかを選択します。このパラメータに値を指定しない場合、プロシージャはロックの取得を待機しません。

REMOVE_CATALOG_ENTITY プロシージャ

このプロシージャは、メジャー・フォルダからメジャーを削除します。

構文

```
REMOVE_CATALOG_ENTITY (
    catalog_name    IN    VARCHAR2,
    cube_owner      IN    VARCHAR2,
    cube_name       IN    VARCHAR2,
    measure_name    IN    VARCHAR2);
```

パラメータ

表 7-6 REMOVE_CATALOG_ENTITY プロシージャのパラメータ

パラメータ	説明
catalog_name	メジャー・フォルダの名前。
cube_owner	キューブの所有者。
cube_name	キューブの名前。
measure_name	メジャー・フォルダから削除するメジャーの名前。

SET_CATALOG_NAME プロシージャ

このプロシージャは、メジャー・フォルダに名前を設定します。

構文

```
SET_CATALOG_NAME (
    old_catalog_name    IN    VARCHAR2,
    new_catalog_name    IN    VARCHAR2);
```

パラメータ

表 7-7 SET_CATALOG_NAME プロシージャのパラメータ

パラメータ	説明
old_catalog_name	古いメジャー・フォルダの名前。
new_catalog_name	新しいメジャー・フォルダの名前。

SET_DESCRIPTION プロシージャ

このプロシージャは、メジャー・フォルダに説明を設定します。

構文

```
SET_DESCRIPTION (
    catalog_name    IN    VARCHAR2,
    description     IN    VARCHAR2);
```

パラメータ

表 7-8 SET_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
catalog_name	メジャー・フォルダの名前。
description	メジャー・フォルダの説明。

SET_PARENT_CATALOG プロシージャ

このプロシージャは、メジャー・フォルダに親メジャー・フォルダを設定します。

構文

```
SET_PARENT_CATALOG (
    catalog_name          IN   VARCHAR2,
    parent_catalog_name  IN   VARCHAR2  DEFAULT NULL);
```

パラメータ

表 7-9 SET_PARENT_CATALOG プロシージャのパラメータ

パラメータ	説明
catalog_name	メジャー・フォルダの名前。
parent_catalog_name	親メジャー・フォルダの名前。メジャー・フォルダがルート・レベルにある場合、このパラメータは NULL です。

CWM2_OLAP_CUBE

CWM2_OLAP_CUBE パッケージでは、キューブを管理するためのプロシージャが提供されます。

参照： 第2章「CWM2によるOLAPカタログ・メタデータの作成」を参照してください。

この章では、次の項目について説明します。

- キューブの理解
- 例：キューブの作成
- CWM2_OLAP_CUBE サブプログラムの概要

キューブの理解

キューブは、OLAP メタデータのエンティティです。これは、キューブが名前と所有者で識別される OLAP カタログ内の論理オブジェクトであることを意味します。

キューブは、メジャーの割当て先となる多次元フレームワークです。メジャーは、ファクト表に格納されているデータを表します。ファクト表は、リレーショナル表またはビューです。ビューは、アナリティック・ワークスペースに格納されているデータを参照する場合があります。

CWM2_OLAP_CUBE パッケージのプロシージャを使用して、キューブの作成、削除およびロック、キューブとディメンションの関連付け、表示に使用する定義の指定を行います。

CWM2_OLAP_MEASURE パッケージを使用してキューブのメジャーを作成する前に、キューブを作成する必要があります。

参照：

- [第 14 章「CWM2_OLAP_MEASURE」](#) を参照してください。
- キューブおよび OLAP メタデータ・モデルの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

例：キューブの作成

次の文は、キューブ SALES_CUBE を削除して再作成し、ディメンション TIME_DIM、GEOG_DIM および PRODUCT_DIM を追加します。

キューブを削除すると、キューブのメジャーとともに、そのエンティティも OLAP カタログから削除されます。ただし、キューブを削除しても、キューブのディメンションは削除されません。

```
execute cwm2_olap_cube.drop_cube('JSMITH', 'SALES_CUBE');
execute cwm2_olap_cube.create_cube
    ('JSMITH', 'SALES_CUBE', 'Sales', 'Sales Cube',
     'Sales dimensioned over geography, product, and time' );
execute cwm2_olap_cube.add_dimension_to_cube
    ('JSMITH', 'SALES_CUBE', 'JSMITH', 'TIME_DIM');
execute cwm2_olap_cube.add_dimension_to_cube
    ('JSMITH', 'SALES_CUBE', 'JSMITH', 'GEOG_DIM');
execute cwm2_olap_cube.add_dimension_to_cube
    ('JSMITH', 'SALES_CUBE', 'JSMITH', 'PRODUCT_DIM');
```

CWM2_OLAP_CUBE サブプログラムの概要

表 8-1 CWM2_OLAP_CUBE サブプログラム

サブプログラム	説明
ADD_DIMENSION_TO_CUBE プロシージャ (8-3 ページ)	ディメンションをキューブに追加します。
CREATE_CUBE プロシージャ (8-4 ページ)	キューブを作成します。
DROP_CUBE プロシージャ (8-5 ページ)	キューブを削除します。
LOCK_CUBE プロシージャ (8-5 ページ)	更新のためにキューブのメタデータをロックします。
REMOVE_DIMENSION_FROM_CUBE プロシージャ (8-6 ページ)	キューブからディメンションを削除します。
SET_AGGREGATION_OPERATOR プロシージャ (8-6 ページ)	キューブ・データのロールアップ用の集計演算子を設定します。
SET_CUBE_NAME プロシージャ (8-8 ページ)	キューブに名前を設定します。
SET_DEFAULT_CUBE_DIM_CALC_HIER プロシージャ (8-9 ページ)	キューブのディメンションにデフォルトの計算階層を設定します。
SET_DESCRIPTION プロシージャ (8-9 ページ)	キューブに説明を設定します。
SET_DISPLAY_NAME プロシージャ (8-10 ページ)	キューブに表示名を設定します。
SET_MV_SUMMARY_CODE プロシージャ (8-10 ページ)	キューブに関連付けられたマテリアライズド・ビューの形式を設定します。
SET_SHORT_DESCRIPTION プロシージャ (8-11 ページ)	キューブに簡単な説明を設定します。

ADD_DIMENSION_TO_CUBE プロシージャ

このプロシージャは、キューブにディメンションを追加します。

構文

```
ADD_DIMENSION_TO_CUBE (
    cube_owner          IN  VARCHAR2,
    cube_name           IN  VARCHAR2,
    dimension_owner     IN  VARCHAR2,
```

```
dimension_name    IN    VARCHAR2);
```

パラメータ

表 8-2 ADD_DIMENSION_TO_CUBE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
dimension_owner	キューブに追加するディメンションの所有者。
dimension_name	キューブに追加するディメンションの名前。

CREATE_CUBE プロシージャ

このプロシージャは、OLAP カタログに新しいキューブを作成します。

キューブ作成の一部として、説明および表示プロパティを指定する必要があります。キューブの作成後、このパッケージの他のプロシージャをコールして、これらのプロパティを上書きできます。

構文

```
CREATE_CUBE (
    cube_owner      IN    VARCHAR2,
    cube_name       IN    VARCHAR2,
    display_name    IN    VARCHAR2,
    short_description IN  VARCHAR2,
    description     IN    VARCHAR2);
```

パラメータ

表 8-3 CREATE_CUBE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
display_name	キューブの表示名。
short_description	キューブの簡単な説明。
description	キューブの説明。

DROP_CUBE プロシージャ

このプロシージャは、OLAP カタログからキューブを削除します。

注意： キューブを削除すると、関連付けられたメジャーも削除されます。ただし、キューブのディメンションは削除されません。別のキューブのコンテキスト内にマップされている場合があります。

構文

```
DROP_CUBE (
    cube_owner      IN  VARCHAR2,
    cube_name       IN  VARCHAR2);
```

パラメータ

表 8-4 DROP_CUBE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。

LOCK_CUBE プロシージャ

このプロシージャは、CWM2 モデル表のキューブを識別する行でデータベース・ロックを取得して、更新のためにキューブのメタデータをロックします。

構文

```
LOCK_CUBE (
    cube_owner      IN  VARCHAR2,
    cube_name       IN  VARCHAR2,
    wait_for_lock   IN  BOOLEAN DEFAULT FALSE);
```

パラメータ

表 8-5 LOCK_CUBE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。

表 8-5 LOCK_CUBE プロシージャのパラメータ (続き)

パラメータ	説明
wait_for_lock	(オプション) キューブが他のユーザーによってすでにロックされている場合、キューブが使用可能になるまで待機するかどうかを選択します。このパラメータに値を指定しない場合、プロシージャはロックの取得を待機しません。

REMOVE_DIMENSION_FROM_CUBE プロシージャ

このプロシージャは、キューブからディメンションを削除します。

構文

```
REMOVE_DIMENSION_FROM_CUBE (
    cube_owner      IN   VARCHAR2,
    cube_name       IN   VARCHAR2,
    dimension_owner IN   VARCHAR2,
    dimension_name  IN   VARCHAR2);
```

パラメータ

表 8-6 REMOVE_DIMENSION_FROM_CUBE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
dimension_owner	キューブから削除するディメンションの所有者。
dimension_name	キューブから削除するディメンションの名前。

SET_AGGREGATION_OPERATOR プロシージャ

このプロシージャは、キューブのディメンションに対するデータのロールアップ用の集計演算子を設定します。キューブは、識別子 'LOWESTLEVEL' の格納タイプでスター・スキーマにマップされる必要があります。2-11 ページの「[ファクト表とディメンション表の結合](#)」を参照してください。

OLAP カタログでサポートされる集計演算子は、1-21 ページの表 1-10「[集計演算子](#)」にリストしています。

集計演算子を指定しない場合、演算子は加算です。

ビュー ALL_OLAP2_AGGREGATION_USES は、キューブに指定されているデフォルトではない集計演算子を表示します。5-3 ページの「[ALL_OLAP2_AGGREGATION_USES](#)」を参照してください。

構文

```
SET_AGGREGATION_OPERATOR (
    cube_owner      IN   VARCHAR2,
    cube_name       IN   VARCHAR2,
    aggop_spec      IN   VARCHAR2);
```

パラメータ

表 8-7 SET_AGGREGATION_OPERATOR プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
aggop_spec	<p>キューブの集計演算子を指定する文字列。</p> <p>指定する各集計演算子は、キューブの特定のディメンションの特定の階層上のキューブのすべてのメジャーに適用されます。階層を指定しない場合、演算子はディメンションのすべての階層に適用されます。デフォルトでは、集計演算子は加算です。</p> <p>次のように、文字列を一重引用符で囲み、ディメンション / 演算子の各句をセミコロンで区切って指定します。</p> <pre>'DIM:dim1_owner.dim1_name/AGGOP:operator; DIM:dim2_owner.dim2_name/AGGOP:operator;.....'</pre> <p>演算子をディメンションの特定の階層に適用する必要がある場合、次のように、オプションの 'HIER' 句を DIM 句の後に指定します。</p> <pre>/HIER:hiername1</pre> <p>加重演算子に対しては、'AGGOP' 句の後に次のようにオプションで WEIGHTBY 句を指定します。</p> <pre>/WEIGHTBY:TblOwner.TblName.ColName;</pre> <p>注意: キューブのデータは、集計仕様内のディメンション句の順序で集計されます。</p>

例

次の例は、Product および Channel ディメンションの標準的な階層に対しては加算を、Geography ディメンションの標準的な階層に対しては MAX 演算子を、Time ディメンションの年度累計階層に対しては AVERAGE 演算子を使用して ANALYTIC_CUBE のデータを集計するように指定します。指定外のすべての階層は加算を使用します。

```
execute cwm2_olap_cube.set_aggregation_operator
('XADEMO', 'ANALYTIC_CUBE',
'DIM:XADEMO.PRODUCT/HIER:STANDARD/AGGOP:SUM;
```

```
DIM:XADEMO.GEOGRAPHY/HIER:STANDARD/AGGOP:MAX;
DIM:XADEMO.TIME/HIER:YTD/AGGOP:AVERAGE;
DIM:XADEMO.CHANNEL/HIER:STANDARD/AGGOP:SUM;');
```

次の例は、Product デイメンションの加重演算子を含む同じ指定を示します。

```
execute cwm2_olap_cube.set_aggregation_operator
('XADEMO', 'ANALYTIC_CUBE',
'DIM:XADEMO.PRODUCT/HIER:STANDARD/AGGOP:SUM/
WEIGHTBY:XADEMO.XADEMO_SALES_VIEW.COSTS;
DIM:XADEMO.GEOGRAPHY/HIER:STANDARD/AGGOP:MAX;
DIM:XADEMO.TIME/HIER:YTD/AGGOP:AVERAGE;
DIM:XADEMO.CHANNEL/HIER:STANDARD/AGGOP:SUM;');
```

次の例では、各デイメンションのすべての階層に集計演算子が指定されます。

```
execute cwm2_olap_cube.set_aggregation_operator
('XADEMO', 'ANALYTIC_CUBE',
DIM:XADEMO.PRODUCT/AGGOP:SUM;
DIM:XADEMO.GEOGRAPHY/AGGOP:MAX;
DIM:XADEMO.TIME/AGGOP:AVERAGE;
DIM:XADEMO.CHANNEL/AGGOP:SUM;');
```

参照

1-5 ページの「[アナリティック・ワークスペースにおけるキューブのデータの集計](#)」を参照してください。

SET_CUBE_NAME プロシージャ

このプロシージャは、キューブに名前を設定します。

構文

```
SET_CUBE_NAME (
    cube_owner      IN   VARCHAR2,
    cube_name       IN   VARCHAR2,
    set_cube_name   IN   VARCHAR2);
```

パラメータ

表 8-8 SET_CUBE_NAME プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの元の名前。

表 8-8 SET_CUBE_NAME プロシージャのパラメータ (続き)

パラメータ	説明
set_cube_name	キューブの新しい名前。

SET_DEFAULT_CUBE_DIM_CALC_HIER プロシージャ

このプロシージャは、このキューブのディメンションにデフォルトの計算階層を設定します。

構文

```
SET_DEFAULT_CUBE_DIM_CALC_HIER (
    cube_owner      IN   VARCHAR2,
    cube_name       IN   VARCHAR2,
    dimension_owner IN   VARCHAR2,
    dimension_name  IN   VARCHAR2,
    hierarchy_name  IN   VARCHAR2);
```

パラメータ

表 8-9 SET_DEFAULT_CUBE_DIM_CALC_HIER プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	このディメンションのデフォルトで使用する階層の名前。

SET_DESCRIPTION プロシージャ

このプロシージャは、キューブに説明を設定します。

構文

```
SET_DESCRIPTION (
    cube_owner      IN   VARCHAR2,
    cube_name       IN   VARCHAR2,
    description     IN   VARCHAR2);
```

パラメータ

表 8-10 SET_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
description	キューブの説明。

SET_DISPLAY_NAME プロシージャ

このプロシージャは、キューブに表示名を設定します。

構文

```
SET_DISPLAY_NAME (
    cube_owner    IN    VARCHAR2,
    cube_name     IN    VARCHAR2,
    display_name  IN    VARCHAR2);
```

パラメータ

表 8-11 SET_DISPLAY_NAME プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
display_name	キューブの表示名。

SET_MV_SUMMARY_CODE プロシージャ

このプロシージャは、このキューブのマテリアライズド・ビューの形式を指定します。マテリアライズド・ビューは、グルーピング・セット (groupingset) またはロールアップ (rollup) 形式に指定できます。

ロールアップ形式のマテリアライズド・ビューでは、すべてのディメンション・キー列が移入され、ディメンションの関係を完全に指定した場合のみデータにアクセスできます。

グルーピング・セット形式のマテリアライズド・ビューでは、ディメンション・キー列に NULL 値が含まれる場合があり、1 つ以上のレベルを指定することによってデータにアクセスできます。

構文

```
SET_MV_SUMMARY_CODE (
    cube_owner          IN   VARCHAR2,
    cube_name           IN   VARCHAR2,
    summary_code       IN   VARCHAR2);
```

パラメータ

表 8-12 SET_MV_SUMMARY_CODE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
summary_code	次のいずれかの値（大 / 小文字は区別されない）。 <ul style="list-style-type: none"> ■ rollup（ロールアップ形式の場合） ■ groupingset（グルーピング・セット形式の場合）

SET_SHORT_DESCRIPTION プロシージャ

このプロシージャは、キューブに簡単な説明を設定します。

構文

```
SET_SHORT_DESCRIPTION (
    cube_owner          IN   VARCHAR2,
    cube_name           IN   VARCHAR2,
    short_description   IN   VARCHAR2);
```

パラメータ

表 8-13 SET_SHORT_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
short_description	キューブの簡単な説明。

CWM2_OLAP_DIMENSION

CWM2_OLAP_DIMENSION パッケージでは、ディメンションを管理するためのプロシージャが提供されます。

参照： 第2章「CWM2によるOLAPカタログ・メタデータの作成」を参照してください。

この章では、次の項目について説明します。

- [ディメンションの理解](#)
- [例：CWM2ディメンションの作成](#)
- [CWM2_OLAP_DIMENSION サブプログラムの概要](#)

ディメンションの理解

ディメンションは、OLAP メタデータのエンティティです。これは、ディメンションが名前と所有者で識別される OLAP カタログ内の論理オブジェクトであることを意味します。論理 OLAP ディメンションの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

注意： CWM2 のディメンションは、ディメンション表の列に直接マップされ、Oracle データベースのディメンション・オブジェクトとは関係がありません。

CWM2_OLAP_DIMENSION パッケージのプロシージャを使用して、CWM2 ディメンション・エンティティの作成、削除およびロック、表示に使用する定義の指定を行います。CWM2 ディメンションを完全に定義するには、2-2 ページの「[ディメンションの作成](#)」に示す手順に従います。

参照： ディメンションおよび OLAP メタデータ・モデルの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

例 : CWM2 ディメンションの作成

次の文は、JSMITH スキーマで CWM2 ディメンションのエンティティ PRODUCT_DIM を作成します。表示名は Product で、複数名は Products です。簡単な説明は Prod で、説明は Product です。

```
execute cwm2_olap_dimension.create_dimension
('JSMITH', 'PRODUCT_DIM', 'Product', 'Products', 'Prod', 'Product');
```

次の文は、簡単な説明を Product に、詳細な説明を Product Dimension に変更します。

```
execute cwm2_olap_dimension.set_short_description
('JSMITH', 'PRODUCT_DIM', 'Product');
execute cwm2_olap_dimension.set_description
('JSMITH', 'PRODUCT_DIM', 'Product Dimension');
```

CWM2_OLAP_DIMENSION サブプログラムの概要

表 9-1 CWM2_OLAP_DIMENSION サブプログラム

サブプログラム	説明
CREATE_DIMENSION プロシージャ (9-3 ページ)	ディメンションを作成します。
DROP_DIMENSION プロシージャ (9-4 ページ)	ディメンションを削除します。
LOCK_DIMENSION プロシージャ (9-5 ページ)	更新のためにディメンションのメタデータをロックします。
SET_DEFAULT_DISPLAY_HIERARCHY プロシージャ (9-5 ページ)	ディメンションにデフォルトの階層を設定します。
SET_DESCRIPTION プロシージャ (9-6 ページ)	ディメンションに説明を設定します。
SET_DIMENSION_NAME プロシージャ (9-6 ページ)	ディメンションに名前を設定します。
SET_DISPLAY_NAME プロシージャ (9-7 ページ)	ディメンションに表示名を設定します。
SET_PLURAL_NAME プロシージャ (9-7 ページ)	ディメンションに複数名を設定します。
SET_SHORT_DESCRIPTION プロシージャ (9-8 ページ)	ディメンションに簡単な説明を設定します。

CREATE_DIMENSION プロシージャ

このプロシージャは、OLAP カタログに新しいディメンション・エンティティを作成します。

デフォルトでは、新しいディメンションは通常のディメンションですが、`dimension_type` パラメータに値 `TIME` を指定して、時間ディメンションを作成できます。

ディメンション作成の一部として、説明および表示プロパティを指定する必要があります。ディメンションの作成後、このパッケージの他のプロシージャをコールして、これらのプロパティを上書きできます。

構文

```
CREATE_DIMENSION (
    dimension_owner          IN   VARCHAR2,
```

```

dimension_name      IN  VARCHAR2,
display_name       IN  VARCHAR2,
plural_name        IN  VARCHAR2,
short_description  IN  VARCHAR2,
description        IN  VARCHAR2,
dimension_type     IN  VARCHAR2 DEFAULT NULL);

```

パラメータ

表 9-2 CREATE_DIMENSION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
display_name	ディメンションの表示名。
plural_name	ディメンションの複数名。
short_description	ディメンションの簡単な説明。
description	ディメンションの説明。
dimension_type	(オプション) ディメンションのタイプ。値 TIME を指定して、時間ディメンションを作成します。このパラメータを指定しない場合、ディメンションは、通常のディメンションとして作成されます。

DROP_DIMENSION プロシージャ

このプロシージャは、OLAP カタログからディメンション・エンティティを削除します。関連するすべてのレベル、階層およびディメンション属性も削除されます。

構文

```

DROP_DIMENSION (
    dimension_owner  IN  VARCHAR2,
    dimension_name   IN  VARCHAR2);

```

パラメータ

表 9-3 DROP_DIMENSION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。

表 9-3 DROP_DIMENSION プロシージャのパラメータ (続き)

パラメータ	説明
dimension_name	ディメンションの名前。

LOCK_DIMENSION プロシージャ

このプロシージャは、CWM2 モデル表のディメンションを識別する行でデータベース・ロックを取得して、更新のためにディメンションのメタデータをロックします。

構文

```
LOCK_DIMENSION (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    wait_for_lock      IN    BOOLEAN DEFAULT FALSE);
```

パラメータ

表 9-4 LOCK_DIMENSION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
wait_for_lock	(オプション) ディメンションが他のユーザーによってすでにロックされている場合、ディメンションが使用可能になるまで待機するかどうかを選択します。このパラメータに値を指定しない場合、プロシージャはロックの取得を待機しません。

SET_DEFAULT_DISPLAY_HIERARCHY プロシージャ

このプロシージャは、表示に使用するデフォルトの階層を設定します。

構文

```
SET_DEFAULT_DISPLAY_HIERARCHY (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    hierarchy_name     IN    VARCHAR2);
```

パラメータ

表 9-5 SET_DEFAULT_DISPLAY_HIERARCHY プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	ディメンションのいずれかの階層の名前。

SET_DESCRIPTION プロシージャ

このプロシージャは、ディメンションに説明を設定します。

構文

```
SET_DESCRIPTION (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    description        IN    VARCHAR2);
```

パラメータ

表 9-6 SET_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
description	ディメンションの説明。

SET_DIMENSION_NAME プロシージャ

このプロシージャは、ディメンションに名前を設定します。

構文

```
SET_DIMENSION_NAME (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    set_dimension_name IN    VARCHAR2);
```

パラメータ

表 9-7 SET_DIMENSION_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの元の名前。
set_dimension_name	ディメンションの新しい名前。

SET_DISPLAY_NAME プロシージャ

このプロシージャは、ディメンションに表示名を設定します。

構文

```
SET_DISPLAY_NAME (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    display_name       IN    VARCHAR2);
```

パラメータ

表 9-8 SET_DISPLAY_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
display_name	ディメンションの表示名。

SET_PLURAL_NAME プロシージャ

このプロシージャは、ディメンションの複数名を設定します。

構文

```
SET_PLURAL_NAME (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    plural_name        IN    VARCHAR2);
```

パラメータ

表 9-9 SET_PLURAL_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
plural_name	ディメンションの複数名。

SET_SHORT_DESCRIPTION プロシージャ

このプロシージャは、ディメンションに簡単な説明を設定します。

構文

```
SET_SHORT_DESCRIPTION (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    short_description  IN    VARCHAR2);
```

パラメータ

表 9-10 SET_SHORT_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
short_description	ディメンションの簡単な説明。

CWM2_OLAP_DIMENSION_ATTRIBUTE

CWM2_OLAP_DIMENSION_ATTRIBUTE パッケージでは、ディメンション属性を管理するためのプロシージャが提供されます。

参照： 第2章「CWM2によるOLAPカタログ・メタデータの作成」を参照してください。

この章では、次の項目について説明します。

- [ディメンション属性の理解](#)
- [例：ディメンション属性の作成](#)
- [CWM2_OLAP_DIMENSION_ATTRIBUTE サブプログラムの概要](#)

ディメンション属性の理解

ディメンション属性は、OLAP メタデータのエンティティです。これは、ディメンション属性が名前と所有者で識別される OLAP カタログ内の論理オブジェクトであることを意味します。

ディメンション属性は、ディメンションにレベル属性のセットを定義します。ディメンション属性には、ディメンションの一部またはすべてのレベルのレベル属性が含まれます。時間ディメンションの場合は、ディメンション属性 end date および time span をすべてのレベルに定義する必要があります。

CWM2_OLAP_DIMENSION_ATTRIBUTE パッケージのプロシージャを使用して、ディメンション・エンティティの作成、削除およびロック、表示に使用する定義の指定を行います。

一部のディメンション属性の名前は CWM2 で特別な意味を持つため予約済です。予約済のディメンション属性で構成されているレベル属性は、特定の情報を含む列にマップされます。表 10-1 に、予約済のディメンション属性を示します。

表 10-1 予約済のディメンション属性

ディメンション属性	説明
Long Description	ディメンション・メンバーの詳細な説明。
Short Description	ディメンション・メンバーの簡単な説明。
End Date	時間ディメンションでの特定の時間間隔の最終日（必須）。
Time Span	時間ディメンションでの特定の時間間隔の日数（必須）。
Prior Period	時間ディメンションでのこの時間間隔の以前の時間間隔。
Year Ago Period	時間ディメンションでのこの時間間隔の1年前の時間間隔。
ET Key	埋込み合計ディメンションでのディメンション・メンバーを識別する埋込み合計キー（必須）。
Parent ET Key	埋込み合計ディメンションでの ET キーの親であるディメンション・メンバー（必須）。
Grouping ID	埋込み合計ディメンションでのグルーピング ID (GID)。ディメンション表の行の階層レベルを識別します（必須）。
Parent Grouping ID	埋込み合計ディメンションでのグルーピング ID の親であるディメンション・メンバー（必須）。

ディメンション属性を作成する前に、親ディメンションはすでに存在している必要があります。ディメンションを完全に定義するには、2-2 ページの「[ディメンションの作成](#)」に示す手順に従います。

参照：

- [第 13 章「CWM2_OLAP_LEVEL_ATTRIBUTE」](#) を参照してください。
- ディメンション属性および OLAP メタデータ・モデルの詳細は、『[Oracle OLAP アプリケーション開発者ガイド](#)』を参照してください。

例：ディメンション属性の作成

次の文は、JSMITH スキーマの PRODUCT_DIM ディメンションに対して、ディメンション属性 PRODUCT_DIM_BRAND を作成します。表示名は Brand です。簡単な説明は Brand Name で、説明は Product Brand Name です。

```
execute cwm2_olap_dimension_attribute.create_dimension_attribute
('JSMITH', 'PRODUCT_DIM', 'PRODUCT_DIM_BRAND',
'Brand', 'Brand Name', 'Product Brand Name');
```

次の文は、JSMITH スキーマの PRODUCT_DIM ディメンションに対して、ディメンション属性 Short Description を作成します。Short Description は、予約済のディメンション属性です。

```
execute cwm2_olap_dimension_attribute.create_dimension_attribute
('JSMITH', 'PRODUCT_DIM', 'Short Description',
 'Short Product Names', 'Short Desc Product',
 'Short Name of Products', TRUE);
```

CWM2_OLAP_DIMENSION_ATTRIBUTE サブプログラムの概要

表 10-2 CWM2_OLAP_DIMENSION_ATTRIBUTE サブプログラム

サブプログラム	説明
CREATE_DIMENSION_ATTRIBUTE プロシージャ (10-4 ページ)	ディメンション属性を作成します。
DROP_DIMENSION_ATTRIBUTE プロシージャ (10-6 ページ)	ディメンション属性を削除します。
LOCK_DIMENSION_ATTRIBUTE プロシージャ (10-6 ページ)	更新のためにディメンション属性をロックします。
SET_DESCRIPTION プロシージャ (10-7 ページ)	ディメンション属性に説明を設定します。
SET_DIMENSION_ATTRIBUTE_NAME プロシージャ (10-7 ページ)	ディメンション属性に名前を設定します。
SET_DISPLAY_NAME プロシージャ (10-8 ページ)	ディメンション属性に表示名を設定します。
SET_SHORT_DESCRIPTION プロシージャ (10-9 ページ)	ディメンション属性に簡単な説明を設定します。

CREATE_DIMENSION_ATTRIBUTE プロシージャ

このプロシージャは、新しいディメンション属性を作成します。

ディメンション属性が予約済の場合、予約済の名前をディメンション属性の名前として、または指定する名前に関連付けられた型として指定できます。表 10-1 「[予約済のディメンション属性](#)」に、予約済のディメンション属性を示します。

ディメンション属性の名前が、特定のグループのレベル属性をマッピングするために予約される必要がある場合は、RESERVED_DIMENSION_ATTRIBUTE 引数を TRUE に設定できます。詳細は、表 10-1 「[予約済のディメンション属性](#)」を参照してください。

属性作成の一部として、説明および表示プロパティを指定する必要があります。ディメンション属性の作成後、このパッケージの他のプロシージャをコールして、これらのプロパティを上書きできます。

構文

```
CREATE_DIMENSION_ATTRIBUTE (
    dimension_owner          IN  VARCHAR2,
    dimension_name           IN  VARCHAR2,
```

```

dimension_attribute_name    IN    VARCHAR2,
display_name                IN    VARCHAR2,
short_description          IN    VARCHAR2,
description                 IN    VARCHAR2,
type                        IN    VARCHAR2          );
use_name_as_type            IN    BOOLEAN DEFAULT FALSE);

```

パラメータ

表 10-3 CREATE_DIMENSION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
display_name	ディメンション属性の表示名。
short_description	ディメンション属性の簡単な説明。
description	ディメンション属性の説明。
type または use_name_as_type	<p>この引数には次のいずれかを指定できます。</p> <ul style="list-style-type: none"> ■ type 値が表 10-1 「予約済のディメンション属性」の予約済の名前の1つである VARCHAR2 引数。予約済のディメンション属性に独自の名前を作成する場合、この引数を指定します。 ■ use_name_as_type BOOLEAN 引数 (デフォルトは FALSE)。この引数は、ディメンション属性の名前が予約済の名前かどうかを指定します。この引数が TRUE の場合、dimension_attribute_name 引数の値は、表 10-1 「予約済のディメンション属性」の予約済の名前である必要があります。 <p>この引数の値を指定しない場合、ディメンション属性は予約されません。</p>

DROP_DIMENSION_ATTRIBUTE プロシージャ

このプロシージャは、ディメンション属性を削除します。

構文

```
DROP_DIMENSION_ATTRIBUTE (
    dimension_owner          IN   VARCHAR2,
    dimension_name          IN   VARCHAR2,
    dimension_attribute_name IN   VARCHAR2);
```

パラメータ

表 10-4 DROP_DIMENSION_ATTRIBUTE プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。

LOCK_DIMENSION_ATTRIBUTE プロシージャ

このプロシージャは、CWM2 モデル表のディメンションを識別する行でデータベース・ロックを取得して、更新のためにディメンション属性をロックします。

構文

```
LOCK_DIMENSION_ATTRIBUTE (
    dimension_owner          IN   VARCHAR2,
    dimension_name          IN   VARCHAR2,
    dimension_attribute_name IN   VARCHAR2,
    wait_for_lock           IN   BOOLEAN DEFAULT FALSE);
```

パラメータ

表 10-5 LOCK_DIMENSION_ATTRIBUTE プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。

表 10-5 LOCK_DIMENSION_ATTRIBUTE プロシージャのパラメータ (続き)

パラメータ	説明
dimension_attribute_name	ディメンション属性の名前。
wait_for_lock	(オプション) ディメンション属性が他のユーザーによってすでにロックされている場合、ディメンション属性が使用可能になるまで待機するかどうかを選択します。このパラメータに値を指定しない場合、プロシージャはロックの取得を待機しません。

SET_DESCRIPTION プロシージャ

このプロシージャは、ディメンション属性に説明を設定します。

構文

```
SET_DESCRIPTION (
    dimension_owner          IN  VARCHAR2,
    dimension_name          IN  VARCHAR2,
    dimension_attribute_name IN  VARCHAR2,
    description             IN  VARCHAR2);
```

パラメータ

表 10-6 SET_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
description	ディメンション属性の説明。

SET_DIMENSION_ATTRIBUTE_NAME プロシージャ

このプロシージャは、ディメンション属性に名前を設定します。

ディメンション属性が予約済の場合、予約済の名前をディメンション属性の名前として、または指定する名前に関連付けられた型として指定できます。表 10-1 「予約済のディメンション属性」に、予約済のディメンション属性を示します。

構文

```
SET_DIMENSION_ATTRIBUTE_NAME (
```

```

dimension_owner          IN  VARCHAR2,
dimension_name           IN  VARCHAR2,
dimension_attribute_name IN  VARCHAR2,
set_dimension_attribute_name IN VARCHAR2,
type                    IN  VARCHAR2          );
use_name_as_type        IN  BOOLEAN DEFAULT FALSE);

```

パラメータ

表 10-7 SET_DIMENSION_ATTRIBUTE_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の元の名前。
set_dimension_attribute_name	ディメンション属性の新しい名前。
type または use_name_as_type	<p>この引数には次のいずれかを指定できます。</p> <ul style="list-style-type: none"> ■ type 値が表 10-1 「予約済のディメンション属性」の予約済の名前の1つである VARCHAR2 引数。予約済のディメンション属性に独自の名前を作成する場合、この引数を指定します。 ■ use_name_as_type BOOLEAN 引数 (デフォルトは FALSE)。この引数は、ディメンション属性の名前が予約済の名前かどうかを指定します。この引数が TRUE の場合、dimension_attribute_name 引数の値は、表 10-1 「予約済のディメンション属性」の予約済の名前である必要があります。 <p>この引数の値を指定しない場合、ディメンション属性は予約されません。</p>

SET_DISPLAY_NAME プロシージャ

このプロシージャは、ディメンション属性に表示名を設定します。

構文

```

SET_DISPLAY_NAME (
    dimension_owner          IN  VARCHAR2,
    dimension_name           IN  VARCHAR2,
    dimension_attribute_name IN  VARCHAR2,
    display_name             IN  VARCHAR2);

```

パラメータ

表 10-8 SET_DISPLAY_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
display_name	ディメンション属性の表示名。

SET_SHORT_DESCRIPTION プロシージャ

このプロシージャは、ディメンション属性に簡単な説明を設定します。

構文

```
SET_SHORT_DESCRIPTION (
    dimension_owner          IN  VARCHAR2,
    dimension_name          IN  VARCHAR2,
    dimension_attribute_name IN  VARCHAR2,
    short_description       IN  VARCHAR2);
```

パラメータ

表 10-9 SET_SHORT_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
short_description	ディメンション属性の簡単な説明。

CWM2_OLAP_HIERARCHY

CWM2_OLAP_HIERARCHY パッケージでは、階層を管理するためのプロシージャが提供されます。

参照： 第2章「CWM2によるOLAPカタログ・メタデータの作成」を参照してください。

この章では、次の項目について説明します。

- 階層の理解
- 例：階層の作成
- CWM2_OLAP_HIERARCHY サブプログラムの概要

階層の理解

階層は、OLAP メタデータのエンティティです。これは、階層が名前と所有者で識別されるOLAP カタログ内の論理オブジェクトであることを意味します。

階層は、ディメンションの一連のレベル間の親子関係を定義します。単一のディメンションに複数の階層が関連付けられている場合もあります。複数の階層で同じレベルを使用することもできます。階層の詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

CWM2_OLAP_HIERARCHY パッケージのプロシージャを使用して、階層の作成、削除およびロック、表示に使用する記述情報の指定を行います。

階層を作成する前に、親ディメンションがOLAP カタログにすでに存在している必要があります。

参照：

- 第 12 章「CWM2_OLAP_LEVEL」を参照してください。
- 階層および OLAP メタデータ・モデルの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

例：階層の作成

次の文は、JSMITH スキーマの PRODUCT_DIM デイメンションに対して、デイメンション階層 PRODUCT_DIM_ROLLUP を作成します。表示名は Standard です。簡単な説明は Std Product、説明は Standard Product Hierarchy です。解決済コードは SOLVED LEVEL-BASED で、この階層が埋込み合計デイメンション表にマップされ、このデイメンション階層に関連付けられたファクト表に完全解決済データが格納されることを表します。

```
execute cwm2_olap_hierarchy.create_hierarchy
('JSMITH', 'PRODUCT_DIM', 'PRODUCT_DIM_ROLLUP',
 'Standard', 'Std Product', 'Standard Product Hierarchy',
 'SOLVED LEVEL-BASED');
```

CWM2_OLAP_HIERARCHY サブプログラムの概要

表 11-1 CWM2_OLAP_HIERARCHY サブプログラム

サブプログラム	説明
CREATE_HIERARCHY プロシージャ (11-3 ページ)	階層を作成します。
DROP_HIERARCHY プロシージャ (11-4 ページ)	階層を削除します。
LOCK_HIERARCHY プロシージャ (11-5 ページ)	更新のために階層をロックします。
SET_DESCRIPTION プロシージャ (11-5 ページ)	階層に説明を設定します。
SET_DISPLAY_NAME プロシージャ (11-6 ページ)	階層に表示名を設定します。
SET_HIERARCHY_NAME プロシージャ (11-7 ページ)	階層に名前を設定します。
SET_SHORT_DESCRIPTION プロシージャ (11-7 ページ)	階層に簡単な説明を設定します。
SET_SOLVED_CODE プロシージャ (11-8 ページ)	階層に解決済コードを設定します。

CREATE_HIERARCHY プロシージャ

このプロシージャは、OLAP カタログに新しい階層を作成します。

階層作成の一部として、説明および表示プロパティを指定する必要があります。階層の作成後、CWM2_OLAP_HIERARCHY パッケージの他のプロシージャをコールして、これらのプロパティを上書きできます。

構文

```
CREATE_HIERARCHY (
    dimension_owner      IN   VARCHAR2,
    dimension_name       IN   VARCHAR2,
    hierarchy_name       IN   VARCHAR2,
    display_name         IN   VARCHAR2,
    short_description    IN   VARCHAR2,
    description          IN   VARCHAR2,
    solved_code          IN   VARCHAR2);
```

パラメータ

表 11-2 CREATE_HIERARCHY プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。
display_name	階層の表示名。
short_description	階層の簡単な説明。
description	階層の説明。
solved_code	階層に埋込み合計が含まれるかどうか、およびレベルベース・ディメンション表または親子ディメンション表にマップされるかどうか。異なる解決済コードでの階層のマッピングについては、2-11 ページの「 ファクト表とディメンション表の結合 」を参照してください。 このパラメータの値は次のとおりです。 <ul style="list-style-type: none"> ■ UNSOLVED LEVEL-BASED は、埋込み合計を含まず、レベルベース・ディメンション表に格納される階層に使用します。 ■ SOLVED LEVEL-BASED は、埋込み合計を含み、レベルベース・ディメンション表に格納される、グルーピング ID を持つ階層に使用します。 ■ SOLVED VALUE-BASED は、埋込み合計を含み、親子ディメンション表に格納される階層に使用します。

DROP_HIERARCHY プロシージャ

このプロシージャは、OLAP カタログから階層を削除します。

構文

```
DROP_HIERARCHY (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    hierarchy_name     IN    VARCHAR2);
```

パラメータ

表 11-3 DROP_HIERARCHY プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。

LOCK_HIERARCHY プロシージャ

このプロシージャは、CWM2 モデル表の階層を識別する行でデータベース・ロックを取得して、更新のために階層のメタデータをロックします。

構文

```
LOCK_HIERARCHY (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    hierarchy_name     IN    VARCHAR2,
    wait_for_lock      IN    BOOLEAN DEFAULT FALSE);
```

パラメータ

表 11-4 LOCK_HIERARCHY プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。
wait_for_lock	(オプション) 階層が他のユーザーによってすでにロックされている場合、階層が使用可能になるまで待機するかどうかを選択します。このパラメータに値を指定しない場合、プロシージャはロックの取得を待機しません。

SET_DESCRIPTION プロシージャ

このプロシージャは、階層に説明を設定します。

構文

```
SET_DESCRIPTION (
```

```

dimension_owner  IN  VARCHAR2,
dimension_name   IN  VARCHAR2,
hierarchy_name  IN  VARCHAR2,
description      IN  VARCHAR2);

```

パラメータ

表 11-5 SET_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。
description	階層の説明。

SET_DISPLAY_NAME プロシージャ

このプロシージャは、ディメンションに表示名を設定します。

構文

```

SET_DISPLAY_NAME (
    dimension_owner  IN  VARCHAR2,
    dimension_name   IN  VARCHAR2,
    hierarchy_name   IN  VARCHAR2,
    display_name     IN  VARCHAR2);

```

パラメータ

表 11-6 SET_DISPLAY_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。
display_name	階層の表示名。

SET_HIERARCHY_NAME プロシージャ

このプロシージャは、階層に名前を設定します。

構文

```
SET_HIERARCHY_NAME (
    dimension_owner      IN   VARCHAR2,
    dimension_name       IN   VARCHAR2,
    hierarchy_name       IN   VARCHAR2,
    set_hierarchy_name   IN   VARCHAR2);
```

パラメータ

表 11-7 SET_HIERARCHY_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の元の名前。
set_hierarchy_name	階層の新しい名前。

SET_SHORT_DESCRIPTION プロシージャ

このプロシージャは、階層に簡単な説明を設定します。

構文

```
SET_SHORT_DESCRIPTION (
    dimension_owner      IN   VARCHAR2,
    dimension_name       IN   VARCHAR2,
    hierarchy_name       IN   VARCHAR2,
    short_description    IN   VARCHAR2);
```

パラメータ

表 11-8 SET_SHORT_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。

表 11-8 SET_SHORT_DESCRIPTION プロシージャのパラメータ (続き)

パラメータ	説明
hierarchy_name	階層の名前。
short_description	階層の簡単な説明。

SET_SOLVED_CODE プロシージャ

このプロシージャは、階層に解決済コードを設定します。解決済コードによって、この階層によってディメンション化されるデータに埋込み合計が含まれるかどうか、およびレベルベース・ディメンション表または親子ディメンション表にマップされるかどうかを指定します。親子ディメンション表にマップする場合、OLAP API からはアクセスできません。

解決済データおよび未解決データのマッピングの詳細は、2-11 ページの「[ファクト表とディメンション表の結合](#)」を参照してください。

構文

```
SET_SOLVED_CODE (
    dimension_owner    IN   VARCHAR2,
    dimension_name     IN   VARCHAR2,
    hierarchy_name     IN   VARCHAR2,
    solved_code        IN   VARCHAR2);
```

パラメータ

表 11-9 SET_SOLVED_CODE プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。

表 11-9 SET_SOLVED_CODE プロシージャのパラメータ (続き)

パラメータ	説明
solved_code	<p data-bbox="601 305 1300 404">階層に埋込み合計が含まれるかどうか、およびレベルベース・ディメンション表または親子ディメンション表にマップされるかどうか。異なる解決済コードでの階層のマッピングについては、2-11 ページの「ファクト表とディメンション表の結合」を参照してください。</p> <p data-bbox="601 418 991 440">このパラメータの値は次のとおりです。</p> <ul data-bbox="601 460 1300 673" style="list-style-type: none"><li data-bbox="601 460 1300 510">■ UNSOLVED LEVEL-BASED は、埋込み合計を含まず、レベルベース・ディメンション表に格納される階層に使用します。<li data-bbox="601 529 1300 609">■ SOLVED LEVEL-BASED は、埋込み合計を含み、レベルベース・ディメンション表に格納される、グルーピング ID を持つ階層に使用します。<li data-bbox="601 621 1300 673">■ SOLVED VALUE-BASED は、埋込み合計を含み、親子ディメンション表に格納される階層に使用します。

CWM2_OLAP_LEVEL

CWM2_OLAP_LEVEL パッケージでは、レベルを管理するためのプロシージャが提供されます。

参照： 第2章「CWM2によるOLAPカタログ・メタデータの作成」を参照してください。

この章では、次の項目について説明します。

- レベルの理解
- 例：レベルの作成
- CWM2_OLAP_LEVEL サブプログラムの概要

レベルの理解

レベルは、OLAP メタデータのエンティティです。これは、レベルが名前と所有者で識別される OLAP カタログ内の論理オブジェクトであることを意味します。

ディメンション・メンバーは、ディメンション表またはビューの列にマップするレベルで編成されます。通常、レベルは階層で編成されます。すべてのディメンションには、1つ以上のレベルが必要です。レベルの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

CWM2_OLAP_LEVEL パッケージのプロシージャを使用して、階層の作成、削除およびロック、階層へのレベルの割当て、表示に使用する定義の指定を行います。

レベルを作成する前に、親ディメンションおよび親階層が OLAP カタログにすでに存在している必要があります。

参照：

- 第 11 章「CWM2_OLAP_HIERARCHY」を参照してください。
- レベルおよび OLAP メタデータ・モデルの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

例：レベルの作成

次の文は、PRODUCT_DIM デイメンションの 4 つのレベルを作成し、それらのレベルを PRODUCT_DIM_ROLLUP 階層に割り当てます。

```
execute cwm2_olap_level.create_level
('JSMITH', 'PRODUCT_DIM', 'TOTALPROD_LVL',
 'Total Product', 'All Products', 'Total',
 'Equipment and Parts of standard product hierarchy');
execute cwm2_olap_level.create_level
('JSMITH', 'PRODUCT_DIM', 'PROD_CATEGORY_LVL',
 'Product Category', 'Product Categories', 'Category',
 'Categories of standard product hierarchy');
execute cwm2_olap_level.create_level
('JSMITH', 'PRODUCT_DIM', 'PROD_SUBCATEGORY_LVL',
 'Product Sub-Category', 'Product Sub-Categories', 'Sub-Category',
 'Sub-Categories of standard product hierarchy');
execute cwm2_olap_level.create_level
('JSMITH', 'PRODUCT_DIM', 'PRODUCT_LVL',
 'Product', 'Products', 'Product',
 'Individual products of standard product hierarchy');

execute cwm2_olap_level.add_level_to_hierarchy
('JSMITH', 'PRODUCT_DIM', 'PRODUCT_DIM_ROLLUP',
 'PRODUCT_LVL', 'PROD_SUBCATEGORY_LVL');
execute cwm2_olap_level.add_level_to_hierarchy
('JSMITH', 'PRODUCT_DIM', 'PRODUCT_DIM_ROLLUP',
 'PROD_SUBCATEGORY_LVL', 'PROD_CATEGORY_LVL');
execute cwm2_olap_level.add_level_to_hierarchy
('JSMITH', 'PRODUCT_DIM', 'PRODUCT_DIM_ROLLUP',
 'PROD_CATEGORY_LVL', 'TOTALPROD_LVL');
execute cwm2_olap_level.add_level_to_hierarchy
('JSMITH', 'PRODUCT_DIM', 'PRODUCT_DIM_ROLLUP', 'TOTALPROD_LVL');
```

CWM2_OLAP_LEVEL サブプログラムの概要

表 12-1 CWM2_OLAP_LEVEL サブプログラム

サブプログラム	説明
ADD_LEVEL_TO_HIERARCHY プロシージャ (12-3 ページ)	階層にレベルを追加します。
CREATE_LEVEL プロシージャ (12-4 ページ)	レベルを作成します。
DROP_LEVEL プロシージャ (12-5 ページ)	レベルを削除します。
LOCK_LEVEL プロシージャ (12-5 ページ)	更新のためにレベルのメタデータをロックします。
REMOVE_LEVEL_FROM_HIERARCHY プロシージャ (12-6 ページ)	階層からレベルを削除します。
SET_DESCRIPTION プロシージャ (12-7 ページ)	レベルに説明を設定します。
SET_DISPLAY_NAME プロシージャ (12-7 ページ)	レベルに表示名を設定します。
SET_LEVEL_NAME プロシージャ (12-8 ページ)	レベルに名前を設定します。
SET_PLURAL_NAME プロシージャ (12-8 ページ)	レベルに複数名を設定します。
SET_SHORT_DESCRIPTION プロシージャ (12-9 ページ)	レベルに簡単な説明を設定します。

ADD_LEVEL_TO_HIERARCHY プロシージャ

このプロシージャは、階層にレベルを追加します。

構文

```
ADD_LEVEL_TO_HIERARCHY (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    hierarchy_name     IN    VARCHAR2,
    level_name         IN    VARCHAR2,
    parent_level_name  IN    VARCHAR2 DEFAULT NULL);
```

パラメータ

表 12-2 ADD_LEVEL_TO_HIERARCHY プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。
level_name	階層に追加するレベルの名前。
parent_level_name	階層内のレベルの親の名前。親を指定しない場合、追加したレベルは階層のルートになります。

CREATE_LEVEL プロシージャ

このプロシージャは、OLAP カタログに新しいレベルを作成します。

レベル作成の一部として、説明および表示プロパティを指定する必要があります。レベルの作成後、CWM2_OLAP_LEVEL パッケージの他のプロシージャをコールして、これらのプロパティを上書きできます。

構文

```
CREATE_LEVEL (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    level_name         IN    VARCHAR2,
    display_name       IN    VARCHAR2,
    plural_name        IN    VARCHAR2,
    short_description  IN    VARCHAR2,
    description        IN    VARCHAR2);
```

パラメータ

表 12-3 CREATE_LEVEL プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
level_name	レベルの名前。
display_name	レベルの表示名。

表 12-3 CREATE_LEVEL プロシージャのパラメータ (続き)

パラメータ	説明
plural_name	レベルの複数名。
short_description	レベルの簡単な説明。
description	レベルの説明。

DROP_LEVEL プロシージャ

このプロシージャは、OLAP カタログからレベルを削除します。関連するすべてのレベル属性も削除されます。

構文

```
DROP_LEVEL (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    level_name         IN    VARCHAR2);
```

パラメータ

表 12-4 DROP_LEVEL プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
level_name	レベルの名前。

LOCK_LEVEL プロシージャ

このプロシージャは、CWM2 モデル表のレベルを識別する行でデータベース・ロックを取得して、更新のためにレベルのメタデータをロックします。

構文

```
LOCK_LEVEL (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    level_name         IN    VARCHAR2,
    wait_for_lock      IN    BOOLEAN DEFAULT FALSE);
```

パラメータ

表 12-5 LOCK_LEVEL プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
level_name	レベルの名前。
wait_for_lock	(オプション) レベルが他のユーザーによってすでにロックされている場合、レベルが使用可能になるまで待機するかどうかを選択します。このパラメータに値を指定しない場合、プロシージャはロックの取得を待機しません。

REMOVE_LEVEL_FROM_HIERARCHY プロシージャ

このプロシージャは、階層からレベルを削除します。

構文

```
REMOVE_LEVEL_FROM_HIERARCHY (
    dimension_owner    IN  VARCHAR2,
    dimension_name     IN  VARCHAR2,
    hierarchy_name     IN  VARCHAR2,
    level_name         IN  VARCHAR2);
```

パラメータ

表 12-6 REMOVE_LEVEL_FROM_HIERARCHY プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。
level_name	階層から削除するレベルの名前。

SET_DESCRIPTION プロシージャ

このプロシージャは、レベルに説明を設定します。

構文

```
SET_DESCRIPTION (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    level_name         IN    VARCHAR2,
    description        IN    VARCHAR2);
```

パラメータ

表 12-7 SET_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
level_name	レベルの名前。
description	レベルの説明。

SET_DISPLAY_NAME プロシージャ

このプロシージャは、レベルに表示名を設定します。

構文

```
SET_DISPLAY_NAME (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    level_name         IN    VARCHAR2,
    display_name       IN    VARCHAR2);
```

パラメータ

表 12-8 SET_DISPLAY_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。

表 12-8 SET_DISPLAY_NAME プロシージャのパラメータ (続き)

パラメータ	説明
level_name	レベルの名前。
display_name	レベルの表示名。

SET_LEVEL_NAME プロシージャ

このプロシージャは、レベルに名前を設定します。

構文

```
SET_LEVEL_NAME (
    dimension_owner  IN  VARCHAR2,
    dimension_name   IN  VARCHAR2,
    level_name       IN  VARCHAR2,
    set_level_name   IN  VARCHAR2);
```

パラメータ

表 12-9 SET_LEVEL_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
level_name	レベルの元の名前。
set_level_name	レベルの新しい名前。

SET_PLURAL_NAME プロシージャ

このプロシージャは、レベルの複数名を設定します。

構文

```
SET_PLURAL_NAME (
    dimension_owner  IN  VARCHAR2,
    dimension_name   IN  VARCHAR2,
    level_name       IN  VARCHAR2,
    plural_name      IN  VARCHAR2);
```

パラメータ

表 12-10 SET_PLURAL_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
level_name	レベルの名前。
plural_name	レベルの複数名。

SET_SHORT_DESCRIPTION プロシージャ

このプロシージャは、レベルに簡単な説明を設定します。

構文

```
SET_SHORT_DESCRIPTION (
    dimension_owner      IN   VARCHAR2,
    dimension_name      IN   VARCHAR2,
    level_name          IN   VARCHAR2,
    short_description   IN   VARCHAR2);
```

パラメータ

表 12-11 SET_SHORT_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
level_name	レベルの名前。
short_description	レベルの簡単な説明。

CWM2_OLAP_LEVEL_ATTRIBUTE

CWM2_OLAP_LEVEL_ATTRIBUTE パッケージでは、レベル属性を管理するためのプロシージャが提供されます。

参照： 第2章「CWM2によるOLAPカタログ・メタデータの作成」を参照してください。

この章では、次の項目について説明します。

- レベル属性の理解
- 例：レベル属性の作成
- CWM2_OLAP_LEVEL_ATTRIBUTE サブプログラムの概要

レベル属性の理解

レベル属性は、OLAP メタデータのエンティティです。これは、レベル属性が名前と所有者で識別される OLAP カタログ内の論理オブジェクトであることを意味します。

レベル属性は、レベルおよびディメンション属性の子エンティティです。レベル属性は、関連するレベルの定義を格納します。たとえば、製品識別子を含むレベルは、各製品の色情報を含む関連付けられたレベル属性を持つ場合があります。

各レベル属性は、ディメンション表の列にマップされます。レベル属性の列は、関連付けられたレベルの列（または複数の列）と同じ表内にある必要があります。レベル属性の詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

CWM2_OLAP_LEVEL_ATTRIBUTE パッケージのプロシージャを使用して、レベル属性の作成、削除およびロック、レベルおよびディメンション属性へのレベル属性の割当て、表示に使用する定義の指定を行います。

一部のレベル属性の名前は CWM2 で特別な意味を持つため予約済です。予約済のレベル属性は、同じ名前の予約済のディメンション属性に関連付けられます。予約済のレベル属性は、特定の情報を含む列にマップされます。表 13-1 に、予約済のレベル属性を示します。

表 13-1 予約済のレベル属性

ディメンション属性	説明
Long Description	ディメンション・メンバーの詳細な説明。
Short Description	ディメンション・メンバーの簡単な説明。
End Date	時間ディメンションでの特定の時間間隔の最終日 (必須)。
Time Span	時間ディメンションでの特定の時間間隔の日数 (必須)。
Prior Period	時間ディメンションでのこの時間間隔の以前の時間間隔。
Year Ago Period	時間ディメンションでのこの時間間隔の 1 年前の時間間隔。
ET Key	埋込み合計ディメンションでの埋込み合計キー。ディメンション表の行で最低レベルのディメンション・メンバーを識別します (必須)。
Parent ET Key	埋込み合計ディメンションでの ET キーの親であるディメンション・メンバー (必須)。
Grouping ID	埋込み合計ディメンションでのグルーピング ID (GID)。ディメンション表の行の階層レベルを識別します (必須)。
Parent Grouping ID	埋込み合計ディメンションでのグルーピング ID の親であるディメンション・メンバー (必須)。

レベル属性を作成する前に、親ディメンション、親レベルおよび親ディメンション属性は、OLAP カタログにすでに存在している必要があります。

参照：

- [第 10 章「CWM2_OLAP_DIMENSION_ATTRIBUTE」](#) を参照してください。
- レベル属性および OLAP メタデータ・モデルの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

例：レベル属性の作成

次の文は、最低レベルにカラー属性、PRODUCT_DIM ディメンションの 4 つのすべてのレベルに詳細な説明を作成します。

```
execute cwm2_olap_level_attribute.create_level_attribute
('JSMITH', 'PRODUCT_DIM', 'Product Color', 'PRODUCT_LVL', 'Product Color',
'PROD_STD_COLOR', 'Prod Color', 'Product Color');
```

```
execute cwm2_olap_level_attribute.create_level_attribute
('JSMITH', 'PRODUCT_DIM', 'Long Description', 'PRODUCT_LVL',
```

```
'Long Description','PRODUCT_STD_LLABEL', 'Product',
'Long Labels for individual products of the PRODUCT hierarchy', TRUE);

execute cwm2_olap_level_attribute.create_level_attribute
('JSMITH', 'PRODUCT_DIM', 'Long Description', 'PROD_SUBCATEGORY_LVL',
'Long Description', 'PROD_STD_LLABEL', 'Product Sub Category',
'Long Labels for subcategories of the PRODUCT hierarchy', TRUE);

execute cwm2_olap_level_attribute.create_level_attribute
('JSMITH', 'PRODUCT_DIM', 'Long Description', 'PROD_CATEGORY_LVL',
'Long Description', 'PROD_STD_LLABEL', 'Product Category',
'Long Labels for categories of the PRODUCT hierarchy', TRUE);

execute cwm2_olap_level_attribute.create_level_attribute
('JSMITH', 'PRODUCT_DIM', 'Long Description', 'TOTALPROD_LVL',
'Long Description', 'PROD_STD_LLABEL', 'Total Product',
'Long Labels for total of the PRODUCT hierarchy', TRUE);
```

CWM2_OLAP_LEVEL_ATTRIBUTE サブプログラムの概要

表 13-2 CWM2_OLAP_LEVEL_ATTRIBUTE サブプログラム

サブプログラム	説明
CREATE_LEVEL_ATTRIBUTE プロシージャ (13-4 ページ)	レベル属性を作成します。
DROP_LEVEL_ATTRIBUTE プロシージャ (13-6 ページ)	レベル属性を削除します。
LOCK_LEVEL_ATTRIBUTE プロシージャ (13-6 ページ)	更新のためにレベル属性のメタデータをロックします。
SET_DESCRIPTION プロシージャ (13-7 ページ)	レベル属性に説明を設定します。
SET_DISPLAY_NAME プロシージャ (13-8 ページ)	レベル属性に表示名を設定します。
SET_LEVEL_ATTRIBUTE_NAME プロシージャ (13-8 ページ)	レベル属性に名前を設定します。
SET_SHORT_DESCRIPTION プロシージャ (13-10 ページ)	レベル属性に簡単な説明を設定します。

CREATE_LEVEL_ATTRIBUTE プロシージャ

このプロシージャは、OLAP カタログに新しいレベル属性を作成し、レベル属性をレベルおよびディメンション属性に関連付けます。

レベル属性が予約済の場合、予約済の名前をレベル属性の名前として、または指定する名前に関連付けられた型として指定できます。表 13-1 「[予約済のレベル属性](#)」に、予約済のレベル属性を示します。

レベル属性の作成の一部として、説明および表示プロパティを指定する必要があります。レベル属性の作成後、CWM2_OLAP_LEVEL_ATTRIBUTE パッケージの他のプロシージャをコールして、これらのプロパティを上書きできます。

構文

```
CREATE_LEVEL_ATTRIBUTE (
    dimension_owner          IN    VARCHAR2,
    dimension_name           IN    VARCHAR2,
    dimension_attribute_name IN    VARCHAR2,
    level_name               IN    VARCHAR2,
```

```

level_attribute_name    IN  VARCHAR2,
display_name           IN  VARCHAR2,
short_description      IN  VARCHAR2,
description            IN  VARCHAR2,
type                  IN  VARCHAR2          );
use_name_as_type       IN  BOOLEAN DEFAULT FALSE);

```

パラメータ

表 13-3 CREATE_LEVEL_ATTRIBUTE プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	このレベル属性を含むディメンション属性の名前。
level_name	レベルの名前。
level_attribute_name	レベル属性の名前。
display_name	レベル属性の表示名。
short_description	レベル属性の簡単な説明。
description	レベル属性の説明。
type または use_name_as_type	<p>この引数には次のいずれかを指定できます。</p> <ul style="list-style-type: none"> ■ type 値が表 13-1 「予約済のレベル属性」 の予約済の名前の 1 つである VARCHAR2 引数。予約済のレベル属性に独自の名前を作成する場合、この引数を指定します。 ■ use_name_as_type BOOLEAN 引数（デフォルトは FALSE）。この引数は、レベル属性の名前が予約済の名前かどうかを指定します。この引数が TRUE の場合、level_attribute_name 引数の値は、表 13-1 「予約済のレベル属性」 の予約済の名前である必要があります。 <p>この引数の値を指定しない場合、レベル属性は予約されません。</p>

DROP_LEVEL_ATTRIBUTE プロシージャ

このプロシージャは、OLAP カタログからレベル属性を削除します。

構文

```
DROP_LEVEL_ATTRIBUTE (
    dimension_owner          IN   VARCHAR2,
    dimension_name           IN   VARCHAR2,
    dimension_attribute_name IN   VARCHAR2,
    level_name               IN   VARCHAR2,
    level_attribute_name     IN   VARCHAR2);
```

パラメータ

表 13-4 DROP_LEVEL_ATTRIBUTE プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
level_name	レベルの名前。
level_attribute_name	レベル属性の名前。

LOCK_LEVEL_ATTRIBUTE プロシージャ

このプロシージャは、CWM2 モデル表のレベル属性を識別する行でデータベース・ロックを取得して、更新のためにレベル属性のメタデータをロックします。

構文

```
LOCK_LEVEL_ATTRIBUTE (
    dimension_owner          IN   VARCHAR2,
    dimension_name           IN   VARCHAR2,
    dimension_attribute_name IN   VARCHAR2,
    level_name               IN   VARCHAR2,
    level_attribute_name     IN   VARCHAR2,
    wait_for_lock            IN   BOOLEAN DEFAULT FALSE);
```

パラメータ

表 13-5 LOCK_LEVEL_ATTRIBUTE プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
level_name	レベルの名前。
level_attribute_name	レベル属性の名前。
wait_for_lock	(オプション) レベル属性が他のユーザーによってすでにロックされている場合、レベル属性が使用可能になるまで待機するかどうかを選択します。このパラメータに値を指定しない場合、プロシージャはロックの取得を待機しません。

SET_DESCRIPTION プロシージャ

このプロシージャは、レベル属性に説明を設定します。

構文

```
SET_DESCRIPTION (
    dimension_owner          IN   VARCHAR2,
    dimension_name           IN   VARCHAR2,
    dimension_attribute_name IN   VARCHAR2,
    level_name               IN   VARCHAR2,
    level_attribute_name     IN   VARCHAR2,
    description              IN   VARCHAR2);
```

パラメータ

表 13-6 SET_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
level_name	レベルの名前。

表 13-6 SET_DESCRIPTION プロシージャのパラメータ (続き)

パラメータ	説明
level_attribute_name	レベル属性の名前。
description	レベル属性の説明。

SET_DISPLAY_NAME プロシージャ

このプロシージャは、レベル属性に表示名を設定します。

構文

```
SET_DISPLAY_NAME (
    dimension_owner          IN   VARCHAR2,
    dimension_name           IN   VARCHAR2,
    dimension_attribute_name IN   VARCHAR2,
    level_name               IN   VARCHAR2,
    level_attribute_name     IN   VARCHAR2,
    display_name             IN   VARCHAR2);
```

パラメータ

表 13-7 SET_DISPLAY_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
level_name	レベルの名前。
level_attribute_name	レベル属性の名前。
display_name	レベル属性の表示名。

SET_LEVEL_ATTRIBUTE_NAME プロシージャ

このプロシージャは、レベル属性に名前を設定します。

レベル属性が予約済の場合、予約済の名前をレベル属性の名前として、または指定する名前に関連付けられた型として指定できます。表 13-1 「予約済のレベル属性」に、予約済のレベル属性を示します。

構文

```
SET_LEVEL_ATTRIBUTE_NAME (
    dimension_owner          IN   VARCHAR2,
    dimension_name           IN   VARCHAR2,
    dimension_attribute_name IN   VARCHAR2,
    level_name               IN   VARCHAR2,
    level_attribute_name     IN   VARCHAR2,
    set_level_attribute_name IN   VARCHAR2,
    type                     IN   VARCHAR2          );
    use_name_as_type        IN   BOOLEAN DEFAULT FALSE);
```

パラメータ

表 13-8 SET_LEVEL_ATTRIBUTE_NAME プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
level_name	レベルの名前。
level_attribute_name	レベル属性の元の名前。
set_level_attribute_name	レベル属性の新しい名前。
type または use_name_as_type	<p>この引数には次のいずれかを指定できます。</p> <ul style="list-style-type: none"> ■ type 値が表 13-1 「予約済のレベル属性」 の予約済の名前の 1 つである VARCHAR2 引数。予約済のレベル属性に独自の名前を作成する場合、この引数を指定します。 ■ use_name_as_type BOOLEAN 引数（デフォルトは FALSE）。この引数は、レベル属性の名前が予約済の名前かどうかを指定します。この引数が TRUE の場合、level_attribute_name 引数の値は、表 13-1 「予約済のレベル属性」 の予約済の名前である必要があります。 <p>この引数の値を指定しない場合、レベル属性は予約されません。</p>

SET_SHORT_DESCRIPTION プロシージャ

このプロシージャは、レベル属性に簡単な説明を設定します。

構文

```
SET_SHORT_DESCRIPTION (  
    dimension_owner          IN  VARCHAR2,  
    dimension_name          IN  VARCHAR2,  
    dimension_attribute_name IN  VARCHAR2,  
    level_name              IN  VARCHAR2,  
    level_attribute_name    IN  VARCHAR2,  
    short_description       IN  VARCHAR2);
```

パラメータ

表 13-9 SET_SHORT_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
level_name	レベルの名前。
level_attribute_name	レベル属性の名前。
short_description	レベル属性の簡単な説明。

CWM2_OLAP_MEASURE

CWM2_OLAP_MEASURE パッケージでは、メジャーを管理するためのプロシージャが提供されません。

参照： 第2章「CWM2によるOLAPカタログ・メタデータの作成」を参照してください。

この章では、次の項目について説明します。

- [メジャーの理解](#)
- [例：メジャーの作成](#)
- [CWM2_OLAP_MEASURE サブプログラムの概要](#)

メジャーの理解

メジャーは、OLAP メタデータのエンティティです。これは、メジャーが名前と所有者で識別される OLAP カタログ内の論理オブジェクトであることを意味します。

メジャーは、ファクト表に格納されたデータを表します。ファクト表は、リレーショナル表またはビューです。ビューは、アナリティック・ワークスペースに格納されているデータを参照する場合があります。

メジャーは、キューブのコンテキスト内に存在します。キューブは、メジャーのデータのディメンションを完全に指定します。メジャーの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

CWM2_OLAP_MEASURE パッケージのプロシージャを使用して、メジャーの作成、削除およびロック、キューブへのメジャーの割当て、表示に使用する定義の指定を行います。

メジャーを作成する前に、親キューブが OLAP カタログにすでに存在している必要があります。

参照：

- 第8章「CWM2_OLAP_CUBE」を参照してください。
- メジャーおよびOLAP メタデータ・モデルの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

例：メジャーの作成

次の文は、SALES_CUBE キューブに SALES_AMOUNT および SALES_QUANTITY メジャーを作成します。

```
execute cwm2_olap_measure.create_measure
('JSMITH', 'SALES_CUBE', 'SALES_AMOUNT', 'Sales Amount',
 '$ Sales', 'Dollar Sales');
execute cwm2_olap_measure.create_measure
('JSMITH', 'SALES_CUBE', 'SALES_QUANTITY', 'Sales Quantity',
 'Sales Quantity', 'Quantity of Items Sold');
```

CWM2_OLAP_MEASURE サブプログラムの概要

表 14-1 CWM2_OLAP_MEASURE サブプログラム

サブプログラム	説明
CREATE_MEASURE プロシージャ (14-3 ページ)	メジャーを作成します。
DROP_MEASURE プロシージャ (14-4 ページ)	メジャーを削除します。
LOCK_MEASURE プロシージャ (14-4 ページ)	更新のためにメジャーのメタデータをロックします。
SET_DESCRIPTION プロシージャ (14-4 ページ)	メジャーに説明を設定します。
SET_DISPLAY_NAME プロシージャ (14-6 ページ)	メジャーに表示名を設定します。
SET_MEASURE_NAME プロシージャ (14-6 ページ)	メジャーに名前を設定します。
SET_SHORT_DESCRIPTION プロシージャ (14-7 ページ)	メジャーに簡単な説明を設定します。

CREATE_MEASURE プロシージャ

このプロシージャは、OLAP カタログに新しいメジャーを作成します。

メジャーは、キューブのコンテキスト内でのみ作成できます。メジャーを作成する前に、キューブが存在している必要があります。

メジャー作成の一部として、説明および表示プロパティを指定する必要があります。メジャーの作成後、このパッケージの他のプロシージャをコールして、これらのプロパティを上書きできます。

構文

```
CREATE_MEASURE (
    cube_owner          IN  VARCHAR2,
    cube_name           IN  VARCHAR2,
    measure_name        IN  VARCHAR2,
    display_name        IN  VARCHAR2,
    short_description   IN  VARCHAR2,
    description         IN  VARCHAR2);
```

パラメータ

表 14-2 CREATE_MEASURE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
measure_name	メジャーの名前。
display_name	メジャーの表示名。
short_description	メジャーの簡単な説明。
description	メジャーの説明。

DROP_MEASURE プロシージャ

このプロシージャは、キューブからメジャーを削除します。

構文

```
DROP_MEASURE (
    cube_owner      IN   VARCHAR2,
    cube_name       IN   VARCHAR2,
    measure_name    IN   VARCHAR2);
```

パラメータ

表 14-3 DROP_MEASURE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
measure_name	キューブから削除するメジャーの名前。

LOCK_MEASURE プロシージャ

このプロシージャは、CWM2 モデル表のメジャーを識別する行でデータベース・ロックを取得して、更新のためにメジャーのメタデータをロックします。

構文

```
LOCK_MEASURE (
```

```

cube_owner      IN  VARCHAR2,
cube_name       IN  VARCHAR2,
measure_name    IN  VARCHAR2,
wait_for_lock   IN  BOOLEAN DEFAULT FALSE);

```

パラメータ

表 14-4 LOCK_MEASURE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
measure_name	ロックするメジャーの名前。
wait_for_lock	(オプション) メジャーが他のユーザーによってすでにロックされている場合、メジャーが使用可能になるまで待機するかどうかを選択します。このパラメータに値を指定しない場合、プロシージャはロックの取得を待機しません。

SET_DESCRIPTION プロシージャ

このプロシージャは、メジャーに説明を設定します。

構文

```

SET_DESCRIPTION (
    cube_owner      IN  VARCHAR2,
    cube_name       IN  VARCHAR2,
    measure_name    IN  VARCHAR2,
    description     IN  VARCHAR2);

```

パラメータ

表 14-5 SET_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
measure_name	メジャーの名前。
description	メジャーの説明。

SET_DISPLAY_NAME プロシージャ

このプロシージャは、メジャーに表示名を設定します。

構文

```
SET_DISPLAY_NAME (
    cube_owner      IN   VARCHAR2,
    cube_name       IN   VARCHAR2,
    measure_name    IN   VARCHAR2,
    display_name    IN   VARCHAR2);
```

パラメータ

表 14-6 SET_DISPLAY_NAME プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
measure_name	メジャーの名前。
display_name	メジャーの表示名。

SET_MEASURE_NAME プロシージャ

このプロシージャは、メジャーに名前を設定します。

構文

```
SET_MEASURE_NAME (
    cube_owner      IN   VARCHAR2,
    cube_name       IN   VARCHAR2,
    measure_name    IN   VARCHAR2,
    set_cube_name   IN   VARCHAR2);
```

パラメータ

表 14-7 SET_MEASURE_NAME プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。

表 14-7 SET_MEASURE_NAME プロシージャのパラメータ (続き)

パラメータ	説明
measure_name	メジャーの元の名前。
set_cube_name	メジャーの新しい名前。

SET_SHORT_DESCRIPTION プロシージャ

このプロシージャは、メジャーに簡単な説明を設定します。

構文

```
SET_SHORT_DESCRIPTION (
    cube_owner          IN   VARCHAR2,
    cube_name           IN   VARCHAR2,
    measure_name        IN   VARCHAR2,
    short_description   IN   VARCHAR2);
```

パラメータ

表 14-8 SET_SHORT_DESCRIPTION プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
measure_name	メジャーの名前。
short_description	メジャーの簡単な説明。

CWM2_OLAP_METADATA_REFRESH

CWM2_OLAP_METADATA_REFRESH パッケージでは、OLAP API が使用するキャッシュされたメタデータ表をリフレッシュするためのプロシージャが提供されます。

参照：

- 2-12 ページの「[OLAP メタデータの検証およびコミット](#)」を参照してください。
- [第 5 章「OLAP カタログ・メタデータのビュー](#)」を参照してください。
- [第 3 章「アクティブ・カタログ・ビュー](#)」を参照してください。

この章では、次の項目について説明します。

- [キャッシュされた OLAP カタログ・メタデータのビュー](#)
- [キャッシュされたアクティブ・カタログ・メタデータのビュー](#)
- [CWM2_OLAP_METADATA_REFRESH サブプログラムの概要](#)

キャッシュされた OLAP カタログ・メタデータのビュー

接頭辞 MRV_OLAP2 が名前に付けられた Metadata Reader のビューは、OLAP カタログ・メタデータの一連のキャッシュ表に対する読み込み API を表示します。これらのビューおよび表は、OLAP API Metadata Reader による問合せが容易になるように構成されています。

キャッシュ表は、OLAP カタログ・モデル表とは異なり、メタデータに変更が加えられても自動的にリフレッシュされません。キャッシュ表をリフレッシュするには、MR_REFRESH プロシージャをコールする必要があります。

OLAP カタログ・モデル表のビュー（[第 5 章「OLAP カタログ・メタデータのビュー](#)」を参照）には、接頭辞 ALL_OLAP2 が付けられています。ほとんどの MRV_OLAP2 ビューは、対応する ALL_OLAP2 ビューと同じ名前および列構造を持ちます。

注意： MRV_OLAP2 ビューの基礎となる表が OLAP カタログ・メタデータ表と整合性がない場合、OLAP API がメタデータにアクセスできなくなることがあります。

キャッシュされたアクティブ・カタログ・メタデータのビュー

MRV_OLAP2_AW ビューは、アクティブ・カタログの一連のキャッシュ表に対する読み込み API を表示します。これらのビューおよび表は、問合せのパフォーマンスが向上するように構成されています。

キャッシュ表は、アクティブ・カタログとは異なり、アナリティック・ワークスペースに変更が加えられても自動的にリフレッシュされません。キャッシュ表をリフレッシュするには、MR_AC_REFRESH プロシージャをコールする必要があります。

アクティブ・カタログ・ビュー（第3章「アクティブ・カタログ・ビュー」を参照）には、接頭辞 ALL_OLAP2_AW が付けられています。MRV_OLAP2_AW ビューは、対応する ALL_OLAP2_AW ビューと同じ名前および列構造を持ちます。

注意： MRV_OLAP2_AW ビューの基礎となる表が、アナリティック・ワークスペースのスタンダード・フォームのメタデータと整合性がない場合、これらのビューから正確な情報を得られなくなることがあります。

CWM2_OLAP_METADATA_REFRESH サブプログラムの概要

表 15-1 CWM2_OLAP_METADATA_REFRESH サブプログラム

サブプログラム	説明
MR_REFRESH プロシージャ	OLAP API Metadata Reader が使用する、キャッシュされたメタデータ表をリフレッシュします。
MR_AC_REFRESH プロシージャ	キャッシュされたアクティブ・カタログ・メタデータ表をリフレッシュします。

MR_REFRESH プロシージャ

このプロシージャは、MRV_OLAP2 ビューの基礎となるメタデータ表をリフレッシュします。OLAP API Metadata Reader による問合せをサポートするには、これらの表を更新する必要があります。

OLAP API の OLAP カタログ・メタデータを作成、削除および更新するスクリプトの最後の文として MR_REFRESH を実行します。

Enterprise Manager で OLAP カタログ・メタデータを作成または変更した後、MR_REFRESH を実行します。

MR_REFRESH プロシージャには COMMIT が含まれます。メタデータ表への更新は、データベースに永続的に保存されます。

構文

```
MR_REFRESH;
```

MR_AC_REFRESH プロシージャ

このプロシージャは、MRV_OLAP2_AW ビューの基礎となるメタデータ表をリフレッシュします。アクティブ・カタログのキャッシュ表に対する問合せをサポートするには、これらの表を更新する必要があります。

MR_AC_REFRESH は、DBMS_AWM パッケージを使用してアナリティック・ワークスペースを作成、変更または有効化するスクリプトの最後の文として実行します。

MR_AC_REFRESH プロシージャには COMMIT が含まれます。

構文

```
MR_AC_REFRESH;
```

CWM2_OLAP_PC_TRANSFORM

CWM2_OLAP_PC_TRANSFORM パッケージには、親子ディメンション表から解決済のレベルベース・ディメンション表を作成する SQL スクリプトを生成するためのプロシージャが含まれます。

スクリプトを実行して新しい表を作成した後、OLAP API アプリケーションがディメンションにアクセスできるように OLAP メタデータを定義できます。

参照：

- OLAP カタログ・メタデータでサポートされるデータ・ウェアハウス表のタイプについては、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。
- ディメンション階層の OLAP カタログ・メタデータの作成方法については、[第 11 章「CWM2_OLAP_HIERARCHY」](#)を参照してください。

この章では、次の項目について説明します。

- [前提条件](#)
- [親子ディメンション](#)
- [解決済のレベルベース・ディメンション](#)
- [例：解決済のレベルベース・ディメンション表の作成](#)
- [CWM2_OLAP_PC_TRANSFORM サブプログラムの概要](#)

前提条件

CWM2_OLAP_PC_TRANSFORM.CREATE_SCRIPT プロシージャを実行する前に、RDBMS でファイルへの書込みが可能であることを確認してください。ディレクトリを指定する際は、自身のユーザー ID に適切なアクセス権が付与されているディレクトリ・オブジェクトか、またはインスタンスの UTL_FILE_DIR 初期化パラメータで設定されているパスを使用できます。

親子ディメンション表が存在し、CWM2_OLAP_PC_TRANSFORM.CREATE_SCRIPT プロシージャにアクセス可能である必要があります。

親子ディメンション

親子ディメンション表は、階層関係が親列と子列によって定義される表です。階層は、親列と子列の値の関係で定義されるため、親子ディメンションは**値ベースの階層**と呼ばれる場合があります。

親子ディメンション表の列の例

次の例では、子列と親列の値の関係を示します。子の属性である説明列も示します。

CHILD	PARENT	DESCRIPTION
-----	-----	-----
World		World
USA	World	United States of America
Northeast	USA	North East Region
Southeast	USA	South East Region
MA	Northeast	Massachusetts
Boston	MA	Boston, MA
Burlington	MA	Burlington, MA
NY	Northeast	New York State
New York City	NY	New York, NY
GA	Southeast	Georgia
Atlanta	GA	Atlanta, GA
Canada	World	Canada

OLAP カタログ・メタデータを作成して、親子ディメンションを表示する場合は、階層の `solved_code` を 'SOLVED VALUE-BASED' に設定します (第 11 章「[CWM2_OLAP_HIERARCHY](#)」を参照)。

注意： OLAP カタログ・メタデータを作成して値ベースの階層を表示できますが、このタイプの階層は OLAP API を使用するアプリケーションにアクセスできません。

解決済のレベルベース・ディメンション

OLAP_PC_TRANSFORM.CREATE_SCRIPT によって生成されるスクリプトは、親子表からの値をレベルに格納する表を作成します。

その結果生成されるレベルベース・ディメンション表には、すべての行のすべてのレベル値の関係が含まれます。このディメンションに関連するファクト表には、レベルの組合せに対する埋込み合計が含まれるため、このタイプのディメンション表は**解決済**です。

OLAP API から親子ディメンション表へのアクセスを可能にする場合、親子ディメンション表を解決済のレベルベース・ディメンション表に変換する必要があります。OLAP API では、ディメンションにレベルがあり、GID（グルーピング ID）列および埋込み合計（ET）キー列を含んでいる必要があります。GID および ET キー列については、「例：解決済のレベルベース・ディメンション表の作成」を参照してください。

次に、「親子ディメンション表の列の例」に示した親子関係が解決済レベルとしてどのように表されるかを示します。

TOT_GEOG	COUNTRY	REGION	STATE	CITY	DESCRIPTION
World	USA	Northeast	MA	Boston	Boston, MA
World	USA	Northeast	MA	Burlington	Burlington, MA
World	USA	Northeast	NY	New York City	New York, NY
World	USA	Southeast	GA	Atlanta	Atlanta, GA
World	USA	Northeast	MA		Massachusetts
World	USA	Northeast	NY		New York State
World	USA	Southeast	GA		Georgia
World	USA	Northeast			North East Region
World	USA	Southeast			South East Region
World	USA				United States of America
World	Canada				Canada
World					World

OLAP カタログ・メタデータを作成して、解決済のレベルベース・ディメンション階層を表示する場合は、`solved_code` を 'SOLVED LEVEL-BASED' に設定します（第 11 章「CWM2_OLAP_HIERARCHY」を参照）。

例：解決済のレベルベース・ディメンション表の作成

親子ディメンション表に、「親子ディメンション表の列の例」に示した PARENT および CHILD 列がある場合、次のようなコマンドを使用すると、解決済のレベルベース・ディメンション表にこれらの列を表示できます。

```
execute cwm2_olap_pc_transform.create_script
  ('/dat1/scripts/myscripts' ,
   'jsmith' ,
   'input_tbl' ,
   'PARENT' ,
   'CHILD' ,
   'output_tbl' ,
   'jsmith_data');
```

この文は、`/dat1/scripts/myscripts` ディレクトリでスクリプトを作成します。スクリプトは、親子表 `input_tbl` を解決済のレベルベース表 `output_tbl` に変換します。両方の表は、`jsmith` スキーマの `jsmith_data` 表領域内にあります。

次のコマンドで結果スクリプトを実行できます。

```
@create_output_tbl
```

次のコマンドで結果表を表示できます。

```
select * from output_tbl_view
```

結果表は、次のように表示されます。

GID	SHORT_DESC	LONG_DESC	CHILD1	CHILD2	CHILD3	CHILD4	CHILD5
0	Boston	Boston	World	USA	Northeast	MA	Boston
0	Burlington	Burlington	World	USA	Northeast	MA	Burlington
0	New York City	New York City	World	USA	Northeast	NY	New York City
0	Atlanta	Atlanta	World	USA	Southeast	GA	Atlanta
1	MA	MA	World	USA	Northeast	MA	
1	NY	MA	World	USA	Northeast	NY	
1	GA	GA	World	USA	Southeast	GA	
3	Northeast	Northeast	World	USA	Northeast		
3	Southeast	Southeast	World	USA	Southeast		
7	USA	USA	World	USA			
7	Canada	Canada	World	Canada			
15	World	World	World				

グループID列

OLAP API での要求に応じて、GID 列はスクリプトによって自動的に作成されます。GID は、レベル列の NULL 以外の値に 0、NULL 値に 1 を割り当てて、各行に関連付けられた階層レベルを識別します。その結果生成された 2 進数が GID 値です。たとえば、上位 3 レベルに 0、下位 2 レベルに 1 が割り当てられているため、3 という GID がレベル値が World、USA、Northeast の行に割り当てられます。

CHILD1	CHILD2	CHILD3	CHILD4	CHILD5
World	USA	Northeast		
0	0	0	1	1

埋込み合計キー列

スクリプトによって、詳細な説明および簡単な説明の列が自動的に生成されます。この情報を含む入力表の列がある場合、これらの列を CREATE_SCRIPT プロシージャへのパラメータとして指定できます。

簡単な説明の列を指定しない場合、スクリプトによって列が作成され、各行に表示される最低レベルの子の値が移入されます。詳細な説明の列を指定しない場合、スクリプトによって簡単な説明がレプリケートされます。

OLAP API によって要求された ET キー列は、デフォルトで作成された簡単な説明の列です。

CWM2_OLAP_PC_TRANSFORM サブプログラムの概要

表 16-1 CWM2_OLAP_PC_TRANSFORM

サブプログラム	説明
CREATE_SCRIPT プロシージャ (16-6 ページ)	親子表を埋込み合計表に変換するスクリプトを生成します。

CREATE_SCRIPT プロシージャ

このプロシージャは、親子ディメンション表を埋込み合計ディメンション表に変換するスクリプトを生成します。

構文

```
CREATE_SCRIPT (
    directory          IN  VARCHAR2,
    schema             IN  VARCHAR2,
    pc_table           IN  VARCHAR2,
    pc_parent         IN  VARCHAR2,
    pc_child          IN  VARCHAR2,
    slb_table         IN  VARCHAR2,
    slb_tablespace    IN  VARCHAR2,
    pc_root           IN  VARCHAR2  DEFAULT NULL,
    number_of_levels  IN  NUMBER    DEFAULT NULL,
    level_names       IN  VARCHAR2  DEFAULT NULL,
    short_description IN  VARCHAR2  DEFAULT NULL,
    long_description  IN  VARCHAR2  DEFAULT NULL,
    attribute_names   IN  VARCHAR2  DEFAULT NULL);
```

パラメータ

表 16-2 CREATE_SCRIPT プロシージャのパラメータ

パラメータ	説明
directory	生成されたスクリプトが格納されるディレクトリ。ディレクトリ・オブジェクト、または UTL_FILE_DIR 初期化パラメータで指定されているディレクトリ・パスを指定できます。
schema	親子表を含むスキーマ。このスキーマには、解決済のレベルベース表も含まれます。
pc_table	親子表の名前。

表 16-2 CREATE_SCRIPT プロシージャのパラメータ (続き)

パラメータ	説明
pc_parent	親の値を含む pc_table の列名。
pc_child	子の値を含む pc_table の列名。
slb_table	作成される解決済のレベルベース表の名前。
slb_tablespace	解決済のレベルベース表が作成される表領域の名前。
pc_root	次のいずれかを選択します。 null - 親列の null で親子階層のルートが識別されます。(デフォルト) condition - 親子階層のルートが条件になります。次に例を示します。 'long_des = "All Countries"'
number_of_levels	次のいずれかを選択します。 null - 解決済のレベルベース表のレベル数が、親子表の階層のすべてのレベルになります。(デフォルト) number - 解決済のレベルベース表に作成されるレベル数です。
level_names	次のいずれかを選択します。 null - 解決済のレベルベース表の列名が、レベル数に連結されるソースの子の列名になります。(デフォルト) list - 解決済のレベルベース表のカンマで区切られた列名のリストです。
short_description	次のいずれかを選択します。 null - 親子表には簡単な説明はありません。解決済のレベルベース表の各行にある最高レベルの NULL 以外の子の値が、簡単な説明として使用されます。この値が ET キー列を構成します。(デフォルト) column name - 簡単な説明を含む親子表の列名です。この列が、親子表から解決済のレベルベース表にコピーされます。
long_description	次のいずれかを選択します。 null - 親子表には詳細な説明はありません。簡単な説明が使用されます。(デフォルト) column name - 詳細な説明を含む親子表の列名です。この列が、親子表から解決済のレベルベース表にコピーされます。

表 16-2 CREATE_SCRIPT プロシージャのパラメータ (続き)

パラメータ	説明
attribute_names	次のいずれかを選択します。 null - 親子表には属性はありません。(デフォルト) list - 親子表のカンマで区切られた属性列のリストです。これらの列が、親子表から解決済のレベルベース表にコピーされます。

使用上の注意

1. 解決済のレベルベース表と同じ名前の表がすでに存在する場合、その表はスクリプトによって削除されます。
2. number_of_levels パラメータにレベル数を指定すると、スクリプトを生成するために必要な時間を削減できます。このパラメータに値を指定しない場合、CREATE_SCRIPT プロシージャは親子表からすべてのレベルを計算します。
3. 生成されたスクリプト・ファイルを実行前に変更すると、解決済のレベルベース表の追加の特性を定義できます。

CWM2_OLAP_TABLE_MAP

CWM2_OLAP_TABLE_MAP パッケージでは、OLAP メタデータのエンティティをデータ・ウェアハウスのディメンション表およびファクト表の列にマッピングするためのプロシージャが提供されます。

参照： 第2章「CWM2によるOLAPカタログ・メタデータの作成」を参照してください。

この章では、次の項目について説明します。

- [OLAP メタデータのマッピングの理解](#)
- [例：ディメンションのマッピング](#)
- [例：キューブのマッピング](#)
- [CWM2_OLAP_TABLE_MAP サブプログラムの概要](#)

OLAP メタデータのマッピングの理解

CWM2_OLAP_TABLE_MAP パッケージでは、OLAP メタデータのエンティティをファクト表およびディメンション表の列にリンクし、ファクト表とファクト表に関連付けられたディメンション表の結合関係を確立するためのプロシージャが提供されます。

ディメンション・レベルおよびレベル属性は、ディメンション表の列にマップされます。通常、これらは階層によってマップされます。メジャーは、ファクト表の列にマップされません。

ファクト表とディメンション表の結合関係は、単一ファクト表に格納されている解決済データまたは未解決データに対して指定するか、あるいは単一ファクト表に格納されている、階層の組合せごとの解決済データに対して指定することができます。

参照： 2-10 ページの「[OLAP メタデータのマッピング](#)」を参照してください。

例：ディメンションのマッピング

次の文は、XADEMO.PRODUCT_AW ディメンションの STANDARD 階層の 4 つのレベルを XADEMO_AW_VIEW_PRODUCT ディメンション表の列にマップします。詳細な説明の属性が各レベルにマップされます。

```
execute cwm2_olap_table_map.Map_DimTbl_HierLevel
('XADEMO', 'PRODUCT_AW', 'STANDARD', 'L4',
 'XADEMO', 'XADEMO_AW_VIEW_PRODUCT', 'L4', 'L3');
execute cwm2_olap_table_map.Map_DimTbl_HierLevelAttr
('XADEMO', 'PRODUCT_AW', 'Long Description', 'STANDARD', 'L4',
 'Long Description', 'XADEMO', 'XADEMO_AW_VIEW_PRODUCT', 'PROD_STD_LLABEL');

execute cwm2_olap_table_map.Map_DimTbl_HierLevel
('XADEMO', 'PRODUCT_AW', 'STANDARD', 'L3',
 'XADEMO', 'XADEMO_AW_VIEW_PRODUCT', 'L3', 'L2');
execute cwm2_olap_table_map.Map_DimTbl_HierLevelAttr
('XADEMO', 'PRODUCT_AW', 'Long Description', 'STANDARD', 'L3',
 'Long Description', 'XADEMO', 'XADEMO_AW_VIEW_PRODUCT', 'PROD_STD_LLABEL');

execute cwm2_olap_table_map.Map_DimTbl_HierLevel
('XADEMO', 'PRODUCT_AW', 'STANDARD', 'L2',
 'XADEMO', 'XADEMO_AW_VIEW_PRODUCT', 'L2', 'L1');
execute cwm2_olap_table_map.Map_DimTbl_HierLevelAttr
('XADEMO', 'PRODUCT_AW', 'Long Description', 'STANDARD', 'L2',
 'Long Description', 'XADEMO', 'XADEMO_AW_VIEW_PRODUCT', 'PROD_STD_LLABEL');

execute cwm2_olap_table_map.Map_DimTbl_HierLevel
('XADEMO', 'PRODUCT_AW', 'STANDARD', 'L1',
 'XADEMO', 'XADEMO_AW_VIEW_PRODUCT', 'L1', null);

execute cwm2_olap_table_map.Map_DimTbl_HierLevelAttr
('XADEMO', 'PRODUCT_AW', 'Long Description', 'STANDARD', 'L1',
 'Long Description', 'XADEMO', 'XADEMO_AW_VIEW_PRODUCT', 'PROD_STD_LLABEL');
```

例：キューブのマッピング

次の文は、ディメンション結合キーを XADEMO スキーマの ANALYTIC_CUBE_AW キューブにマップします。結合キーの関係は、4 つのディメンション / 階層の組合せで指定します。

```
PRODUCT_AW/STANDARD
CHANNEL_AW/STANDARD
TIME_AW/YTD
GEOGRAPHY_AW/CONSOLIDATED.
```

ファクト表は、XADEMO_AW_SALES_VIEW_4 と呼ばれます。このファクト表には、最低レベルのデータおよびすべてのレベルの組合せの埋込み合計が格納されます。

```
execute cwm2_olap_table_map.Map_FactTbl_LevelKey
('XADEMO', 'ANALYTIC_CUBE_AW', 'XADEMO', 'XADEMO_AW_SALES_VIEW_4', 'ET',
'DIM:XADEMO.PRODUCT_AW/HIER:STANDARD/GID:PRODUCT_GID/LVL:L4/COL:PRODUCT_ET;
DIM:XADEMO.CHANNEL_AW/HIER:STANDARD/GID:CHANNEL_GID/LVL:STANDARD_1/COL:CHANNEL_ET;
DIM:XADEMO.TIME_AW/HIER:YTD/GID:TIME_YTD_GID/LVL:L3/COL:TIME_YTD_ET;
DIM:XADEMO.GEOGRAPHY_AW/HIER:CONSOLIDATED/GID:GEOG_CONS_GID/LVL:L4/COL:GEOG_CONS_ET;');
```

次の文は、F.SALES_AW メジャーをファクト表の SALES 列にマップします。

```
execute cwm2_olap_table_map.Map_FactTbl_Measure
('XADEMO', 'ANALYTIC_CUBE_AW', 'F.SALES_AW',
'XADEMO', 'XADEMO_AW_SALES_VIEW_4', 'SALES',
'DIM:XADEMO.PRODUCT_AW/HIER:STANDARD/LVL:L4/COL:PRODUCT_ET;
DIM:XADEMO.CHANNEL_AW/HIER:STANDARD/LVL:STANDARD_1/COL:CHANNEL_ET;
DIM:XADEMO.TIME_AW/HIER:YTD/LVL:L3/COL:TIME_YTD_ET;
DIM:XADEMO.GEOGRAPHY_AW/HIER:CONSOLIDATED/LVL:L4/COL:GEOG_CONS_ET;');
```

CWM2_OLAP_TABLE_MAP サブプログラムの概要

表 17-1 CWM2_OLAP_TABLE_MAP

サブプログラム	説明
MAP_DIMTBL_HIERLEVELATTR プロシージャ (17-5 ページ)	階層レベル属性をディメンション表の列にマップします。
MAP_DIMTBL_HIERLEVEL プロシージャ (17-5 ページ)	階層レベルをディメンション表の 1 つ以上の列にマップします。
MAP_DIMTBL_HIERSORTKEY プロシージャ (17-6 ページ)	ディメンション表の列で階層のメンバーをソートします。
MAP_DIMTBL_LEVELATTR プロシージャ (17-7 ページ)	非階層レベル属性をディメンション表の列にマップします。
MAP_DIMTBL_LEVEL プロシージャ (17-8 ページ)	非階層レベルをディメンション表の 1 つ以上の列にマップします。
MAP_FACTTBL_LEVELKEY プロシージャ (17-9 ページ)	キューブのディメンションをファクト表にマップします。
MAP_FACTTBL_MEASURE プロシージャ (17-11 ページ)	メジャーをファクト表の列にマップします。
REMOVEMAP_DIMTBL_HIERLEVELATTR プロシージャ (17-12 ページ)	ディメンション表の列から階層レベル属性のマッピングを削除します。
REMOVEMAP_DIMTBL_HIERLEVEL プロシージャ (17-13 ページ)	ディメンション表の 1 つ以上の列から階層レベルのマッピングを削除します。
REMOVEMAP_DIMTBL_HIERSORTKEY プロシージャ (17-14 ページ)	ディメンション表の列に関連付けられたカスタム・ソート基準を削除します。
REMOVEMAP_DIMTBL_LEVELATTR プロシージャ (17-14 ページ)	ディメンション表の列から非階層レベル属性のマッピングを削除します。
REMOVEMAP_DIMTBL_LEVEL プロシージャ (17-15 ページ)	ディメンション表の 1 つ以上の列から非階層レベルのマッピングを削除します。
REMOVEMAP_FACTTBL_LEVELKEY プロシージャ (17-15 ページ)	ファクト表からキューブのディメンションのマッピングを削除します。
REMOVEMAP_FACTTBL_MEASURE プロシージャ (17-16 ページ)	ファクト表の列からメジャーのマッピングを削除します。

MAP_DIMTBL_HIERLEVELATTR プロシージャ

このプロシージャは、レベル属性をディメンション表の列にマップします。
マップされる属性は、階層のコンテキスト内のレベルに関連付けられます。

構文

```
MAP_DIMTBL_HIERLEVELATTR (
    dimension_owner          IN   VARCHAR2,
    dimension_name           IN   VARCHAR2,
    dimension_attribute_name IN   VARCHAR2,
    hierarchy_name           IN   VARCHAR2,
    level_name               IN   VARCHAR2,
    level_attribute_name     IN   VARCHAR2,
    table_owner              IN   VARCHAR2,
    table_name               IN   VARCHAR2,
    attrcol                  IN   VARCHAR2);
```

パラメータ

表 17-2 MAP_DIMTBL_HIERLEVELATTR プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
hierarchy_name	階層の名前。
level_name	レベルの名前。
level_attribute_name	レベルに関連付けられているレベル属性の名前。
table_owner	ディメンション表の所有者。
table_name	ディメンション表の名前。
attrcol	レベル属性がマップされる必要があるディメンション表の列。

MAP_DIMTBL_HIERLEVEL プロシージャ

このプロシージャは、レベルをディメンション表の1つ以上の列にマップします。
マップされるレベルは、階層のコンテキスト内で識別されます。

構文

```
MAP_DIMTBL_HIERLEVEL (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    hierarchy_name     IN    VARCHAR2,
    level_name         IN    VARCHAR2,
    table_owner        IN    VARCHAR2,
    table_name         IN    VARCHAR2,
    keycol             IN    VARCHAR2,
    parentcol         IN    VARCHAR2 DEFAULT NULL);
```

パラメータ

表 17-3 MAP_DIMTBL_HIERLEVEL プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。
level_name	レベルの名前。
table_owner	ディメンション表の所有者。
table_name	ディメンション表の名前。
keycol	このレベルがマップされる必要があるディメンション表の列。 この列は、ファクト表のこのレベル列のキーになります。 レベルが複数の列に格納されている場合、カンマで列名を区切ります。これらの列は、ファクト表のこれらのレベル列の複数列からなるキーになります。
parentcol	階層の親レベルが格納される列。このパラメータを指定しない場合、レベルは階層のルートになります。

MAP_DIMTBL_HIERSORTKEY プロシージャ

このプロシージャは、ディメンション表の列で階層のメンバーをソートする方法を指定します。列は、キー列または関連する属性列である場合があります。カスタム・ソートでは、列を昇順または降順のどちらかでソートするか、また、その場合に NULL を先頭に置くか最後に置くかを指定できます。

カスタム・ソート情報はオプションで、ディメンションの複数のレベルで適用できます。

構文

```
MAP_DIMTBL_HIERSORTKEY (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    hierarchy_name     IN    VARCHAR2,
    sortcol            IN    VARCHAR2);
```

パラメータ

表 17-4 MAP_DIMTBL_HIERSORTKEY プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。
sortcol	ディメンション表の指定した列に格納されている値のソート方法を指定する文字列。文字列は、表の名前、列の名前、昇順または降順でソートするか、および NULL を最初または最後のどちらに置くかを指定します。 文字列は一重引用符で囲む必要があります。形式は次のとおりです。 'TBL:tableowner.tablename/COL:columnname/ORD:ASC DSC/NULL:FIRST LAST;'

MAP_DIMTBL_LEVELATTR プロシージャ

このプロシージャは、レベル属性をディメンション表の列にマップします。

マップされる属性は、階層コンテキストを持たないレベルに関連付けられます。通常、階層コンテキストを持たないレベルのみがこのディメンションに定義されます。

構文

```
MAP_DIMTBL_LEVELATTR (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    dimension_attribute_name IN VARCHAR2,
    level_name         IN    VARCHAR2,
    level_attribute_name IN VARCHAR2,
    table_owner        IN    VARCHAR2,
    table_name         IN    VARCHAR2,
    attrcol            IN    VARCHAR2);
```

パラメータ

表 17-5 MAP_DIMTBL_LEVELATTR プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
level_name	レベルの名前。
level_attribute_name	レベルに関連付けられているレベル属性の名前。
table_owner	ディメンション表の所有者。
table_name	ディメンション表の名前。
attrcol	レベル属性がマップされる必要があるディメンション表の列。

MAP_DIMTBL_LEVEL プロシージャ

このプロシージャは、レベルをディメンション表の1つ以上の列にマップします。

マップされるレベルには、階層コンテキストがありません。通常、階層コンテキストを持たないレベルのみがこのディメンションに定義されます。

構文

```
MAP_DIMTBL_LEVEL (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    level_name         IN    VARCHAR2,
    table_owner        IN    VARCHAR2,
    table_name         IN    VARCHAR2,
    keycol             IN    VARCHAR2);
```

パラメータ

表 17-6 MAP_DIMTBL_LEVEL プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。

表 17-6 MAP_DIMTBL_LEVEL プロシージャのパラメータ (続き)

パラメータ	説明
level_name	レベルの名前。
table_owner	ディメンション表の所有者。
table_name	ディメンション表の名前。
keycol	このレベルがマップされる必要があるディメンション表の列。 この列は、ファクト表のこのレベル列のキーになります。 レベルが複数の列に格納されている場合、カンマで列名を区切り ます。これらの列は、ファクト表のこれらのレベル列の複数 列からなるキーになります。

MAP_FACTTBL_LEVELKEY プロシージャ

このプロシージャは、ファクト表と一連のディメンション表の結合関係を作成します。キューブのディメンションごとに結合を指定する必要があります。各ディメンションは、いずれかの階層のコンテキスト内で結合されます。

たとえば、3つのディメンションが存在するキューブがあり、各ディメンションには1つの階層のみが存在する場合、MAP_FACTTBL_LEVELKEY を1回コールするのみで完全にキューブをマップできます。

ただし、3つのディメンションが存在するキューブがあり、そのうちの2つのディメンションにそれぞれ2つの階層が存在する場合、キューブを完全にマップするには、MAP_FACTTBL_LEVELKEY を4回コールする必要があります。たとえば、Dim1、Dim2 および Dim3 というディメンションがあり、Dim1 と Dim3 にそれぞれ2つの階層が存在する場合は、MAP_FACTTBL_LEVELKEY への各コールで次のマッピング文字列を指定します。

```
Dim1_Hier1, Dim2_Hier, Dim3_Hier1
Dim1_Hier1, Dim2_Hier, Dim3_Hier2
Dim1_Hier2, Dim2_Hier, Dim3_Hier1
Dim1_Hier2, Dim2_Hier, Dim3_Hier2
```

通常、各階層の組合せに対するデータは、個別のファクト表に格納されます。

詳細は、2-11 ページの「[ファクト表とディメンション表の結合](#)」を参照してください。

構文

```
MAP_FACTTBL_LEVELKEY (
    cube_owner          IN   VARCHAR2,
    cube_name           IN   VARCHAR2,
    facttable_owner     IN   VARCHAR2,
    facttable_name      IN   VARCHAR2,
    storetype           IN   VARCHAR2,
```

```
dimkeymap      IN  VARCHAR2,
dimktype      IN  VARCHAR2 DEFAULT NULL);
```

パラメータ

表 17-7 MAP_FACTTBL_LEVELKEY プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
facttable_owner	ファクト表の所有者。
facttable_name	ファクト表の名前。
storetype	次のいずれかを選択します。 'LOWESTLEVEL' は、最低レベルのデータのみを格納するファクト表に使用します。 'ET' は、最低レベルのデータに加えて、すべてのレベルの組合せの埋込み合計を格納するファクト表に使用します。
dimkeymap	ファクト表のデータの各ディメンションに対するマッピングを指定する文字列。各ディメンションに、階層および階層内でマップされる最低レベルを指定する必要があります。 次のように、文字列を一重引用符で囲み、ディメンションをセミコロンで区切って指定します。 'DIM:dimname1/HIER:hiername1 /GID:gid_columnname1/LVL:levelname1 /COL:map_columnname1; DIM:dimname2/HIER:hiername2 /GID:gid_columnname2/LVL:levelname2 /COL:map_columnname2;.....' マッピング文字列の GID 句は、埋込み合計に対してのみ適用されます。storetype 引数に 'LOWESTLEVEL' を指定する場合は、マッピング文字列に GID 句を含めないでください。 また、この文字列は MAP_FACTTBL_MEASURE プロシージャに対する引数として指定する必要があります。
dimktype	このパラメータは、現在使用されていません。

MAP_FACTTBL_MEASURE プロシージャ

このプロシージャは、メジャーをファクト表の列にマップします。

構文

```
MAP_FACTTBL_MEASURE (  
    cube_owner          IN   VARCHAR2,  
    cube_name           IN   VARCHAR2,  
    measure_name        IN   VARCHAR2,  
    facttable_owner     IN   VARCHAR2,  
    facttable_name      IN   VARCHAR2,  
    column_name         IN   VARCHAR2,  
    dimkeymap           IN   VARCHAR2);
```

パラメータ

表 17-8 MAP_FACTTBL_MEASURE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
measure_name	マップするメジャーの名前。
facttable_owner	ファクト表の所有者。
facttable_name	ファクト表の名前。
column_name	メジャーがマップされるファクト表の列。

表 17-8 MAP_FACTTBL_MEASURE プロシージャのパラメータ (続き)

パラメータ	説明
dimkeymap	<p>各メジャーのディメンションに対するマッピングを指定する文字列。各ディメンションに、階層および階層内でマップされる最低レベルを指定する必要があります。</p> <p>次のように、文字列を一重引用符で囲み、ディメンションをセミコロンで区切って指定します。</p> <pre>'DIM:dimname1/HIER:hiername1 /GID:gid_columnname1/LVL:levelname1 /COL:map_columnname1; DIM:dimname2/HIER:hiername2 /GID:gid_columnname2/LVL:levelname2 /COL:map_columnname2;.....'</pre> <p>マッピング文字列の GID 句は、埋込み合計に対してのみ適用されます。storetype 引数に 'LOWESTLEVEL' を指定する場合は、マッピング文字列に GID 句を含めないでください。</p> <p>また、この文字列は MAP_FACTTBL_LEVELKEY プロシージャへの引数として指定する必要があります。</p>

REMOVEMAP_DIMTBL_HIERLEVELATTR プロシージャ

このプロシージャは、レベル属性とディメンション表の列の関係を削除します。属性は、関連付けられたレベルを含む階層によって識別されます。

このプロシージャが正常に完了すると、レベル属性は純粋な論理メタデータのエンティティになります。レベル属性には、関連付けられたデータはありません。

構文

```
REMOVEMAP_DIMTBL_HIERLEVELATTR (
    dimension_owner          IN   VARCHAR2,
    dimension_name           IN   VARCHAR2,
    dimension_attribute_name IN   VARCHAR2,
    hierarchy_name          IN   VARCHAR2,
    level_name               IN   VARCHAR2,
    level_attribute_name     IN   VARCHAR2);
```

パラメータ

表 17-9 REMOVEMAP_DIMTBL_HIERLEVELATTR プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。

表 17-9 REMOVEMAP_DIMTBL_HIERLEVELATTR プロシージャのパラメータ (続き)

パラメータ	説明
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
hierarchy_name	階層の名前。
level_name	レベルの名前。
level_attribute_name	レベルに関連付けられているレベル属性の名前。

REMOVEMAP_DIMTBL_HIERLEVEL プロシージャ

このプロシージャは、階層のレベルとディメンション表の1つ以上の列の関係を削除します。

このプロシージャが正常に完了すると、レベルは純粋な論理メタデータのエンティティになります。レベルには、関連付けられたデータはありません。

構文

```
REMOVEMAP_DIMTBL_HIERLEVEL (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    hierarchy_name     IN    VARCHAR2,
    level_name         IN    VARCHAR2);
```

パラメータ

表 17-10 REMOVEMAP_DIMTBL_HIERLEVEL プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。
level_name	レベルの名前。

REMOVEMAP_DIMTBL_HIERSORTKEY プロシージャ

このプロシージャは、ディメンション表の列に関連付けられたカスタム・ソート基準を削除します。

構文

```
REMOVEMAP_DIMTBL_HIERSORTKEY (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    hierarchy_name     IN    VARCHAR2);
```

パラメータ

表 17-11 REMOVEMAP_DIMTBL_HIERSORTKEY プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
hierarchy_name	階層の名前。

REMOVEMAP_DIMTBL_LEVELATTR プロシージャ

このプロシージャは、レベル属性とディメンション表の列の関係を削除します。

このプロシージャが正常に完了すると、レベル属性は純粋な論理メタデータのエンティティになります。レベル属性には、関連付けられたデータはありません。

構文

```
REMOVEMAP_DIMTBL_LEVELATTR (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    dimension_attribute_name IN  VARCHAR2,
    level_name         IN    VARCHAR2,
    level_attribute_name IN  VARCHAR2);
```

パラメータ

表 17-12 REMOVEMAP_DIMTBL_LEVELATTR プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。

表 17-12 REMOVEMAP_DIMTBL_LEVELATTR プロシージャのパラメータ (続き)

パラメータ	説明
dimension_name	ディメンションの名前。
dimension_attribute_name	ディメンション属性の名前。
level_name	レベルの名前。
level_attribute_name	レベルに関連付けられているレベル属性の名前。

REMOVEMAP_DIMTBL_LEVEL プロシージャ

このプロシージャは、レベルとディメンション表の1つ以上の列の関係を削除します。

このプロシージャが正常に完了すると、レベルは純粋な論理メタデータのエンティティになります。レベルには、関連付けられたデータはありません。

構文

```
REMOVEMAP_DIMTBL_LEVEL (
    dimension_owner    IN    VARCHAR2,
    dimension_name     IN    VARCHAR2,
    level_name        IN    VARCHAR2);
```

パラメータ

表 17-13 REMOVEMAP_DIMTBL_LEVEL プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
level_name	レベルの名前。

REMOVEMAP_FACTTBL_LEVELKEY プロシージャ

このプロシージャは、ファクト表のキー列とディメンション表のディメンション階層のレベル列の関係を削除します。

構文

```
REMOVEMAP_FACTTBL_LEVELKEY (
    cube_owner    IN    VARCHAR2,
    cube_name     IN    VARCHAR2,
```

```

facttable_owner  IN  VARCHAR2,
facttable_name   IN  VARCHAR2 DEFAULT );

```

パラメータ

表 17-14 REMOVEFACTTBL_LEVELKEY プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
facttable_owner	ファクト表の所有者。
facttable_name	ファクト表の名前。

REMOVEFACTTBL_MEASURE プロシージャ

このプロシージャは、ファクト表のメジャー列とキューブに関連付けられた論理メジャーの関係を削除します。

このプロシージャが正常に完了すると、メジャーは純粋な論理メタデータのエンティティになります。メジャーには、関連付けられたデータはありません。

構文

```

REMOVEFACTTBL_MEASURE (
    cube_owner      IN  VARCHAR2,
    cube_name       IN  VARCHAR2,
    measure_name    IN  VARCHAR2,
    facttable_owner IN  VARCHAR2,
    facttable_name  IN  VARCHAR2,
    column_name     IN  VARCHAR2,
    dimkeymap       IN  VARCHAR2);

```

パラメータ

表 17-15 REMOVEFACTTBL_MEASURE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
measure_name	メジャーの名前。
facttable_owner	ファクト表の所有者。

表 17-15 REMOVEFACTBL_MEASURE プロシージャのパラメータ (続き)

パラメータ	説明
facttable_name	ファクト表の名前。
column_name	メジャーがマップされるファクト表の列。
dimkeymap	<p>各メジャーのディメンションに対するマッピングを指定する文字列。各ディメンションに、階層および階層内でマップされる最低レベルを指定する必要があります。</p> <p>次のように、文字列を一重引用符で囲み、ディメンションをセミコロンで区切って指定します。</p> <pre>'DIM:dimname1/HIER:hiername1 /GID:gid_columnname1/LVL:levelname1 /COL:map_columnname1; DIM:dimname2/HIER:hiername2 /GID:gid_columnname2/LVL:levelname2 /COL:map_columnname2;.....'</pre> <p>マッピング文字列の GID 句は、埋込み合計に対してのみ適用されます。メジャーに含まれているのが詳細データのみで、メジャーが格納タイプ 'LOWESTLEVEL' でマップされている場合は、マッピング文字列に GID 句を含めないでください。</p> <p>また、この文字列は MAP_FACTBL_MEASURE プロシージャおよび MAP_FACTBL_LEVELKEY プロシージャへの引数として指定する必要があります。</p>

CWM2_OLAP_VALIDATE

CWM2_OLAP_VALIDATE パッケージでは、OLAP メタデータを検証するためのプロシージャが提供されます。

参照：

- 2-12 ページの「[OLAP メタデータの検証](#)」を参照してください。
- 第 19 章「[CWM2_OLAP_VERIFY_ACCESS](#)」を参照してください。

この章では、次の項目について説明します。

- [OLAP カタログ・メタデータの検証について](#)
- [CWM2_OLAP_VALIDATE サブプログラムの概要](#)

OLAP カタログ・メタデータの検証について

検証プロセスでは、メタデータの構造的な整合性がチェックされ、メタデータがディメンション表およびファクト表の列に適切にマップされているかどうかを確認されます。必要に応じて、OLAP API に固有の追加の検証プロセスを実行することもできます。

CWM2_OLAP_VALIDATE のプロシージャは、CWM2 プロシージャで作成されたメタデータに加え、Enterprise Manager で作成された OLAP メタデータも検証します。

参照： 詳細は、2-12 ページの「[OLAP メタデータの検証およびコミット](#)」を参照してください。

構造の検証

構造の検証では、キューブおよびディメンションに必要なすべての構成部品が含まれていることを確認します。CWM2_OLAP_VALIDATE のすべてのプロシージャは、デフォルトで構造の検証を実行します。

キューブ

キューブを構造的に有効にするには、次の条件が満たされている必要があります。

- 1つ以上の有効なディメンションが含まれている。
- 1つ以上のメジャーが含まれている。

ディメンション

ディメンションを構造的に有効にするには、次の条件が満たされている必要があります。

- 1つ以上のレベルが含まれている。
- 1つ以上の階層が含まれている（各階層には1つ以上のレベルが必要）。
- 1つ以上のディメンション属性が含まれている（各ディメンション属性には1つ以上のレベル属性が必要）。

マッピングの検証

マッピングの検証では、メタデータが、表またはビューの列に適切にマップされていることを確認します。CWM2_OLAP_VALIDATE のすべてのプロシージャは、デフォルトでマッピングの検証を実行します。

キューブ

キューブを有効にするには、キューブのマッピングで次の条件が満たされている必要があります。

- キューブが1つ以上のファクト表にマップされている。
- キューブのすべてのメジャーがファクト表の既存の列にマップされている（複数のファクト表が存在する場合は、各ファクト表にすべてのメジャーが必要）。
- ディメンション / 階層のすべての組合せがいずれかのファクト表にマップされている。

ディメンション

ディメンションを有効にするには、ディメンションのマッピングで次の条件が満たされている必要があります。

- すべてのレベルがディメンション表の既存の列にマップされている。
- レベル属性が対応するレベルと同じ表の列にマップされている。

検証タイプ

CWM2_OLAP_VALIDATE パッケージのすべてのプロシージャは、引数として検証タイプを取ります。検証タイプには次のいずれかを指定できます。

DEFAULT -- メタデータの基本構造およびそのソース表へのマッピングを検証します。メタデータを有効にするには、18-2 ページの「[構造の検証](#)」と「[マッピングの検証](#)」に示した条件が満たされている必要があります。

OLAP API -- デフォルトの検証に加え、次の検証を実行します。

- ET 形式のキューブの各ディメンションが、すべてのレベルで、ディメンション属性およびレベル属性の 'ET KEY' と 'GROUPING ID' を持っているかどうかの検証。
- 時間ディメンションが、すべてのレベルで、ディメンション属性およびレベル属性の 'END DATE' と 'TIME SPAN' を持っているかどうかの検証。

CWM2_OLAP_VALIDATE サブプログラムの概要

表 18-1 CWM2_OLAP_VALIDATE

サブプログラム	説明
VALIDATE_ALL_CUBES プロシージャ (18-4 ページ)	OLAP カタログのすべてのキューブを検証します。
VALIDATE_ALL_DIMENSIONS プロシージャ (18-5 ページ)	OLAP カタログのすべてのディメンションを検証します。
VALIDATE_CUBE プロシージャ (18-5 ページ)	OLAP カタログ・キューブを検証します。
VALIDATE_DIMENSION プロシージャ (18-6 ページ)	OLAP カタログ・ディメンションを検証します。
VALIDATE_OLAP_CATALOG プロシージャ (18-7 ページ)	OLAP カタログのすべてのキューブおよびディメンションを検証します。

VALIDATE_ALL_CUBES プロシージャ

このプロシージャは、OLAP カタログのすべてのキューブを検証します。キューブに関連付けられているすべてのディメンションも検証されます。

キューブの妥当性ステータスは [ALL_OLAP2_CUBES](#) ビューに表示されます。

構文

```
VALIDATE_ALL_CUBES (
    type_of_validation    IN    VARCHAR2 DEFAULT 'DEFAULT',
    verbose_report        IN    VARCHAR2 DEFAULT 'YES');
```

パラメータ

表 18-2 VALIDATE_ALL_CUBES プロシージャのパラメータ

パラメータ	説明
type_of_validation	'DEFAULT' または 'OLAP API'。18-3 ページの「 検証タイプ 」を参照してください。
verbose_report	'YES' または 'NO'。すべての検証チェックをレポートするか、または主要なイベントおよびエラーのみをレポートするかを指定します。デフォルトでは、すべての検証チェックがレポートされます。

VALIDATE_ALL_DIMENSIONS プロシージャ

このプロシージャは、OLAP カタログのすべてのディメンションを検証します。

ディメンションの妥当性ステータスは [ALL_OLAP2_DIMENSIONS](#) ビューに表示されます。

構文

```
VALIDATE_ALL_DIMENSIONS (
    type_of_validation    IN    VARCHAR2 DEFAULT 'DEFAULT',
    verbose_report        IN    VARCHAR2 DEFAULT 'YES');
```

パラメータ

表 18-3 VALIDATE_ALL_DIMENSIONS プロシージャのパラメータ

パラメータ	説明
type_of_validation	'DEFAULT' または 'OLAP API'。18-3 ページの「 検証タイプ 」を参照してください。
verbose_report	'YES' または 'NO'。すべての検証チェックをレポートするか、または主要なイベントおよびエラーのみをレポートするかを指定します。デフォルトでは、すべての検証チェックがレポートされます。

VALIDATE_CUBE プロシージャ

このプロシージャは、OLAP カタログ・キューブを検証します。キューブに関連付けられているすべてのディメンションも検証されます。

キューブの妥当性ステータスは [ALL_OLAP2_CUBES](#) ビューに表示されます。

構文

```
VALIDATE_CUBE (
    cube_owner            IN    VARCHAR2,
    cube_name             IN    VARCHAR2,
    type_of_validation    IN    VARCHAR2 DEFAULT 'DEFAULT',
    verbose_report        IN    VARCHAR2 DEFAULT 'YES');
```

パラメータ

表 18-4 VALIDATE_CUBE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。

表 18-4 VALIDATE_CUBE プロシージャのパラメータ (続き)

パラメータ	説明
cube_name	キューブの名前。
type_of_validation	'DEFAULT' または 'OLAP API'。18-3 ページの「 検証タイプ 」を参照してください。
verbose_report	'YES' または 'NO'。すべての検証チェックをレポートするか、または主要なイベントおよびエラーのみをレポートするかを指定します。デフォルトでは、すべての検証チェックがレポートされます。

VALIDATE_DIMENSION プロシージャ

このプロシージャは、OLAP カタログ・ディメンションを検証します。

OLAP ディメンションの妥当性ステータスは [ALL_OLAP2_DIMENSIONS](#) ビューに表示されます。

構文

```
VALIDATE_DIMENSION (
    dimension_owner      IN   VARCHAR2,
    dimension_name       IN   VARCHAR2,
    type_of_validation   IN   VARCHAR2 DEFAULT 'DEFAULT',
    verbose_report       IN   VARCHAR2 DEFAULT 'YES');
```

パラメータ

表 18-5 VALIDATE_DIMENSION プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
type_of_validation	'DEFAULT' または 'OLAP API'。18-3 ページの「 検証タイプ 」を参照してください。
verbose_report	'YES' または 'NO'。すべての検証チェックをレポートするか、または主要なイベントおよびエラーのみをレポートするかを指定します。デフォルトでは、すべての検証チェックがレポートされます。

VALIDATE_OLAP_CATALOG プロシージャ

このプロシージャは、OLAP カタログのすべてのメタデータを検証します。すべてのキューブ（およびそのディメンション）と、キューブに関連付けられていないすべてのディメンションも検証されます。

VALIDATE_OLAP_CATALOG は、キューブに関連付けられていない各ディメンションをアルファベット順に検証してから、各キューブをアルファベット順に検証します。

構文

```
VALIDATE_OLAP_CATALOG (
    type_of_validation    IN  VARCHAR2 DEFAULT 'DEFAULT',
    verbose_report       IN  VARCHAR2 DEFAULT 'YES');
```

パラメータ

表 18-6 VALIDATE_OLAP_CATALOG プロシージャのパラメータ

パラメータ	説明
type_of_validation	'DEFAULT' または 'OLAP API'。18-3 ページの「 検証タイプ 」を参照してください。
verbose_report	'YES' または 'NO'。すべての検証チェックをレポートするか、または主要なイベントおよびエラーのみをレポートするかを指定します。デフォルトでは、すべての検証チェックがレポートされます。

CWM2_OLAP_VERIFY_ACCESS

CWM2_OLAP_VERIFY_ACCESS パッケージでは、OLAP キューブを検証し、OLAP API からのアクセス可能性を確認するためのプロシージャが提供されます。

参照：

- 2-12 ページの「[OLAP メタデータの検証およびコミット](#)」を参照してください。
- [第 15 章「CWM2_OLAP_METADATA_REFRESH」](#)を参照してください。
- [第 18 章「CWM2_OLAP_VALIDATE」](#)を参照してください。

この章では、次の項目について説明します。

- [OLAP キューブのアクセス可能性の検証](#)
- [CWM2_OLAP_VERIFY_ACCESS サブプログラムの概要](#)

OLAP キューブのアクセス可能性の検証

CWM2_OLAP_VALIDATE パッケージのキューブ検証プロシージャは、OLAP キューブの論理構造を検証し、それがディメンション表およびファクト表の列に適切にマップされているかどうかをチェックします。しかし、キューブがこの基準に照らして完全に妥当であっても、アプリケーションから利用することができない場合もあります。

このため、CWM2_OLAP_VERIFY_ACCESS パッケージを使用して、次の追加条件が満たされているかどうかをチェックすることが必要な場合もあります。

- OLAP API Metadata Reader が使用するメタデータ表に、キューブのメタデータに加えられた最新の変更が反映されている。これらの MRV\$ 表が更新されていない場合は、CWM2_OLAP_METADATA_REFRESH パッケージのプロシージャを実行して、OLAP API によるアクセスを可能にする必要があります。

- アプリケーションの識別情報が、キューブの基礎となるソース・データに対するアクセス権を所有している。CWM2_OLAP_VALIDATE の検証プロシージャは、SYS 識別情報の下で実行されます。これらのプロシージャによってキューブが完全に妥当であることが示されても、アプリケーションがそのキューブにアクセスできない場合もあります。この場合には、コール元ユーザーに適切な権限を付与する必要があります。

CWM2_OLAP_VERIFY_ACCESS サブプログラムの概要

表 19-1 CWM2_OLAP_VERIFY_ACCESS

サブプログラム	説明
VERIFY_CUBE_ACCESS プロシージャ (19-3 ページ)	キューブを検証し、OLAP アプリケーションからのアクセス可能性を確認します。

VERIFY_CUBE_ACCESS プロシージャ

このプロシージャは、まず CWM2_OLAP_VALIDATE パッケージの VALIDATE_CUBE プロシージャをコールしてキューブを検証します。そして、コール元ユーザーの識別情報の下で実行されている OLAP API アプリケーションがそのキューブに対するアクセス権を所有しているかどうかをチェックします。

キューブのアクセス可能性の要件については、19-1 ページの「[OLAP キューブのアクセス可能性の検証](#)」を参照してください。

構文

```
VERIFY_CUBE_ACCESS (
    cube_owner          IN  VARCHAR2,
    cube_name          IN  VARCHAR2,
    type_of_validation IN  VARCHAR2 DEFAULT 'DEFAULT',
    verbose_report     IN  VARCHAR2 DEFAULT 'YES');
```

パラメータ

表 19-2 VERIFY_CUBE_ACCESS プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
type_of_validation	'DEFAULT' または 'OLAP API'。18-3 ページの「 検証タイプ 」を参照してください。
verbose_report	'YES' または 'NO'。すべての検証チェックをレポートするか、または主要なイベントおよびエラーのみをレポートするかを指定します。デフォルトでは、すべての検証チェックがレポートされます。

DBMS_AW パッケージでは、アナリティック・ワークスペース内で各種操作を実行するためのプロシージャおよびファンクションが提供されます。DBMS_AW を使用すると、次のことを行うことができます。

- OLAP DML コマンドの SQL 文への埋込み
- ワークスペース内での計算によって得られるデータを戻す問合せの記述
- ワークスペース内の集計データの管理に有用な情報の取得

参照：

- パッケージについては、『PL/SQL ユーザーズ・ガイドおよびリファレンス』を参照してください。

この章では、次の項目について説明します。

- [OLAP DML の SQL 文への埋込み](#)
- [カスタム・メジャーの SELECT 文への埋込み](#)
- [集計アドバイザの使用](#)
- [DBMS_AW サブプログラムの概要](#)

OLAP DML の SQL 文への埋込み

DBMS_AW パッケージを使用すると、アナリティック・ワークスペース内であらゆる OLAP 処理を実行できます。レガシー・ワークスペース、リレーショナル表またはフラット・ファイルからデータをインポートできます。OLAP オブジェクトを定義して、複雑な計算を実行することもできます。

注意： DBMS_AW パッケージを使用してアナリティック・ワークスペースを新規に作成した場合、スタンダード・フォームを必要とする OLAP ユーティリティを使用できなくなることがあります。この場合、OLAP_TABLE ファンクションを使用して、そのワークスペースのリレーショナル・ビューを独自に作成する必要があります。OLAP API からワークスペースにアクセスできるようにするには、CWM2 パッケージを使用して、そのビュー用のメタデータを独自に作成する必要があります。

OLAP DML コマンドの実行方法

DBMS_AW パッケージでは、非定型の OLAP DML コマンドを実行するためのプロシージャがいくつか提供されています。EXECUTE または INTERP_SILENT プロシージャ、あるいは INTERP または INTERCLOB ファンクションを使用して、単一の OLAP DML コマンドまたはセミコロンで区切った一連のコマンドを実行できます。

どのプロシージャを使用するかは、出力の送信方法と、入力バッファおよび出力バッファのサイズによって決まります。たとえば、EXECUTE プロシージャは出力をプリンタ・バッファに送信します。INTERP_SILENT プロシージャは出力を抑制し、INTERP ファンクションはセッション・ログを戻します。

OLAP DML コマンドで引用符を使用する場合のガイドライン

SQL プロセッサは、埋め込まれた OLAP DML コマンドを全体的または部分的に評価した後、Oracle OLAP に送信して処理します。DBMS_AW の各プロシージャの `olap-commands` パラメータで OLAP DML コマンドの書式を設定する場合は、次のガイドラインに従います。

- OLAP DML コマンド内で一重引用符 (') を使用する場合は、常に、2 つの一重引用符 (') を使用します。SQL プロセッサは、OLAP DML コマンドを Oracle OLAP に送信する前に、これらの一重引用符の 1 つを削除します。
- OLAP DML では、二重引用符 (") はコメントの始まりを示します。

カスタム・メジャーの SELECT 文への埋込み

DBMS_AW パッケージの OLAP_EXPRESSION ファンクションは、アナリティック・ワークスペース内で単一行の数値ファンクションを動的に実行し、結果を戻します。

OLAP_EXPRESSION ファンクションは、SELECT 文の WHERE 句および ORDER BY 句に埋め込むことができます。

OLAP_EXPRESSION のバリエーションを使用すると、テキスト、日付またはブールの式を計算できます。

次のスクリプトは、MEASURE_VIEW というビューを作成するために使用されたものです。OLAP_EXPRESSION の使用例を示すために、[例 20-1](#) と [例 20-2](#) でこのビューを使用します。

サンプル・ビュー : MEASURE_VIEW

```

CREATE TYPE measure_row AS OBJECT (
    time                VARCHAR2(12),
    geography            VARCHAR2(30),
    product              VARCHAR2(30),
    channel              VARCHAR2(30),
    sales                NUMBER(16),
    cost                 NUMBER(16),
    promotions           NUMBER(16),
    quota                NUMBER(16),
    units                NUMBER(16),
    r2c                  RAW(32));
/

CREATE TYPE measure_table AS TABLE OF measure_row;
/

CREATE OR REPLACE VIEW measure_view AS
SELECT sales, cost, promotions, quota, units,
       time, geography, product, channel, r2c
FROM TABLE(CAST(OLAP_TABLE(
    'xademo DURATION SESSION',
    'measure_table',
    '',
    'MEASURE sales FROM analytic_cube_f.sales
    MEASURE cost FROM analytic_cube_f.costs
    MEASURE promotions FROM analytic_cube_f.promo
    MEASURE quota FROM analytic_cube_f.quota
    MEASURE units FROM analytic_cube_f.units
    DIMENSION time FROM time WITH
        HIERARCHY time_member_parentrel
        INHIERARCHY time_member_inhier
    DIMENSION geography FROM geography WITH
        HIERARCHY geography_member_parentrel
        INHIERARCHY geography_member_inhier
    DIMENSION product FROM product WITH
        HIERARCHY product_member_parentrel
        INHIERARCHY product_member_inhier
    DIMENSION channel FROM channel WITH
        HIERARCHY channel_member_parentrel
        INHIERARCHY channel_member_inhier
    ROW2CELL r2c')
    AS measure_table))
WHERE sales IS NOT NULL;
/
COMMIT
/

```

```
GRANT SELECT ON measure_view TO PUBLIC;
```

例 20-1 OLAP_EXPRESSION: WHERE 句が定義された時系列ファンクション

この例では、20-3 ページの「サンプル・ビュー: MEASURE_VIEW」に示したビューを使用します。

次の SELECT 文は、PERIODAGO という別名を付けて式を計算し、その結果セットを 200,000 より大きい値に制限します。この計算では、LAG ファンクションを使用して、以前の時間間隔の値が戻されます。

```
SELECT time, cost, OLAP_EXPRESSION(r2c,
    'LAG(analytic_cube_f.costs, 1, time,
        LEVELREL time_member_levelrel)') periodago
FROM measure_view
WHERE geography = 'L1.WORLD' AND
CHANNEL = 'STANDARD_2.TOTALCHANNEL' AND
PRODUCT = 'L1.TOTALPROD' and
OLAP_EXPRESSION(r2c, 'LAG(analytic_cube_f.costs, 1, time,
    LEVELREL time_member_levelrel)') > 200000;
```

この SELECT 文は、次の結果を戻します。

TIME	COST	PERIODAGO
-----	-----	-----
L1.1997	1078031	2490243.07
L2.Q1.97	615399	560379.445
L2.Q2.96	649004	615398.858
L2.Q2.97	462632	649004.473
L2.Q3.96	582693	462632.064
L2.Q4.96	698166	582693.091
L3.AUG96	194498	209476.344
L3.FEB96	186762	252738.981
L3.JAN96	185755	205214.946
.	.	.
.	.	.
.	.	.

例 20-2 OLAP_EXPRESSION: ORDER BY 句が定義された数値計算

この例では、20-3 ページの「サンプル・ビュー: MEASURE_VIEW」に示したビューを使用します。

この例では、売上げからコストを引いて利益を計算し、この式に PROFIT という別名を付けます。行は、地域ごとに、利益の多い地域から少ない地域に順序付けされます。

```
SELECT geography, sales, cost, OLAP_EXPRESSION(r2c,
    'analytic_cube_f.sales - analytic_cube_f.costs') profit
FROM measure_view
```

```

WHERE
channel = 'STANDARD_2.TOTALCHANNEL' AND
product = 'L1.TOTALPROD' AND
time = 'L3.APR97'
ORDER BY OLAP_EXPRESSION(r2c,
'analytic_cube_f.sales - analytic_cube_f.costs') DESC;

```

この SELECT 文は、次の結果を戻します。

GEOGRAPHY	SALES	COST	PROFIT
L1.WORLD	9010260	209476	8800783.17
L2.EUROPE	3884776	95204	3789571.85
L2.AMERICAS	2734436	55322	2679114.66
L2.ASIA	1625379	37259	1588120.61
L3.USA	1603043	27547	1575496.86
L2.AUSTRALIA	765668	21692	743976.058
L3.UK	733090	19144	713945.952
L3.CANADA	731734	19666	712067.455
L4.NEWYORK	684008	8020	675987.377
L3.GERMANY	659428	12440	646988.197
L3.FRANCE	596767	19307	577460.113
.	.	.	.
.	.	.	.
.	.	.	.

集計アドバイザの使用

アナリティック・ワークスペース内の集計データの管理は、パフォーマンスに大きな影響を与える場合があります。DBMS_AW パッケージの ADVISE_REL プロシージャおよび ADVISE_CUBE プロシージャを使用すると、事前集計対象として最適なディメンション・メンバーの組合せのセットを判断できます。これらのプロシージャのことを、まとめて**集計アドバイザ**と呼びます。

ADVISE_REL は、指定されたパーセンテージに基づいて、事前集計するべきディメンション・メンバーのセットを提示します。ADVISE_CUBE プロシージャは、事前集計するべき、キューブの各ディメンションのメンバーのセットを提示します。集計アドバイザ・プロシージャは、データベース・スタンダード・フォームを必要とします。

参照： スタンダード・フォームのアナリティック・ワークスペースについては、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

ワークスペース内での集計機能

集計データの格納に関する指示は、`aggmap` というワークスペース・オブジェクトで指定します。OLAP DML の `AGGREGATE` コマンドは、`aggmap` を使用してデータを事前集計します。事前集計されていないデータは、そのデータに対する問合せが発行されたときに、`AGGREGATE` ファンクションによって動的に集計されます。

静的な集計と動的な集計の割合は、ディスク領域、利用可能なメモリー、そのデータに対して発行される問合せの性質と頻度など、多くの要因によって決まります。これらの要因を比較検討することによって、事前集計するデータのパーセンテージを決定できます。

事前集計するデータのパーセンテージが決まったら、集計アドバイザーを使用します。集計アドバイザーのプロシージャは、階層内におけるディメンション・メンバーの分布を分析し、事前集計対象として最適なディメンション・メンバーのセットを判断します。

例 : `ADVISE_REL` プロシージャの使用

`ADVISE_REL` プロシージャは、指定された事前計算のパーセンテージに基づいて、ディメンションとそのすべての階層関係を表すファミリ・リレーションを分析し、ディメンション・メンバーのリストを戻します。

`ADVISE_CUBE` も、キューブの `aggmap` の各ディメンションに対して同様の経験則を適用します。

参照 :

- 20-12 ページの「[ADVISE_REL プロシージャ](#)」を参照してください。
- 20-11 ページの「[ADVISE_CUBE プロシージャ](#)」を参照してください。

20-8 ページの例 20-3 では、サンプルの `Customer` ディメンションを使用して、`ADVISE_REL` プロシージャの使用例を示しています。

サンプル・ディメンション : GLOBAL アナリティック・ワークスペースの `Customer`

`GLOBAL_AW.GLOBAL` の `Customer` ディメンションには、4 つのレベルを持つ `SHIPMENTS_ROLLUP` と、3 つのレベルを持つ `MARKET_ROLLUP` という 2 つの階層があります。このディメンションには 106 のメンバーがあります。この中には、各レベルの全メンバーおよびすべてのレベル名が含まれます。

`Customer` ディメンションのメンバーは、テキスト値が詳細な説明および簡単な説明で定義されている整数キーです。

次の OLAP DML コマンドには、`Customer` ディメンションのスタンダード・フォーム表現の特徴が示されています。

```
" ---- Number of members of Customer dimension
>show statlen(customer)
106
```

```

" ---- Hierarchies in Customer dimension;
>rpr w 40 customer_hierlist
CUSTOMER_HIERLIST
-----
MARKET_ROLLUP
SHIPMENTS_ROLLUP

" ---- Levels in Customer dimension
>rpr w 40 customer_levellist
CUSTOMER_LEVELLIST
-----
ALL_CUSTOMERS
REGION
WAREHOUSE
TOTAL_MARKET
MARKET_SEGMENT
ACCOUNT
SHIP_TO
" ---- In the MARKET_ROLLUP hierarchy, ACCOUNT is the leaf level.
" ---- In the SHIPMENTS_HIER hierarchy, SHIP_TO is the leaf level.
" ---- MARKET_HIER                SHIPMENTS_HIER
" ----
" ---- TOTAL_MARKET                ALL_CUSTOMERS
" ---- MARKET_SEGMENT            REGIONS
" ---- ACCOUNT                    WAREHOUSE
" ----                            SHIP_TO
" ----
" ---- Parent relation showing parent-child relationships in the Customer dimension
>limit customer to last 20          "Only show the last 20 members
>rpr w 10 down customer w 20 customer_parentrel
-----CUSTOMER_PARENTREL-----
-----CUSTOMER_HIERLIST-----
CUSTOMER      MARKET_ROLLUP      SHIPMENTS_ROLLUP
-----
103           44                 21
104           45                 21
105           45                 21
106           45                 21
7             NA                 NA
1             NA                 NA
8             NA                 1
9             NA                 1
10            NA                 1
11            NA                 8
12            NA                 10
13            NA                 9
14            NA                 9
15            NA                 8

```

```

16      NA      9
17      NA      8
18      NA      8
19      NA      9
20      NA      9
21      NA     10
" ---- Show text descriptions for the same twenty dimension members
>report w 15 down customer w 35 across customer_hierlist: <customer_short_description>
ALL_LANGUAGES: AMERICAN_AMERICA
-----CUSTOMER_HIERLIST-----
-----MARKET_ROLLUP-----
-----SHIPMENTS_ROLLUP-----
CUSTOMER      CUSTOMER_SHORT_DESCRIPTION      CUSTOMER_SHORT_DESCRIPTION
-----
103      US Marine Svcs Washington      US Marine Svcs Washington
104      Warren Systems New York      Warren Systems New York
105      Warren Systems Philladelphia      Warren Systems Philladelphia
106      Warren Systems Boston      Warren Systems Boston
7      Total Market      NA
1      NA      All Customers
8      NA      Asia Pacific
9      NA      Europe
10     NA      North America
11     NA      Australia
12     NA      Canada
13     NA      France
14     NA      Germany
15     NA      Hong Kong
16     NA      Italy
17     NA      Japan
18     NA      Singapore
19     NA      Spain
20     NA      United Kingdom
21     NA      United States

```

例 20-3 ADVISE_REL: Customer ディメンションについて提示された事前集計

この例では、20-6 ページの「サンプル・ディメンション:GLOBAL アナリティック・ワークスペースの Customer」で説明した GLOBAL の Customer ディメンションを使用します。

次の PL/SQL 文は、Customer ディメンションの 25% を事前集計することを前提としています。ADVISE_REL は、提示するメンバーのセットを値セットに戻します。

```

SQL>SET SERVEROUTPUT ON
SQL>EXECUTE dbms_aw.execute('aw attach global_aw.global');
SQL>EXECUTE dbms_aw.execute('define customer_preagg valueset customer');
SQL>EXECUTE dbms_aw.advise_rel('customer_parentrel', 'customer_preagg', 25);
SQL>EXECUTE dbms_aw.execute('show values(customer_preagg)');
31
2
4

```

5
6
7
1
8
9
20
21

戻された Customer のメンバーを、説明、関連レベルおよび関連階層とともに次に示します。

Customer の メンバー	説明	階層	レベル
31	Kosh Enterprises	MARKET_ROLLUP	ACCOUNT
2	Consulting	MARKET_ROLLUP	MARKET_SEGMENT
4	Government	MARKET_ROLLUP	MARKET_SEGMENT
5	Manufacturing	MARKET_ROLLUP	MARKET_SEGMENT
6	Reseller	MARKET_ROLLUP	MARKET_SEGMENT
7	TOTAL_MARKET	MARKET_ROLLUP	TOTAL_MARKET
1	ALL_CUSTOMERS	SHIPMENTS_ROLLUP	ALL_CUSTOMERS
8	Asia Pacific	SHIPMENTS_ROLLUP	REGION
9	Europe	SHIPMENTS_ROLLUP	REGION
20	United Kingdom	SHIPMENTS_ROLLUP	WAREHOUSE
21	United States	SHIPMENTS_ROLLUP	WAREHOUSE

DBMS_AW サブプログラムの概要

次の表に、DBMS_AW で提供されるサブプログラムを示します。

表 20-1 DBMS_AW サブプログラム

サブプログラム	説明
ADVISE_CUBE プロシージャ (20-11 ページ)	指定されたキューブのデータのパーセンテージに基づいて、スタンダード・フォームのキューブをどのように事前集計するべきかを提示します。
ADVISE_REL プロシージャ (20-12 ページ)	指定されたディメンションのメンバーのパーセンテージに基づいて、スタンダード・フォームのディメンションをどのように事前集計するべきかを提示します。
EXECUTE プロシージャ (20-13 ページ)	1 つ以上の OLAP DML コマンドを実行します。入力および出力は、4KB に制限されます。通常、アナリティック・ワークスペースを使用して対話型セッションで使用されます。
GETLOG ファンクション (20-14 ページ)	最後に実行した INTERP または INTERPCLOB ファンクションからセッション・ログを戻します。
INTERP ファンクション (20-15 ページ)	1 つ以上の OLAP DML コマンドを実行します。入力は 4KB に、出力は 4GB に制限されます。通常、EXECUTE プロシージャに対する 4KB の出力制限が非常に厳密な場合にアプリケーションで使用されます。
INTERPCLOB ファンクション (20-16 ページ)	1 つ以上の OLAP DML コマンドを実行します。入力および出力は、4GB に制限されます。通常、INTERP ファンクションに対する 4KB の入力制限が非常に厳密な場合にアプリケーションで使用されます。
INTERP_SILENT プロシージャ (20-17 ページ)	1 つ以上の OLAP DML コマンドを実行し、出力を抑止します。入力は 4KB に、出力は 4GB に制限されます。
OLAP_EXPRESSION ファンクション (20-18 ページ)	アナリティック・ワークスペースで計算された単一行の数値ファンクションの結果セットを戻します。
OLAP_EXPRESSION_BOOL ファンクション (20-19 ページ)	アナリティック・ワークスペースで計算された単一行のブール・ファンクションの結果セットを戻します。
OLAP_EXPRESSION_DATE ファンクション (20-20 ページ)	アナリティック・ワークスペースで計算された単一行の日付ファンクションの結果セットを戻します。
OLAP_EXPRESSION_TEXT ファンクション (20-21 ページ)	アナリティック・ワークスペースで計算された単一行のテキスト・ファンクションの結果セットを戻します。

表 20-1 DBMS_AW サブプログラム (続き)

サブプログラム	説明
PRINTLOG プロシージャ (20-22 ページ)	INTERP、INTERCLOB または GETLOG ファンクションによつて戻されるセッション・ログを出力します。

ADVISE_CUBE プロシージャ

ADVISE_CUBE プロシージャは、アナリティック・ワークスペース内のスタンダード・フォームのキューブをどのように事前集計すべきかを判断する際に役立ちます。事前集計するキューブのデータのパーセンテージを指定すると、キューブの各ディメンションの中から事前集計を推奨するメンバーのセットが ADVISE_CUBE によって提示されます。

ADVISE_CUBE プロシージャは、入力として、aggmap および事前計算のパーセンテージを取ります。aggmap の各 RELATION 文には、PRECOMPUTE 句を指定する必要があります。PRECOMPUTE 句は、値セットを含んでいなければなりません。ADVISE_CUBE は、指定された事前計算のパーセンテージに基づいて、ディメンション・メンバーのセットを各値セットに戻します。

構文

```
ADVISE_CUBE (
    aggmap_name          IN   VARCHAR2
    precompute_percentage IN   INTEGER DEFAULT 20);
```

パラメータ

表 20-2 ADVISE_CUBE プロシージャのパラメータ

パラメータ	説明
aggmap_name	キューブに関連付けられた aggmap の名前。 aggmap の各 RELATION 文には、値セットを含む PRECOMPUTE 句を指定する必要があります。ADVISE_CUBE は、ディメンション・メンバーのリストを各値セットに戻します。値セットが空でない場合、ADVISE_CUBE は、新しい値を追加する前に値セットの内容を削除します。
precompute_percentage	事前集計するキューブのデータのパーセンテージ。デフォルトは 20% です。

例

この例では、PRODUCT と TIME でディメンション化された UNITS というキューブに対して ADVISE_CUBE プロシージャを使用します。ADVISE_CUBE は、キューブのデータの 40% を事前集計したい場合にその対象とするべきディメンションの組合せを戻します。

```

SET SERVEROUTPUT ON
--- View valuesets
SQL>EXECUTE dbms_aw.execute('describe prodvals');
      DEFINE PRODVALS VALUESET PRODUCT
SQL>EXECUTE dbms_aw.execute('describe timevals');
      DEFINE TIMEVALS VALUESET TIME
--- View aggmap
SQL>EXECUTE dbms_aw.execute ('describe units_agg');
      DEFINE UNITS_AGG AGGMAP
      RELATION product_parentrel PRECOMPUTE (prodvals)
      RELATION time_parentrel PRECOMPUTE (timevals)
SQL>EXECUTE dbms_aw.advise_cube ('units_agg', 40);
----
---- The results are returned in the prodvals and timevals valuesets

```

参照

20-5 ページの「[集計アドバイザーの使用](#)」を参照してください。

ADVISE_REL プロシージャ

ADVISE_REL プロシージャは、アナリティック・ワークスペース内のスタンダード・フォームのディメンションをどのように事前集計するべきかを判断する際に役立ちます。事前集計するディメンションのパーセンテージを指定すると、ADVISE_REL によって、推奨するディメンション・メンバーのセットが提示されます。

ADVISE_REL プロシージャは、入力として、ファミリー・リレーション、値セットおよび事前計算のパーセンテージを取ります。ファミリー・リレーションは、ディメンションのメンバー間の階層関係を指定するスタンダード・フォームのオブジェクトです。値セットは、分析対象のディメンションから定義する必要があります。ADVISE_REL は、指定された事前計算のパーセンテージに基づいて、ディメンション・メンバーのセットを値セットに戻します。

構文

```

ADVISE_REL (
    family_relation_name    IN    VARCHAR2,
    valueset_name           IN    VARCHAR2,
    precompute_percentage   IN    INTEGER DEFAULT 20);

```

パラメータ

表 20-3 ADVISE_REL プロシージャのパラメータ

パラメータ	説明
family_relation_name	ファミリー・リレーションの名前。ファミリー・リレーションは、ディメンションと、ディメンション・メンバー間の階層関係を指定するものです。

表 20-3 ADVISE_REL プロシージャのパラメータ (続き)

パラメータ	説明
valueset_name	プロシージャの結果を格納する値セットの名前。値セットは、ファミリー・リレーションのディメンションから定義する必要があります。値セットが空でない場合、ADVISE_REL は、新しい値を追加する前に値セットの内容を削除します。
precompute_percentage	事前集計するディメンションのパーセンテージ。デフォルトは 20% です。

参照

20-5 ページの「[集計アドバイザの使用](#)」を参照してください。

EXECUTE プロシージャ

EXECUTE プロシージャは、1 つ以上の OLAP DML コマンドを実行し、出力をプリンタ・バッファに送信します。通常、対話型 SQL セッションでのアナリティック・ワークスペース・データの操作に使用されます。

SQL*Plus を使用する場合、次のコマンドを発行して、プリンタ・バッファを画面に送信できます。

```
SET SERVEROUT ON
```

別のプログラムを使用している場合に同様の設定を行う方法については、そのプログラムのドキュメントを参照してください。

入力および出力は、4KB に制限されます。より大きい値については、このパッケージの INTERP および INTERPCLOB ファンクションを参照してください。

このプロシージャは、OLAP DML の OUTFILE コマンドを使用して出力をリダイレクトした場合、DML コマンドの出力を戻しません。

構文

```
EXECUTE (
    olap_commands    IN    VARCHAR2
    text              OUT   VARCHAR2);
```

パラメータ

表 20-4 EXECUTE プロシージャのパラメータ

パラメータ	説明
olap-commands	セミコロンで区切られた 1 つ以上の OLAP DML コマンド。20-2 ページの「 OLAP DML コマンドで引用符を使用する場合のガイドライン 」を参照してください。
text	OLAP コマンドに回答して実行される OLAP 計算エンジンからの出力。

例

次の SQL*Plus セッションの例では、XADEMO というアナリティック・ワークスペースをアタッチし、XADEMO に COST_PP という式を作成した後、その新しい式の定義を表示します。

```
SQL> SET SERVEROUT ON
```

```
SQL> EXECUTE DBMS_AW.EXECUTE('AW ATTACH xademo RW; DEFINE cost_pp FORMULA
LAG(analytic_cube_f.costs, 1, time, LEVELREL time_levelrel)');
```

```
PL/SQL procedure successfully completed.
```

```
SQL> EXECUTE DBMS_AW.EXECUTE('DESCRIBE cost_pp');
```

```
DEFINE COST_PP FORMULA DECIMAL <CHANNEL GEOGRAPHY PRODUCT TIME>
EQ lag(analytic_cube_f.costs, 1, time, levelrel time.levelrel)
```

```
PL/SQL procedure successfully completed.
```

GETLOG ファンクション

このファンクションは、このパッケージで最後に実行した INTERP または INTERPCLOB ファンクションからセッション・ログを戻します。

このファンクションによって戻されるセッション・ログを出力するには、DBMS_AW.PRINTLOG プロシージャを使用します。

構文

```
GETLOG()
      RETURN CLOB;
```

戻される内容

INTERP または INTERPCLOB への最新のコールからのセッション・ログ。

例

次の例では、INTERP をコールして戻されるセッション・ログを示し、次に、GETLOG をコールして戻される同一のセッション・ログを示します。

```
SQL> SET SERVEROUT ON SIZE 1000000
SQL> EXECUTE DBMS_AW.PRINTLOG(DBMS_AW.INTERP('AW ATTACH xademo; LISTNAMES AGGMAP'));
2 AGGMAPs
```

```
-----
ANALYTIC_CUBE.AGGMAP.1
SALES_MULTIKEY_CUBE.AGGMAP.1
```

PL/SQL procedure successfully completed.

```
SQL> EXECUTE DBMS_AW.PRINTLOG(DBMS_AW.GETLOG());
2 AGGMAPs
```

```
-----
ANALYTIC_CUBE.AGGMAP.1
SALES_MULTIKEY_CUBE.AGGMAP.1
```

PL/SQL procedure successfully completed.

INTERP ファンクション

INTERP ファンクションは、1つ以上の OLAP DML コマンドを実行し、コマンドが実行されるセッション・ログを戻します。通常、EXECUTE プロシージャに対する 4KB の出力制限が非常に厳密な場合にアプリケーションで使用されます。

INTERP ファンクションへの入力は、4KB に制限されます。より大きい入力値については、このパッケージの INTERPCLOB ファンクションを参照してください。

このファンクションは、OLAP DML の OUTFILE コマンドを使用して出力をリダイレクトした場合、DML コマンドの出力を戻しません。

このパッケージの PRINTLOG プロシージャに対する引数として INTERP ファンクションを使用して、セッション・ログを表示できます。例を参照してください。

構文

```
INTERP (
    olap-commands      IN  VARCHAR2)
RETURN CLOB;
```

パラメータ

表 20-5 INTERP ファンクションのパラメータ

パラメータ	説明
olap-commands	セミコロンで区切られた 1 つ以上の OLAP DML コマンド。20-2 ページの「 OLAP DML コマンドで引用符を使用する場合のガイドライン 」を参照してください。

戻される内容

OLAP DML コマンドが実行された Oracle OLAP セッションのログ・ファイル。

例

次の SQL*Plus セッションの例では、XADEMO というアナリティック・ワークスペースをアタッチし、PRODUCT ディメンションのメンバーを表示します。

```
SQL> SET SERVEROUT ON SIZE 1000000
SQL> EXECUTE DBMS_AW.PRINTLOG(DBMS_AW.INTERP('AW ATTACH cloned; REPORT product'));
PRODUCT
-----
L1.TOTALPROD
L2.ACCDIV
L2.AUDIODIV
L2.VIDEODIV
L3.AUDIOCOMP
L3.AUDIOTAPE
.
.
.
PL/SQL procedure successfully completed.
```

INTERPCLOB ファンクション

INTERPCLOB ファンクションは、1 つ以上の OLAP DML コマンドを実行し、コマンドが実行されるセッション・ログを戻します。通常、INTERP ファンクションに対する 4KB の入力制限が非常に厳密な場合にアプリケーションで使用されます。

このファンクションは、OLAP DML の OUTFILE コマンドを使用して出力をリダイレクトした場合、OLAP DML コマンドの出力を戻しません。

このパッケージの PRINTLOG プロシージャに対する引数として INTERPCLOB ファンクションを使用して、セッション・ログを表示できます。例を参照してください。

構文

```
INTERPCLOB (
```

```

        olap-commands      IN      CLOB)
RETURN CLOB;

```

パラメータ

表 20-6 INTERPCLOB ファンクションのパラメータ

パラメータ	説明
olap-commands	セミコロンで区切られた 1 つ以上の OLAP DML コマンド。20-2 ページの「 OLAP DML コマンドで引用符を使用する場合のガイドライン 」を参照してください。

戻される内容

OLAP DML コマンドが実行された Oracle OLAP セッションのログ・ファイル。

例

次の SQL*Plus セッションの例では、ELECTRONICS というアナリティック・ワークスペースを作成し、dbs ディレクトリ別名に格納された EIF ファイルの内容をインポートして、アナリティック・ワークスペースの内容を表示します。

```

SQL> SET SERVEROUT ON SIZE 1000000
SQL> EXECUTE DBMS_AW.PRINTLOG(DBMS_AW.INTERPCLOB('AW CREATE electronics; IMPORT ALL
FROM EIF FILE ''dbs/electronics.eif'' DATA DFNS; DESCRIBE'));

```

```

DEFINE GEOGRAPHY DIMENSION TEXT WIDTH 12
LD Geography Dimension Values
DEFINE PRODUCT DIMENSION TEXT WIDTH 12
LD Product Dimension Values
DEFINE TIME DIMENSION TEXT WIDTH 12
LD Time Dimension Values
DEFINE CHANNEL DIMENSION TEXT WIDTH 12
LD Channel Dimension Values

```

```

.
.
.

```

```

PL/SQL procedure successfully completed.

```

INTERP_SILENT プロシージャ

INTERP_SILENT プロシージャは、1 つ以上の OLAP DML コマンドを実行し、それらのコマンドからのすべての出力を抑止します。OLAP コマンド・インタプリタからのエラー・メッセージは抑止されません。

INTERP_SILENT ファンクションへの入力は、4KB に制限されます。OLAP DML コマンドの出力を表示する場合は、EXECUTE プロシージャ、INTERP ファンクションまたは INTERPCLOB ファンクションを使用します。

構文

```
INTERP_SILENT (  
    olap-commands      IN  VARCHAR2);
```

パラメータ

表 20-7 INTERP_SILENT ファンクションのパラメータ

パラメータ	説明
olap-commands	セミコロンで区切られた 1 つ以上の OLAP DML コマンド。20-2 ページの「 OLAP DML コマンドで引用符を使用する場合のガイドライン 」を参照してください。

例

次のコマンドは、EXECUTE と INTERP_SILENT でのメッセージ処理の相違点を示します。いずれのコマンドを使用しても、アナリティック・ワークスペース XADemo が読込み専用モードでアタッチされます。ただし、EXECUTE は警告メッセージを表示しますが、INTERP_SILENT は表示しません。

```
SQL> EXECUTE DBMS_AW.EXECUTE('AW ATTACH xademo');  
IMPORTANT: Analytic workspace XADemo is read-only. Therefore, you will  
not be able to use the UPDATE command to save changes to it.
```

```
PL/SQL procedure successfully completed.
```

```
SQL> EXECUTE DBMS_AW.INTERP_SILENT('AW ATTACH xademo');
```

```
PL/SQL procedure successfully completed.
```

OLAP_EXPRESSION ファンクション

OLAP_EXPRESSION ファンクションを使用すると、アナリティック・ワークスペースで単一行の数値ファンクションを実行して SELECT 文にカスタム・メジャーを生成できます。OLAP_EXPRESSION は、式を計算することに加えて、WHERE および ORDER BY 句内で使用すると、SELECT の結果セットを変更することもできます。

構文

```
OLAP_EXPRESSION(  
    ...
```

```

r2c          IN  RAW(32),
expression   IN  VARCHAR2 )
RETURN NUMBER;

```

パラメータ

表 20-8 OLAP_EXPRESSION ファンクションのパラメータ

パラメータ	説明
r2c	OLAP_TABLE へのコールの ROW2CELL 句によって移入された列の名前。 ROW2CELL は、OLAP_TABLE ファンクションの制限マップ・パラメータのコンポーネントです。25-1 ページの「 OLAP_TABLE の使用 」を参照してください。
expression	アナリティック・ワークスペースで実行される数値計算。

戻される内容

OLAP_TABLE ファンクションによって戻される表オブジェクトの各行に対する式の評価。テキスト、ブールまたは日付の各データを戻す場合は、このパッケージの OLAP_EXPRESSION_TEXT、OLAP_EXPRESSION_BOOL または OLAP_EXPRESSION_DATE の各ファンクションを使用します。

注意

OLAP_EXPRESSION は、OLAP_TABLE ファンクションによって戻される表オブジェクトとともに使用します。戻される表オブジェクトには、ROW2CELL によって移入された列が含まれている必要があります。このファンクションの使用の詳細は、[第 25 章「OLAP_TABLE」](#)を参照してください。

例

20-2 ページの「[カスタム・メジャーの SELECT 文への埋込み](#)」を参照してください。

OLAP_EXPRESSION_BOOL ファンクション

OLAP_EXPRESSION_BOOL ファンクションを使用すると、アナリティック・ワークスペースで単一行のブール・ファンクションを実行して SELECT 文にカスタム・メジャーを生成できます。OLAP_EXPRESSION_BOOL は、式を計算することに加えて、WHERE および ORDER BY 句内で使用すると、SELECT の結果セットを変更することもできます。

構文

```
OLAP_EXPRESSION_BOOL(
```

```

        r2c          IN  RAW(32),
        expression  IN  VARCHAR2 )
RETURN NUMBER;
```

パラメータ

表 20-9 OLAP_EXPRESSION_BOOL ファンクションのパラメータ

パラメータ	説明
r2c	OLAP_TABLE へのコールの ROW2CELL 句によって移入された列の名前。 ROW2CELL は、OLAP_TABLE ファンクションの制限マップ・パラメータのコンポーネントです。25-1 ページの「 OLAP_TABLE の使用 」を参照してください。
expression	アナリティック・ワークスペースで実行されるブール計算。

戻される内容

OLAP_TABLE ファンクションによって戻される表オブジェクトの各行に対する式の評価。

戻り値は、数値 1 (true) または 0 (false) です。

テキスト、数値または日付の各データを戻す場合は、このパッケージの OLAP_EXPRESSION_TEXT、OLAP_EXPRESSION または OLAP_EXPRESSION_DATE の各ファンクションを使用します。

注意

OLAP_EXPRESSION_BOOL は、OLAP_TABLE ファンクションによって戻される表オブジェクトとともに使用します。戻される表オブジェクトには、ROW2CELL によって移入された列が含まれている必要があります。このファンクションの使用の詳細は、[第 25 章「OLAP_TABLE」](#)を参照してください。

例

20-2 ページの「[カスタム・メジャーの SELECT 文への埋込み](#)」を参照してください。

OLAP_EXPRESSION_DATE ファンクション

OLAP_EXPRESSION_DATE ファンクションを使用すると、アナリティック・ワークスペースで単一行の日付ファンクションを実行して SELECT 文にカスタム・メジャーを生成できます。OLAP_EXPRESSION_DATE は、式を計算することに加えて、WHERE および ORDER BY 句内で使用すると、SELECT の結果セットを変更することもできます。

構文

```
OLAP_EXPRESSION_DATE (
    r2c          IN  RAW(32) ,
    expression   IN  VARCHAR2 )
RETURN DATE;
```

パラメータ

表 20-10 OLAP_EXPRESSION_DATE ファンクションのパラメータ

パラメータ	説明
r2c	OLAP_TABLE へのコールの ROW2CELL 句によって移入された列の名前。 ROW2CELL は、OLAP_TABLE ファンクションの制限マップ・パラメータのコンポーネントです。25-1 ページの「 OLAP_TABLE の使用 」を参照してください。
expression	アナリティック・ワークスペースで実行される日付計算。

戻される内容

OLAP_TABLE ファンクションによって戻される表オブジェクトの各行に対する式の評価。テキスト、ブールまたは数値の各データを戻す場合は、このパッケージの OLAP_EXPRESSION_TEXT、OLAP_EXPRESSION_BOOL または OLAP_EXPRESSION の各ファンクションを使用します。

注意

OLAP_EXPRESSION_DATE は、OLAP_TABLE ファンクションによって戻される表オブジェクトとともに使用します。戻される表オブジェクトには、ROW2CELL によって移入された列が含まれている必要があります。このファンクションの使用の詳細は、[第 25 章「OLAP_TABLE」](#) を参照してください。

例

20-2 ページの「[カスタム・メジャーの SELECT 文への埋込み](#)」を参照してください。

OLAP_EXPRESSION_TEXT ファンクション

OLAP_EXPRESSION_TEXT ファンクションを使用すると、アナリティック・ワークスペースで単一行のテキスト・ファンクションを実行して SELECT 文にカスタム・メジャーを生成できます。OLAP_EXPRESSION_TEXT は、式を計算することに加えて、WHERE および ORDER BY 句内で使用すると、SELECT の結果セットを変更することもできます。

構文

```
OLAP_EXPRESSION_TEXT (  
    r2c          IN    RAW(32),  
    expression   IN    VARCHAR2 )  
RETURN VARCHAR2;
```

パラメータ

表 20-11 OLAP_EXPRESSION_TEXT ファンクションのパラメータ

パラメータ	説明
r2c	OLAP_TABLE へのコールの ROW2CELL 句によって移入された列の名前。 ROW2CELL は、OLAP_TABLE ファンクションの制限マップ・パラメータのコンポーネントです。25-1 ページの「 OLAP_TABLE の使用 」を参照してください。
expression	アナリティック・ワークスペースで実行されるテキスト計算。

戻される内容

OLAP_TABLE ファンクションによって戻される表オブジェクトの各行に対する式の評価。

数値、ブールまたは日付の各データを戻す場合は、このパッケージの OLAP_EXPRESSION、OLAP_EXPRESSION_BOOL または OLAP_EXPRESSION_DATE の各ファンクションを使用します。

注意

OLAP_EXPRESSION_TEXT は、OLAP_TABLE ファンクションによって戻される表オブジェクトとともに使用します。戻される表オブジェクトには、ROW2CELL によって移入された列が含まれている必要があります。このファンクションの使用方法の詳細は、[第 25 章「OLAP_TABLE」](#) を参照してください。

例

20-2 ページの「[カスタム・メジャーの SELECT 文への埋込み](#)」を参照してください。

PRINTLOG プロシージャ

このプロシージャは、PL/SQL の DBMS_OUTPUT パッケージを使用して、このパッケージの INTERP、INTERPCLOB または GETLOG ファンクションによって戻されたセッション・ログをプリンタ・バッファに送信します。

SQL*Plus を使用する場合、次のコマンドを発行して、プリンタ・バッファを画面に送信できます。

```
SET SERVEROUT ON SIZE 1000000
```

SIZE 句を使用すると、このバッファを、デフォルト・サイズの 4KB から増やすことができます。

別のプログラムを使用している場合に同様の設定を行う方法については、そのプログラムのドキュメントを参照してください。

構文

```
DBMS_AW.PRINTLOG (  
    session-log    IN    CLOB);
```

パラメータ

表 20-12 PRINTLOG プロシージャのパラメータ

パラメータ	説明
session-log	セッションのログ。

例

次の例では、INTERP ファンクションによって戻されるセッション・ログを示します。

```
SQL> SET SERVEROUT ON SIZE 1000000  
SQL> EXECUTE DBMS_AW.PRINTLOG(DBMS_AW.INTERP('DESCRIBE analytic_cube_f.profit'));  
  
DEFINE ANALYTIC_CUBE.F.PROFIT FORMULA DECIMAL <CHANNEL  
GEOGRAPHY PRODUCT TIME>  
EQ analytic_cube.f.sales - analytic_cube.f.costs  
  
PL/SQL procedure successfully completed.
```

DBMS_AW_UTILITIES

DBMS_AW_UTILITIES パッケージには、アナリティック・ワークスペースのカスタム・メジャーを管理するためのプロシージャが含まれています。

参照：

- アナリティック・ワークスペースの詳細は、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。
- アナリティック・ワークスペースのリレーショナル・ビューを作成する方法については、第 1 章「DBMS_AWM を使用したアナリティック・ワークスペースの作成」を参照してください。

この章では、次の項目について説明します。

- [カスタム・メジャーについて](#)
- [カスタム・メジャーに対する問合せ](#)
- [例：カスタム・メジャーの作成](#)
- [DBMS_AW_UTILITIES サブプログラムの概要](#)

カスタム・メジャーについて

DBMS_AW_UTILITIES パッケージを使用すると、データベース・スタンダード・フォームのアナリティック・ワークスペース内にカスタム・メジャーを定義し、そのカスタム・メジャーをリレーショナル・ビューの列に関連付けることができます。現行セッション用に一時的なカスタム・メジャーを定義することも、カスタム・メジャーを永続的に保存することもできます。

注意： スタンダード・フォームのアナリティック・ワークスペースと、その内容を公開するリレーショナル・ビューは、DBMS_AWMパッケージのプロシージャで作成します。

カスタム・メジャーは、1つ以上の格納済メジャーから導出されます。カスタム・メジャーは実行時に計算され、ファクト表と同じように構成されたビューの列に戻されます。PROFITS はカスタム・メジャーの例で、SALES メジャーから COSTS メジャーを差し引くことで計算されています。

DBMS_AW_UTILITIES で作成されたカスタム・メジャーは、アナリティック・ワークスペース内の式として定義されます。式は、計算を表すワークスペース・スキーマ・オブジェクトです。式の結果セットには、ステータスが設定されている各ワークスペース・ディメンション・メンバーの値が含まれます。

カスタム・メジャーに対する問合せ

CREATE_CUSTOM_MEASURE プロシージャによって新しいカスタム・メジャーが正常に作成されると、次の情報が表示されます。

```
Custom Measure cust_meas_name created in Workspace workspace_name.  
Custom Measure cust_meas_name mapped to column col_name in View view_name.
```

指定された列に問合せを発行することで、カスタム・メジャー計算の結果を取得できます。

また、次に示す表に問合せを発行すると、CREATE_CUSTOM_MEASURE で作成されたカスタム・メジャーに関する情報を取得できます。これらの表から、カスタム・メジャー計算の結果を格納する列の名前を取得することもできます。

- olapsys.CWM2\$_AW_PERM_CUST_MEAS_MAP — 永続的なカスタム・メジャーに関する情報を提供します。この表を使用できるのは DBA 権限を持つユーザーのみです。
- olapsys.CWM2\$_AW_TEMP_CUST_MEAS_MAP — 一時的なカスタム・メジャーに関する情報を提供します。この表には現行ユーザーがアクセスできます。

CWM2\$_AW_PERM_CUST_MEAS_MAP

次の表に、CWM2\$_AW_PERM_CUST_MEAS_MAP 表の列を示します。

列	データ型	NULL	説明
AW_ACCESS_VIEW_NAME	VARCHAR2(61)	NOT NULL	永続的なカスタム・メジャーを格納するビューの名前。
CUST_ADT_COLUMN	VARCHAR2(30)	NOT NULL	ビューの列。

列	データ型	NULL	説明
WORKSPACE_NAME	VARCHAR2 (61)		カスタム・メジャーの基礎となるメジャー、およびカスタム・メジャー計算を定義する式を格納するアナリティック・ワークスペースの名前。
AW_MEASURE_NAME	VARCHAR2 (64)		導出された (カスタム) メジャーの名前。
SESSIONID	VARCHAR2 (10)		カスタム・メジャーが作成されたセッションの ID。
USERNAME	VARCHAR2 (30)		カスタム・メジャーを作成したユーザー。

CWM2\$_AW_TEMP_CUST_MEAS_MAP

次の表に、CWM2\$_AW_TEMP_CUST_MEAS_MAP 表の列を示します。

列	データ型	NULL	説明
AW_ACCESS_VIEW_NAME	VARCHAR2 (61)	NOT NULL	一時的なカスタム・メジャーを格納するビューの名前。
CUST_ADT_COLUMN	VARCHAR2 (30)	NOT NULL	ビューの列。
WORKSPACE_NAME	VARCHAR2 (61)		カスタム・メジャーの基礎となるメジャー、およびカスタム・メジャー計算を定義する式を格納するアナリティック・ワークスペースの名前。
AW_MEASURE_NAME	VARCHAR2 (64)		導出された (カスタム) メジャーの名前。
SESSIONID	VARCHAR2 (10)		現行セッションの ID。一時的なカスタム・メジャーが存在しているのは現行セッションの間のみです。
USERNAME	VARCHAR2 (30)		カスタム・メジャーを作成したユーザー。

例 : カスタム・メジャーの作成

次の例では、アナリティック・ワークスペース GLOBAL_AW.GLOBAL に一時的なカスタム・メジャーを作成します。このメジャーは、キューブ PRICE_CUBE の単位価格と単位原価の差を戻します。このカスタム・メジャーは、ビュー GLOBAL_AW.GLOB_GLOBA_UNITS_CU10VIEW に戻されます。

問合せの出力を表示するために、出力を画面に送信します。

```
SQL>set serverout on
SQL>exec cwm2_olap_manager.set_echo_on;
```

例 : カスタム・メジャーの作成

次の問合せを使用すると、使用可能なアナリティック・ワークスペースのリストを取得できます。

```
SQL>select * from all_olap2_aws where aw = 'GLOBAL';
```

OWNER	AW	AW_NUMBER
GLOBAL_AW	GLOBAL	1005

次の問合せは、アナリティック・ワークスペースのキューブに対して有効になっているビューのリストを戻します。

```
SQL>select * from all_aw_cube_enabled_views where aw_name = 'GLOBAL';
```

AW_OWNER	AW_NAME	CUBE_NAME	HIERCOMBO_NU	HIERCOMBO_STR	SYSTEM_VIEWNAME	USERP_VIE
GLOBAL_AW	GLOBAL	PRICE_CUBE	#####	DIM:PRODUCT/HIER:PRODUCT_ROLLUP; DIM:TIME/HIER:CALENDAR	GLOB_GLOBA_PRICE_CU4VIEW	
GLOBAL_AW	GLOBAL	UNITS_CUBE	#####	DIM:CHANNEL/HIER:CHANNEL_ROLLUP; DIM:CUSTOMER/HIER:MARKET_ROLLUP; DIM:PRODUCT/HIER:PRODUCT_ROLLUP; DIM:TIME/HIER:CALENDAR	GLOB_GLOBA_UNITS_CU9VIEW	
GLOBAL_AW	GLOBAL	UNITS_CUBE	#####	DIM:CHANNEL/HIER:CHANNEL_ROLLUP; DIM:CUSTOMER/HIER:SHIPMENTS_ROLLUP; DIM:PRODUCT/HIER:PRODUCT_ROLLUP; DIM:TIME/HIER:CALENDAR	GLOB_GLOBA_UNITS_CU10VIEW	

次のアクティブ・カタログ・ビューに対して問合せを発行すると、キューブ内のメジャーの名前を取得できます。

```
SQL>select * from all_olap2_aw_cube_measures where aw_name = 'GLOBAL';
```

AW_OWNER	AW_NAM	AW_CUBE_NAM	AW_MEASURE_	AW_PHYSICAL_	MEASURE_SOU	DISPLAY_NAM	DESCRIPTI	IS_AGGR
GLOBAL_AW	GLOBAL	PRICE_CUBE	UNIT_COST	UNIT_COST	UNIT_COST	UNIT COST	Unit Cost	YES
GLOBAL_AW	GLOBAL	PRICE_CUBE	UNIT_PRICE	UNIT_PRICE	UNIT_PRICE	UNIT PRICE	Unit Price	YES
GLOBAL_AW	GLOBAL	UNITS_CUBE	UNITS	UNITS	UNITS	UNITS	Units Sold	YES

次の文は、GLOBAL_AW スキーマのアナリティック・ワークスペース GLOBAL に、数値式 PRICE_COST を作成します。この式は、単位価格と単位原価の差を計算します。結果のデータは、ビュー GLOBAL_AW.GLOB_GLOBA_UNITS_CU10VIEW に戻されます。

```
SQL>execute dbms_aw_utilities.create_custom_measure  
('GLOBAL_AW.GLOBAL', 'PRICE_COST',
```

```
'UNIT_PRICE - UNIT_COST', 'temporary',
'GLOBAL_AW.GLOB_GLOBA_UNITS_CU10VIEW');
```

```
Custom Measure 'PRICE_COST' created in Workspace 'GLOBAL_AW.GLOBAL'.
Custom Measure 'PRICE_COST' mapped to column 'CUST_MEAS_NUM1'
in View 'GLOBAL_AW.GLOB_GLOBA_UNITS_CU10VIEW'.
```

次の問合せでは、新たに作成したカスタム・メジャーが
CWM2\$_AW_temp_CUST_MEAS_MAP 表にリストされていることを確認できます。

```
SQL>select * from olapsys.CWM2$_AW_TEMP_CUST_MEAS_MAP
      where workspace_name = 'GLOBAL_AW.GLOBAL';
```

AW_ACCESS_VIEW_NAME	CUST_ADT_COLUMN	WORKSPACE_NAME	AW_MEASURE_NAME	SESSIONID	USERNAME
GLOBAL_AW.GLOB_GLOBA_UNITS_CU10VIEW	CUST_MEAS_NUM1	GLOBAL_AW.GLOBAL	PRICE_COST	325	MYUSER

カスタム計算によって得られたデータを取得するには、次の問合せを使用します。

```
SQL>select CUST_MEAS_NUM1 from GLOBAL_AW.GLOB_GLOBA_UNITS_CU10VIEW;
```

DBMS_AW_UTILITIES サブプログラムの概要

表 21-1 に、DBMS_AW_UTILITIES で提供されるサブプログラムを示します。

表 21-1 DBMS_AW_UTILITIES

サブプログラム	説明
CREATE_CUSTOM_MEASURE プロシージャ (21-6 ページ)	OLAP 式を作成し、その式をアナリティック・ワークスペースのファクト・ビューの列に関連付けます。
DELETE_CUSTOM_MEASURE プロシージャ (21-8 ページ)	CREATE_CUSTOM_MEASURE で作成されたカスタム・メジャーを削除します。
UPDATE_CUSTOM_MEASURE プロシージャ (21-8 ページ)	CREATE_CUSTOM_MEASURE で作成された OLAP 式の定義を変更します。

CREATE_CUSTOM_MEASURE プロシージャ

CREATE_CUSTOM_MEASURE プロシージャでは、アナリティック・ワークスペース内の式オブジェクトに作成および格納する計算を指定します。式は、アナリティック・ワークスペース内に永続的に定義することも、ワークスペースが閉じられるまで一時的に存在するようにすることもできます。

CREATE_CUSTOM_MEASURE は、式をファクト・ビューの列に関連付けます。これらの列に対して問合せを発行すると、式によってカスタム・メジャーが計算され、結果セットが列に移入されます。CREATE_CUSTOM_MEASURE は、ファクト・ビューが、DBMS_AWM.CREATE_AWCUBE_ACCESS プロシージャによって生成された有効化スクリプトで作成されていることを前提としています。ファクト・ビューは、アナリティック・ワークスペース・キューブのメジャーを論理ファクト表のセットとして表示します。階層の各組合せを表すための別のビューもあります。

作成されるビューは、カスタム・メジャー用に使用可能な空のテキスト列と数値列を持ちます。空のテキスト列と数値列は、それぞれ 100 個あります。

テキスト列は CUST_MEAS_TEXT n という名前で、 n は 1 ~ 100 の数字です。データ型は VARCHAR2 (1000) です。

数値列は CUST_MEAS_NUM n という名前で、 n は 1 ~ 100 の数字です。データ型は NUMBER です。

構文

```
CREATE_CUSTOM_MEASURE(
    aw_name                VARCHAR2,
    aw_formula_name        VARCHAR2,
    aw_formula_expr        VARCHAR2,
```

```
aw_formula_create_mode  VARCHAR2,
view_name                VARCHAR2);
```

パラメータ

表 21-2 CREATE_CUSTOM_MEASURE プロシージャのパラメータ

パラメータ	説明
aw_name	アナリティック・ワークスペースの名前。この名前は、 <code>owner.name</code> の形式で指定する必要があります。owner はスキーマ名で、name はワークスペース名です。
aw_formula_name	アナリティック・ワークスペースに作成する式の名前。
aw_formula_expr	式に格納するテキスト式または数値式。
aw_formula_create_mode	次のいずれかの値を選択します。 'PERMANENT' -- 式を永続的にアナリティック・ワークスペースに作成します。ワークスペースは読み込み / 書き込みモードで開かれ、更新およびコミットされます。 'TEMPORARY' -- 式を一時的にアナリティック・ワークスペースに作成します。ワークスペースは読み込み専用モードで開かれ、式はワークスペースが閉じられるときに破棄されます。
view_name	OLAP_TABLE ファンクションを使用してアナリティック・ワークスペースにアクセスし、カスタム・メジャーのデータを読み込むビューの名前。 テキスト・データは <code>CUST_MEAS_TEXTn</code> という列に戻されます。n は、次に利用可能な列の連番です。 数値データは <code>CUST_MEAS_NUMn</code> という列に戻されます。n は、次に利用可能な列の連番です。

参照

- 22-21 ページの「[CREATE_AWCUBE_ACCESS プロシージャ](#)」を参照してください。
- 1-5 ページの「[ワークスペース・キューブへのリレーショナル・アクセスの有効化](#)」を参照してください。
- 1-22 ページの「[ワークスペース・キューブへのリレーショナル・アクセスの作成](#)」を参照してください。

DELETE_CUSTOM_MEASURE プロシージャ

DELETE_CUSTOM_MEASURE プロシージャは、CREATE_CUSTOM_MEASURE によって作成されたカスタム・メジャーを削除します。このプロシージャは、アナリティック・ワークスペースのカスタム・メジャーを計算する式を削除し、ファクト・ビューの列からその式を削除します。

構文

```
DELETE_CUSTOM_MEASURE(
    aw_name          VARCHAR2,
    aw_formula_name  VARCHAR2,
    view_name        VARCHAR2);
```

パラメータ

表 21-3 DELETE_CUSTOM_MEASURE プロシージャのパラメータ

パラメータ	説明
aw_name	アナリティック・ワークスペースの名前。この名前は、 <code>owner.name</code> の形式で指定する必要があります。owner はスキーマ名で、name はワークスペース名です。
aw_formula_name	アナリティック・ワークスペースから削除する式の名前。
view_name	CREATE_CUSTOM_MEASURE で指定したビューの名前。カスタム・メジャーへの参照がこのビューの列から削除されます。

UPDATE_CUSTOM_MEASURE プロシージャ

このプロシージャは、アナリティック・ワークスペースのカスタム・メジャーの式を更新します。

式は、CREATE_CUSTOM_MEASURE プロシージャによって定義され、ビューに関連付けられています。

構文

```
UPDATE_CUSTOM_MEASURE(
    aw_name          VARCHAR2,
    aw_formula_name  VARCHAR2,
    aw_formula_expr  VARCHAR2);
```

パラメータ

表 21-4 UPDATE_CUSTOM_MEASURE プロシージャのパラメータ

パラメータ	説明
aw_name	アナリティック・ワークスペースの名前。この名前は、 <code>owner.name</code> の形式で指定する必要があります。owner はスキーマ名で、name はワークスペース名です。
aw_formula_name	アナリティック・ワークスペースの式の名前。
aw_formula_expr	式で実行する新しい計算。

Analytic Workspace Manager のパッケージである DBMS_AWM では、リレーショナル・データ・ウェアハウスのデータをアナリティック・ワークスペースにロードし、OLAP API および BI Beans がそのワークスペースにアクセスできるようにするためのプロシージャが提供されます。

注意： DBMS_AWM パッケージのほとんどの機能は、Analytic Workspace Manager の Graphical User Interface (GUI) から使用できます。

参照：

- [第 1 章「DBMS_AWM を使用したアナリティック・ワークスペースの作成」](#) を参照してください。
- [第 2 章「CWM2 による OLAP カタログ・メタデータの作成」](#) を参照してください。

この章では、次の項目について説明します。

- [DBMS_AWM サブプログラムのパラメータ](#)
- [DBMS_AWM サブプログラムの概要](#)

DBMS_AWM サブプログラムのパラメータ

cube_name、dimension_name、measure_name および level_name の各パラメータは、リレーショナル・ソース・キューブにマップされる、OLAP カタログ内のメタデータ・エンティティを参照します。

aw_cube_name または aw_dimension_name の各パラメータは、アナリティック・ワークスペース内のターゲット・キューブまたはディメンションを参照します。

接尾辞 `_spec` が付くパラメータは、アナリティック・ワークスペース内のターゲット・キューブをロード、集計および最適化するための特定の仕様を参照します。

参照：「リレーショナル・ソース・キューブ」、「多次元ターゲット・キューブ」および「リレーショナル・ターゲット・キューブ」の各用語の定義については、1-2 ページの「概要」を参照してください。

表 22-1 に、DBMS_AWM のパラメータを示します。

表 22-1 DBMS_AWM プロシージャのパラメータ

パラメータ	説明
<code>cube_owner</code>	リレーショナル・ソース表（スター・スキーマ）に関連付けられた OLAP カタログ・キューブの所有者。
<code>cube_name</code>	リレーショナル・ソース表（スター・スキーマ）に関連付けられた OLAP カタログ・キューブの名前。
<code>dimension_owner</code>	ソース・ディメンション参照表に関連付けられた OLAP カタログ・ディメンションの所有者。
<code>dimension_name</code>	ソース・ディメンション参照表に関連付けられた OLAP カタログ・ディメンションの名前。
<code>aw_owner</code>	アナリティック・ワークスペースの所有者。ワークスペース内のキューブおよびディメンションの所有者でもあります。
<code>aw_cube_name</code>	アナリティック・ワークスペース内のターゲット・キューブの名前。ネーミング要件については、表 22-13 「CREATE_AWCUBE プロシージャのパラメータ」を参照してください。
<code>aw_dimension_name</code>	アナリティック・ワークスペース内のターゲット・ディメンションの名前。ネーミング要件については、表 22-18 「CREATE_AWDIMENSION プロシージャのパラメータ」を参照してください。
<code>dimension_load_spec</code>	OLAP カタログ・ソース・ディメンションをアナリティック・ワークスペース内のターゲット・ディメンションにロードするための仕様の名前。
<code>cube_load_spec</code>	OLAP カタログ・ソース・キューブをアナリティック・ワークスペース内のターゲット・キューブにロードするための仕様の名前。
<code>aggregation_spec</code>	アナリティック・ワークスペース内のターゲット・キューブのストアド・サマリーを作成するための仕様の名前。
<code>composite_spec</code>	アナリティック・ワークスペース内のターゲット・キューブのコンポジットおよびディメンション順序を定義するための仕様の名前。

DBMS_AWM サブプログラムの概要

表 22-2 に、DBMS_AWM サブプログラムをアルファベット順に示します。各サブプログラムの詳細は、この章の後半で説明します。

DBMS_AWM サブプログラムの機能別のリストは、1-6 ページの「DBMS_AWM プロシージャの理解」で確認できます。

表 22-2 DBMS_AWM サブプログラム

サブプログラム	説明
ADD_AWCOMP_SPEC_COMP_MEMBER プロシージャ (22-6 ページ)	コンポジット仕様のコンポジットにメンバーを追加します。
ADD_AWCOMP_SPEC_MEMBER プロシージャ (22-8 ページ)	コンポジット仕様にメンバーを追加します。
ADD_AWCUBEAGG_SPEC_LEVEL プロシージャ (22-9 ページ)	集計仕様にレベルを追加します。
ADD_AWCUBEAGG_SPEC_MEASURE プロシージャ (22-10 ページ)	集計仕様にメジャーを追加します。
ADD_AWCUBELOAD_SPEC_COMP プロシージャ (22-11 ページ)	キューブ・ロード仕様にコンポジット仕様を追加します。
ADD_AWCUBELOAD_SPEC_FILTER プロシージャ (22-12 ページ)	キューブ・ロード仕様に WHERE 句を追加します。
ADD_AWCUBELOAD_SPEC_MEASURE プロシージャ (22-13 ページ)	キューブ・ロード仕様にメジャーを追加します。
ADD_AWDIMLOAD_SPEC_FILTER プロシージャ (22-15 ページ)	ディメンション・ロード仕様に WHERE 句を追加します。
AGGREGATE_AWCUBE プロシージャ (22-16 ページ)	アナリティック・ワークスペース内のキューブのストアド・サマリーを作成します。
CREATE_AWCOMP_SPEC プロシージャ (22-17 ページ)	キューブのコンポジット仕様を作成します。
CREATE_AWCUBE プロシージャ (22-19 ページ)	アナリティック・ワークスペース内に、OLAP カタログで定義されているキューブを保持するためのコンテンツを作成します。
CREATE_AWCUBE_ACCESS プロシージャ (22-21 ページ)	アナリティック・ワークスペース内のキューブへのリレーショナル・アクセスを有効化するスクリプトを作成します。

表 22-2 DBMS_AWM サブプログラム (続き)

サブプログラム	説明
CREATE_AWCUBE_ACCESS_FULL プロシージャ (22-23 ページ)	アナリティック・ワークスペース内のキューブへのリレーショナル・アクセスを有効にします。
CREATE_AWCUBEAGG_SPEC プロシージャ (22-24 ページ)	キューブの集計仕様を作成します。
CREATE_AWCUBELOAD_SPEC プロシージャ (22-26 ページ)	キューブのロード仕様を作成します。
CREATE_AWDIMENSION プロシージャ (22-28 ページ)	アナリティック・ワークスペース内に、OLAP カタログで定義されているディメンションを保持するためのコンテナを作成します。
CREATE_AWDIMENSION_ACCESS プロシージャ (22-30 ページ)	アナリティック・ワークスペース内のディメンションへのリレーショナル・アクセスを有効化するスクリプトを作成します。
CREATE_AWDIMENSION_ACCESS_FULL プロシージャ (22-32 ページ)	アナリティック・ワークスペース内のディメンションへのリレーショナル・アクセスを有効にします。
CREATE_AWDIMLOAD_SPEC プロシージャ (22-33 ページ)	ディメンションのロード仕様を作成します。
DELETE_AWCOMP_SPEC プロシージャ (22-35 ページ)	コンポジット仕様を削除します。
DELETE_AWCOMP_SPEC_MEMBER プロシージャ (22-35 ページ)	コンポジット仕様のメンバーを削除します。
DELETE_AWCUBE_ACCESS プロシージャ (22-36 ページ)	アナリティック・ワークスペース内のキューブへのリレーショナル・アクセスを無効化するスクリプトを作成します。
DELETE_AWCUBEAGG_SPEC プロシージャ (22-37 ページ)	集計仕様を削除します。
DELETE_AWCUBEAGG_SPEC_LEVEL プロシージャ (22-38 ページ)	集計仕様からレベルを削除します。
DELETE_AWCUBEAGG_SPEC_MEASURE プロシージャ (22-39 ページ)	集計仕様からメジャーを削除します。
DELETE_AWCUBELOAD_SPEC プロシージャ (22-40 ページ)	キューブ・ロード仕様を削除します。
DELETE_AWCUBELOAD_SPEC_COMP プロシージャ (22-40 ページ)	キューブ・ロード仕様からコンポジット仕様を削除します。

表 22-2 DBMS_AWM サブプログラム (続き)

サブプログラム	説明
DELETE_AWCUBELOAD_SPEC_FILTER プロシージャ (22-41 ページ)	キューブ・ロード仕様から WHERE 句を削除します。
DELETE_AWCUBELOAD_SPEC_MEASURE プロシージャ (22-42 ページ)	キューブ・ロード仕様からメジャーを削除します。
DELETE_AWDIMENSION_ACCESS プロシージャ (22-42 ページ)	アナリティック・ワークスペース内のディメンションへのリレーショナル・アクセスを無効化するスクリプトを作成します。
DELETE_AWDIMLOAD_SPEC プロシージャ (22-44 ページ)	ディメンション・ロード仕様を削除します。
DELETE_AWDIMLOAD_SPEC_FILTER プロシージャ (22-44 ページ)	ディメンション・ロード仕様から WHERE 句を削除します。
REFRESH_AWCUBE プロシージャ (22-45 ページ)	OLAP カタログ・ソース・キューブのデータおよびメタデータを、アナリティック・ワークスペース内のターゲット・キューブにロードします。
REFRESH_AWDIMENSION プロシージャ (22-47 ページ)	OLAP カタログ・ソース・ディメンションのデータおよびメタデータを、アナリティック・ワークスペース内のターゲット・ディメンションにロードします。
SET_AWCOMP_SPEC_CUBE プロシージャ (22-49 ページ)	コンポジット仕様に関連付けられているキューブを変更します。
SET_AWCOMP_SPEC_MEMBER_NAME プロシージャ (22-50 ページ)	コンポジット仕様のメンバーの名前を変更します。
SET_AWCOMP_SPEC_MEMBER_POS プロシージャ (22-51 ページ)	コンポジット仕様内でのメンバーの位置を変更します。
SET_AWCOMP_SPEC_MEMBER_SEG プロシージャ (22-52 ページ)	コンポジット仕様のメンバーに関連付けられているセグメント・サイズを変更します。
SET_AWCOMP_SPEC_NAME プロシージャ (22-54 ページ)	コンポジット仕様の名前を変更します。
SET_AWCUBE_VIEW_NAME プロシージャ (22-55 ページ)	アナリティック・ワークスペース・キューブのリレーショナル・ビューの名前を変更します。
SET_AWCUBEAGG_SPEC_AGGOP プロシージャ (22-56 ページ)	キューブのディメンションに対するメジャー集計の集計演算子を指定します。
SET_AWCUBELOAD_SPEC_CUBE プロシージャ (22-57 ページ)	キューブ・ロード仕様に関連付けられているキューブを変更します。

表 22-2 DBMS_AWM サブプログラム (続き)

サブプログラム	説明
SET_AWCUBELOAD_SPEC_LOADTYPE プロシージャ (22-58 ページ)	キューブ・ロード仕様のタイプを変更します。
SET_AWCUBELOAD_SPEC_NAME プロシージャ (22-58 ページ)	キューブ・ロード仕様の名前を変更します。
SET_AWCUBELOAD_SPEC_PARAMETER プロシージャ (22-59 ページ)	キューブ・ロード仕様のパラメータを設定します。
SET_AWDIMENSION_VIEW_NAME プロシージャ (22-60 ページ)	アナリティック・ワークスペース・ディメンションのリレーショナル・ビューの名前を変更します。
SET_AWDIMLOAD_SPEC_DIMENSION プロシージャ (22-61 ページ)	ディメンション・ロード仕様に関連付けられているディメンションを変更します。
SET_AWDIMLOAD_SPEC_LOADTYPE プロシージャ (22-62 ページ)	ディメンション・ロード仕様のタイプを変更します。
SET_AWDIMLOAD_SPEC_NAME プロシージャ (22-63 ページ)	ディメンション・ロード仕様の名前を変更します。
SET_AWDIMLOAD_SPEC_PARAMETER プロシージャ (22-64 ページ)	ディメンション・ロード仕様のパラメータを設定します。

ADD_AWCOMP_SPEC_COMP_MEMBER プロシージャ

このプロシージャは、コンポジット仕様のコンポジットにメンバーを追加します。メンバーは、ディメンションまたはネストしたコンポジットです。

コンポジットのメンバーは適切な順序で追加する必要があります。メンバーの順序を変更する場合は、そのコンポジットを削除してから再作成する必要があります。このためには、DELETE_AWCOMP_SPEC_MEMBER および ADD_AWCOMP_SPEC_MEMBER をコールします。

構文

```
ADD_AWCOMP_SPEC_COMP_MEMBER (
    composite_spec          IN  VARCHAR2,
    cube_owner              IN  VARCHAR2,
    cube_name               IN  VARCHAR2,
    composite_name          IN  VARCHAR2,
    nested_member_name      IN  VARCHAR2,
    nested_member_type      IN  VARCHARs,
    dimension_owner         IN  VARCHAR2 DEFAULT NULL,
    dimension_name          IN  VARCHAR2 DEFAULT NULL);
```

パラメータ

表 22-3 ADD_AWCOMP_SPEC_COMP_MEMBER プロシージャのパラメータ

パラメータ	説明
composite_spec	キューブのコンポジット仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
composite_name	コンポジット仕様のコンポジットの名前。
nested_member_name	コンポジットに追加するメンバーの名前。
nested_member_type	新しいメンバーのタイプ。タイプには、'DIMENSION' または 'COMPOSITE' を指定できます。
dimension_owner	コンポジットに追加する OLAP カタログ・ソース・ディメンションの所有者。新しいメンバーがディメンションではなくネストしたコンポジットである場合は、このパラメータを NULL (デフォルト) にする必要があります。
dimension_name	コンポジットに追加する OLAP カタログ・ソース・ディメンションの名前。新しいメンバーがディメンションではなくネストしたコンポジットである場合は、このパラメータを NULL (デフォルト) にする必要があります。

例

次の文は、PRODUCT および GEOGRAPHY というディメンションからなるコンポジット COMP1 を、コンポジット仕様 AC_COMPSPEC に追加します。

```
execute DBMS_AWM.Create_AWComp_spec
    ('AC_COMPSPEC', 'XADEMO', 'ANALYTIC_CUBE');
execute DBMS_AWM.Add_AWComp_Spec_Member
    ('AC_COMPSPEC', 'XADEMO', 'ANALYTIC_CUBE', 'COMP1', 'COMPOSITE');
execute DBMS_AWM.Add_AWComp_Spec_Comp_Member
    ('AC_COMPSPEC', 'XADEMO', 'ANALYTIC_CUBE', 'COMP1', 'PROD_COMP',
    'DIMENSION', 'XADEMO', 'PRODUCT');
execute DBMS_AWM.Add_AWComp_Spec_Comp_Member
    ('AC_COMPSPEC', 'XADEMO', 'ANALYTIC_CUBE', 'COMP1', 'GEOG_COMP',
    'DIMENSION', 'XADEMO', 'GEOGRAPHY');
```

参照

- 1-16 ページの「スパースなデータの管理およびワークスペース・キューブの最適化」を参照してください。

- 22-35 ページの「[DELETE_AWCOMP_SPEC_MEMBER プロシージャ](#)」を参照してください。
- 22-8 ページの「[ADD_AWCOMP_SPEC_MEMBER プロシージャ](#)」を参照してください。
- 22-17 ページの「[CREATE_AWCOMP_SPEC プロシージャ](#)」を参照してください。

ADD_AWCOMP_SPEC_MEMBER プロシージャ

このプロシージャは、コンポジット仕様にメンバーを追加します。コンポジット仕様のメンバーは、コンポジットとディメンションです。

構文

```
ADD_AWCOMP_SPEC_MEMBER (
    composite_spec      IN    VARCHAR2,
    cube_owner         IN    VARCHAR2,
    cube_name          IN    VARCHAR2,
    member_name        IN    VARCHAR2,
    member_type        IN    VARCHAR2,
    dimension_owner    IN    VARCHAR2 DEFAULT NULL,
    diimension_name    IN    VARCHAR2 DEFAULT NULL);
```

パラメータ

表 22-4 ADD_AWCOMP_SPEC_MEMBER プロシージャのパラメータ

パラメータ	説明
composite_spec	キューブのコンポジット仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
member_name	コンポジット仕様のメンバーの名前。
member_type	メンバーのタイプ。タイプには、'DIMENSION' または 'COMPOSITE' を指定できます。
dimension_owner	コンポジット仕様に追加する OLAP カタログ・ソース・ディメンションの所有者。新しいメンバーがディメンションではなくコンポジットである場合は、このパラメータを NULL (デフォルト) にする必要があります。
dimension_name	コンポジット仕様に追加する OLAP カタログ・ソース・ディメンションの名前。新しいメンバーがディメンションではなくコンポジットである場合は、このパラメータを NULL (デフォルト) にする必要があります。

例

次の文は、Time ディメンションと COMP1 というコンポジットを、コンポジット仕様 AC_COMPSPEC に追加します。

```
execute DBMS_AWM.Add_AWComp_Spec_Member
    ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'TIMECOMP_MEMBER' ,
     'DIMENSION' , 'XADEMO' , 'TIME');
execute DBMS_AWM.Add_AWComp_Spec_Member
    ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'COMP1' , 'COMPOSITE');
```

参照

- 1-16 ページの「[スパースなデータの管理およびワークスペース・キューブの最適化](#)」を参照してください。
- 22-17 ページの「[CREATE_AWCOMP_SPEC プロシージャ](#)」を参照してください。

ADD_AWCUBEAGG_SPEC_LEVEL プロシージャ

このプロシージャは、集計仕様にレベルを追加します。

構文

```
ADD_AWCUBEAGG_SPEC_LEVEL (
    aggregation_spec    IN    VARCHAR2,
    aw_owner            IN    VARCHAR2,
    aw_name             IN    VARCHAR2,
    aw_cube_name       IN    VARCHAR2,
    aw_dimension_name  IN    VARCHAR2,
    aw_level_name      IN    VARCHAR2);
```

パラメータ

表 22-5 ADD_AWCUBEAGG_SPEC_LEVEL プロシージャのパラメータ

パラメータ	説明
aggregation_spec	アナリティック・ワークスペース内のキューブの集計仕様の名前。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のキューブの名前。
aw_dimension_name	キューブのディメンションの名前。
aw_level_name	ディメンションのレベルの名前。

例

次の文は、Product の 2 つのレベル、Channel の 1 つのレベルおよび Time の 1 つのレベルを、集計仕様 AC_AGGSPEC に追加します。

```
execute dbms_awm.add_awcubeagg_spec_level
      ('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'AW_PROD', 'L3')
execute dbms_awm.add_awcubeagg_spec_level
      ('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'AW_PROD', 'L2')
execute dbms_awm.add_awcubeagg_spec_level
      ('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'AW_CHAN', 'STANDARD_2')
execute dbms_awm.add_awcubeagg_spec_level
      ('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'AW_TIME', 'L2')
```

参照

- 1-18 ページの「アナリティック・ワークスペースにおけるデータの集計」を参照してください。
- 22-24 ページの「CREATE_AWCUBEAGG_SPEC プロシージャ」を参照してください。

ADD_AWCUBEAGG_SPEC_MEASURE プロシージャ

このプロシージャは、集計仕様にメジャーを追加します。

構文

```
ADD_AWCUBEAGG_SPEC_MEASURE (
      aggregation_spec      IN      VARCHAR2,
      aw_owner              IN      VARCHAR2,
      aw_name               IN      VARCHAR2,
      aw_cube_name         IN      VARCHAR2,
      aw_measure_name      IN      VARCHAR2);
```

パラメータ

表 22-6 ADD_AWCUBEAGG_SPEC_MEASURE プロシージャのパラメータ

パラメータ	説明
aggregation_spec	アナリティック・ワークスペース内のキューブの集計仕様の名前。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のキューブの名前。
aw_measure_name	キューブのいずれかのメジャーの名前。

例

次の文は、Costs メジャーおよび Quota メジャーを、アナリティック・ワークスペース MYAW 内のキューブ AW_ANACUBE の集計仕様に追加します。

```
execute dbms_awm.add_awcubeagg_spec_measure
('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'XXF.COSTS')
execute dbms_awm.add_awcubeagg_spec_measure
('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'XXF.QUOTA')
```

参照

- 1-18 ページの「[アナリティック・ワークスペースにおけるデータの集計](#)」を参照してください。
- 22-24 ページの「[CREATE_AWCUBEAGG_SPEC プロシージャ](#)」を参照してください。

ADD_AWCUBELOAD_SPEC_COMP プロシージャ

このプロシージャは、キューブ・ロード仕様にコンポジット仕様を追加します。

構文

```
ADD_AWCUBELOAD_SPEC_COMP (
    cube_load_spec      IN  VARCHAR2,
    cube_owner          IN  VARCHAR2,
    cube_name           IN  VARCHAR2,
    composite_spec      IN  VARCHAR2);
```

パラメータ

表 22-7 ADD_AWCUBELOAD_SPEC_COMP プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブ・ロード仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
composite_spec	キューブ・ロード仕様に追加するコンポジット仕様の名前。

例

次の文は、コンポジット仕様 AC_COMPSPEC をキューブ・ロード仕様 AC_CUBELOADSPEC に追加します。

```
execute DBMS_AWM.add_AWCubeLoad_Spec_Comp
('AC_CUBELOADSPEC', 'XADEMO', 'ANALYTIC_CUBE', 'AC_COMPSPEC');
```

参照

- 1-4 ページの「ワークスペース・キューブの作成および移入」を参照してください。
- 22-26 ページの「CREATE_AWCUBELOAD_SPEC プロシージャ」を参照してください。
- 22-17 ページの「CREATE_AWCOMP_SPEC プロシージャ」を参照してください。

ADD_AWCUBELOAD_SPEC_FILTER プロシージャ

このプロシージャは、キューブ・ロード仕様にフィルタ条件を追加します。フィルタは SQL の WHERE 句です。ソース・ファクト表に対する問合せで使用します。

構文

```
ADD_AWCUBELOAD_SPEC_FILTER (
    cube_load_spec      IN  VARCHAR2,
    cube_owner          IN  VARCHAR2,
    cube_name           IN  VARCHAR2,
    fact_table_owner    IN  VARCHAR2,
    fact_table_name     IN  VARCHAR2,
    where_clause        IN  VARCHAR2);
```

パラメータ

表 22-8 ADD_AWCUBELOAD_SPEC_FILTER プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブ・ロード仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
fact_table_owner	OLAP カタログ・ソース・キューブにマップされているファクト表の所有者。
fact_table_name	OLAP カタログ・ソース・キューブにマップされているファクト表の名前。
where_clause	ファクト表からロードする行を指定する SQL の WHERE 句。

例

次の文は、AC_CUBELOADSPEC2 というキューブ・ロード仕様を作成します。アナリティック・ワークスペース内のターゲット・キューブがこの仕様でリフレッシュされると、25 未満の SALES のみがロードされます。

```
execute dbms_awm.create_awcubeload_spec
    ('AC_CUBELOADSPEC2', 'XADEMO', 'ANALYTIC_CUBE', 'LOAD_DATA');
execute dbms_awm.add_awcubeload_spec_measure
    ('AC_CUBELOADSPEC2', 'XADEMO', 'ANALYTIC_CUBE', 'F.SALES',
    'AW_SALES', 'Sales');
execute dbms_awm.add_awcubeload_spec_filter
    ('AC_CUBELOADSPEC2', 'XADEMO', 'ANALYTIC_CUBE',
    'XADEMO', 'XADEMO_ANALYTIC_FACTS', ''SALES' < 25');
```

参照

- 1-4 ページの「[ワークスペース・キューブの作成および移入](#)」を参照してください。
- 22-26 ページの「[CREATE_AWCUBELOAD_SPEC プロシージャ](#)」を参照してください。

ADD_AWCUBELOAD_SPEC_MEASURE プロシージャ

このプロシージャは、キューブ・ロード仕様にメジャーを追加します。

1 つ以上のメジャーをキューブ・ロード仕様に追加した場合、そのメジャーのみがロードされます。メジャーをキューブ・ロード仕様に追加しない場合は、キューブのすべてのメジャーがロードされます。

このプロシージャでは、アナリティック・ワークスペースでのメジャー名、表示名および説明を指定できます。ターゲット名を指定しない場合、またはこのプロシージャをまったくコールしない場合は、OLAP カタログのソース名が使用されます。

構文

```
ADD_AWCUBELOAD_SPEC_MEASURE (
    cube_load_spec          IN  VARCHAR2,
    cube_owner              IN  VARCHAR2,
    cube_name               IN  VARCHAR2,
    measure_name            IN  VARCHAR2,
    aw_measure_name         IN  VARCHAR2 DEFAULT NULL,
    aw_measure_display_name IN  VARCHAR2 DEFAULT NULL,
    aw_measure_description  IN  VARCHAR2 DEFAULT NULL);
```

パラメータ

表 22-9 ADD_AWCUBELOAD_SPEC_MEASURE プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブ・ロード仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
measure_name	OLAP カタログ・ソース・メジャーの名前。
aw_measure_name	アナリティック・ワークスペース内のターゲット・メジャーの名前。名前を指定しない場合は、OLAP カタログのメジャーの名前が使用されます。
aw_measure_display_name	アナリティック・ワークスペース内のターゲット・メジャーの表示名。表示名を指定しない場合は、OLAP カタログ内でのメジャーの表示名が使用されます。
aw_measure_description	アナリティック・ワークスペース内のターゲット・メジャーの説明。説明を指定しない場合は、OLAP カタログ内でのメジャーの説明が使用されます。

例

次の文は、AC_CUBELOADSPEC2 というキューブ・ロード仕様を作成します。アナリティック・ワークスペース内のターゲット・キューブがこの仕様でリフレッシュされると、Sales メジャーのみがロードされます。

ターゲットの Sales メジャーには、AW_SALES という論理名と、'Sales' という説明が付けられます。

```
execute dbms_awm.create_awcubeload_spec
      ('AC_CUBELOADSPEC2', 'XADAMO', 'ANALYTIC_CUBE', 'LOAD_DATA');
execute dbms_awm.add_awcubeload_spec_measure
      ('AC_CUBELOADSPEC2', 'XADAMO', 'ANALYTIC_CUBE', 'F.SALES',
      'AW_SALES', 'Sales');
```

参照

- 22-26 ページの「[CREATE_AWCUBELOAD_SPEC プロシージャ](#)」を参照してください。
- 22-45 ページの「[REFRESH_AWCUBE プロシージャ](#)」を参照してください。

ADD_AWDIMLOAD_SPEC_FILTER プロシージャ

このプロシージャは、ディメンション・ロード仕様にフィルタ条件を追加します。フィルタは SQL の WHERE 句です。ソース・ディメンション表に対する問合せで使用します。

構文

```
ADD_AWDIMLOAD_SPEC_FILTER (
    dimension_load_spec    IN    VARCHAR2,
    dimension_owner        IN    VARCHAR2,
    dimension_name         IN    VARCHAR2,
    dimension_table_owner  IN    VARCHAR2,
    dimension_table_name   IN    VARCHAR2,
    where_clause           IN    VARCHAR2);
```

パラメータ

表 22-10 ADD_AWDIMLOAD_SPEC_FILTER プロシージャのパラメータ

パラメータ	説明
dimension_load_spec	ディメンション・ロード仕様の名前。
dimension_owner	OLAP カタログ・ソース・ディメンションの所有者。
dimension_name	OLAP カタログ・ソース・ディメンションの名前。
dimension_table_owner	OLAP カタログ・ソース・ディメンションにマップされているディメンション表の所有者。
dimension_table_name	OLAP カタログ・ソース・ディメンションにマップされているディメンション表の名前。
where_clause	ディメンション表からアナリティック・ワークスペースにロードする行を指定する SQL の WHERE 句。

例

次の文は、XADEMO 内の CHANNEL ディメンションのロード仕様を作成します。ターゲット・ディメンションがこの仕様でリフレッシュされると、メンバー DIRECT のみがロードされません。

```
execute dbms_awm.create_awdimload_spec
    ('CHAN_DIMLOADSPEC', 'XADEMO', 'CHANNEL', 'FULL_LOAD');
execute dbms_awm.add_awdimload_spec_filter
    ('CHAN_DIMLOADSPEC', 'XADEMO', 'CHANNEL', 'XADEMO',
    'XADEMO_CHANNEL', ''CHAN_STD_CHANNEL' = 'DIRECT'');
```

参照

- 1-4 ページの「ワークスペース・ディメンションの作成および移入」を参照してください。
- 22-33 ページの「CREATE_AWDIMLOAD_SPEC プロシージャ」を参照してください。

AGGREGATE_AWCUBE プロシージャ

このプロシージャは、集計仕様を使用して、アナリティック・ワークスペース内のキューブの集計データを事前計算および格納します。

REFRESH_AWCUBE プロシージャは、詳細データをロードし、動的な集計をサポートする内部ワークスペース構造を設定します。キューブのサマリー・データを事前計算および格納する場合は、AGGREGATE_AWCUBE プロシージャを使用する必要があります。

リフレッシュを行ったら、ストアド・サマリーとそのデータの整合性を保証するため、AGGREGATE_AWCUBE を再実行する必要があります。

AGGREGATE_AWCUBE は、OLAP DML の UPDATE コマンドを実行して、変更内容をアナリティック・ワークスペースに保存します。AGGREGATE_AWCUBE は、SQL の COMMIT は実行しません。

構文

```
AGGREGATE_AWCUBE (
    aw_owner          IN   VARCHAR2,
    aw_name           IN   VARCHAR2,
    aw_cube_name      IN   VARCHAR2,
    aggregation_spec  IN   VARCHAR2);
```

パラメータ

表 22-11 AGGREGATE_AWCUBE プロシージャのパラメータ

パラメータ	説明
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のキューブの名前。
aggregation_spec	キューブの集計仕様の名前。

例

次の文は、アナリティック・ワークスペース MYSCHEMA.MYAW 内のターゲット・キューブ AC2 の集計計画 AGG1 を作成します。このターゲット・キューブは、ソース・キューブ XADEMO.ANALYTIC_CUBE から作成されたものです。

```

----- Create agg plan for analytic cube -----
----- with levels 2 and 3 of product, standard_2 of channel, and 2 of time -----
----- with measures costs and quota -----

execute dbms_awm.create_awcubeagg_spec
      ('AGG1', 'MYSCHEMA', 'MYAW', 'AC2')
execute dbms_awm.add_awcubeagg_spec_level
      ('AGG1', 'MYSCHEMA', 'MYAW', 'AC2', 'PRODUCT', 'L3')
execute dbms_awm.add_awcubeagg_spec_level
      ('AGG1', 'MYSCHEMA', 'MYAW', 'AC2', 'PRODUCT', 'L2')
execute dbms_awm.add_awcubeagg_spec_level
      ('AGG1', 'MYSCHEMA', 'MYAW', 'AC2', 'CHANNEL', 'STANDARD_2')
execute dbms_awm.add_awcubeagg_spec_level
      ('AGG1', 'MYSCHEMA', 'MYAW', 'AC2', 'TIME', 'L2')
execute dbms_awm.add_awcubeagg_spec_measure
      ('AGG1', 'MYSCHEMA', 'MYAW', 'AC2', 'XXF.COSTS')
execute dbms_awm.add_awcubeagg_spec_measure
      ('AGG1', 'MYSCHEMA', 'MYAW', 'AC2', 'XXF.QUOTA')
execute dbms_awm.aggregate_awcube('MYSCHEMA', 'MYAW', 'AC2', 'AGG1')

```

参照

- 1-18 ページの「[アナリティック・ワークスペースにおけるデータの集計](#)」を参照してください。
- 22-24 ページの「[CREATE_AWCUBEAGG_SPEC プロシージャ](#)」を参照してください。

CREATE_AWCOMP_SPEC プロシージャ

このプロシージャは、OLAP カタログ・ソース・キューブの**コンポジット仕様**を作成します。コンポジット仕様は、スパースなデータを、アナリティック・ワークスペース内のターゲット・キューブにどのように格納するかを決定します。また、データのロードおよび問合せの効率に影響を与えるディメンション順序も決定します。

コンポジットは、1つ以上のスパースなメジャーへの索引を提供する、ディメンション値の組合せのリストです。コンポジットは、アナリティック・ワークスペース内の名前付きオブジェクトです。コンポジットは OLAP DML コマンドで定義およびメンテナンスします。

コンポジット仕様の**メンバー**は、コンポジット（そのメンバーはディメンション）と個々のディメンションです。

構文

```
CREATE_AWCOMP_SPEC (
    composite_spec      IN  VARCHAR2,
    cube_owner         IN  VARCHAR2,
    cube_name          IN  VARCHAR2);
```

パラメータ

表 22-12 CREATE_AWCOMP_SPEC プロシージャのパラメータ

パラメータ	説明
composite_spec	キューブのコンポジット仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。

注意

既存のコンポジット仕様を変更する場合は、次のプロシージャを使用できます。

- [SET_AWCOMP_SPEC_CUBE](#) プロシージャ
- [SET_AWCOMP_SPEC_MEMBER_NAME](#) プロシージャ
- [SET_AWCOMP_SPEC_MEMBER_POS](#) プロシージャ
- [SET_AWCOMP_SPEC_MEMBER_SEG](#) プロシージャ
- [SET_AWCOMP_SPEC_NAME](#) プロシージャ

例

次の文は、XADEMO 内の ANALYTIC_CUBE のコンポジット仕様を作成します。このコンポジット仕様は、Time ディメンションとそれに続く COMP1 というコンポジットで構成されます。

```
execute DBMS_AWM.Create_AWComp_spec
    ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE');
execute DBMS_AWM.Add_AWComp_Spec_Member
    ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'TIMECOMP_MEMBER' ,
    'DIMENSION' , 'XADEMO' , 'TIME');
execute DBMS_AWM.Add_AWComp_Spec_Member
    ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'COMP1' , 'COMPOSITE');
```

参照

- 1-16 ページの「[スパースなデータの管理およびワークスペース・キューブの最適化](#)」を参照してください。
- 22-8 ページの「[ADD_AWCAMP_SPEC_MEMBER プロシージャ](#)」を参照してください。
- 22-6 ページの「[ADD_AWCAMP_SPEC_COMP_MEMBER プロシージャ](#)」を参照してください。
- 22-11 ページの「[ADD_AWCUBELOAD_SPEC_COMP プロシージャ](#)」を参照してください。

CREATE_AWCUBE プロシージャ

このプロシージャは、リレーショナル・キューブを保持するための多次元フレームワークをアナリティック・ワークスペースに作成します。

スター・スキーマおよび OLAP カタログ・メタデータからなるリレーショナル・キューブは、アナリティック・ワークスペース内のターゲット多次元キューブのソースです。データおよびメタデータは、REFRESH_AWCUBE プロシージャによって、ソース・キューブからターゲット・キューブにロードされます。

CREATE_AWCUBE は、OLAP DML の UPDATE コマンドを実行して、変更内容をアナリティック・ワークスペースに保存します。CREATE_AWCUBE は、SQL の COMMIT は実行しません。

キューブの多次元フレームワークはデータベース・スタンダード・フォームです。このため、OLAP API イネーブラや、その他の OLAP 管理ツールおよびユーティリティとの互換性が保証されます。

注意： CREATE_AWCUBE を実行して新しいワークスペース・キューブを作成する前に、キューブのディメンションごとに CREATE_AWDIMENSION を実行する必要があります。

構文

```
CREATE_AWCUBE (
    cube_owner      IN   VARCHAR2,
    cube_name       IN   VARCHAR2,
    aw_owner        IN   VARCHAR2,
    aw_name         IN   VARCHAR2,
    aw_cube_name    IN   VARCHAR2  DEFAULT NULL);
```

パラメータ

表 22-13 CREATE_AWCUBE プロシージャのパラメータ

パラメータ	説明
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のターゲット・キューブの名前。 アナリティック・ワークスペース内のキューブの名前を指定する場合、その名前は SQL の汎用オブジェクト・ネーミング規則に準拠している必要があります。また、そのアナリティック・ワークスペースを所有するスキーマ内で一意でなければなりません。一意性をテストするには、次のような文を使用します。 <pre>select owner, cube_name from all_olap2_cubes union all select aw_owner, aw_logical_name from all_olap2_aw_cubes;</pre> 通常、アナリティック・ワークスペース内では、簡単なターゲット・キューブ名でキューブを参照できます。ただし、データベース・スタンダード・フォームは、論理オブジェクトの完全名もサポートしています。キューブの完全名は次のようになります。 <pre>aw_owner.aw_cube_name.CUBE</pre>

例

次の文は、アナリティック・ワークスペース MYSCHEMA.MYAW 内に XADEMO.ANALYTIC_CUBE のための構造を作成します。ワークスペース内のキューブの名前は AW_ANACUBE です。

```
--- Create the dimensions in the analytic workspace ---

execute dbms_awm.create_awdimension
      ('XADEMO','CHANNEL','MYSCHEMA', 'MYAW', 'AW_CHAN');
execute dbms_awm.create_awdimension
      ('XADEMO','GEOGRAPHY','MYSCHEMA', 'MYAW', 'AW_GEOG');
execute dbms_awm.create_awdimension
      ('XADEMO','PRODUCT','MYSCHEMA', 'MYAW', 'AW_PROD');
execute dbms_awm.create_awdimension
      ('XADEMO','TIME','MYSCHEMA', 'MYAW', 'AW_TIME');
```

```

--- Create the cube in the analytic workspace ----

execute dbms_awm.create_awcube
      ('XADEMO', 'ANALYTIC_CUBE', 'MYSHEMA', 'MYAW', 'AW_ANACUBE');

```

次のような文を使用すると、キューブがアナリティック・ワークスペースに作成されているかどうかを確認できます。

```

--- View the cube in the analytic workspace ----

execute dbms_aw.execute
      ('aw attach MYSHEMA.MYAW');
execute dbms_aw.execute
      ('limit name to obj(property'AW$ROLE') eq 'CUBEDEF');
execute dbms_aw.execute
      ('report w 40 name');

```

NAME

```

-----
AW_ANACUBE

```

または、アクティブ・カタログに問合せを発行して、キューブが作成されているかどうかを確認することもできます。

```

select * from all_olap2_aw_cubes
      where owner in 'myschema' and
             aw_name in 'myaw' and
             aw_logical_name in 'aw_anacube';

```

参照

- 1-13 ページの「[ワークスペース・キューブの作成およびリフレッシュ](#)」を参照してください。
- 22-28 ページの「[CREATE_AWDIMENSION プロシージャ](#)」を参照してください。
- 22-45 ページの「[REFRESH_AWCUBE プロシージャ](#)」を参照してください。
- 22-21 ページの「[CREATE_AWCUBE_ACCESS プロシージャ](#)」を参照してください。
- [第3章「アクティブ・カタログ・ビュー」](#)を参照してください。

CREATE_AWCUBE_ACCESS プロシージャ

このプロシージャは、アナリティック・ワークスペース内のキューブのリレーショナル・ファクト・ビューを作成するスクリプトを生成します。このビューは、OLAP API で必要な埋込み合計形式です。

このスクリプトでは、オプションとして、ワークスペース・キューブのビューにマップされる OLAP カタログ・メタデータを生成することもできます。このメタデータは OLAP API に必要です。

ワークスペース・キューブへのリレーショナル・アクセスには、ディメンション・ビューとファクト・ビューの両方が必要です。CREATE_AWDIMENSION_ACCESS プロシージャを使用すると、ディメンション・ビューを作成するスクリプトを生成できます。

キューブの有効化プロセスを 1 つの手順で完了するには、CREATE_AWCUBE_ACCESS_FULL プロシージャを使用します。このプロシージャは、有効化スクリプトの作成と実行の両方を行います。

構文

```
CREATE_AWCUBE_ACCESS (
    aw_owner          IN   VARCHAR2,
    aw_name           IN   VARCHAR2,
    aw_cube_name      IN   VARCHAR2,
    access_type       IN   VARCHAR2,
    script_directory  IN   VARCHAR2,
    script_name       IN   VARCHAR2,
    open_mode         IN   VARCHAR2);
```

パラメータ

表 22-14 CREATE_AWCUBE_ACCESS プロシージャのパラメータ

パラメータ	説明
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のキューブの名前。
access_type	スクリプトでビューの OLAP カタログ・メタデータを生成するかどうかを指定します。次のいずれかの値を指定します。 <ul style="list-style-type: none"> ■ 'SQL' はメタデータを生成しません。 ■ 'OLAP' はメタデータを生成します。
script_directory	スクリプトが格納されるディレクトリ。ディレクトリ・オブジェクト、または UTL_FILE_DIR パラメータで設定されているパスを指定できます。
script_name	スクリプト・ファイルの名前。

表 22-14 CREATE_AWCUBE_ACCESS プロシージャのパラメータ (続き)

パラメータ	説明
open_mode	<p>スクリプト・ファイルの開き方を指定する次のいずれかのモードを指定します。</p> <ul style="list-style-type: none"> ■ 'W' は、スクリプト・ファイルの既存の内容をすべて上書きします。 ■ 'A' は、スクリプト・ファイルの既存の内容に新しいスクリプトを追記します。

例

次の文は、aw_anacube_enable.sql という有効化スクリプトを /dat1/scripts ディレクトリに作成します。このスクリプトを実行すると、ワークスペース XADEMO.MYAW 内の AW_ANACUBE キューブのファクト・ビューを作成できます。このスクリプトは、ビューにマップされる AW_ANACUBE という OLAP カタログ・キューブも生成します。

```
execute dbms_awm.create_awcube_access
('XADEMO', 'MYAW', 'AW_ANACUBE', 'OLAP',
'/dat1/scripts/', 'aw_anacube_enable.sql', 'w');
```

参照

- 1-22 ページの「[ワークスペース・キューブへのリレーショナル・アクセスの作成](#)」を参照してください。
- 22-23 ページの「[CREATE_AWCUBE_ACCESS_FULL プロシージャ](#)」を参照してください。
- 22-36 ページの「[DELETE_AWCUBE_ACCESS プロシージャ](#)」を参照してください。
- 22-55 ページの「[SET_AWCUBE_VIEW_NAME プロシージャ](#)」を参照してください。
- 22-30 ページの「[CREATE_AWDIMENSION_ACCESS プロシージャ](#)」を参照してください。
- 22-45 ページの「[REFRESH_AWCUBE プロシージャ](#)」を参照してください。
- [第 25 章「OLAP_TABLE」](#)を参照してください。

CREATE_AWCUBE_ACCESS_FULL プロシージャ

このプロシージャは、OLAP API でワークスペース・キューブにアクセスできるようにするためのプロセス全体を実行します。CREATE_AWCUBE_ACCESS と同じように、このプロシージャも有効化スクリプトを生成します。ただし、このプロシージャはスクリプトをファイルに書き込みません。かわりに、一時メモリーにスクリプトを書き込み、これを実行します。

作成されるビューおよびメタデータは、CREATE_AWCUBE_ACCESS によって生成される有効化スクリプトで作成されるものと同じです。

構文

```
CREATE_AWCUBE_ACCESS_FULL (
    run_id          IN    NUMBER,
    aw_owner        IN    VARCHAR2,
    aw_name         IN    VARCHAR2,
    aw_cube_name    IN    VARCHAR2,
    access_type     IN    VARCHAR2);
```

パラメータ

表 22-15 CREATE_AWCUBE_ACCESS_FULL プロシージャのパラメータ

パラメータ	説明
run_id	一時メモリー内の位置を識別するためのランダムな数値。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のキューブの名前。
access_type	有効化ビューに加えて、OLAP カタログ・メタデータも生成するかどうかを指定します。次のいずれかの値を指定します。 <ul style="list-style-type: none">■ 'SQL' はメタデータを生成しません。■ 'OLAP' はメタデータを生成します。

参照

- 1-22 ページの「ワークスペース・キューブへのリレーショナル・アクセスの作成」を参照してください。
- 22-21 ページの「CREATE_AWCUBE_ACCESS プロシージャ」を参照してください。
- 22-45 ページの「REFRESH_AWCUBE プロシージャ」を参照してください。
- 第 25 章「OLAP_TABLE」を参照してください。

CREATE_AWCUBEAGG_SPEC プロシージャ

このプロシージャは、OLAP カタログ・キューブの**集計仕様**を作成します。集計仕様は、ターゲット・キューブとともにアナリティック・ワークスペースに格納するサマリー・データを決定します。

集計仕様は、キューブのどのレベルを事前集計するかを決定します。キューブのすべてのメジャーをこれらのレベルに集計することも、個々のメジャーを選択することもできます。メジャーは、すべて同じレベルに集計されます。

事前集計されていないレベルは、問合せが発行されたときに動的に集計されます。事前集計するデータを判断するには、記憶域とメモリーの制約、およびクライアントによる典型的な問合せを評価する必要があります。集計仕様を作成しない場合、サマリーは格納されず、すべての集計がオンデマンドで実行されます。

集計仕様は、OLAP DML の集計サブシステムを使用します。これには、AGGREGATE コマンド、集計マップおよび関連機能が含まれます。

構文

```
CREATE_AWCUBEAGG_SPEC (
    aggregation_spec      IN   VARCHAR2,
    aw_owner              IN   VARCHAR2,
    aw_name               IN   VARCHAR2,
    aw_cube_name          IN   VARCHAR2);
```

パラメータ

表 22-16 CREATE_AWCUBEAGG_SPEC プロシージャのパラメータ

パラメータ	説明
aggregation_spec	アナリティック・ワークスペース内のキューブの集計仕様の名前。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のキューブの名前。

注意

既存の集計仕様を変更する場合は、**SET_AWCUBEAGG_SPEC_AGGOP** プロシージャを使用できます。

例

次の文は、アナリティック・ワークスペース MYSCHEMA.MYAW 内のターゲット・キューブ AW_ANACUBE の集計仕様を作成します。この集計仕様は、PRODUCT の第 3 レベル、CHANNEL の STANDARD_2 レベル、および TIME の第 2 レベルの格納済の合計を Costs メジャーおよび Sales メジャーに含めるよう指定します。

```
execute dbms_awm.create_awcubeagg_spec
```

```

('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE');
execute dbms_awn.add_awnbeagg_spec_level
('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'AW_PROD', 'L3');
execute dbms_awn.add_awnbeagg_spec_level
('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'AW_CHAN',
'STANDARD_2');
execute dbms_awn.add_awnbeagg_spec_level
('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'AW_TIME', 'L2');
execute dbms_awn.add_awnbeagg_spec_measure
('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'XXF.COSTS');
execute dbms_awn.add_awnbeagg_spec_measure
('AC_AGGSPEC', 'MYSHEMA', 'MYAW', 'AW_ANACUBE', 'XXF.SALES');

```

参照

- 1-18 ページの「アナリティック・ワークスペースにおけるデータの集計」を参照してください。
- 22-9 ページの「ADD_AWCUBEAGG_SPEC_LEVEL プロシージャ」を参照してください。
- 22-10 ページの「ADD_AWCUBEAGG_SPEC_MEASURE プロシージャ」を参照してください。
- 22-16 ページの「AGGREGATE_AWCUBE プロシージャ」を参照してください。

CREATE_AWCUBELOAD_SPEC プロシージャ

このプロシージャは、OLAP カタログ・キューブのロード仕様を作成します。ロード仕様は、キューブのデータを、リレーショナル・ファクト表からアナリティック・ワークスペースに REFRESH_AWCUBE プロシージャでどのようにロードするかを決定します。

キューブ・ロード仕様では、ロード・タイプを定義します。ロード・タイプは、アナリティック・ワークスペースにデータをロードするかロード手順のみをロードするかを指定するものです。ロード手順は OLAP DML プログラムです。ロード手順のみをロードする場合は、これらのプログラムを実行することで、後からデータをロードできます。

CREATE_AWCOMP_SPEC で作成した別の仕様を、キューブ・ロード仕様に関連付けることもできます。この仕様は、ディメンション順序を指定し、スパースなデータのアナリティック・ワークスペース内での格納方法を決定します。

構文

```

CREATE_AWCUBELOAD_SPEC (
    cube_load_spec    IN    VARCHAR2,
    cube_owner        IN    VARCHAR2,
    cube_name         IN    VARCHAR2,
    load_type         IN    VARCHAR2);

```

パラメータ

表 22-17 CREATE_AWCUBELOAD_SPEC プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブ・ロード仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
load_type	'LOAD_DATA' -- ファクト表のデータをアナリティック・ワークスペースのターゲット・キューブにロードします。 'LOAD_PROGRAM' -- アナリティック・ワークスペースにロード・プログラムを作成しますが、実行はしません。このプログラムを手動で実行すると、データをロードできます。キューブ・ロード・プログラムの名前は、アナリティック・ワークスペース内のスタンダード・フォームのキューブの AW\$LOADPRGS プロパティに格納されます。 ->show obj (property 'aw\$loadprgs' 'my_awcube_name')

注意

既存のキューブ・ロード仕様を変更する場合は、次のプロシージャを使用できます。

- [SET_AWCUBELOAD_SPEC_CUBE プロシージャ](#)
- [SET_AWCUBELOAD_SPEC_LOADTYPE プロシージャ](#)
- [SET_AWCUBELOAD_SPEC_NAME プロシージャ](#)
- [SET_AWCUBELOAD_SPEC_PARAMETER プロシージャ](#)

例

次の文は、ソース・キューブ XADEMO.ANALYTIC_CUBE のキューブ・ロード仕様を作成します。このロード仕様は、MYSHEMA.MYAW 内のターゲット・キューブ AW_ANACUBE をリフレッシュするために使用されます。

```
execute dbms_awm.create_awcubeload_spec
    ('AC_CUBELOADSPEC', 'XADEMO', 'ANALYTIC_CUBE', 'LOAD_DATA');
execute dbms_awm.refresh_awcube
    ('MYSHEMA', 'MYAW', 'AW_ANACUBE', 'AC_CUBELOADSPEC');
```

参照

- 1-4 ページの「[ワークスペース・キューブの作成および移入](#)」を参照してください。

- 22-11 ページの「[ADD_AWCUBELOAD_SPEC_COMP プロシージャ](#)」を参照してください。
- 22-45 ページの「[REFRESH_AWCUBE プロシージャ](#)」を参照してください。

CREATE_AWDIMENSION プロシージャ

このプロシージャは、リレーショナル・ディメンションを保持するための多次元フレームワークをアナリティック・ワークスペースに作成します。

ディメンション参照表および OLAP カタログ・メタデータからなるリレーショナル・ディメンションは、アナリティック・ワークスペース内のターゲット・ディメンションのソースです。データおよびメタデータは、REFRESH_AWDIMENSION プロシージャによって、ソース・ディメンションからターゲット・ディメンションにロードされます。

CREATE_AWDIMENSION は、OLAP DML の UPDATE コマンドを実行して、変更内容をアナリティック・ワークスペースに保存します。CREATE_AWDIMENSION は、SQL の COMMIT は実行しません。

ディメンションの多次元フレームワークはデータベース・スタンダード・フォームです。このため、OLAP API イネーブラや、その他の OLAP 管理ツールおよびユーティリティとの互換性が保証されます。

注意： CREATE_AWCUBE を実行して新しいワークスペース・キューブを作成する前に、キューブのディメンションごとに CREATE_AWDIMENSION を実行する必要があります。

ワークスペースは、CREATE_AWDIMENSION の最初のコールの時点で存在している必要があります。

構文

```
CREATE_AWDIMENSION (
    dimension_owner      IN   VARCHAR2,
    dimension_name       IN   VARCHAR2,
    aw_owner             IN   VARCHAR2,
    aw_name              IN   VARCHAR2,
    aw_dimension_name    IN   VARCHAR2  DEFAULT NULL),
```

パラメータ

表 22-18 CREATE_AWDIMENSION プロシージャのパラメータ

パラメータ	説明
dimension_owner	OLAP カタログ・ソース・ディメンションの所有者。

表 22-18 CREATE_AWDIMENSION プロシージャのパラメータ (続き)

パラメータ	説明
dimension_name	OLAP カタログ・ソース・ディメンションの名前。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_dimension_name	<p>アナリティック・ワークスペース内のターゲット・ディメンションの名前。</p> <p>アナリティック・ワークスペース内のディメンションの名前を指定する場合、その名前は SQL の汎用オブジェクト・ネーミング規則に準拠している必要があります。また、そのアナリティック・ワークスペースを所有するスキーマ内で一意でなければなりません。一意性をテストするには、次のような文を使用します。</p> <pre>select owner, dimension_name from all_olap2_dimensions union all select aw_owner, aw_logical_name from all_olap2_aw_dimensions;</pre> <p>通常、アナリティック・ワークスペース内では、簡単なターゲット・ディメンション名でディメンションを参照できます。ただし、データベース・スタンダード・フォームは、論理オブジェクトの完全名もサポートしています。ディメンションの完全名は次のようになります。</p> <pre>aw_owner.aw_dimension_name.DIMENSION</pre>

例

次の文は、XADEMO スキーマのワークスペース MYAW 内に、CHANNEL、GEOGRAPHY、PRODUCT、TIME および DIVISION のアナリティック・ワークスペース・ディメンションを作成します。

```
execute dbms_awm.create_awdimension
      ('XADEMO','CHANNEL','MYSHEMA', 'MYAW', 'AW_CHAN');
execute dbms_awm.create_awdimension
      ('XADEMO','GEOGRAPHY','MYSHEMA','MYAW', 'AW_GEOG');
execute dbms_awm.create_awdimension
      ('XADEMO','PRODUCT','MYSHEMA', 'MYAW', 'AW_PROD');
execute dbms_awm.create_awdimension
      ('XADEMO','TIME','MYSHEMA', 'MYAW', 'AW_TIME');
execute dbms_awm.create_awdimension
      ('XADEMO','DIVISION','MYSHEMA', 'MYAW', 'AW_DIV');
```

次のような文を使用すると、ディメンションがアナリティック・ワークスペースに作成されているかどうかを確認できます。

```
execute dbms_aw.execute
      ('aw attach MYSHEMA.MYAW');
execute dbms_aw.execute
      ('limit name to obj(property'AW$ROLE') eq ''DIMDEF'');
execute dbms_aw.execute
      ('report w 40 name');
```

NAME

```
-----
AW_CHAN
AW_GEOG
AW_PROD
AW_TIME
AW_DIV
```

または、アクティブ・カタログに問合せを発行して、ディメンションが作成されているかどうかを確認することもできます。

```
select * from all_olap2_aw_dimensions
      where aw_owner in 'myschema' and aw_name in 'myaw';
```

参照

- 1-11 ページの「ワークスペース・ディメンションの作成およびリフレッシュ」を参照してください。
- 22-47 ページの「REFRESH_AWDIMENSION プロシージャ」を参照してください。
- 22-30 ページの「CREATE_AWDIMENSION_ACCESS プロシージャ」を参照してください。
- 22-19 ページの「CREATE_AWCUBE プロシージャ」を参照してください。
- 第3章「アクティブ・カタログ・ビュー」を参照してください。

CREATE_AWDIMENSION_ACCESS プロシージャ

このプロシージャは、アナリティック・ワークスペース内のディメンションのリレーショナル・ビューを作成するスクリプトを生成します。このビューは、OLAP API で必要な埋込み合計形式です。

このスクリプトでは、オプションとして、ワークスペース・ディメンションのビューにマップされる OLAP カタログ・メタデータを生成することもできます。このメタデータは OLAP API に必要です。

ワークスペース・キューブへのリレーショナル・アクセスには、ファクト・ビューとディメンション・ビューの両方が必要です。CREATE_AWCUBE_ACCESS プロシージャを使用すると、ファクト・ビューを作成するスクリプトを生成できます。

有効化プロセスを1つの手順で完了するには、CREATE_AWDIMENSION_ACCESS_FULL プロシージャを使用します。このプロシージャは、有効化スクリプトの作成と実行の両方を行います。

構文

```
CREATE_AWDIMENSION_ACCESS (
    aw_owner          IN   VARCHAR2,
    aw_name           IN   VARCHAR2,
    aw_dimension_name IN   VARCHAR2,
    access_type       IN   VARCHAR2,
    script_directory  IN   VARCHAR2,
    script_name       IN   VARCHAR2,
    open_mode         IN   VARCHAR2);
```

パラメータ

表 22-19 CREATE_AWDIMENSION_ACCESS プロシージャのパラメータ

パラメータ	説明
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_dimension_name	アナリティック・ワークスペース内のディメンションの名前。
access_type	スクリプトでビューの OLAP カタログ・メタデータを生成するかどうかを指定します。次のいずれかの値を指定します。 <ul style="list-style-type: none"> ■ 'SQL' はメタデータを生成しません。 ■ 'OLAP' はメタデータを生成します。
script_directory	スクリプトが格納されるディレクトリ。ディレクトリ・オブジェクト、または UTL_FILE_DIR パラメータで設定されているパスを指定できます。
script_name	スクリプト・ファイルの名前。
open_mode	スクリプト・ファイルの開き方を指定する次のいずれかのモードを指定します。 <ul style="list-style-type: none"> ■ 'w' は、スクリプト・ファイルの既存の内容をすべて上書きします。 ■ 'a' は、スクリプト・ファイルの既存の内容に新しいスクリプトを追記します。

例

次の文は、`aw_prod_enable` という有効化スクリプトを `/dat1/scripts` ディレクトリに作成します。このスクリプトを実行すると、ワークスペース `XADEMO.MYAW` 内の `AW_PROD` ディメンションのビューを作成できます。このスクリプトは、ビューにマップされる `AW_PROD` という OLAP カタログ・ディメンションも生成します。

```
execute dbms_awm.create_awdimension_access
('XADEMO', 'MYAW', 'AW_PROD', 'OLAP',
'/dat1/scripts/', 'aw_prod_enable', 'w');
```

参照

- 1-5 ページの「ワークスペース・キューブへのリレーショナル・アクセスの有効化」を参照してください。
- 22-42 ページの「`DELETE_AWDIMENSION_ACCESS` プロシージャ」を参照してください。
- 22-60 ページの「`SET_AWDIMENSION_VIEW_NAME` プロシージャ」を参照してください。
- 第 25 章「`OLAP_TABLE`」を参照してください。

CREATE_AWDIMENSION_ACCESS_FULL プロシージャ

このプロシージャは、OLAP API でワークスペース・ディメンションにアクセスできるようにするためのプロセス全体を実行します。`CREATE_AWDIMENSION_ACCESS` と同じように、このプロシージャも有効化スクリプトを生成します。ただし、このプロシージャはスクリプトをファイルに書き込みません。かわりに、一時メモリーにスクリプトを書き込み、これを実行します。

作成されるビューおよびメタデータは、`CREATE_AWDIMENSION_ACCESS` によって生成される有効化スクリプトで作成されるものと同じです。

構文

```
CREATE_AWDIMENSION_ACCESS_FULL (
    run_id                IN    NUMBER,
    aw_owner              IN    VARCHAR2,
    aw_name               IN    VARCHAR2,
    aw_dimension_name    IN    VARCHAR2,
    access_type          IN    VARCHAR2);
```

パラメータ

表 22-20 CCREATE_AWDIMENSION_ACCESS_FULL プロシージャのパラメータ

パラメータ	説明
run_id	一時メモリー内の位置を識別するためのランダムな数値。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_dimension_name	アナリティック・ワークスペース内のディメンションの名前。
access_type	有効化ビューに加えて、OLAP カタログ・メタデータも生成するかどうかを指定します。次のいずれかの値を指定します。 <ul style="list-style-type: none"> ■ 'SQL' はメタデータを生成しません。 ■ 'OLAP' はメタデータを生成します。

参照

- 1-22 ページの「[ワークスペース・キューブへのリレーショナル・アクセスの作成](#)」を参照してください。
- 22-30 ページの「[CREATE_AWDIMENSION_ACCESS プロシージャ](#)」を参照してください。
- 22-47 ページの「[REFRESH_AWDIMENSION プロシージャ](#)」を参照してください。
- [第 25 章「OLAP_TABLE」](#) を参照してください。

CREATE_AWDIMLOAD_SPEC プロシージャ

このプロシージャは、OLAP カタログ・ディメンションの**ロード仕様**を作成します。ロード仕様は、ディメンションを、リレーショナル・ディメンション表からアナリティック・ワークスペースに REFRESH_AWDIMENSION プロシージャでどのようにロードするかを決定します。

ロード仕様なしでディメンションをリフレッシュした場合、新しいディメンション・メンバーのみがロードされます。

構文

```
CREATE_AWDIMLOAD_SPEC (
    dimension_load_spec    IN    VARCHAR2,
    dimension_owner        IN    VARCHAR2,
    dimension_name         IN    VARCHAR2,
    load_type              IN    VARCHAR2);
```

パラメータ

表 22-21 CREATE_AWDIMLOAD_SPEC プロシージャのパラメータ

パラメータ	説明
dimension_load_spec	ロード仕様の名前。 SET_AWDIMLOAD_SPEC_NAME プロシージャを使用すると、名前を変更できます。
dimension_owner	OLAP カタログ・ソース・ディメンションの所有者。
dimension_name	OLAP カタログ・ソース・ディメンションの名前。
load_type	次のいずれかを指定します。 'FULL_LOAD_ADDITIONS_ONLY' -- ディメンションをリフレッシュする際、新しいディメンション・メンバーのみがロードされます。(デフォルト) 'FULL_LOAD' -- ワークスペース内のすべてのディメンション・メンバーが削除され、ソース・ディメンションのすべてのメンバーがロードされます。

注意

既存のディメンション・ロード仕様を変更する場合は、次のプロシージャを使用できます。

- [SET_AWDIMLOAD_SPEC_DIMENSION](#) プロシージャ
- [SET_AWDIMLOAD_SPEC_LOADTYPE](#) プロシージャ
- [SET_AWDIMLOAD_SPEC_NAME](#) プロシージャ
- [SET_AWDIMLOAD_SPEC_PARAMETER](#) プロシージャ

例

次の文は、ソース・ディメンション XADEMO.CHANNEL のロード仕様を作成し、これを使用して、アナリティック・ワークスペース MYSCHEMA.MYAW 内のターゲット・ディメンション AW_CHAN にディメンション・メンバーをロードします。このロード仕様には、ディメンション・メンバー 'DIRECT' のみをロードするためのフィルタ条件 (WHERE 句) が含まれています。

```
execute dbms_awm.create_awdimload_spec
  ('CHAN_DIMLOADSPEC', 'XADEMO', 'CHANNEL', 'FULL_LOAD');
execute dbms_awm.add_awdimload_spec_filter
  ('CHAN_DIMLOADSPEC', 'XADEMO', 'CHANNEL', 'XADEMO',
  'XADEMO_CHANNEL', ''''CHAN_STD_CHANNEL'' = 'DIRECT'' );
execute dbms_awm.refresh_awdimension
  ('MYSCHEMA', 'MYAW', 'AW_CHAN', 'CHAN_DIMLOADSPEC');
```

参照

- 1-4 ページの「ワークスペース・ディメンションの作成および移入」を参照してください。
- 22-47 ページの「REFRESH_AWDIMENSION プロシージャ」を参照してください。

DELETE_AWCOMP_SPEC プロシージャ

このプロシージャは、コンポジット仕様を削除します。

構文

```
DELETE_AWCOMP_SPEC (
    composite_spec    IN    VARCHAR2,
    cube_owner       IN    VARCHAR2,
    cube_name        IN    VARCHAR2);
```

パラメータ

表 22-22 DELETE_AWCOMP_SPEC プロシージャのパラメータ

パラメータ	説明
composite_spec	キューブのコンポジット仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。

参照

22-17 ページの「CREATE_AWCOMP_SPEC プロシージャ」を参照してください。

DELETE_AWCOMP_SPEC_MEMBER プロシージャ

このプロシージャは、コンポジット仕様のメンバーを削除します。メンバーは、ディメンションまたはコンポジットのいずれかです。

構文

```
DELETE_AWCOMP_SPEC_MEMBER (
    composite_spec    IN    VARCHAR2,
    cube_owner       IN    VARCHAR2,
    cube_name        IN    VARCHAR2,
    member_name      IN    VARCHAR2,);
```

パラメータ

表 22-23 DELETE_AWCOMP_SPEC_MEMBER プロシージャのパラメータ

パラメータ	説明
composite_spec	キューブのコンポジット仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
member_name	削除するディメンションまたはコンポジットの名前。

参照

22-8 ページの「[ADD_AWCOMP_SPEC_MEMBER プロシージャ](#)」を参照してください。

DELETE_AWCUBE_ACCESS プロシージャ

このプロシージャは、ワークスペース・キューブに関連付けられているビューおよび OLAP カタログ・メタデータを削除するためのスクリプトを生成します。このスクリプトは、アナリティック・ワークスペースに格納されている有効化メタデータは削除しません。

ワークスペース・キューブまたはワークスペース自体を削除した場合は、このプロシージャを実行して、関連付けられている有効化ビューおよびメタデータをクリーン・アップする必要があります。

新しい世代の有効化ビューおよびメタデータを作成する場合、このプロシージャを実行する必要はありません。有効化プロセスは、新しいビューおよびメタデータを作成する前に古い世代を削除します。

構文

```
DELETE_AWCUBE_ACCESS (
    aw_owner          IN  VARCHAR2,
    aw_name           IN  VARCHAR2,
    aw_cube_name      IN  VARCHAR2,
    access_type       IN  VARCHAR2,
    script_directory  IN  VARCHAR2,
    script_name       IN  VARCHAR2,
    open_mode         IN  VARCHAR2);
```

パラメータ

表 22-24 DELETE_AWCUBE_ACCESS プロシージャのパラメータ

パラメータ	説明
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のキューブの名前。
access_type	ビューに OLAP カタログ・メタデータがあるかどうかを指定します。 <ul style="list-style-type: none"> ■ 'SQL' -- メタデータが存在しない。 ■ 'OLAP' -- OLAP カタログ・メタデータが存在する。
script_directory	スクリプトが格納されるディレクトリ。ディレクトリ・オブジェクト、または UTL_FILE_DIR パラメータで設定されているパスを指定できます。
script_name	スクリプト・ファイルの名前。
open_mode	スクリプト・ファイルの開き方を指定する次のいずれかのモードを指定します。 <ul style="list-style-type: none"> ■ 'W' は、スクリプト・ファイルの既存の内容をすべて上書きします。 ■ 'A' は、スクリプト・ファイルの既存の内容に新しいスクリプトを追記します。

参照

- 1-22 ページの「[ワークスペース・キューブへのリレーショナル・アクセスの作成](#)」を参照してください。
- 22-21 ページの「[CREATE_AWCUBE_ACCESS プロシージャ](#)」を参照してください。
- 22-23 ページの「[CREATE_AWCUBE_ACCESS_FULL プロシージャ](#)」を参照してください。
- 22-55 ページの「[SET_AWCUBE_VIEW_NAME プロシージャ](#)」を参照してください。

DELETE_AWCUBEAGG_SPEC プロシージャ

このプロシージャは、集計仕様を削除します。

構文

```
DELETE_AWCUBEAGG_SPEC (
    aggregation_spec    IN    VARCHAR2,
```

```

aw_owner          IN   VARCHAR2,
aw_name           IN   VARCHAR2,
aw_cube_name      IN   VARCHAR2);

```

パラメータ

表 22-25 DELETE_AWCUBEAGG_SPEC プロシージャのパラメータ

パラメータ	説明
aggregation_spec	アナリティック・ワークスペース内のキューブの集計仕様の名前。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のキューブの名前。

参照

22-24 ページの「[CREATE_AWCUBEAGG_SPEC プロシージャ](#)」を参照してください。

DELETE_AWCUBEAGG_SPEC_LEVEL プロシージャ

このプロシージャは、集計仕様からレベルを削除します。

構文

```

DELETE_AWCUBEAGG_SPEC_LEVEL (
    aggregation_spec  IN   VARCHAR2,
    aw_owner          IN   VARCHAR2,
    aw_name           IN   VARCHAR2,
    aw_cube_name      IN   VARCHAR2,
    aw_dimension_name IN   VARCHAR2,
    aw_level_name     IN   VARCHAR2);

```

パラメータ

表 22-26 DELETE_AWCUBEAGG_SPEC_LEVEL プロシージャのパラメータ

パラメータ	説明
aggregation_spec	アナリティック・ワークスペース内のキューブの集計仕様の名前。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。

表 22-26 DELETE_AWCUBEAGG_SPEC_LEVEL プロシージャのパラメータ (続き)

パラメータ	説明
aw_cube_name	アナリティック・ワークスペース内のキューブの名前。
aw_dimension_name	キューブのディメンションの名前。
aw_level_name	ディメンションのレベルの名前。

参照

22-9 ページの「[ADD_AWCUBEAGG_SPEC_LEVEL プロシージャ](#)」を参照してください。

DELETE_AWCUBEAGG_SPEC_MEASURE プロシージャ

このプロシージャは、集計仕様からメジャーを削除します。

構文

```
DELETE_AWCUBEAGG_SPEC_MEASURE (
    aggregation_spec    IN    VARCHAR2,
    aw_owner            IN    VARCHAR2,
    aw_name             IN    VARCHAR2,
    aw_cube_name       IN    VARCHAR2,
    aw_measure_name    IN    VARCHAR2);
```

パラメータ

表 22-27 DELETE_AWCUBEAGG_SPEC_MEASURE プロシージャのパラメータ

パラメータ	説明
aggregation_spec	アナリティック・ワークスペース内のキューブの集計仕様の名前。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のターゲット・キューブの名前。
aw_measure_name	削除するメジャーの名前。

参照

22-10 ページの「[ADD_AWCUBEAGG_SPEC_MEASURE プロシージャ](#)」を参照してください。

DELETE_AWCUBELOAD_SPEC プロシージャ

このプロシージャは、キューブ・ロード仕様を削除します。

構文

```
DELETE_AWCUBELOAD_SPEC (
    cube_load_spec    IN    VARCHAR2,
    cube_owner        IN    VARCHAR2,
    cube_name         IN    VARCHAR2);
```

パラメータ

表 22-28 DELETE_AWCUBELOAD_SPEC プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブ・ロード仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。

参照

22-26 ページの「[CREATE_AWCUBELOAD_SPEC プロシージャ](#)」を参照してください。

DELETE_AWCUBELOAD_SPEC_COMP プロシージャ

このプロシージャは、キューブ・ロード仕様からコンポジット仕様を削除します。

構文

```
DELETE_AWCUBELOAD_SPEC_COMP (
    cube_load_spec    IN    VARCHAR2,
    cube_owner        IN    VARCHAR2,
    cube_name         IN    VARCHAR2,
    composite_spec    IN    VARCHAR2);
```

パラメータ

表 22-29 DELETE_AWCUBELOAD_SPEC_COMP プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブ・ロード仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。

表 22-29 DELETE_AWCUBELOAD_SPEC_COMP プロシージャのパラメータ (続き)

パラメータ	説明
cube_name	OLAP カタログ・ソース・キューブの名前。
composite_spec	削除するコンポジット仕様の名前。

参照

22-11 ページの「[ADD_AWCUBELOAD_SPEC_COMP プロシージャ](#)」を参照してください。

DELETE_AWCUBELOAD_SPEC_FILTER プロシージャ

このプロシージャは、キューブ・ロード仕様からフィルタ条件 (WHERE 句) を削除します。

構文

```
DELETE_AWCUBELOAD_SPEC_FILTER (
    cube_load_spec      IN  VARCHAR2,
    cube_owner          IN  VARCHAR2,
    cube_name           IN  VARCHAR2,
    fact_table_owner    IN  VARCHAR2,
    fact_table_name     IN  VARCHAR2);
```

パラメータ

表 22-30 DELETE_AWCUBELOAD_SPEC_FILTER プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブ・ロード仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
fact_table_owner	この OLAP カタログ・ソース・キューブにマップされているファクト表の所有者。
fact_table_name	この OLAP カタログ・ソース・キューブにマップされているファクト表の名前。

参照

22-12 ページの「[ADD_AWCUBELOAD_SPEC_FILTER プロシージャ](#)」を参照してください。

DELETE_AWCUBELOAD_SPEC_MEASURE プロシージャ

このプロシージャは、キューブ・ロード仕様からメジャーを削除します。

構文

```
DELETE_AWCUBELOAD_SPEC_MEASURE (  
    cube_load_spec    IN    VARCHAR2,  
    cube_owner        IN    VARCHAR2,  
    cube_name         IN    VARCHAR2,  
    measure_name      IN    VARCHAR2);
```

パラメータ

表 22-31 DELETE_AWCUBELOAD_SPEC_MEASURE プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブ・ロード仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
measure_name	削除するメジャーの名前。

参照

22-13 ページの「[ADD_AWCUBELOAD_SPEC_MEASURE プロシージャ](#)」を参照してください。

DELETE_AWDIMENSION_ACCESS プロシージャ

このプロシージャは、ワークスペース・ディメンションに関連付けられているビューおよび OLAP カタログ・メタデータを削除するためのスクリプトを生成します。このスクリプトは、アナリティック・ワークスペースに格納されている有効化メタデータは削除しません。

ワークスペース・ディメンションまたはワークスペース自体を削除した場合は、このプロシージャを実行して、関連付けられている有効化ビューおよびメタデータをクリーン・アップする必要があります。

新しい世代の有効化ビューおよびメタデータを作成する場合、このプロシージャを実行する必要はありません。有効化プロセスは、新しいビューおよびメタデータを作成する前に古い世代を削除します。

構文

```
DELETE_AWDIMENSION_ACCESS (  
    aw_owner          IN    VARCHAR2,
```

```

aw_name          IN  VARCHAR2,
aw_dimension_name IN  VARCHAR2,
access_type      IN  VARCHAR2,
script_directory IN  VARCHAR2,
script_name      IN  VARCHAR2,
open_mode        IN  VARCHAR2);

```

パラメータ

表 22-32 DELETE_AWDIMENSION_ACCESS プロシージャのパラメータ

パラメータ	説明
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_dimension_name	アナリティック・ワークスペース内のディメンションの名前。
access_type	ビューに OLAP カタログ・メタデータがあるかどうかを指定します。 <ul style="list-style-type: none"> ■ 'SQL' -- メタデータが存在しない。 ■ 'OLAP' -- OLAP カタログ・メタデータが存在する。
script_directory	スクリプトが格納されるディレクトリ。ディレクトリ・オブジェクト、または UTL_FILE_DIR パラメータで設定されているパスを指定できます。
script_name	スクリプト・ファイルの名前。
open_mode	スクリプト・ファイルの開き方を指定する次のいずれかのモードを指定します。 <ul style="list-style-type: none"> ■ 'W' は、スクリプト・ファイルの既存の内容をすべて上書きします。 ■ 'A' は、スクリプト・ファイルの既存の内容に新しいスクリプトを追記します。

参照

- 22-30 ページの「[CREATE_AWDIMENSION_ACCESS プロシージャ](#)」を参照してください。
- 22-23 ページの「[CREATE_AWCUBE_ACCESS_FULL プロシージャ](#)」を参照してください。
- 22-60 ページの「[SET_AWDIMENSION_VIEW_NAME プロシージャ](#)」を参照してください。

- 1-11 ページの「[ワークスペース・ディメンションの作成およびリフレッシュ](#)」を参照してください。

DELETE_AWDIMLOAD_SPEC プロシージャ

このプロシージャは、ディメンション・ロード仕様を削除します。

構文

```
DELETE_AWDIMLOAD_SPEC (  
    dimension_load_spec      IN  VARCHAR2,  
    dimension_owner          IN  VARCHAR2,  
    dimension_name           IN  VARCHAR2);
```

パラメータ

表 22-33 DELETE_AWDIMLOAD_SPEC プロシージャのパラメータ

パラメータ	説明
dimension_load_spec	ディメンション・ロード仕様の名前。
dimension_owner	OLAP カタログ・ソース・ディメンションの所有者。
dimension_name	OLAP カタログ・ソース・ディメンションの名前。

参照

22-33 ページの「[CREATE_AWDIMLOAD_SPEC プロシージャ](#)」を参照してください。

DELETE_AWDIMLOAD_SPEC_FILTER プロシージャ

このプロシージャは、ディメンション・ロード仕様からフィルタ条件（WHERE 句）を削除します。

構文

```
DELETE_AWDIMLOAD_SPEC_FILTER (  
    dimension_load_spec      IN  VARCHAR2,  
    dimension_owner          IN  VARCHAR2,  
    dimension_name           IN  VARCHAR2);  
    dimension_table_owner    IN  VARCHAR2,  
    dimension_table_name     IN  VARCHAR2);
```

パラメータ

表 22-34 DELETE_AWDIMLOAD_SPEC_FILTER プロシージャのパラメータ

パラメータ	説明
dimension_load_spec	ディメンション・ロード仕様の名前。
dimension_owner	OLAP カタログ・ソース・ディメンションの所有者。
dimension_name	OLAP カタログ・ソース・ディメンションの名前。
dimension_table_owner	OLAP カタログ・ソース・ディメンションにマップされているディメンション表の所有者。
dimension_table_name	OLAP カタログ・ソース・ディメンションにマップされているディメンション表の名前。

参照

22-15 ページの「[ADD_AWDIMLOAD_SPEC_FILTER プロシージャ](#)」を参照してください。

REFRESH_AWCUBE プロシージャ

このプロシージャは、OLAP カタログ・ソース・キューブのデータおよびメタデータを、アナリティック・ワークスペース内のターゲット・キューブにロードします。

REFRESH_AWCUBE は、OLAP DML の UPDATE コマンドを実行して、変更内容をアナリティック・ワークスペースに保存します。REFRESH_AWCUBE は、SQL の COMMIT は実行しません。

キューブのデータのリフレッシュ方法を決定するキューブ・ロード仕様を指定することもできます。キューブ・ロード仕様は、データをロードするか、後で実行するためにロード・プログラムのみをロードするかを決定します。キューブ・ロード仕様には、ディメンション順序およびスパースなデータの処理方法を決定するコンポジット仕様を含めることができます。

ロード仕様を指定しない場合は、すべてのデータがロードされます。ロード仕様にコンポジット仕様を含めない場合、ディメンションは最初に変化するものとして **Time** で順序付けられ、その後任何其他すべてのディメンションのコンポジットが続きます。コンポジット内のディメンションは、サイズ（ディメンション・メンバーの数）の降順で順序付けされます。

キューブのロード仕様が個々のメジャーを識別できない場合（ADD_AWCUBELOAD_SPEC_MEASURE）、キューブのすべてのメジャーがワークスペースにロードされます。キューブのロード仕様にフィルタ条件（ファクト表に対する WHERE 句）が含まれていない場合、メジャーのすべてのデータがワークスペースにロードされます。

REFRESH_AWCUBE への最初のコールの前に、キューブのディメンションごとに REFRESH_AWDIMENSION をコールする必要があります。すでにデータが格納されている

キューブをリフレッシュする前には、最後のリフレッシュ以降に変更が加えられた、そのキューブのディメンションすべてをリフレッシュする必要があります。

構文

```
REFRESH_AWCUBE (
    aw_owner          IN   VARCHAR2,
    aw_name           IN   VARCHAR2,
    aw_cube_name      IN   VARCHAR2,
    cube_load_spec    IN   VARCHAR2 DEFAULT NULL);
```

パラメータ

表 22-35 REFRESH_AWCUBE プロシージャのパラメータ

パラメータ	説明
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のターゲット・キューブの名前。
cube_load_spec	キューブ・ロード仕様の名前。ロード仕様を指定しない場合は、すべてのファクト・データがロードされます (デフォルト)。

注意

ワークスペース・キューブをリフレッシュすると、論理キューブを定義するすべての OLAP カタログ・メタデータ (その次元性、メジャー、説明など) がリフレッシュされます。キューブのデータは、ロード仕様に従ってリフレッシュされます。詳細は、1-15 ページの「[キューブのメタデータのリフレッシュ](#)」を参照してください。

リフレッシュと有効化プロセスの関係については、1-25 ページの「[アナリティック・ワークスペースの有効化メタデータ](#)」を参照してください。

リフレッシュと集計プロセスの関係については、1-18 ページの「[アナリティック・ワークスペースにおけるデータの集計](#)」を参照してください。

例

次の文は、ソース・キューブ XADEMO.ANALYTIC_CUBE からターゲット・キューブ AW_ANACUBE を作成します。ターゲット・キューブのすべてのディメンションをリフレッシュしてからロード仕様を作成し、ターゲット・キューブのデータをリフレッシュします。

```
-- create cube, cube load spec, and refresh
execute dbms_awm.create_awcube
    ('XADEMO', 'ANALYTIC_CUBE', 'MYSHEMA', 'MYAW', 'AW_ANACUBE');
```

```
execute dbms_awm.create_awcubeload_spec
    ('AC_CUBELOADSPEC', 'XADEMO', 'ANALYTIC_CUBE', 'LOAD_DATA')
execute dbms_awm.refresh_awdimension
    ('MYSHEMA', 'MYAW', 'AW_CHAN');
execute dbms_awm.refresh_awdimension
    ('MYSHEMA', 'MYAW', 'AW_PROD');
execute dbms_awm.refresh_awdimension
    ('MYSHEMA', 'MYAW', 'AW_GEOG');
execute dbms_awm.refresh_awdimension
    ('MYSHEMA', 'MYAW', 'AW_TIME');
execute dbms_awm.refresh_awcube
    ('MYSHEMA', 'MYAW', 'AW_ANACUBE', 'AC_CUBELOADSPEC')
```

参照

- 1-13 ページの「ワークスペース・キューブの作成およびリフレッシュ」を参照してください。
- 22-19 ページの「CREATE_AWCUBE プロシージャ」を参照してください。
- 22-45 ページの「REFRESH_AWCUBE プロシージャ」を参照してください。
- 22-17 ページの「CREATE_AWCOMP_SPEC プロシージャ」を参照してください。
- 22-21 ページの「CREATE_AWCUBE_ACCESS プロシージャ」を参照してください。

REFRESH_AWDIMENSION プロシージャ

このプロシージャは、OLAP カタログ・ソース・ディメンションのデータおよびメタデータを、アナリティック・ワークスペース内のターゲット・ディメンションにロードします。

REFRESH_AWDIMENSION は、OLAP DML の UPDATE コマンドを実行して、変更内容をアナリティック・ワークスペースに保存します。REFRESH_AWDIMENSION は、SQL の COMMIT は実行しません。

ワークスペース内のディメンションのメンバーのリフレッシュ方法を決定するディメンション・ロード仕様を指定することもできます。ロード仕様を指定しない場合は、すべてのディメンション・メンバーがロード対象として選択されますが、実際には新しいメンバーのみがターゲット・ディメンションに追加されます。

フィルタ条件（ディメンション表に対する WHERE 句）を指定すると、ソース表からロードするディメンション・メンバーを個々に選択できます。

REFRESH_AWCUBE への最初のコールの前に、キューブのディメンションごとに REFRESH_AWDIMENSION をコールする必要があります。それ以降のキューブのリフレッシュでは、新しい時間間隔が Time ディメンションに追加されたなど、ソース・ディメンションに対して変更が加えられた場合、コールする必要があるのは REFRESH_AWDIMENSION のみです。

構文

```
REFRESH_AWDIMENSION (
    aw_owner          IN   VARCHAR2,
    aw_name           IN   VARCHAR2,
    aw_dimension_name IN   VARCHAR2,
    dimension_load_spec IN  VARCHAR2 DEFAULT NULL);
```

パラメータ

表 22-36 REFRESH_AWDIMENSION プロシージャのパラメータ

パラメータ	説明
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_dimension_name	アナリティック・ワークスペース内のターゲット・ディメンションの名前。
dimension_load_spec	ディメンション・ロード仕様の名前。ロード仕様を指定しない場合は、新しいメンバーがターゲット・ディメンションに追加されます (デフォルト)。

注意

ワークスペース・ディメンションをリフレッシュすると、論理ディメンションを定義するすべての OLAP カタログ・メタデータ (そのレベル、階層、属性、説明など) がリフレッシュされます。ディメンションのデータは、ロード仕様に従ってリフレッシュされます。詳細は、1-12 ページの「[ディメンションのメタデータのリフレッシュ](#)」を参照してください。

リフレッシュと有効化プロセスの関係については、1-25 ページの「[アナリティック・ワークスペースの有効化メタデータ](#)」を参照してください。

例

次の文は、アナリティック・ワークスペース MYSCHEMA.MYAW 内の、XADEMO.ANALYTIC_CUBE ソース・キューブのディメンションをリフレッシュします。

```
-- Create dimension load specs and refresh dimensions

-- CHANNEL dimension
execute dbms_awm.create_awdimload_spec
    ('CHAN_DIMLOADSPEC', 'XADEMO', 'CHANNEL', 'FULL_LOAD');
execute dbms_awm.add_awdimload_spec_filter
    ('CHAN_DIMLOADSPEC', 'XADEMO', 'CHANNEL', 'XADEMO',
    'XADEMO_CHANNEL', ''CHAN_STD_CHANNEL'' = 'DIRECT');
```

```

('MYSHEMA', 'MYAW', 'AW_CHAN', 'CHAN_DIMLOADSPEC');

-- PRODUCT dimension
execute dbms_awm.create_awdimload_spec
('PROD_DIMLOADSPEC', 'XADEMO', 'PRODUCT', 'FULL_LOAD');
execute dbms_awm.Set_AWDimLoad_Spec_Parameter
('PROD_DIMLOADSPEC', 'XADEMO', 'PRODUCT', 'UNIQUE_RDBMS_KEY', 'YES');
execute dbms_awm.refresh_awdimension
('MYSHEMA', 'MYAW', 'AW_PROD', 'PROD_DIMLOADSPEC');

-- GEOGRAPHY dimension
execute dbms_awm.create_awdimload_spec
('GEOG_DIMLOADSPEC', 'XADEMO', 'GEOGRAPHY', 'FULL_LOAD');
execute dbms_awm.refresh_awdimension
('MYSHEMA', 'MYAW', 'AW_GEOG', 'GEOG_DIMLOADSPEC');

-- TIME dimension
execute dbms_awm.create_awdimload_spec
('TIME_DIMLOADSPEC', 'XADEMO', 'TIME', 'FULL_LOAD');
execute dbms_awm.refresh_awdimension
('MYSHEMA', 'MYAW', 'AW_TIME', 'TIME_DIMLOADSPEC');

```

参照

- 1-11 ページの「ワークスペース・ディメンションの作成およびリフレッシュ」を参照してください。
- 22-28 ページの「CREATE_AWDIMENSION プロシージャ」を参照してください。
- 22-33 ページの「CREATE_AWDIMLOAD_SPEC プロシージャ」を参照してください。
- 22-30 ページの「CREATE_AWDIMENSION_ACCESS プロシージャ」を参照してください。

SET_AWCOMP_SPEC_CUBE プロシージャ

このプロシージャは、コンポジット仕様を別のキューブに関連付けます。

構文

```

SET_AWCOMP_SPEC_CUBE (
    composite_spec      IN  VARCHAR2,
    old_cube_owner     IN  VARCHAR2,
    old_cube_name      IN  VARCHAR2,
    new_cube_owner     IN  VARCHAR2,
    new_cube_name      IN  VARCHAR2);

```

パラメータ

表 22-37 SET_AWCOMP_SPEC_CUBE プロシージャのパラメータ

パラメータ	説明
composite_spec	コンポジット仕様の名前。
old_cube_owner	古い OLAP カタログ・ソース・キューブの所有者。
old_cube_name	古い OLAP カタログ・ソース・キューブの名前。
new_cube_owner	新しい OLAP カタログ・ソース・キューブの所有者。
new_cube_name	新しい OLAP カタログ・ソース・キューブの名前。

参照

- 1-16 ページの「[スパースなデータの管理およびワークスペース・キューブの最適化](#)」を参照してください。
- 22-17 ページの「[CREATE_AWCOMP_SPEC プロシージャ](#)」を参照してください。

SET_AWCOMP_SPEC_MEMBER_NAME プロシージャ

このプロシージャは、コンポジット仕様のメンバーの名前を変更します。メンバーは、ディメンションまたはコンポジットのいずれかです。

構文

```
SET_AWCOMP_SPEC_MEMBER_NAME (
    composite_spec      IN   VARCHAR2,
    cube_owner         IN   VARCHAR2,
    cube_name          IN   VARCHAR2,
    old_member_name    IN   VARCHAR2,
    new_member_name    IN   VARCHAR2);
```

パラメータ

表 22-38 SET_AWCOMP_SPEC_MEMBER_NAME プロシージャのパラメータ

パラメータ	説明
composite_spec	キューブのコンポジット仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。

表 22-38 SET_AWCOMP_SPEC_MEMBER_NAME プロシージャのパラメータ (続き)

パラメータ	説明
old_member_name	古いメンバー名。ディメンションまたはコンポジットのいずれかです。
new_member_name	新しいメンバー名。

参照

- 1-16 ページの「[スパースなデータの管理およびワークスペース・キューブの最適化](#)」を参照してください。
- 22-17 ページの「[CREATE_AWCOMP_SPEC プロシージャ](#)」を参照してください。

SET_AWCOMP_SPEC_MEMBER_POS プロシージャ

このプロシージャは、コンポジット仕様のメンバーの位置を設定します。メンバーは、ディメンションまたはコンポジットのいずれかです。

構文

```
SET_AWCOMP_SPEC_MEMBER_POS (
    composite_spec      IN  VARCHAR2,
    cube_owner         IN  VARCHAR2,
    cube_name          IN  VARCHAR2,
    member_name        IN  VARCHAR2,
    member_position    IN  NUMBER);
```

パラメータ

表 22-39 SET_AWCOMP_SPEC_MEMBER_POS プロシージャのパラメータ

パラメータ	説明
composite_spec	キューブのコンポジット仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
member_name	コンポジット仕様のメンバー。ディメンションまたはコンポジットのいずれかです。
member_position	コンポジット仕様内でのメンバーの位置。

例

次の文は、XADEMO 内の ANALYTIC_CUBE のコンポジット仕様を作成します。このコンポジット仕様には、TIMECOMP_MEMBER という Time ディメンションと COMP1 というコンポジットの 2 つのメンバーが含まれます。

```
---- The logical members of the specification are:
----      <TIME COMP1<PRODUCT, GEOGRAPHY>.
-----
execute DBMS_AWM.Create_AWComp_spec
      ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE');
execute DBMS_AWM.Add_AWComp_Spec_Member
      ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'TIMECOMP_MEMBER' ,
      'DIMENSION' , 'XADEMO' , 'TIME');
execute DBMS_AWM.Add_AWComp_Spec_Member
      ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'COMP1' , 'COMPOSITE');
execute DBMS_AWM.Add_AWComp_Spec_Comp_Member
      ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'COMP1' , 'PROD_COMP' ,
      'DIMENSION' , 'XADEMO' , 'PRODUCT');
execute DBMS_AWM.Add_AWComp_Spec_Comp_Member
      ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'COMP1' , 'GEOG_COMP' ,
      'DIMENSION' , 'XADEMO' , 'GEOGRAPHY');

---- With the following statement, the logical members of the specification
---- are reordered as follows.
----      <COMP1<PRODUCT, GEOGRAPHY> TIME>.
-----

execute DBMS_AWM.Set_AWComp_Spec_Member_Pos
      ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'COMP1' , 1);
```

参照

- 1-16 ページの「[スパースなデータの管理およびワークスペース・キューブの最適化](#)」を参照してください。
- 22-17 ページの「[CREATE_AWCOMP_SPEC プロシージャ](#)」を参照してください。

SET_AWCOMP_SPEC_MEMBER_SEG プロシージャ

このプロシージャは、コンポジット仕様のメンバーのセグメント・サイズを設定します。メンバーは、ディメンションまたはコンポジットのいずれかです。

セグメントとは、OLAP エンジンがデータの格納に使用する内部バッファのことです。セグメントのサイズは、データのロードおよびデータに対する問合せのパフォーマンスに影響します。

構文

```
SET_AWCOMP_SPEC_MEMBER_SEG (
    composite_spec      IN   VARCHAR2,
    cube_owner         IN   VARCHAR2,
    cube_name          IN   VARCHAR2,
    member_name        IN   VARCHAR2,
    member_segwidth    IN   NUMBER DEFAULT NULL);
```

パラメータ

表 22-40 SET_AWCOMP_SPEC_MEMBER_SEG プロシージャのパラメータ

パラメータ	説明
composite_spec	コンポジット仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
member_name	ディメンションまたはコンポジットの名前。
member_segwidth	ディメンションまたはコンポジットに関連付けるセグメント・サイズ。ディメンションのセグメント・サイズを指定しない場合、値はディメンションの最大サイズ (ディメンション・メンバーの数) になります。コンポジットのセグメント・サイズを指定しない場合、値は 10,000,000 になります。

例

次の文は、Time ディメンションのセグメント・サイズをゼロ (アナリティック・ワークスペースでのデフォルト設定) に、COMP1 コンポジットのセグメント・サイズを 10,000,000 に設定します。

```
execute DBMS_AWM.Create_AWComp_spec
    ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE');
execute DBMS_AWM.Add_AWComp_Spec_Member
    ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'TIME_DIM' ,
    'DIMENSION' , 'XADEMO' , 'time');
execute DBMS_AWM.Add_AWComp_Spec_Member
    ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'COMP1' , 'COMPOSITE');
execute DBMS_AWM.Add_AWComp_Spec_Comp_Member
    ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'COMP1' , 'COMP1_PROD' ,
    'DIMENSION' , 'XADEMO' , 'product');
execute DBMS_AWM.Add_AWComp_Spec_Comp_Member
    ('AC_COMPSPEC' , 'XADEMO' , 'ANALYTIC_CUBE' , 'COMP1' , 'COMP1_GEOG' ,
    'DIMENSION' , 'XADEMO' , 'geography');
execute DBMS_AWM.Set_AWComp_Spec_Member_Seg
```

```

('AC_COMPSPEC', 'XADEMO', 'ANALYTIC_CUBE', 'TIME_DIM', 0);
execute DBMS_AWM.Set_AWComp_Spec_Member_Seg
('AC_COMPSPEC', 'XADEMO', 'ANALYTIC_CUBE', 'COMP1', NULL);

```

参照

- 1-16 ページの「[スパースなデータの管理およびワークスペース・キューブの最適化](#)」を参照してください。
- 『Oracle9i OLAP DML Reference』のオンライン・ヘルプで「segment width」を検索してください。
- 22-17 ページの「[CREATE_AWCOMP_SPEC プロシージャ](#)」を参照してください。

SET_AWCOMP_SPEC_NAME プロシージャ

このプロシージャは、コンポジット仕様の名前を変更します。

構文

```

SET_AWCOMP_SPEC_NAME (
    old_composite_spec      IN  VARCHAR2,
    cube_owner              IN  VARCHAR2,
    cube_name               IN  VARCHAR2,
    new_composite_spec      IN  VARCHAR2);

```

パラメータ

表 22-41 SET_AWCOMP_SPEC_NAME プロシージャのパラメータ

パラメータ	説明
old_composite_spec	キューブのコンポジット仕様の古い名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
new_composite_spec	コンポジット仕様の新しい名前。

参照

- 1-16 ページの「[スパースなデータの管理およびワークスペース・キューブの最適化](#)」を参照してください。
- 22-17 ページの「[CREATE_AWCOMP_SPEC プロシージャ](#)」を参照してください。

SET_AWCUBE_VIEW_NAME プロシージャ

このプロシージャは、アナリティック・ワークスペース・キューブのリレーショナル・ビューの名前を変更します。名前はアナリティック・ワークスペースに格納され、新しい有効化スクリプトを生成および実行する際にインスタンス化されます。

このプロシージャを使用して、キューブのリフレッシュ時に設定されるデフォルトのビュー名を上書きできます。

構文

```
SET_AWCUBE_VIEW_NAME (
    aw_owner           IN  VARCHAR2,
    aw_name            IN  VARCHAR2,
    aw_cube_name       IN  VARCHAR2,
    hierarchy_combo_number IN  NUMBER,
    view_name          IN  VARCHAR2);
```

パラメータ

表 22-42 SET_AWCUBE_VIEW_NAME プロシージャのパラメータ

パラメータ	説明
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のキューブの名前。
hierarchy_combo_number	階層の組合せの番号。
view_name	この階層の組合せのファクト・ビューの名前。

注意

有効化ビュー名の詳細は、1-26 ページの「[ファクト・ビューのデフォルト名](#)」を参照してください。

参照

- 1-22 ページの「[ワークスペース・キューブへのリレーショナル・アクセスの作成](#)」を参照してください。
- 22-21 ページの「[CREATE_AWCUBE_ACCESS プロシージャ](#)」を参照してください。
- 22-36 ページの「[DELETE_AWCUBE_ACCESS プロシージャ](#)」を参照してください。

SET_AWCUBEAGG_SPEC_AGGOP プロシージャ

このプロシージャは、集計仕様のディメンションの1つに対する集計の演算子を設定します。

OLAP DML の RELATION コマンドで使用可能な任意の集計演算子を指定できます。デフォルトの演算子は加算 (SUM) です。このプロシージャを使用して、OLAP カタログ内のソース・キューブに関連付けられた集計演算子を上書きできます。

注意： DBMS_AWM パッケージは現在、加重集計演算子をサポートしていません。たとえば、OLAP カタログでキューブのディメンションの1つに対する集計に加重合計または加重平均を指定した場合、これらはアナリティック・ワークスペースでは同等のスカラー演算 (合計または平均) に変換されます。SET_AWCUBEAGG_SPEC_AGGOP で指定された加重演算子も同様に変換されます。

構文

```
SET_AWCUBEAGG_SPEC_AGGOP (
    aggregation_spec      IN  VARCHAR2,
    aw_owner              IN  VARCHAR2,
    aw_name               IN  VARCHAR2,
    aw_cube_name         IN  VARCHAR2,
    aw_measure_name      IN  VARCHAR2,
    aw_dimension_name    IN  VARCHAR2,
    aggregation_operator IN  VARCHAR2);
```

パラメータ

表 22-43 SET_AWCUBEAGG_SPEC_AGGOP プロシージャのパラメータ

パラメータ	説明
aggregation_spec	アナリティック・ワークスペース内の集計仕様の名前。
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_cube_name	アナリティック・ワークスペース内のターゲット・キューブの名前。
aw_measure_name	集計するメジャーの名前。
aw_dimension_name	キューブのディメンションの名前。
aggregation_operator	このディメンションに対する集計の集計演算子。表 1-10 「集計演算子」を参照してください。

注意

OLAP カタログおよびアナリティック・ワークスペースでサポートされている集計メソッドの詳細は、1-20 ページの「[集計メソッドの選択](#)」を参照してください。

参照

- 1-18 ページの「[アナリティック・ワークスペースにおけるデータの集計](#)」を参照してください。
- 22-24 ページの「[CREATE_AWCUBEAGG_SPEC プロシージャ](#)」を参照してください。
- 『Oracle9i OLAP DML Reference』のオンライン・ヘルプの RELATION コマンドを参照してください。

SET_AWCUBELOAD_SPEC_CUBE プロシージャ

このプロシージャは、キューブ・ロード仕様を別のキューブに関連付けます。

構文

```
SET_AWCUBELOAD_SPEC_CUBE (
    cube_load_spec      IN   VARCHAR2,
    old_cube_owner      IN   VARCHAR2,
    old_cube_name       IN   VARCHAR2,
    new_cube_owner      IN   VARCHAR2,
    new_cube_name       IN   VARCHAR2);
```

パラメータ

表 22-44 SET_AWCUBELOAD_SPEC_CUBE プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブ・ロード仕様の名前。
old_cube_owner	古い OLAP カタログ・ソース・キューブの所有者。
old_cube_name	古い OLAP カタログ・ソース・キューブの名前。
new_cube_owner	新しい OLAP カタログ・ソース・キューブの所有者。
new_cube_name	新しい OLAP カタログ・ソース・キューブの名前。

参照

22-26 ページの「[CREATE_AWCUBELOAD_SPEC プロシージャ](#)」を参照してください。

SET_AWCUBELOAD_SPEC_LOADTYPE プロシージャ

このプロシージャは、キューブ・ロード仕様のロード・タイプを再設定します。ロード・タイプは、アナリティック・ワークスペースへのデータのロード方法を指定します。

構文

```
SET_AWCUBELOAD_SPEC_LOADTYPE (
    cube_load_spec    IN  VARCHAR2,
    cube_owner        IN  VARCHAR2,
    cube_name         IN  VARCHAR2,
    load_type         IN  VARCHAR2);
```

パラメータ

表 22-45 SET_AWCUBELOAD_SPEC_LOADTYPE プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブのロード仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
load_type	'LOAD_DATA' -- ファクト表のデータをアナリティック・ワークスペースのターゲット・キューブにロードします。 'LOAD_PROGRAM' -- アナリティック・ワークスペースにロード・プログラムを作成しますが、実行はしません。このプログラムを手動で実行すると、データをロードできます。キューブ・ロード・プログラムの名前は、アナリティック・ワークスペース内のスタンダード・フォームのキューブの AW\$LOADPRGS プロパティに格納されます。ロード・プログラムの名前は、次のような OLAP DML コマンドで表示できます。 ->show obj(property 'aw\$loadprgs' 'my_awcube_name')

参照

22-26 ページの「[CREATE_AWCUBELOAD_SPEC プロシージャ](#)」を参照してください。

SET_AWCUBELOAD_SPEC_NAME プロシージャ

このプロシージャは、キューブ・ロード仕様の名前を変更します。

構文

```
SET_AWCUBELOAD_SPEC_NAME (
    old_cube_load_spec    IN  VARCHAR2,
    cube_owner            IN  VARCHAR2,
```

```
cube_name          IN  VARCHAR2,
new_cube_load_spec IN  VARCHAR2);
```

パラメータ

表 22-46 SET_AWCUBELOAD_SPEC_NAME プロシージャのパラメータ

パラメータ	説明
old_cube_load_spec	キューブ・ロード仕様の古い名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。
new_cube_load_spec	キューブ・ロード仕様の新しい名前。

参照

22-26 ページの「[CREATE_AWCUBELOAD_SPEC プロシージャ](#)」を参照してください。

SET_AWCUBELOAD_SPEC_PARAMETER プロシージャ

このプロシージャは、キューブ・ロード仕様のパラメータを設定します。

構文

```
SET_AWCUBELOAD_SPEC_PARAMETER (
    cube_load_spec  IN  VARCHAR2,
    cube_owner      IN  VARCHAR2,
    cube_name       IN  VARCHAR2,
    parameter_name  IN  VARCHAR2,
    parameter_value IN  VARCHAR2 DEFAULT NULL);
```

パラメータ

表 22-47 SET_AWCUBELOAD_SPEC_PARAMETER プロシージャのパラメータ

パラメータ	説明
cube_load_spec	キューブ・ロード仕様の名前。
cube_owner	OLAP カタログ・ソース・キューブの所有者。
cube_name	OLAP カタログ・ソース・キューブの名前。

表 22-47 SET_AWCUBELOAD_SPEC_PARAMETER プロシージャのパラメータ (続き)

パラメータ	説明
parameter_name	'DISPLAY_NAME' -- アナリティック・ワークスペース内のターゲット・キューブの表示名として、OLAP カタログ・ソース・キューブの名前を使用するか、またはターゲット・キューブの表示名を使用するかを指定します。
parameter_value	DISPLAY_NAME の値は、アナリティック・ワークスペース内のターゲット・キューブの表示名です。このパラメータを指定しない場合、アナリティック・ワークスペース内のターゲット・キューブの表示名として、OLAP カタログ内のソース・キューブの表示名が使用されます。

例

次の文は、AC_CUBELOADSPEC キューブ・ロード仕様にターゲット・キューブの表示名を指定します。

```
execute dbms_awn.set_awcubeload_spec_parameter
('AC_CUBELOADSPEC', 'XADemo', 'ANALYTIC_CUBE',
'DISPLAY_NAME', 'My AW Analytic Cube Display Name')
```

参照

22-26 ページの「[CREATE_AWCUBELOAD_SPEC プロシージャ](#)」を参照してください。

SET_AWDIMENSION_VIEW_NAME プロシージャ

このプロシージャは、アナリティック・ワークスペース・ディメンションのリレーショナル・ビューの名前を変更します。名前はアナリティック・ワークスペースに格納され、新しい有効化スクリプトを生成および実行する際にインスタンス化されます。

このプロシージャを使用して、ディメンションのリフレッシュ時に設定されるデフォルトのビュー名を上書きできます。

構文

```
SET_AWDIMENSION_VIEW_NAME (
    aw_owner          IN    VARCHAR2,
    aw_name           IN    VARCHAR2,
    aw_dimension_name IN    VARCHAR2,
    hierarchy_name    IN    VARCHAR2,
    view_name         IN    VARCHAR2);
```

パラメータ

表 22-48 SET_AWDIMENSION_VIEW_NAME プロシージャのパラメータ

パラメータ	説明
aw_owner	アナリティック・ワークスペースの所有者。
aw_name	アナリティック・ワークスペースの名前。
aw_dimension_name	アナリティック・ワークスペース内のディメンションの名前。
hierarchy_name	アナリティック・ワークスペース内の階層の名前。
view_name	ディメンション階層のビューの名前。

注意

有効化ビュー名の詳細は、1-25 ページの「[ディメンション・ビューのデフォルト名](#)」を参照してください。

参照

- 1-22 ページの「[ワークスペース・キューブへのリレーショナル・アクセスの作成](#)」を参照してください。
- 22-30 ページの「[CREATE_AWDIMENSION_ACCESS プロシージャ](#)」を参照してください。
- 22-42 ページの「[DELETE_AWDIMENSION_ACCESS プロシージャ](#)」を参照してください。

SET_AWDIMLOAD_SPEC_DIMENSION プロシージャ

このプロシージャは、ディメンション・ロード仕様を別のディメンションに関連付けます。

構文

```
SET_AWDIMLOAD_SPEC_DIMENSION (
    dimension_load_spec    IN    VARCHAR2,
    old_dimension_owner    IN    VARCHAR2,
    old_dimension_name     IN    VARCHAR2,
    new_dimension_owner    IN    VARCHAR2,
    new_dimension_name     IN    VARCHAR2);
```

パラメータ

表 22-49 SET_AWDIMLOAD_SPEC_DIMENSION プロシージャのパラメータ

パラメータ	説明
dimension_load_spec	ディメンション・ロード仕様の名前。
old_dimension_owner	古い OLAP カタログ・ソース・ディメンションの所有者。
old_dimension_name	古い OLAP カタログ・ソース・ディメンションの名前。
new_dimension_owner	新しい OLAP カタログ・ソース・ディメンションの所有者。
new_dimension_name	新しい OLAP カタログ・ソース・ディメンションの名前。

参照

22-33 ページの「[CREATE_AWDIMLOAD_SPEC プロシージャ](#)」を参照してください。

SET_AWDIMLOAD_SPEC_LOADTYPE プロシージャ

このプロシージャは、ディメンション・ロード仕様のロード・タイプを再設定します。ロード・タイプは、アナリティック・ワークスペースへのディメンション・メンバーのロード方法を指定します。

ディメンションをリフレッシュする際、デフォルトでは新しいメンバーのみがロードされません。

構文

```
SET_AWDIMLOAD_SPEC_LOADTYPE (
    dimension_load_spec    IN    VARCHAR2,
    dimension_owner        IN    VARCHAR2,
    dimension_name         IN    VARCHAR2,
    load_type              IN    VARCHAR2);
```

パラメータ

表 22-50 SET_AWDIMLOAD_SPEC_LOADTYPE プロシージャのパラメータ

パラメータ	説明
dimension_load_spec	ディメンション・ロード仕様の名前。
dimension_owner	OLAP カタログ・ソース・ディメンションの所有者。
dimension_name	OLAP カタログ・ソース・ディメンションの名前。

表 22-50 SET_AWDIMLOAD_SPEC_LOADTYPE プロシージャのパラメータ (続き)

パラメータ	説明
load_type	次のいずれかを指定します。 'FULL_LOAD_ADDITIONS_ONLY' -- デイメンションをリフレッシュする際、新しいデイメンション・メンバーのみがロードされます。(デフォルト) 'FULL_LOAD' -- デイメンションをリフレッシュする際、ワークスペース内のすべてのデイメンション・メンバーが削除され、ソース・デイメンションのすべてのメンバーがロードされます。

参照

22-33 ページの「[DELETE_AWCUBELOAD_SPEC プロシージャ](#)」を参照してください。

SET_AWDIMLOAD_SPEC_NAME プロシージャ

このプロシージャは、デイメンション・ロード仕様の名前を変更します。

構文

```
SET_AWDIMLOAD_SPEC_NAME (
    old_dimension_load_spec    IN    VARCHAR2,
    dimension_owner            IN    VARCHAR2,
    dimension_name              IN    VARCHAR2,
    new_dimension_load_spec    IN    VARCHAR2);
```

パラメータ

表 22-51 SET_AWDIMLOAD_SPEC_NAME プロシージャのパラメータ

パラメータ	説明
old_dimension_load_spec	デイメンション・ロード仕様の古い名前。
dimension_owner	OLAP カタログ・ソース・デイメンションの所有者。
dimension_name	OLAP カタログ・ソース・デイメンションの名前。
new_dimension_load_spec	デイメンション・ロード仕様の新しい名前。

参照

22-33 ページの「[CREATE_AWDIMLOAD_SPEC プロシージャ](#)」を参照してください。

SET_AWDIMLOAD_SPEC_PARAMETER プロシージャ

このプロシージャは、ディメンション・ロード仕様のパラメータを設定します。

構文

```
SET_AWDIMLOAD_SPEC_PARAMETER (
    dimension_load_spec    IN    VARCHAR2,
    dimension_owner        IN    VARCHAR2,
    dimension_name         IN    VARCHAR2,
    parameter_name         IN    VARCHAR2,
    parameter_value        IN    VARCHAR2 DEFAULT NULL);
```

パラメータ

表 22-52 SET_AWDIMLOAD_SPEC_PARAMETER プロシージャのパラメータ

パラメータ	説明
dimension_load_spec	ディメンション・ロード仕様の名前。
dimension_owner	OLAP カタログ・ソース・ディメンションの所有者。
dimension_name	OLAP カタログ・ソース・ディメンションの名前。
parameter_name	次のいずれかを選択します。 'UNIQUE_RDBMS_KEY' -- このディメンションのメンバーがソース表のすべてのレベルを通じて一意であるかどうか。 'DISPLAY_NAME' -- アナリティック・ワークスペース内のターゲット・ディメンションの表示名。 'P_DISPLAY_NAME' -- アナリティック・ワークスペース内のターゲット・ディメンションの複数表示名。

表 22-52 SET_AWDIMLOAD_SPEC_PARAMETER プロシージャのパラメータ (続き)

パラメータ	説明
parameter_value	<p>UNIQUE_RDBMS_KEY の値は、'YES' または 'NO' のいずれかです。デフォルトは 'NO' です。</p> <p>NO -- デイメンション・メンバー名が、RDBMS 表の各レベルを通じて一意でない。アナリティック・ワークスペース内の対応するデイメンション・メンバー名には、接頭辞としてレベル名が含まれます。(デフォルト)</p> <p>YES -- デイメンション・メンバー名が、RDBMS 表の各レベルを通じて一意である。アナリティック・ワークスペース内の対応するデイメンション・メンバー名は、ソース・リレーショナル・デイメンションでの名前と同じになります。</p> <p>DISPLAY_NAME の値は、アナリティック・ワークスペース内のターゲット・デイメンションの表示名です。このパラメータを指定しない場合、アナリティック・ワークスペース内のターゲット・デイメンションの表示名として、OLAP カタログ内のソース・デイメンションの表示名が使用されます。</p> <p>P_DISPLAY_NAME の値は、アナリティック・ワークスペース内のターゲット・デイメンションの複数表示名です。このパラメータを指定しない場合、アナリティック・ワークスペース内のターゲット・デイメンションの複数表示名として、OLAP カタログ内のソース・デイメンションの複数表示名が使用されます。</p>

例

次の文は、ロード仕様 PROD_LOADSPEC の Product デイメンションのパラメータを設定します。これらのパラメータでは、デイメンション・メンバー名に接頭辞としてレベル名を付けることを抑止し、ターゲット・デイメンションの表示名および複数表示名を指定します。

```
execute dbms_awm.Set_AWDimLoad_Spec_Parameter
('PROD_LOADSPEC', 'XADEMO', 'PRODUCT', 'UNIQUE_RDBMS_KEY', 'YES')
execute dbms_awm.Set_AWDimLoad_Spec_Parameter
('PROD_LOADSPEC', 'XADEMO', 'PRODUCT', 'DISPLAY_NAME',
'My AW Product Display Name')
execute dbms_awm.Set_AWDimLoad_Spec_Parameter
('PROD_LOADSPEC', 'XADEMO', 'PRODUCT', 'P_DISPLAY_NAME',
'My AW Product Plural Display Name')
```

参照

22-33 ページの「[DELETE_AWCUBEAGG_SPEC プロシージャ](#)」を参照してください。

OLAP データ管理パッケージの DBMS_ODM では、OLAP API 固有の要件を満たすマテリアライズド・ビューを作成するためのプロシージャが提供されます。

参照：

- マテリアライズド・ビューの作成および管理については、『Oracle9i データ・ウェアハウス・ガイド』を参照してください。
- Oracle OLAP のサマリー管理については、『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

この章では、次の項目について説明します。

- [マテリアライズド・ビューでのサマリー管理](#)
- [ファクト表の集計](#)
- [例: Sales キューブのマテリアライズド・ビューの作成](#)
- [DBMS_ODM サブプログラムの概要](#)

マテリアライズド・ビューでのサマリー管理

リレーショナル・ウェアハウスのサマリー管理は、データベースのクエリー・リライト機能によって管理されます。クエリー・リライトを使用すると、集計の実行時に再計算するのではなく、問合せで集計データをマテリアライズド・ビューからフェッチできます。

OLAP API がリレーショナル表に格納されているウェアハウスを問い合わせる場合、可能な場合はクエリー・リライトが使用されます。ただし、OLAP API がクエリー・リライトを使用できるのは、マテリアライズド・ビューが特定の形式になっている場合のみです。

DBMS_ODM パッケージのプロシージャは、OLAP API の要件を満たすマテリアライズド・ビューを作成します。

ソース・データがアナリティック・ワークスペースに格納されている場合、マテリアライズド・ビューは必要ありません。アナリティック・ワークスペース内のネイティブな多次元構

造は、ストアド・サマリーおよび実行時の集計の両方をサポートしています。DBMS_AWM パッケージまたは Analytic Workspace Manager を使用して、スター・スキーマのデータをアナリティック・ワークスペースに移動できます。

グルーピング・セット

DBMS_ODM パッケージは、OLAP カタログに定義されているキューブに基づいてマテリアライズド・ビューのセットを作成します。キューブは、最低レベルのデータのみを格納する単一ファクト表を持つスター・スキーマにマップされている必要があります。

DBMS_ODM プロシージャによって生成されるスクリプトは、次のマテリアライズド・ビューを作成します。

- キューブの各ディメンションの各階層のディメンション・マテリアライズド・ビュー
- GROUP BY GROUPING SETS 構文で作成された、キューブのメジャーのファクト・マテリアライズド・ビュー

CREATE MATERIALIZED VIEW 文で生成された各グルーピング・セットは、レベルの一意の組合せを識別します。グルーピング・セットを使用すると、データを対称的に集計（地理の全レベルにわたる月レベルの売上など）したり、非対称的に集計（各都市の月レベルの売上と各州の四半期レベルの売上など）したりできます。

ファクト表の集計

DBMS_ODM は、キューブのファクト表に対応するグルーピング・セットのマテリアライズド・ビューの作成に関して、複数の方法をサポートしています。次のオプションの中から選択できます。

- キューブのすべてのレベルの組合せの集計を定義するマテリアライズド・ビューを自動生成する。

このオプションでは、ファクト表のサイズによっては、非常に大きなマテリアライズド・ビューが生成される場合があります。一般的に、このオプションはディスク領域に余裕がある場合にのみ使用します。

- キューブの最小の集計を定義するマテリアライズド・ビューを自動生成する。マテリアライズド・ビューには、各ディメンションの最大集計レベルと、最小集計レベルの1つ上のレベルのみが含まれます。

このオプションでは、ファクト表のサイズに応じて適度なサイズのマテリアライズド・ビューが生成されます。集計は対称型になります。

- キューブのレベルの組合せのあるパーセンテージ分の集計を定義するマテリアライズド・ビューを自動生成する。

このオプションでは、ファクト表のサイズおよび指定したパーセンテージに応じて適度なサイズのマテリアライズド・ビューが生成されます。マテリアライズド・ビューに含まれるレベルの組合せはランダムに決まります。通常、集計は非対称型になります。

- キューブのマテリアライズド・ビューに含めるレベルの組合せを手動で選択する。
このオプションでは、マテリアライズド・ビューの内容およびサイズの両方を細かく調整できます。集計は、対称型にも非対称型にもなります。

注意： キューブの各ディメンションに同じ集計演算子を指定している場合は、その演算子を使用してファクト・マテリアライズド・ビューのデータが集計されます。キューブの集計演算子は Enterprise Manager で設定できます。または、CWM2 プロシージャの [SET_AGGREGATION_OPERATOR](#) プロシージャ (8-6 ページ) を使用します。

キューブのディメンションの一部にのみ集計演算子を指定している場合、またはまったく指定していない場合、データは加算で集計されます。

OLAP カタログでサポートされている集計演算子のリストは、1-21 ページの表 1-10 「集計演算子」を参照してください。

手順：マテリアライズド・ビューの自動生成

キューブのマテリアライズド・ビューを自動的に作成するには、次の手順を実行します。

1. OLAP カタログにキューブを作成します。Enterprise Manager または CWM2 のプロシージャを使用できます。CWM2 のプロシージャを使用する場合は、必ずキューブをスター・スキーマにマップしてください。
2. ファイルに書き込みを行うようにデータベースを構成します。DBMS_ODM のプロシージャには、自身のユーザー ID に適切なアクセス権が付与されているディレクトリ・オブジェクトか、またはインスタンスの UTL_FILE_DIR 初期化パラメータで指定されているディレクトリ・パスを指定できます。
3. メタデータ所有者の識別情報を使用して、SQL*Plus にログインします。
4. 現在、キューブに存在するマテリアライズド・ビューを削除します。削除するマテリアライズド・ビューごとに、`DROP MATERIALIZED VIEW mv_name` を実行します。
5. ディメンション・マテリアライズド・ビューを生成するスクリプトを作成します。キューブのディメンションごとに `DBMS_ODM.CREATEDIMMV_GS` を実行します。
6. ファクト・マテリアライズド・ビューを生成するスクリプトを作成します。`DBMS_ODM.CREATESTDFACTMV` を実行し、実体化レベル・パラメータに次のいずれかの値を指定します。
 - FULL — キューブのデータを完全に実体化します。すべてのレベルの組合せがマテリアライズド・ビューに含まれます。
 - MINIMUM — キューブの最小限のデータを実体化します。各ディメンションのリーフ・レベルの 1 つ上のレベルと最大集計レベルがマテリアライズド・ビューに含まれます。

- PERCENT — キューブのレベルの組合せのあるパーセンテージに基づいて、キューブのデータを実体化します。
7. 次のコマンドを使用して、SQL*Plus でスクリプトを実行します。

```
@/users/oracle/OraHome1/olap/mvscript.sql;
```

手順 : マテリアライズド・ビューの手動生成

特定のレベルの組合せを持つマテリアライズド・ビューを作成するには、次の手順を実行します。

1. 23-3 ページの「[手順 : マテリアライズド・ビューの自動生成](#)」に示した最初の5つの手順を実行します。
2. 次の3つの手順を実行して、ファクト・マテリアライズド・ビューを生成するスクリプトを作成します。
 - a. DBMS_ODM.CREATEDIMLEVTUPLE を実行して、sys.olaptablelevels 表を作成します。この表には、キューブのすべてのディメンションおよび各ディメンションのすべてのレベルが示されます。表を編集して、マテリアライズド・ビューに含めないレベルを選択解除します。
 - b. DBMS_ODM.CREATECUBELEVELTUPLE を実行し、sys.olaptableleveltuples 表を作成します。この表には、1つ前の手順で選択したレベルの考え得るすべての組合せ（グルーピング・セット）が示されます。表を編集して、マテリアライズド・ビューに含めないレベルの組合せを選択解除します。
 - c. DBMS_ODM.CREATEFACTMV_GS を実行して、スクリプトを作成します。
3. 次のコマンドを使用して、SQL*Plus でスクリプトを実行します。

```
@/users/oracle/OraHome1/olap/mvscript_fact.sql;
```

例 : Sales キューブのマテリアライズド・ビューの作成

この例では、GLOBAL スキーマの PRICE_CUBE のマテリアライズド・ビューを作成します。

このキューブには、ある期間における様々な商品の単位原価および単位価格が格納されています。ディメンションは、商品、商品のファミリー、商品のクラスおよび合計のレベルを持つ PRODUCT と、月、四半期および年のレベルを持つ TIME です。

この例では、商品ファミリーを月別に、商品クラスを四半期別に集計し、このデータをマテリアライズド・ビューで使用できるようにします。

1. まず、ディメンション・マテリアライズド・ビュー用のスクリプトを生成します。次の文は、スクリプト prodmv および timemv をディレクトリ /users/global/scripts に作成します。

```
exec dbms_odm.createdimmv_gs
```

```
( 'global', 'product', 'prodmv', '/users/global/scripts' );
exec dbms_olap.create_dim_mv(
  'global', 'time', 'timemv', '/users/global/scripts' );
```

- このスクリプトを実行して、ディメンション・マテリアライズド・ビューを作成します。
- 次に、ファクト・マテリアライズド・ビュー用のディメンション・レベルの表を作成します。

```
exec dbms_olap.create_dim_mv_tuple( 'global', 'price_cube' );
```

レベルの表 `sys.olaptablelevels` は、このセッション専用の一時的表です。次の文を使用すると、この表を表示できます。

```
select * from sys.olaptablelevels;
```

SCHEMA_NAME	DIMENSION_NAME	DIMENSION_OWNER	CUBE_NAME	LEVEL_NAME	SELECTED
GLOBAL	TIME	GLOBAL	PRICE_CUBE	Year	1
GLOBAL	TIME	GLOBAL	PRICE_CUBE	Quarter	1
GLOBAL	TIME	GLOBAL	PRICE_CUBE	Month	1
GLOBAL	PRODUCT	GLOBAL	PRICE_CUBE	TOTAL_PRODUCT	1
GLOBAL	PRODUCT	GLOBAL	PRICE_CUBE	CLASS	1
GLOBAL	PRODUCT	GLOBAL	PRICE_CUBE	FAMILY	1
GLOBAL	PRODUCT	GLOBAL	PRICE_CUBE	ITEM	1

初期状態ではすべてのレベルが選択されています (SELECTED 列の「1」)。

- この例でマテリアライズド・ビューに含めるのは、月別の商品ファミリと四半期別の商品クラスのみなので、それ以外のレベルはすべて選択解除します。次のような文を使用して、この表を編集します。

```
update SYS.OLAPTABLELEVELS set selected = 0
  where LEVEL_NAME in ('ITEM', 'TOTAL_PRODUCT', 'Year');
select * from sys.olaptablelevels;
```

SCHEMA_NAME	DIMENSION_NAME	DIMENSION_OWNER	CUBE_NAME	LEVEL_NAME	SELECTED
GLOBAL	TIME	GLOBAL	PRICE_CUBE	Year	0
GLOBAL	TIME	GLOBAL	PRICE_CUBE	Quarter	1
GLOBAL	TIME	GLOBAL	PRICE_CUBE	Month	1
GLOBAL	PRODUCT	GLOBAL	PRICE_CUBE	TOTAL_PRODUCT	0
GLOBAL	PRODUCT	GLOBAL	PRICE_CUBE	CLASS	1
GLOBAL	PRODUCT	GLOBAL	PRICE_CUBE	FAMILY	1
GLOBAL	PRODUCT	GLOBAL	PRICE_CUBE	ITEM	0

- 次に、`sys.olaptableleveltuples` 表を作成します。この表もセッション専用の一時的表で、1つ前の手順で選択したレベルの考え得るすべての組合せが格納されます。レベ

ルの組合せ（グルーピング・セット）には、それぞれ ID 番号が付けられています。初期状態ではすべてのグルーピング・セットが選択されています（SELECTED 列の「1」）。

```
exec dbms_odm.createcubeleveltuple('global','price_cube');
select * from sys.olaptableleveltuples;
```

ID	SCHEMA_NAME	CUBE_NAME	DIMENSION_NAME	DIMENSION_OWNER	LEVEL_NAME	SELECTED
7	GLOBAL	PRICE_CUBE	PRODUCT	GLOBAL	CLASS	1
7	GLOBAL	PRICE_CUBE	TIME	GLOBAL	Quarter	1
6	GLOBAL	PRICE_CUBE	PRODUCT	GLOBAL	FAMILY	1
6	GLOBAL	PRICE_CUBE	TIME	GLOBAL	Quarter	1
3	GLOBAL	PRICE_CUBE	PRODUCT	GLOBAL	CLASS	1
3	GLOBAL	PRICE_CUBE	TIME	GLOBAL	Month	1
2	GLOBAL	PRICE_CUBE	PRODUCT	GLOBAL	FAMILY	1
2	GLOBAL	PRICE_CUBE	TIME	GLOBAL	Month	1

6. この例でマテリアライズド・ビューに含めるのは、月別の商品ファミリーと四半期別の商品クラスのみなので、それ以外のレベルの組合せは選択解除します。次のような文を使用して、sys.olaptableleveltuples 表を編集します。

```
update SYS.OLAPTABLELEVELTUPLES set selected = 0
  where ID in ('6', '3');
select * from sys.olaptableleveltuples where SELECTED = 1;
```

ID	SCHEMA_NAME	CUBE_NAME	DIMENSION_NAME	DIMENSION_OWNER	LEVEL_NAME	SELECTED
7	GLOBAL	PRICE_CUBE	PRODUCT	GLOBAL	CLASS	1
7	GLOBAL	PRICE_CUBE	TIME	GLOBAL	Quarter	1
2	GLOBAL	PRICE_CUBE	PRODUCT	GLOBAL	FAMILY	1
2	GLOBAL	PRICE_CUBE	TIME	GLOBAL	Month	1

7. ファクト・マテリアライズド・ビューを生成するスクリプトを作成するため、CREATEFACTMV_GS プロシージャを実行します。

```
exec dbms_odm.createfactmv_gs
  ('global','price_cube',
   'price_cost_mv','/users/global/scripts',TRUE);
```

作成されるスクリプトの CREATE MATERIALIZED VIEW 文の GROUP BY GROUPING SETS 句には、次の 2 つのグルーピング・セットが含まれます。

```
GROUP BY GROUPING SETS (
  (TIME_DIM.YEAR_ID, TIME_DIM.QUARTER_ID, TIME_DIM.MONTH_ID,
   PRODUCT_DIM.TOTAL_PRODUCT_ID, PRODUCT_DIM.CLASS_ID, PRODUCT_DIM.FAMILY_ID),
```

```
(TIME_DIM.YEAR_ID, TIME_DIM.QUARTER_ID,  
PRODUCT_DIM.TOTAL_PRODUCT_ID, PRODUCT_DIM.CLASS_ID) )
```

スクリプトの最後の文は、OLAP カタログのキューブに関連付けられた `mv_summary_code` を設定します。この設定は、このキューブに関連付けられたマテリアライズド・ビューがグルーピング・セット形式であることを指定します。

```
execute cwm2_olap_cube.set_mv_summary_code  
('GLOBAL', 'PRICE_CUBE', 'GROUPINGSET') ;
```

8. `/users/global/scripts` ディレクトリに移動し、`price_cost_mv` スクリプトを実行してファクト・マテリアライズド・ビューを作成します。

DBMS_ODM サブプログラムの概要

表 23-1 DBMS_ODM サブプログラム

サブプログラム	説明
CREATECUBELEVELTUPLE プロシージャ (23-8 ページ)	キューブのマテリアライズド・ビューに含められるレベルの組合せの表を作成します。
CREATEDIMLEVTUPLE プロシージャ (23-9 ページ)	キューブのマテリアライズド・ビューに含められるレベルの表を作成します。
CREATEDIMMV_GS プロシージャ (23-10 ページ)	ディメンションの各階層のマテリアライズド・ビューを作成するスクリプトを生成します。
CREATEFACTMV_GS プロシージャ (23-11 ページ)	キューブに関連付けられたファクト表のマテリアライズド・ビューを作成するスクリプトを生成します。マテリアライズド・ビューには、事前に指定した個々のレベルの組合せが含まれます。
CREATESTDFACTMV プロシージャ (23-12 ページ)	キューブに関連付けられたファクト表のマテリアライズド・ビューを作成するスクリプトを生成します。マテリアライズド・ビューは、指定した一般的な指示に従って自動的に構成されます。

CREATECUBELEVELTUPLE プロシージャ

このプロシージャは、`sys.olaptableleveltuples` 表を作成します。この表は、キューブのマテリアライズド・ビューに含まれるすべてのレベルの組合せを示します。デフォルトでは、すべてのレベルの組合せがマテリアライズド・ビューに含まれます。表を編集すると、マテリアライズド・ビューに含めないレベルの組合せを選択解除できます。

このプロシージャをコールする前に、`CREATEDIMLEVTUPLE` をコールしてキューブのレベルの表を作成します。

構文

```
CREATECUBELEVELTUPLE (
    cube_owner    IN    VARCHAR2,
    cube_name     IN    VARCHAR2);
```

パラメータ

表 23-2 CREATECUBELEVELTUPLE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。

参照

23-4 ページの「[手順: マテリアライズド・ビューの手動生成](#)」を参照してください。

23-4 ページの「[例: Sales キューブのマテリアライズド・ビューの作成](#)」を参照してください。

CREATEDIMLEVTUPLE プロシージャ

このプロシージャは、`sys.olaptablelevels` 表を作成します。この表は、キューブのすべてのディメンションのすべてのレベルを示します。デフォルトでは、すべてのレベルがマテリアライズド・ビューに含まれます。表を編集すると、マテリアライズド・ビューに含めないレベルを選択解除できます。

このプロシージャをコールした後、`CREATECUBELEVELTUPLE` をコールしてキューブのレベルの組合せ (レベル・タプル) の表を作成します。

構文

```
CREATEDIMLEVTUPLE (
    cube_owner    IN varchar2,
    cube_name     IN varchar2);
```

パラメータ

表 23-3 CREATEDIMLEVTUPLE プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。

参照

23-4 ページの「[手順: マテリアライズド・ビューの手動生成](#)」を参照してください。

23-4 ページの「例: Sales キューブのマテリアライズド・ビューの作成」を参照してください。

CREATEDIMMV_GS プロシージャ

このプロシージャは、ディメンションの各階層のマテリアライズド・ビューを作成するスクリプトを生成します。このプロシージャは、キューブのディメンションごとにコールする必要があります。

ディメンション・マテリアライズド・ビューを作成するプロセスは、ファクト・マテリアライズド・ビューを自動で生成するか手動で生成するかに関係なく同じです。

構文

```
CREATEDIMMV_GS (  
    dimension_owner    IN    VARCHAR2,  
    dimension_name     IN    VARCHAR2,  
    output_file        IN    VARCHAR2,  
    output_path        IN    VARCHAR2,  
    tablespace_mv      IN    VARCHAR2 DEFAULT NULL,  
    tablespace_index   IN    VARCHAR2 DEFAULT NULL);
```

パラメータ

表 23-4 CREATEDIMMV_GS プロシージャのパラメータ

パラメータ	説明
dimension_owner	ディメンションの所有者。
dimension_name	ディメンションの名前。
output_file	出力するスクリプトのファイル名。
output_path	output_file が作成されるディレクトリのパス。ディレクトリ・オブジェクト、または UTL_FILE_DIR パラメータで設定されているパスを指定できます。
tablespace_mv	マテリアライズド・ビューが作成される表領域の名前。このパラメータを省略すると、マテリアライズド・ビューはユーザーのデフォルトの表領域に作成されます。
tablespace_index	マテリアライズド・ビューの索引が作成される表領域の名前。このパラメータを省略すると、索引はユーザーのデフォルトの表領域に作成されます。

参照

23-3 ページの「手順: マテリアライズド・ビューの自動生成」を参照してください。

23-4 ページの「手順: マテリアライズド・ビューの手動生成」を参照してください。

23-4 ページの「例: Sales キューブのマテリアライズド・ビューの作成」を参照してください。

CREATEFACTMV_GS プロシージャ

このプロシージャは、キューブに関連付けられたファクト表のマテリアライズド・ビューを作成するスクリプトを生成します。

このプロシージャをコールする前に、CREATEDIMLEVTUPLE および CREATECUBELEVELTUPLE をコールして sys.olaptableleveltuples 表を作成する必要があります。マテリアライズド・ビューには、sys.olaptableleveltuples 表で選択したすべてのレベルの組合せが含まれます。

構文

```
CREATEFACTMV_GS (
    cube_owner          IN    VARCHAR2,
    cube_name           IN    VARCHAR2,
    outfile             IN    VARCHAR2,
    outfile_path        IN    VARCHAR2,
    partitioning        IN    BOOLEAN,
    tablespace_mv       IN    VARCHAR2 DEFAULT NULL,
    tablespace_index    IN    VARCHAR2 DEFAULT NULL);
```

パラメータ

表 23-5 CREATEFACTMV_GS プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
output_file	出力するスクリプトのファイル名。
output_path	output_file が作成されるディレクトリのパス。ディレクトリ・オブジェクト、または UTL_FILE_DIR パラメータで設定されているパスを指定できます。
partitioning	TRUE を設定すると、索引パーティション化はオンになります。FALSE を設定すると、索引パーティション化はオフになります。
tablespace_mv	マテリアライズド・ビューが作成される表領域の名前。このパラメータを省略すると、マテリアライズド・ビューはユーザーのデフォルトの表領域に作成されます。

表 23-5 CREATEFACTMV_GS プロシージャのパラメータ (続き)

パラメータ	説明
tablespace_index	マテリアライズド・ビューの索引が作成される表領域の名前。 このパラメータを省略すると、索引はユーザーのデフォルトの表領域に作成されます。

参照

23-2 ページの「[ファクト表の集計](#)」を参照してください。

23-4 ページの「[例: Sales キューブのマテリアライズド・ビューの作成](#)」を参照してください。

CREATESTDFACTMV プロシージャ

このプロシージャは、キューブに関連付けられたファクト表のマテリアライズド・ビューを作成するスクリプトを生成します。

このプロシージャは、レベルおよびレベル・タプルの表を自動的に生成および更新します。これらの表を自分で編集する場合は、CREATEDIMLEVTUPLE、CREATECUBELEVELTUPLE および CREATEFACTMV_GS の各プロシージャを使用する必要があります。

構文

```

CREATESTDFACTMV (
    cube_owner          IN    VARCHAR2,
    cube_name           IN    VARCHAR2,
    outfile              IN    VARCHAR2,
    outfile_path         IN    VARCHAR2,
    partitioning        IN    BOOLEAN,
    materialization_level IN  VARCHAR2,
    tablespace_mv       IN    VARCHAR2 DEFAULT NULL,
    tablespace_index    IN    VARCHAR2 DEFAULT NULL);

```

パラメータ

表 23-6 CREATESTDFACTMV プロシージャのパラメータ

パラメータ	説明
cube_owner	キューブの所有者。
cube_name	キューブの名前。
output_file	出力するスクリプトのファイル名。

表 23-6 CREATESTDFACTMV プロシージャのパラメータ (続き)

パラメータ	説明
output_path	output_file が作成されるディレクトリのパス。ディレクトリ・オブジェクト、または UTL_FILE_DIR パラメータで設定されているパスを指定できます。
partitioning	TRUE を設定すると、索引パーティション化はオンになります。FALSE を設定すると、索引パーティション化はオフになります。
tablespace_mv	マテリアライズド・ビューが作成される表領域の名前。このパラメータを省略すると、マテリアライズド・ビューはユーザーのデフォルトの表領域に作成されます。
materialization_level	<p>実体化のレベル。このパラメータは、マテリアライズド・ビューに含まれるレベルの組合せを指定します。次のいずれかの値を指定します。</p> <ul style="list-style-type: none"> ■ FULL — キューブのデータを完全に実体化します。すべてのレベルの組合せがマテリアライズド・ビューに含まれます。 ■ MINIMUM — キューブの最小限のデータを実体化します。各ディメンションのリーフ・レベルの 1 つ上のレベルと最大集計レベルがマテリアライズド・ビューに含まれます。 ■ PERCENT — キューブのレベルの組合せのあるパーセンテージに基づいて、キューブのデータを実体化します。たとえば、キューブに、3 つのレベルを持つディメンションが 2 つと、4 つのレベルを持つディメンションが 1 つあるとします。このキューブのレベルの組合せは 36 通り考えられます (3 × 3 × 4)。キューブの 30% を実体化する場合は、12 個のレベルの組合せがマテリアライズド・ビューに含まれることとなります。
tablespace_index	マテリアライズド・ビューの索引が作成される表領域の名前。このパラメータを省略すると、索引はユーザーのデフォルトの表領域に作成されます。

参照

23-2 ページの「[ファクト表の集計](#)」を参照してください。

OLAP_API_SESSION_INIT

OLAP_API_SESSION_INIT パッケージでは、OLAP API の初期化パラメータの表をメンテナンスするためのプロシージャが提供されます。

この章では、次の項目について説明します。

- OLAP API の初期化パラメータ
- 構成表の表示
- OLAP_API_SESSION_INIT サブプログラムの概要

OLAP API の初期化パラメータ

OLAP_API_SESSION_INIT パッケージには、初期化パラメータの構成表をメンテナンスするためのプロシージャが含まれています。OLAP API がセッションを開くと、特定のロールを持つ任意のユーザーに対して、表にリストされた ALTER SESSION コマンドが実行されます。OLAP API のみがこの表を使用します。その他のタイプのアプリケーションは、この表に格納されているコマンドを実行しません。

この機能は、すべてのユーザーの環境を変更するデータベース初期化ファイルや init.ora ファイルでこれらのパラメータを設定するかわりに使用できます。

インストール時に、OLAP API のパフォーマンス向上が示された ALTER SESSION コマンドが表に移入されます。設定を変更した方が有効な場合でも、表を変更する必要はありません。

表の情報は、この章で後述する ALL_OLAP_ALTER_SESSION ビューの別名から問い合わせることができます。

注意： このパッケージの所有者は SYS ユーザーです。このパッケージを使用するには、実行権限が明示的に付与されている必要があります。

構成表の表示

ALL_OLAP_ALTER_SESSION は、OLAP\$ALTER_SESSION 表のビューである V\$OLAP_ALTER_SESSION のパブリック・シノニムです。ビューおよび表は SYS ユーザーが所有します。

ALL_OLAP_ALTER_SESSION ビュー

ALL_OLAP_ALTER_SESSION の各行は、ロールおよびセッションの初期化パラメータを識別します。ユーザーが OLAP API を使用してセッションを開くと、セッションは、そのユーザーに付与されたロールのパラメータを使用して初期化されます。たとえば、OLAP_DBA ロールおよび SELECT_CATALOG_ROLE に行があり、ユーザーが OLAP_DBA ロールを持つ場合、OLAP_DBA ロールのパラメータは設定されますが、SELECT_CATALOG_ROLE のパラメータは無視されます。

表 24-1 ALL_OLAP_ALTER_SESSION の列の説明

列	データ型	NULL	説明
ROLE	VARCHAR2(30)	NOT NULL	データベース・ロール
CLAUSE_TEXT	VARCHAR2(3000)		ALTER SESSION コマンド

OLAP_API_SESSION_INIT サブプログラムの概要

次の表に、OLAP_API_SESSION_INIT で提供されるサブプログラムを示します。

表 24-2 OLAP_API_SESSION_INIT サブプログラム

サブプログラム	説明
ADD_ALTER_SESSION プロシージャ (24-3 ページ)	特定のデータベース・ロールを持つ OLAP API ユーザーに対し、ALTER SESSION パラメータを指定します。
CLEAN_ALTER_SESSION プロシージャ (24-4 ページ)	孤立データ (データベースに定義されていないロールに対する ALTER SESSION パラメータ) を削除します。
DELETE_ALTER_SESSION プロシージャ (24-4 ページ)	特定のデータベース・ロールを持つ OLAP API ユーザーに対して事前に定義された ALTER SESSION パラメータを削除します。

ADD_ALTER_SESSION プロシージャ

このプロシージャは、特定のデータベース・ロールを持つ OLAP API ユーザーに対し、ALTER SESSION パラメータを指定します。このプロシージャは、OLAP\$ALTER_SESSION 表に行を追加します。

構文

```
ADD_ALTER_SESSION (
    role_name          IN    VARCHAR2,
    session_parameter IN    VARCHAR2);
```

パラメータ

role_name および session_parameter が行として OLAP\$ALTER_SESSION に追加されます。

表 24-3 ADD_ALTER_SESSION プロシージャのパラメータ

パラメータ	説明
role_name	データベースの有効なロールの名前。必須です。
session_parameter	SQL ALTER SESSION コマンドで設定できるパラメータ。必須です。

例

次のコールは、OLAP_DBA ロールを持つユーザーに対するクエリー・リライトを有効にする OLAP\$ALTER_SESSION に行を挿入します。

```
call olap_api_session_init.add_alter_session(
    'OLAP_DBA', 'SET QUERY_REWRITE_ENABLED=TRUE');
```

これで、ALL_OLAP_ALTER_SESSION ビューに次の行が含まれます。

ROLE	CLAUSE TEST
OLAP_DBA	ALTER SESSION SET QUERY_REWRITE_ENABLED=TRUE

CLEAN_ALTER_SESSION プロシージャ

このプロシージャは、現在データベースに定義されていない任意のロールに対するすべての ALTER SESSION パラメータを削除します。このプロシージャは、それらのロールの OLAP\$ALTER_SESSION 表からすべての孤立行を削除します。

構文

```
CLEAN_ALTER_SESSION ();
```

例

次のコールは、すべての孤立行を削除します。

```
call olap_api_session_init.clean_alter_session();
```

DELETE_ALTER_SESSION プロシージャ

このプロシージャは、特定のデータベース・ロールを持つ OLAP API ユーザーに対して事前に定義された ALTER SESSION パラメータを削除します。このプロシージャは、OLAP\$ALTER_SESSION 表の行を削除します。

構文

```
DELETE_ALTER_SESSION (
    role_name          IN      VARCHAR2,
    session_parameter IN      VARCHAR2);
```

パラメータ

role_name および session_parameter の両方が一意に OLAP\$ALTER_SESSION の行を識別します。

表 24-4 DELETE_ALTER_SESSION プロシージャのパラメータ

パラメータ	説明
role_name	データベースの有効なロールの名前。必須です。
session_parameter	SQL ALTER SESSION コマンドで設定できるパラメータ。必須です。

例

次のコールは、1 列目に OLAP_DBA、2 列目に QUERY_REWRITE_ENABLED=TRUE という値を含む OLAP\$ALTER_SESSION の行を削除します。

```
call olap_api_session_init.delete_alter_session(  
    'OLAP_DBA', 'SET QUERY_REWRITE_ENABLED=TRUE');
```

OLAP_TABLE

OLAP_TABLE ファンクションは、アナリティック・ワークスペースから多次元データを抽出し、リレーショナル表の2次元形式で表示します。

参照：

- 『Oracle OLAP アプリケーション開発者ガイド』を参照してください。

この章では、次の項目について説明します。

- [OLAP_TABLE の使用](#)
- [例：ビューの作成](#)
- [例：埋込み合計ディメンションのビューの作成](#)
- [例：埋込み合計メジャーのビューの作成](#)
- [例：ロールアップ形式のビューの作成](#)
- [例：OLAP_TABLE での FETCH コマンドの使用](#)
- [OLAP_TABLE の構文](#)

OLAP_TABLE の使用

OLAP_TABLE ファンクションを SQL の SELECT 文で使用すると、アナリティック・ワークスペースに格納されている多次元データに問合せを発行できます。OLAP_TABLE ファンクションは、表またはビューの名前を使用する場合に使用できます。

OLAP_TABLE は、リレーショナル表およびビュー、または OLAP_TABLE で移入された他のオブジェクトの表に結合可能なオブジェクトの表を戻します。

OLAP_TABLE は、格納済のワークスペース・データを戻したり、格納済のデータに計算を実行してその結果を戻したりできます。

例：ビューの作成

アプリケーションによって要件が異なるため、アナリティック・ワークスペースから SQL にデータをフェッチする場合は、いくつかの形式を使用します。この章の例では、様々な形式を使用してビューを作成する方法を示します。これらの例で使用する構文の詳細は、25-9 ページの「[OLAP_TABLE の構文](#)」を参照してください。

これらの例はビューとして示されますが、これらの例から SELECT 文を抽出し、アナリティック・ワークスペースからアプリケーションにデータをフェッチするために直接使用することができます。

ビューを作成するには、テキスト・エディタを使用して、行、表およびビューを定義する PL/SQL スクリプトを作成します。例 25-1 は、アナリティック・ワークスペースのビュー用に SQL スクリプトを作成するためのテンプレートです。このスクリプトは、SQL*Plus の @ コマンドを使用して実行できます。

例 25-1 OLAP_TABLE を使用したビュー作成用テンプレート

```
SET ECHO ON
SET SERVEROUT ON

DROP TYPE table_obj;
DROP TYPE row_obj;

CREATE TYPE row_obj AS OBJECT (
    column_first    datatype,
    column_next     datatype,
    column_last     datatype);
/
CREATE TYPE table_obj AS TABLE OF row_obj;
/
CREATE OR REPLACE VIEW view_name AS
SELECT column1, column2, columnn
   FROM TABLE(OLAP_TABLE(
        'connection',
        'table_obj',
        'olap_command',
        'limit_map'));
/
COMMIT
/
GRANT SELECT ON view_name TO PUBLIC;
```

例：埋込み合計ディメンションのビューの作成

例 25-2 では、TIME ディメンションの STANDARD 階層のビューを作成するために使用する PL/SQL スクリプトを示します。

参照： 25-9 ページの「[OLAP_TABLE の構文](#)」を参照してください。

例 25-2 OLAP_TABLE を使用したディメンション・ビュー作成用スクリプト

```
CREATE TYPE time_std_row AS OBJECT (
  time_id          VARCHAR2(16),
  standard_short_label  VARCHAR2(16),
  standard_end_date  DATE,
  standard_timespan  NUMBER(6));
/

CREATE TYPE time_std_table AS TABLE OF time_std_row;
/

CREATE OR REPLACE VIEW time_std_view AS
SELECT time_id, standard_short_label, standard_end_date, standard_timespan
FROM TABLE(OLAP_TABLE('xademo DURATION SESSION', 'time_std_table',
  'LIMIT time_hierlist TO ''STANDARD'',
  'DIMENSION time_id FROM time WITH
  HIERARCHY time_member_parentrel
  INHIERARCHY time_member_inhier
  ATTRIBUTE standard_short_label FROM time_short.description
  ATTRIBUTE standard_end_date FROM time_end_date
  ATTRIBUTE standard_timespan FROM time_time_span')));
/
```

```
SQL> SELECT * FROM time_std_view;
```

TIME_ID	STANDARD	STANDARD_	STANDARD_TIMESPAN
-----	-----	-----	-----
L1.1996	1996	31-DEC-96	366
L1.1997	1997	31-MAY-97	151
L2.Q1.96	Q1.96	31-MAR-96	91
L2.Q2.96	Q2.96	30-JUN-96	91
L2.Q3.96	Q3.96	30-SEP-96	92
L2.Q4.96	Q4.96	31-DEC-96	92
L2.Q1.97	Q1.97	31-MAR-97	90
L2.Q2.97	Q2.97	31-MAY-97	61
L3.JAN96	Jan96	31-JAN-96	31
L3.FEB96	Feb96	29-FEB-96	29
L3.MAR96	Mar96	31-MAR-96	31

.
.
.

注意： ビューに対して SELECT 文を発行して、ビューが適切に作成されたことを確認する必要があります。この時点でのみ、OLAP_TABLE へのコール時に発生したエラーが検出されます。

例：埋込み合計メジャーのビューの作成

スター・スキーマでは、各ディメンション・ビューに結合可能な列に個別のメジャー・ビューが必要です。例 25-3 では、カスタム・メジャーをサポートする ROW2CELL によって移入された列を含むメジャー・ビューを作成するための PL/SQL スクリプトを示します。

参照：

- 20-2 ページの「カスタム・メジャーの SELECT 文への埋込み」を参照してください。
- ROW2CELL については、25-14 ページの表 25-2 「OLAP_TABLE の LIMIT_MAP パラメータの構成要素」を参照してください。

例 25-3 OLAP_TABLE を使用したメジャー・ビュー作成用スクリプト

```
CREATE TYPE measure_row AS OBJECT (  
    time                VARCHAR2(12),  
    geography           VARCHAR2(30),  
    product             VARCHAR2(30),  
    channel             VARCHAR2(30),  
    sales               NUMBER(16),  
    cost                NUMBER(16),  
    promotions         NUMBER(16),  
    quota               NUMBER(16),  
    units               NUMBER(16),  
    r2c                 RAW(32));  
  
/  
  
CREATE TYPE measure_table AS TABLE OF measure_row;  
  
/  
  
CREATE OR REPLACE VIEW measure_view AS  
SELECT sales, cost, promotions, quota, units,  
       time, geography, product, channel, r2c  
FROM TABLE(OLAP_TABLE(  
    'xademo DURATION SESSION',  
    'measure_table',  
    '',  
    'MEASURE sales FROM analytic_cube_f.sales  
    MEASURE cost FROM analytic_cube_f.costs  
    MEASURE promotions FROM analytic_cube_f.promo  
    MEASURE quota FROM analytic_cube_f.quota
```

```

MEASURE units FROM analytic_cube_f.units
DIMENSION time FROM time WITH
  HIERARCHY time_member_parentrel
  INHIERARCHY time_member_inhier
DIMENSION geography FROM geography WITH
  HIERARCHY geography_member_parentrel
  INHIERARCHY geography_member_inhier
DIMENSION product FROM product WITH
  HIERARCHY product_member_parentrel
  INHIERARCHY product_member_inhier
DIMENSION channel FROM channel WITH
  HIERARCHY channel_member_parentrel
  INHIERARCHY channel_member_inhier
ROW2CELL r2c'))
WHERE sales IS NOT NULL;
/

SQL> SELECT channel, sales, cost, promotions, quota, units FROM measure_view
WHERE   product = 'L1.TOTALPROD'
AND     geography = 'L1.WORLD'
AND     time = 'L1.1996';

```

CHANNEL	SALES	COST	PROMOTIONS	QUOTA	UNITS
STANDARD_1.CATALOG	76843552	125398	110249	16525	25209
STANDARD_1.DIRECT	41403560	2364845	518649	5458917	118851
STANDARD_2.TOTALCHANNEL	118247112	2490243	628898	5475442	144060

例：ロールアップ形式のビューの作成

ロールアップ形式では、各階層レベルの列を使用して、各ディメンション・メンバーの完全な親子関係が示されます。ロールアップ形式の構文と埋込み合計形式の構文の違いは、制限マップの各ディメンションの定義に FAMILYREL 句が追加されていることのみです。

参照： FAMILYREL については、表 25-3 を参照してください。

例 25-4 では、PRODUCT ディメンションのロールアップ・ビューを作成するために使用する PL/SQL を示します。この例では、埋込み合計形式に使用する構文と制限マップの構文の違いを表示するディメンション・ビューを示します（例 25-2 「OLAP_TABLE を使用したディメンション・ビュー作成用スクリプト」を参照）。これらのレベルのターゲット列が、基本レベルから最大集計レベルの順序（レベル・リスト・ディメンションで表示される順序）で FAMILYREL に表示されていることに注意してください。ファミリー・リレーションは 4 つの列に戻します。最大集計レベル（すべての製品）は、NULL にマッピングするとビューから省略されます。

例 25-5 では、FAMILYREL 句の代替構文を示します。この構文では、QDR を使用してファミリー・リレーションからマップされる列を識別します。

例 25-4 と例 25-5 の制限マップは、同一のビューを生成します。

例 25-4 OLAP_TABLE を使用した製品のロールアップ・ビュー作成用スクリプト

```
CREATE TYPE product_row AS OBJECT (
    equipment      VARCHAR2(20),
    components     VARCHAR2(20),
    divisions      VARCHAR2(20));
/

CREATE TYPE product_table AS TABLE OF product_row;
/

CREATE OR REPLACE VIEW product_view AS
SELECT equipment, components, divisions
   FROM TABLE(OLAP_TABLE('xademo DURATION QUERY', 'product_table',
    '' ,
    'DIMENSION product WITH
      HIERARCHY product_member_parentrel
      FAMILYREL equipment, components, divisions, null
      FROM product_member_familyrel USING product_levellist
      LABEL product_short.description
    ''));

SQL> SELECT * FROM product_view
      ORDER BY divisions, components, equipment;
```

EQUIPMENT	COMPONENTS	DIVISIONS
Chrm Cas	Audio Tape	Accessory Div
Mtl Cassette	Audio Tape	Accessory Div
Std Cassette	Audio Tape	Accessory Div
	Audio Tape	Accessory Div
	.	
	.	
	.	
Standard VCR	VCR	Video Div
Stereo VCR	VCR	Video Div
	VCR	Video Div
		Video Div

例 25-5 OLAP_TABLE の FAMILYREL 句で QDR を使用するスクリプト

```
CREATE TYPE product_row AS OBJECT (
```

```

equipment      VARCHAR2(15),
components     VARCHAR2(15),
divisions      VARCHAR2(15));
/

CREATE TYPE product_table AS TABLE OF product_row;
/

CREATE OR REPLACE VIEW product_view AS
SELECT equipment, components, divisions
   FROM TABLE(OLAP_TABLE('xademo DURATION QUERY', 'product_table',
   ',
   'DIMENSION product WITH
   HIERARCHY product_member_parentrel
   FAMILYREL equipment, components, divisions FROM
   product_member_familyrel(product_levellist 'L4'),
   product_member_familyrel(product_levellist 'L3'),
   product_member_familyrel(product_levellist 'L2')
   LABEL product_short.description
   '));
/

SQL> SELECT * FROM product_view
      ORDER BY divisions, components, equipment;

```

EQUIPMENT	COMPONENTS	DIVISIONS
Chrm Cas	Audio Tape	Accessory Div
Mtl Cassette	Audio Tape	Accessory Div
Std Cassette	Audio Tape	Accessory Div
	Audio Tape	Accessory Div
	.	
	.	
	.	
Standard VCR	VCR	Video Div
Stereo VCR	VCR	Video Div
	VCR	Video Div
		Video Div

例 : OLAP_TABLE での FETCH コマンドの使用

次の例は、2つの変数（SALES および COST）からデータをフェッチして、2つのカスタム・メジャー（COST_PRIOR_PERIOD および PROFIT）を計算します。この例では、ビューを使用せずにアプリケーションで直接 OLAP_TABLE を使用方法も示します。

参照 : 25-11 ページの「[OLAP_COMMAND](#) パラメータでの [FETCH](#) の使用」を参照してください。

例 25-6 OLAP_TABLE で FETCH を使用するスクリプト

```
CREATE TYPE measure_row AS OBJECT (
    time                VARCHAR2(20),
    geography            VARCHAR2(30),
    product              VARCHAR2(30),
    channel              VARCHAR2(30),
    sales                NUMBER(16),
    cost                NUMBER(16),
    cost_prior_period   NUMBER(16),
    profit              NUMBER(16));
/

CREATE TYPE measure_table AS TABLE OF measure_row;
/

SELECT time, geography, product, channel,
       sales, cost, cost_prior_period, profit
   FROM TABLE(OLAP_TABLE(
    'xademo DURATION SESSION',
    'measure_table',
    'FETCH time, geography, product, channel, analytic_cube_f.sales,
    analytic_cube_f.costs, LAG(analytic_cube_f.costs, 1, time, LEVELREL time_member_levelrel),
    analytic_cube_f.sales - analytic_cube_f.costs',
    ''))
  WHERE channel = 'STANDARD_2.TOTALCHANNEL' AND
         product = 'L1.TOTALPROD' AND
         geography = 'L1.WORLD'
 ORDER BY time;
```

この SQL SELECT 文は、次の結果セットを戻します。

TIME	GEOGRAPHY	PRODUCT	CHANNEL	SALES	COST	COST_PRIOR_PERIOD	PROFIT
L1.1996	L1.WORLD	L1.TOTALPROD	STANDARD_2.TOTALCHANNEL	118247112	2490243		115756869
L1.1997	L1.WORLD	L1.TOTALPROD	STANDARD_2.TOTALCHANNEL	46412113	1078031	2490243	45334082
L2.Q1.96	L1.WORLD	L1.TOTALPROD	STANDARD_2.TOTALCHANNEL	26084848	560379		25524469
L2.Q1.97	L1.WORLD	L1.TOTALPROD	STANDARD_2.TOTALCHANNEL	26501765	615399	560379	25886367
L2.Q2.96	L1.WORLD	L1.TOTALPROD	STANDARD_2.TOTALCHANNEL	30468054	649004	615399	29819049
L2.Q2.97	L1.WORLD	L1.TOTALPROD	STANDARD_2.TOTALCHANNEL	19910347	462632	649004	19447715
L2.Q3.96	L1.WORLD	L1.TOTALPROD	STANDARD_2.TOTALCHANNEL	27781702	582693	462632	27199009
L2.Q4.96	L1.WORLD	L1.TOTALPROD	STANDARD_2.TOTALCHANNEL	33912508	698166	582693	33214342
L3.APR96	L1.WORLD	L1.TOTALPROD	STANDARD_2.TOTALCHANNEL	8859808	188851		8670957

⋮

27 rows selected.

OLAP_TABLE の構文

OLAP_TABLE ファンクションは、*limit_map* に定義されている規則に従って移入された *table_name* で識別されたオブジェクトの表を戻します。

入力パラメータに指定された情報を OLAP_TABLE が処理する順序については、25-18 ページの「[注意: OLAP_TABLE における処理の順序](#)」を参照してください。

構文

```
OLAP_TABLE(
    aw_attach          IN   VARCHAR2,
    table_name         IN   VARCHAR2,
    olap_command       IN   VARCHAR2,
    limit_map          IN   VARCHAR2);
```

パラメータ

表 25-1 OLAP_TABLE ファンクションのパラメータ

パラメータ	説明
<i>aw_attach</i>	ソース・データが含まれているアナリティック・ワークスペースの名前。25-9 ページの「 AW_ATTACH パラメータ 」を参照してください。
<i>table_name</i>	多次元データを表形式で構成するために定義された表オブジェクトの名前。25-10 ページの「 TABLE_NAME パラメータ 」を参照してください。
<i>olap_command</i>	データがフェッチされる前に実行される OLAP DML コマンド。25-10 ページの「 OLAP_COMMAND パラメータ 」を参照してください。
<i>limit_map</i>	<i>aw_attach</i> のソース・オブジェクトおよび <i>table_name</i> のターゲット列を識別するキーワード・ベースのマップ。25-12 ページの「 LIMIT_MAP パラメータ 」を参照してください。

AW_ATTACH パラメータ

OLAP_TABLE ファンクションの最初のパラメータは、ソース・データが格納されているアナリティック・ワークスペースの名前を指定し、OLAP_TABLE への最初のコール時にオープンする OLAP セッションにアナリティック・ワークスペースをアタッチしておく期間を指定します。アナリティック・ワークスペースは、問合せの終了時またはセッションの終了時にデタッチできます。次に、このパラメータの完全な構文を示します。

```
'[owner.]aw_name DURATION QUERY | SESSION'
```

次に例を示します。

```
'sys.xademo DURATION QUERY'
```

owner は、他のユーザーによってアクセスされるビューを作成する場合に指定します。そのようなビューを作成しない場合で、アナリティック・ワークスペースを所有している場合は、*owner* を省略できます。このパラメータは、所有者以外のユーザー名でログインしている場合にのみ必要です。

SESSION を指定する場合は、アナリティック・ワークスペースがすでにアタッチされているため、OLAP_TABLE への後続のコールでこのパラメータに空の文字列を使用できます。接続文字列は、必要以上に繰り返すと無視されます。

SESSION を指定すると、アナリティック・ワークスペースがセッションに 1 回のみアタッチされるため、QUERY より高いパフォーマンスが提供されます。ただし、この場合、他のユーザーが行った変更は表示されません。

TABLE_NAME パラメータ

2 番目のパラメータは、オブジェクトの表の名前を識別します。このパラメータの構文は次のとおりです。

```
'table_name'
```

次に例を示します。

```
'product_table'
```

OLAP_COMMAND パラメータ

OLAP_TABLE ファンクションの 3 番目のパラメータは、単一の OLAP DML コマンドです。複数のコマンドを実行する場合は、アナリティック・ワークスペースにプログラムを作成し、このパラメータでそのプログラムをコールします。このパラメータの機能および柔軟性は、OLAP DML で使用可能なデータ操作コマンドを仮想的に処理する機能に基づいています。

olap_command パラメータを OLAP_TABLE が処理する順序については、25-18 ページの「[注意: OLAP_TABLE における処理の順序](#)」を参照してください。

このパラメータの構文は次のとおりです。

```
'olap_command'
```

olap_command パラメータには、次の 2 つの異なる用途があります。

- 制限マップを実行する前に、ワークスペース・セッションで変更を加える
- 制限マップを使用するかわりに、ソース・データを直接指定する

以降の項で、これらの方法について説明します。

OLAP_COMMAND と LIMIT_MAP の併用

制限マップを実行する前に、アプリケーションでただちにアナリティック・ワークスペースに変更を加えたい場合があります。この場合は、*olap_command* パラメータを使用するか、または制限マップで PREDMLCMD キーワードを使用します。詳細は、表 25-2 「OLAP_TABLE の LIMIT_MAP パラメータの構成要素」を参照してください。

通常、*olap_command* パラメータは、1 つ以上のディメンションを制限する場合に使用します。DIMENSION 句で指定したディメンションのいずれかを制限する場合、ディメンションのステータスは、OLAP_TABLE へのコール中のみ変更されます。それ以外の OLAP セッションは影響されません。ただし、他のコマンドがセッションに影響を与える場合があります。

LIMIT コマンドの効力を、コマンドの間だけでなくセッションを通じて有効にしたい場合は、このコマンドを制限マップの PREDMLCMD 句に指定します。

olap_command パラメータでの LIMIT コマンドの使用例を次に示します。

```
'LIMIT product TO product_member_levelrel 'L2''
```

OLAP_COMMAND パラメータでの FETCH の使用

olap_command パラメータには、FETCH コマンドを指定することもできます。このコマンドは、表オブジェクトのソース・データを指定します。FETCH を使用する場合、制限マップは指定しません。25-7 ページの「例: OLAP_TABLE での FETCH コマンドの使用」を参照してください。

FETCH コマンドは、Oracle データベースに移行した OLAP Server アプリケーション用に用意されています。

注意： SNAPI 用に FETCH コマンドを使用する OLAP Server アプリケーションをアップグレードした場合にのみ、OLAP_TABLE で FETCH キーワードを使用します。アップグレードした場合、OLAP Server 6.3 と同じ完全な構文を Oracle で使用できます。以前に SNAPI で使用していたものと同じ FETCH コマンドを OLAP_TABLE で使用できます。

FETCH は、アナリティック・ワークスペースのデータをどのように表オブジェクトにマップするかを明示的に指定します。基本的な構文は次のとおりです。

FETCH *expression*...

ターゲット列ごとに式 (*expression*) を 1 つ入力します。式は、行定義に指定されているとおりの順序で入力します。式は、空白またはカンマで区切ります。改行や継続記号を使用せずに、文全体を 1 行で記述する必要があります。

LIMIT_MAP パラメータ

OLAP_TABLE ファンクションの 4 番目（最後）のパラメータは、アナリティック・ワークスペース・オブジェクトを表の列にマップし、各オブジェクトのロールを識別します。このパラメータは、SELECT 文の WHERE 句と組み合わされて、一連の LIMIT コマンドをアナリティック・ワークスペースに発行するため、制限マップと呼ばれます。制限マップの内容は、*table_name* に指定された表に移入されます。

制限マップの情報を OLAP_TABLE が処理する順序については、25-18 ページの「[注意：OLAP_TABLE における処理の順序](#)」を参照してください。

FETCH コマンドを *olap_command* パラメータで使用する場合、制限マップは省略します。

制限マップのすべてまたは一部をアナリティック・ワークスペースのテキスト変数に格納できます。制限マップに変数を挿入するには、変数名の前にアンパサンド (&) を付けます。この操作は、OLAP DML のアンパサンド置換と呼ばれます。

制限マップの最大長は 2000 文字です。これは、PL/SQL の制限によるものです。

制限マップの構文には、主にディメンション階層を定義するために、多くの句が含まれています。構文エラーが発生すると制限マップが解析されないため、カンマの有無に注意してください。

例 25-7 OLAP_TABLE の LIMIT_MAP パラメータの構文

```
' [MEASURE column FROM {measure | AW_EXPR expression}]
.
.
.
DIMENSION [column FROM] dimension
[WITH
  [HIERARCHY [column FROM] hierarchy_relation[(hierarchy_dimension 'hierarchy')]
  [INHIERARCHY inhierarchy_variable]
  [GID column FROM gid_variable]
  [PARENTGID column FROM gid_variable]
  [FAMILYREL col1, col2, coln FROM
    {expression1, expression2, expressionn |
    family_relation USING level_dimension }
  [LABEL label_variable]]
.
.
.
]
[ATTRIBUTE column FROM attribute_variable]
.
.
.
]
[ROW2CELL column]
[LOOP composite_dimension]
```

```
[PREDMLCMD olap_command]  
[POSTDMLCMD olap_command]  
,
```

各項目の意味は次のとおりです。

column は、ターゲット表の列の名前です。

measure は、アナリティック・ワークスペースに格納されるビジネス・メジャーです。

dimension は、アナリティック・ワークスペースのディメンションです。

expression は、式、またはアナリティック・ワークスペースのオブジェクトに対する修飾データ参照です。

hierarchy_relation は、*dimension* の階層を定義するアナリティック・ワークスペースのセルフ・リレーションです。

hierarchy_dimension は、*dimension* の階層の名前を含むアナリティック・ワークスペースのディメンションです。

hierarchy は、*hierarchy_dimension* のメンバーです。

inhierarchy_variable は、ディメンション・メンバーが *hierarchy* に存在するかどうかを識別するアナリティック・ワークスペースのブール変数です。

gid_variable は、各ディメンション・メンバーのグルーピング ID を含むアナリティック・ワークスペースの変数の名前です。

attribute_variable は、*dimension* の属性値を含むアナリティック・ワークスペースの変数の名前です。

composite_dimension は、*measure* の定義で使用されるコンポジット・ディメンションの名前です。

olap_command は、OLAP DML コマンドです。

表 25-2 OLAP_TABLE の LIMIT_MAP パラメータの構成要素

キーワード	キーワード句の構文と説明
MEASURE	<p>MEASURE column FROM {measure AW_EXPR expression}</p> <p>MEASURE 句は、アナリティック・ワークスペースの変数、式またはリレーションをターゲット表の列にマップします。</p> <p>また、AW_EXPR キーワードも、1つ以上のこれらのオブジェクトに対して OLAP 計算エンジンで実行された計算を列にマップできます。たとえば、計算は次のとおり指定できます。</p> <pre>analytic_cube_sales - analytic_cube_cost</pre> <p>または</p> <pre>LAGDIF(analytic_cube_sales, 1, time, LEVELREL time.lvlrel)</pre> <p>任意の数の MEASURE 句を表示できます。この句は、ディメンション・ビューを作成する場合はオプションになります。</p>

表 25-2 OLAP_TABLE の LIMIT_MAP パラメータの構成要素 (続き)

キーワード	キーワード句の構文と説明
DIMENSION	<p>DIMENSION [column FROM] dimension...</p> <p>DIMENSION 句は、制限マップの1つ以上のメジャー、属性または階層をディメンション化するアナリティック・ワークスペースのディメンションまたは結合ディメンションを識別します。</p> <p><i>column</i> 副次句は、ディメンション・メンバーを表に表示しない場合はオプションになります。この場合、データ選択に使用可能なディメンション属性を含める必要があります。</p> <p>すべての制限マップに、1つ以上の DIMENSION 句を含める必要があります。制限マップに MEASURE 句が含まれている場合は、ディメンションが単一の値に制限されていないかぎり、メジャーの各ディメンションに対する単一の DIMENSION 句も含まれている必要があります。メジャーがコンポジット・ディメンションでディメンション化されている場合は、DIMENSION 句を使用してコンポジット・ディメンションの各ディメンションを識別する必要があります。大量の結果セットをフェッチする場合に高パフォーマンスを得るには、LOOP 句のコンポジット・ディメンションを識別します。</p> <p>ディメンションの名前は、1つの DIMENSION 句でのみ指定できます。DIMENSION の副次句は、ディメンション階層および属性を識別します。</p> <p>WITH 句は、HIERARCHY または ATTRIBUTE 副次句を導入します。制限マップからこれらの副次句を省略する場合は、WITH 句も省略します。ただし、これらの副次句のいずれかまたは両方を含める場合は、単一の WITH 句をそれらの前に付けます。WITH 句の構文は次のとおりです。各構成要素の詳細は、表 25-3 を参照してください。</p> <p>[WITH</p> <pre> [HIERARCHY [column FROM] hierarchy_relation[(hierarchy_dimension 'hierarchy')] [INHIERARCHY inhierarchy_variable] [GID column FROM gid_variable] [PARENTGID column FROM gid_variable] [FAMILYREL col1, col2, coln FROM {expression1, expression2, expressionn family_relation USING level_dimension } [LABEL label_variable]] ] [ATTRIBUTE column FROM attribute_variable] ] </pre> <p>ROW2CELL ROW2CELL column</p> <p>ROW2CELL 句は、DBMS_AW パッケージの単一行ファンクションに必要な情報を RAW(32) 列に移入します。これらのファンクションで使用するビューを作成する場合にこの句を使用します。20-18 ページの「OLAP_EXPRESSION ファンクション」を参照してください。</p> <p>LOOP LOOP sparse_dimension</p> <p>LOOP 句は、制限マップで指定された1つ以上のメジャーをディメンション化する単一の名前付きコンポジット・ディメンションを識別します。大量の結果セットをフェッチする場合にパフォーマンスが向上します。ただし、少数の値の取得には時間がかかる場合があります。</p>

表 25-2 OLAP_TABLE の LIMIT_MAP パラメータの構成要素 (続き)

キーワード	キーワード句の構文と説明
PREDMLCMD	<p>PREDMLCMD <code>olap_command</code></p> <p>PREDMLCMD は、データがアナリティック・ワークスペースからターゲット表にフェッチされる前に実行される OLAP DML コマンドを指定します。たとえば、結果が表にフェッチされるモデルまたは予測を実行する場合に使用できます。コマンドの結果は、制限マップを実行している間有効で、OLAP_TABLE の実行が完了した後のセッションにも引き継がれます。25-18 ページの「注意: OLAP_TABLE における処理の順序」を参照してください。</p>
POSTDMLCMD	<p>POSTDMLCMD <code>olap_command</code></p> <p>POSTDMLCMD は、データがアナリティック・ワークスペースからターゲット表にフェッチされた後に実行される OLAP DML コマンドを指定します。たとえば、PREDMLCMD 句のコマンドで作成されたオブジェクトまたはデータを削除する場合、あるいは PREDMLCMD 句で変更されたディメンションのステータスをリストアする場合に使用できます。「注意: OLAP_TABLE における処理の順序」を参照してください。</p>

表 25-3 OLAP_TABLE の LIMIT_MAP パラメータの DIMENSION 句の WITH 副次句

キーワード	構成要素
HIERARCHY	<p>HIERARCHY [column FROM] hierarchy_relation[(hierarchy_dimension 'hierarchy')]...</p> <p>HIERARCHY 副次句は、ディメンションの階層を定義するアナリティック・ワークスペースの親セルフ・リレーションを識別します。</p> <p><i>hierarchy_dimension</i> に複数のメンバーが存在する場合は、(<i>hierarchy_dimension</i> 'hierarchy') 句を使用してメンバーを指定できます。複数の階層を含めるには、各メンバーに HIERARCHY 副次句を指定します。<i>hierarchy_dimension</i> は、後続の副次句 (INHIERARCHY、GID、PARENTGID および FAMILYREL) で参照されるすべてのアナリティック・ワークスペース・オブジェクトの <i>hierarchy</i> に制限されます。</p> <p>HIERARCHY 副次句は、<i>dimension</i> に階層が含まれない場合、または <i>dimension</i> のステータスが階層の単一レベルに制限されている場合はオプションになります。</p> <ul style="list-style-type: none"> ■ INHIERARCHY inhierarchy_variable <p>INHIERARCHY 副次句は、階層にディメンションが存在するかどうかを識別するアナリティック・ワークスペースのブール変数を識別します。階層から省略されるディメンション・メンバーが存在する場合にのみ必要です。これは、ディメンションに複数の階層が存在する場合に特有です。</p> ■ GID column FROM gid_variable <p>GID 副次句は、各ディメンション・メンバーのグルーピング ID を含むアナリティック・ワークスペースの整数変数をターゲット表の列にマップします。OLAP API を使用する Java アプリケーションの場合に必要です。</p> ■ PARENTGID column FROM gid_variable <p>PARENTGID 副次句は、アナリティック・ワークスペースの GID 変数を使用して親リレーションのグルーピング ID を計算します。親の GID は、アナリティック・ワークスペース・オブジェクトには格納されません。かわりに、GID 句で使用する変数と同じ GID 変数を PARENTGID 句に指定します。</p> <p>PARENTGID 句は、OLAP API を使用する Java アプリケーションに使用することをお勧めします。</p> ■ FAMILYREL col1, col2, coln FROM {expression1, expression2, expressionn family_relation USING level_dimension } [LABEL label_variable] <p>FAMILYREL 副次句は、主に、アナリティック・ワークスペースのファミリー・リレーションをターゲット表の複数列にマップするために使用します。level_dimension の順序で列を表示します。特定のレベルを含めない場合は、ターゲット列に NULL を指定します。結果ビューは、ロールアップ形式です。この形式では、階層の各レベルが別々の列に表示され、各ディメンション・メンバーの完全な親子関係が行内で識別されます。</p> <p>FAMILYREL 副次句は、修飾データ参照 (QDR) のリストを複数列にマップするためにも使用できます。この場合、最初の QDR が最初の列に、2 番目の QDR が 2 番目の列にというようにマップされます。</p> <p>LABEL キーワードは、ディメンション・メンバーに意味のある名前を指定するテキスト属性を識別します。</p> <p>各階層に対して複数の FAMILYREL 句を使用できます。</p>

表 25-3 OLAP_TABLE の LIMIT_MAP パラメータの DIMENSION 句の WITH 副次句 (続き)

キーワード	構成要素
ATTRIBUTE	<p>ATTRIBUTE column FROM attribute_variable</p> <p>ATTRIBUTE 句は、アナリティック・ワークスペースの変数をターゲット表の列にマップします。attribute_variable に複数のディメンションが含まれている場合、値は、<i>dimension</i> のすべてのメンバーに対してマップされますが、追加ディメンションの現在のステータスでは最初のメンバーに対してのみマップされます。たとえば、属性に言語ディメンションが含まれている場合は、そのディメンションのステータスを特定の言語に設定する必要があります。PREDMLCMD 句にディメンションのステータスを設定できます。</p>

注意 : OLAP_TABLE における処理の順序

次に、アナリティック・ワークスペース内のディメンションのステータスを変更する可能性のある命令を、OLAP_TABLE ファンクションが処理する順序を示します。

1. 制限マップの PREDMLCMD パラメータに指定されている OLAP DML コマンドをすべて実行します。
2. 後でリストアできるように、すべてのディメンションの現在のステータスを保存します (ステータスの PUSH)。
3. INHIERARCHY 句で指定されている階層内にあるディメンション値のステータスのみを維持します (LIMIT KEEP)。
4. OLAP_TABLE ファンクションを含む SQL SELECT 文の WHERE 句を満たすディメンション値のステータスのみを維持します。
5. OLAP_TABLE ファンクションの *olap_command* パラメータに指定されている OLAP DML コマンドをすべて実行します。
6. データをフェッチします。
7. すべてのディメンションのステータスをリストアします (ステータスの POP)。
8. 制限マップの POSTDMLCMD パラメータに指定されている OLAP DML コマンドをすべて実行します。

索引

A

- ADVISE_CUBE プロシージャ, 20-11, 20-5, 20-11
ADVISE_REL プロシージャ, 20-5, 20-6, 20-12
ALL_AW_CUBE_AGG_LEVELS ビュー, 4-4
ALL_AW_CUBE_AGG_MEASURES ビュー, 4-4
ALL_AW_CUBE_AGG_PLANS ビュー, 4-5
ALL_AW_CUBE_ENABLED_HIERCOMBO ビュー, 4-5
ALL_AW_CUBE_ENABLED_VIEWS ビュー, 4-5, 21-4
ALL_AW_DIM_ENABLED_VIEWS ビュー, 4-6
ALL_AW_LOAD_CUBE_DIMS ビュー, 4-7
ALL_AW_LOAD_CUBE_FILTS ビュー, 4-8
ALL_AW_LOAD_CUBE_MEASURES ビュー, 4-9
ALL_AW_LOAD_CUBE_PARMS ビュー, 4-10
ALL_AW_LOAD_CUBE_CUBES ビュー, 4-7
ALL_AW_LOAD_DIM_FILTERS ビュー, 4-11
ALL_AW_LOAD_DIM_PARMS ビュー, 4-11
ALL_AW_LOAD_DIMENSIONS ビュー, 4-10
ALL_OLAP2_AGGREGATION_USES ビュー, 5-3
ALL_OLAP2_AW_ATTRIBUTES ビュー, 3-5
ALL_OLAP2_AW_CUBE_AGG_LVL ビュー, 3-6
ALL_OLAP2_AW_CUBE_AGG_MEAS ビュー, 3-7
ALL_OLAP2_AW_CUBE_AGG_OP ビュー, 3-7
ALL_OLAP2_AW_CUBE_AGG_SPECS ビュー, 3-8
ALL_OLAP2_AW_CUBE_DIM_USES ビュー, 3-8
ALL_OLAP2_AW_CUBE_MEASURES ビュー, 3-9
ALL_OLAP2_AW_CUBE_CUBES ビュー, 3-6
ALL_OLAP2_AW_DIM_HIER_LVL_ORD ビュー, 3-10
ALL_OLAP2_AW_DIM_LEVELS ビュー, 3-11
ALL_OLAP2_AW_DIMENSIONS ビュー, 3-10
ALL_OLAP2_AW_MAP_ATTR_USE ビュー (廃止)
「ALL_OLAP2_AW_ATTRIBUTES ビュー」を参照
ALL_OLAP2_AW_MAP_DIM_USE ビュー (廃止)
「ALL_OLAP2_AW_DIMENSIONS ビュー」を参照
ALL_OLAP2_AW_MAP_MEAS_USE ビュー (廃止)
「ALL_OLAP2_AW_CUBE_MEASURES ビュー」を参照
ALL_OLAP2_AW_PHYS_OBJ_PROP ビュー, 3-12
ALL_OLAP2_AW_PHYS_OBJ ビュー, 3-11
ALL_OLAP2_AWS ビュー, 3-5, 21-4
ALL_OLAP2_CATALOG_ENTITY_USES ビュー, 5-4
ALL_OLAP2_CATALOGS ビュー, 5-4
ALL_OLAP2_CUBE_DIM_USES ビュー, 5-5
ALL_OLAP2_CUBE_MEAS_DIM_USES ビュー, 5-7
ALL_OLAP2_CUBE_MEASURE_MAPS ビュー, 5-6
ALL_OLAP2_CUBE_MEASURES ビュー, 5-6
ALL_OLAP2_CUBE_CUBES ビュー, 5-5
ALL_OLAP2_DIM_ATTR_USES ビュー, 5-8
ALL_OLAP2_DIM_ATTRIBUTES ビュー, 5-8
ALL_OLAP2_DIM_HIER_LEVEL_USES ビュー, 5-9
ALL_OLAP2_DIM_HIERARCHIES ビュー, 5-9
ALL_OLAP2_DIM_LEVEL_ATTR_MAPS ビュー, 5-11
ALL_OLAP2_DIM_LEVEL_ATTRIBUTES ビュー, 5-10
ALL_OLAP2_DIM_LEVELS ビュー, 5-10
ALL_OLAP2_DIMENSIONS ビュー, 5-7
ALL_OLAP2_FACT_LEVEL_USES ビュー, 5-12
ALL_OLAP2_FACT_TABLE_GID ビュー, 5-13
ALL_OLAP2_HIER_CUSTOM_SORT ビュー, 5-13
ALL_OLAP2_JOIN_KEY_COLUMN_USES ビュー, 5-14
ALTER SESSION コマンド, 24-1
Analytic Workspace Manager, 1-1, 1-24, 22-1
ATTRIBUTE 副次句 (制限マップ), 25-18

C

CWM2, 1-4, 2-3 ~ 2-10
書込み API, 2-1 ~ 2-10
出力の送信, 2-17
CWM2\$_AW_PERM_CUST_MEAS_MAP 表, 21-2
CWM2\$_AW_TEMP_CUST_MEAS_MAP 表, 21-3
CWM2_OLAP_CATALOG パッケージ, 7-1 ~ 7-2
CWM2_OLAP_CUBE パッケージ, 8-1 ~ 8-2
CWM2_OLAP_DIMENSION_ATTRIBUTE パッケージ, 10-1 ~ 10-3
CWM2_OLAP_DIMENSION パッケージ, 9-1 ~ 9-2
CWM2_OLAP_HIERARCHY パッケージ, 11-1 ~ 11-2
CWM2_OLAP_LEVEL_ATTRIBUTE パッケージ, 13-1 ~ 13-3
CWM2_OLAP_LEVEL パッケージ, 12-1 ~ 12-2
CWM2_OLAP_MANAGER パッケージ, 1-11, 1-13, 2-13, 2-17
CWM2_OLAP_MEASURE パッケージ, 14-1 ~ 14-2
CWM2_OLAP_METADATA_REFRESH パッケージ, 15-1
CWM2_OLAP_PC_TRANSFORM パッケージ, 16-1 ~ 16-8
CWM2_OLAP_TABLE_MAP パッケージ, 17-1 ~ 17-3
CWM2_OLAP_VERIFY_ACCESS パッケージ, 19-1

D

DBMS_AWM パッケージ, 1-1, 1-30, 22-1 ~ 22-65
DBMS_AW パッケージ
ADVISE_CUBE プロシージャ, 20-11
ADVISE_REL プロシージャ, 20-12
EXECUTE プロシージャ, 20-13
GETLOG ファンクション, 20-14
INTERP_SILENT ファンクション, 20-17
INTERPCLOB ファンクション, 20-16
INTERP ファンクション, 20-15
OLAP_EXPRESSION ファンクション, 20-18
PRINTLOG プロシージャ, 20-22
概要, 20-1
カスタム・メジャー, 20-18, 20-19, 20-20, 20-21
DBMS_ODM パッケージ, 23-3, 23-4
DIMENSION 句 (制限マップ), 25-15
DISPLAY_NAME, 4-12, 22-64

E

End Date, 10-2, 13-2
ET Key, 10-2, 13-2

F

FAMILYREL 副次句 (制限マップ), 25-17
FETCH コマンド (DML), 25-11

G

GETLOG ファンクション, 20-14
GID 副次句 (制限マップ), 25-17
Grouping ID, 10-2, 13-2

H

HIERARCHY 副次句 (制限マップ), 25-17

I

CWM2_OLAP_VALIDATE パッケージ, 18-1
INHIERARCHY 副次句 (制限マップ), 25-17
init.ora ファイル, 24-1

L

Long Description, 10-2, 13-2
LOOP 句 (制限マップ), 25-15

M

MEASURE 句 (制限マップ), 25-14
Metadata Reader の表: リフレッシュ, 2-12
Metadata Reader の表: リフレッシュ, 2-15, 2-18
MR_REFRESH プロシージャ, 15-3
MRV_OLAP ビュー, 2-15, 2-18, 15-1, 15-2

O

OLAP API
Metadata Reader の表, 2-15, 2-18
最適化, 24-1
OLAP API イネーブラ, 1-5, 22-23, 22-32
OLAP DML
SQL での実行, 20-1 ~ 20-20

OLAP_API_SESSION_INIT パッケージ, 24-1 ~ 24-2
OLAP_PAGE_POOL_SIZE パラメータ, 6-3
OLAP_TABLE ファンクション
 カスタム・メジャー, 20-19, 20-20, 20-21, 20-22
 構文, 25-9
 セッション・ログの取得, 20-14
 説明, 25-1 ~ 25-6
OLAP カタログ
 OLAP API のビューのリフレッシュ, 15-1
 書き込み API, 2-1 ~ 2-10
 ビュー, 15-1
 表示, 2-18, 5-1
 プリプロセッサ, 16-1
 メタデータ・エンティティのサイズ制限, 2-16
 メタデータのエンティティ, 2-1
 メタデータのエンティティのサイズ, 2-2
 読み込み API, 2-18, 4-1, 5-1, 15-1
OLAP パフォーマンス・ビュー, 6-1
OLAP メタデータ
 DBMS_AWM 用に作成, 1-3
 検証, 2-12 ~ 2-15, 18-1, 19-1
 コミット, 2-12
 ディメンション表に作成, 2-3
 ファクト表に作成, 2-9
 マッピング, 2-5, 2-8, 2-10, 5-6, 5-11, 5-15,
 18-2
OLAP メタデータの検証, 2-12 ~ 2-15
Oracle Enterprise Manager, 1-4
Oracle Warehouse Builder, 1-4
OUTFILE コマンド
 DBMS_AW プロシージャでの効果, 20-13

P

P_DISPLAY_NAME, 4-12, 22-64
Parent ET Key, 10-2, 13-2
Parent Grouping ID, 10-2, 13-2
PARENTGID 副次句 (制限マップ), 25-17
PGA の割当て, 6-3
POSTDMLCMD 句 (制限マップ), 25-16
PREDMLCMD 句 (制限マップ), 25-16
Prior Period, 10-2, 13-2

R

ROW2CELL 句 (制限マップ), 25-15

S

segwidth, 22-52
SERVEROUTPUT オプション, 1-11, 1-13, 2-13,
 2-17, 20-13, 20-22
Short Description, 10-2, 13-2
solved_code, 2-12
SQL
 OLAP コマンドの埋込み, 20-1 ~ 20-20

T

Time Span, 10-2, 13-2

U

UNIQUE_RDBMS_KEY, 4-12, 22-64
UTL_FILE_DIR パラメータ, 1-23, 2-17, 16-1, 16-6,
 22-22, 22-31, 22-37, 22-43, 23-3, 23-10, 23-11,
 23-13

V

V\$AW_CALC ビュー, 6-2
V\$AW_OLAP ビュー, 6-4
V\$AW_SESSION_INFO ビュー, 6-5

Y

Year Ago Period, 10-2, 13-2

あ

アクティブ・カタログ, 3-1, 21-4
アナリティック・ワークスペース
 DBMS_AWM で作成, 1-30
 OLAP API の有効化, 1-5, 1-22, 22-1, 22-21,
 22-23, 22-30, 22-32
 アクティブ・カタログ, 3-1
 集計, 1-5, 1-18, 22-16, 22-24
 ディメンション, 1-16
 パフォーマンス・カウンタ, 6-4
 ビュー, 1-26, 22-23, 22-32
 メンテナンス・ビュー, 4-1
 リスト, 3-5
 リストの取得, 21-4
 リフレッシュ, 22-45, 22-47

アナリティック・ワークスペース管理 API, 22-1 ~ 22-65

アナリティック・ワークスペース・メンテナンス・ビュー, 4-1

い

引用符

OLAP DML, 20-2

う

埋込み合計キー, 2-12, 18-3

埋込み合計キューブ, 18-3

埋込み合計ディメンション・ビュー, 1-22
作成, 1-27, 25-2

埋込み合計ファクト・ビュー, 1-22

埋込み合計ファクト表, 2-11

か

解決済データ, 1-2, 1-22, 1-27, 1-28, 2-11
階層

カスタム・ソート, 5-13, 17-6

作成, 11-1

定義済, 11-1

表示, 5-9, 5-14

カスタム・メジャー, 20-19, 20-20, 20-21, 21-1 ~ 21-9

更新, 21-8

削除, 21-8

作成, 21-5, 21-6

問合せ, 21-2

き

期間属性, 18-3

キャッシュ

パフォーマンス統計, 6-2

キューブ, 1-6

DBMS_AWM メソッド, 1-7

アナリティック・ワークスペース, 3-6

アナリティック・ワークスペースに移入, 22-45

アナリティック・ワークスペースにおける作成およびリフレッシュ, 1-4

アナリティック・ワークスペースにおけるネーミング, 22-2

アナリティック・ワークスペースに作成, 22-19
作成, 2-9, 7-1, 8-1

ソース, 22-1

ターゲット, 22-1

表示, 5-5

キューブのロード仕様, 4-7

キューブ・ロード仕様, 1-4, 1-6, 22-26

DBMS_AWM メソッド, 1-9

く

グルーピング ID, 1-27, 1-28, 2-12, 5-13, 16-4, 18-3
親, 1-27

こ

固定ビュー, 6-2

コンポジット, 1-16, 22-17

コンポジット仕様, 1-6, 4-1, 4-3, 4-7, 22-17

DBMS_AWM メソッド, 1-10

さ

最適化

OLAP API, 24-1

し

時間ディメンション

作成, 2-6

集計

アナリティック・ワークスペース, 1-5, 1-18, 20-11, 20-12, 22-16, 22-24

演算子, 1-20, 5-3, 23-3

集計アドバイザ, 20-5

集計キャッシュ

パフォーマンス統計, 6-3

集計計画, 1-5, 3-8, 22-24

集計仕様, 1-5, 1-6, 1-13, 1-16

DBMS_AWM メソッド, 1-9

作成, 1-19, 22-24

終了日属性, 18-3

初期化パラメータ, 24-1

す

スター・スキーマ, 2-11

スノーflake・スキーマ, 2-11
スパースなデータ, 1-16, 22-17

せ

制限マップ, 25-12 ~ 25-16
構文, 25-12
セッション・カウンタ, 6-5
セッション・キャッシュ
パフォーマンス統計, 6-3
セッション統計, 6-4
セッション・ログ
出力, 20-22
取得, 20-14

そ

属性
表示, 5-8

た

多次元ターゲット・キューブ, 1-2
多次元データ
SQL アクセスの有効化, 1-22, 21-4, 22-1, 22-21,
22-23, 22-30, 22-32
多次元データ・モデル
アクティブ・カタログ, 3-1
データベース・スタンダード・フォーム, 3-1

て

ディメンション, 1-6
DBMS_AWM メソッド, 1-7
アナリティック・ワークスペース, 3-8, 3-10
アナリティック・ワークスペースに移入, 22-47
アナリティック・ワークスペースにおける作成およびリフレッシュ, 1-4
アナリティック・ワークスペースにおける順序付け, 1-4, 1-17
アナリティック・ワークスペースにおけるネーミング, 22-2
アナリティック・ワークスペースに作成, 22-28
埋込み合計, 16-6
親子, 16-1
作成, 2-2, 9-1
表示, 5-7

有効, 18-2
ディメンション属性
作成, 10-1
表示, 5-8
予約済, 10-1
ディメンションの別名, 5-5
ディメンション・ビュー
アナリティック・ワークスペース・オブジェクトの定義, 1-27, 22-30
ディメンション表, 2-10
OLAP メタデータの定義, 2-2
ディメンション・ロード仕様, 1-4, 1-6, 22-33, 22-47
DBMS_AWM メソッド, 1-8
ディレクトリ・オブジェクト, 1-23, 2-17, 16-1,
16-6, 22-22, 22-31, 22-37, 22-43, 23-3, 23-10,
23-11, 23-13
データベース・キャッシュ, 6-3
データベース・スタンダード・フォーム, 22-19, 22-28
オブジェクトの表示, 3-1
データベースの初期化, 24-1

と

動的パフォーマンス・ビュー, 6-1 ~ 6-5
トランザクション統計, 6-5

は

パフォーマンス・カウンタ, 6-1 ~ 6-5

ひ

ビュー
OLAP カタログ・メタデータ, 5-1
アクティブ・カタログ, 3-1
アナリティック・ワークスペースのオブジェクト,
3-1
アナリティック・ワークスペースのメンテナンス情報, 4-1
アナリティック・ワークスペース用に作成, 1-22
埋込み合計ディメンションの作成, 1-27, 25-2
埋込み合計メジャーの作成, 25-4
作成用テンプレート, 25-2
ロールアップ形式の作成, 25-5

ふ

- ファクト・ビュー
 - アナリティック・ワークスペース・オブジェクトからの定義, 1-28, 22-21, 22-23, 22-32
- ファクト表, 2-10, 5-12
 - OLAP メタデータの定義, 2-9
 - サポートされている構成, 2-11
 - ディメンション表との結合, 2-11
- プリンタ・バッファ, 20-13

へ

- ページ・プール
 - パフォーマンス統計, 6-3

ま

- マテリアライズド・ビュー
 - CWM2, 23-3, 23-4
 - OLAP API, 23-1
 - グルーピング・セット, 23-3, 23-4

み

- 未解決のデータ, 1-2, 2-11

め

- メジャー
 - アナリティック・ワークスペース, 3-7, 3-9
 - 作成, 14-1
 - 定義済, 14-1
 - 表示, 5-6
- メジャー・フォルダ
 - 作成, 7-1
 - 定義済, 5-4, 7-2
 - 表示, 5-4

よ

- 予約済のディメンション属性, 10-1
- 予約済のレベル属性, 13-1

り

- リレーショナル・アクセスの有効化, 1-5, 1-22

- 「アナリティック・ワークスペース」を参照
 - SQL アクセスの有効化
- リレーショナル・ソース・キューブ, 1-2
- リレーショナル・ターゲット・キューブ, 1-2

れ

- レベル
 - アナリティック・ワークスペース, 3-6, 3-11
 - 作成, 12-1
 - 表示, 5-10
- レベル属性
 - 作成, 13-1
 - 定義済, 13-1
 - 表示, 5-10, 5-11
 - 予約済, 13-1

ろ

- ロールアップ形式
 - 定義済, 25-17